Aras Innovator 2025 Release

Graph Navigation Administrator Guide

Document #: *D-008109*

Last Modified: *5/27/2025*

Copyright Information

Copyright © 2025 Aras Corporation. All Rights Reserved.

Aras Corporation
100 Brickstone Square
Suite 100
Andover, MA 01810
Phone: 978-691-8900

E-mail: support@aras.com

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

Notice of Rights

Copyright © 2025 by Aras Corporation and/or its affiliates. All rights reserved.

This document is protected by U.S. and international copyright laws and conventions. No copyright may be obscured or removed from this document. This document may not be modified or altered, or reproduced or transmitted in any form, without the explicit permission of 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.

Notice of Liability

THIS DOCUMENT IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY, AND THE CONTENTS HEREOF ARE SUBJECT TO CHANGE WITHOUT NOTICE. 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.

Table of Contents

Send Us Your Comments **4**

1 Terminology **5**

2 Overview **6**

3 Creating Graph Views **7**

3.1 Creating an Ad Hoc Graph View Definition 7
3.2 Creating a Query-Based Graph View Definition 12

3.2.1 Creating a Query Definition *12*
3.2.2 Creating a Query-Based Graph View Definition *12*

3.3 Creating a View Card 16
3.4 Creating Node View and Connector View Definitions 17
3.5 Attaching the Open Graph Action to an ItemType 19
3.6 Exporting Graph Views 19

4 Appendix **21**

4.1 View Card Template 21

4.1.1 Node Template Structure *21*
4.1.2 Node Global Styling *22*
4.1.3 Node Cell Styling *23*
4.1.4 Simple Connector Styling *24*
4.1.5 Connector with Label Styling *25*
4.1.6 Sample View Card *26*

©2025 Aras Corporation All Copyrights Reserved.

Send Us Your Comments

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

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

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

You can send comments to us in the following ways:

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

Or,

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

Or,

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

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

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

©2025 Aras Corporation All Copyrights Reserved.

1 Terminology

Table 1 describes the terms used in this Administrator Guide.

Term	Definition
Connector	A connector represents the link between nodes. A connector can be implicit or represent a collection of ItemType and Relationship information.
Connector Type (CT)	A definition for the dynamic aggregation of data to present in a Connector.
Connector View Definition (CVD)	A definition for the presentation styling of a Connector which uses View Cards.
Context Item	The item selected when starting up a graph view is the context item for the graph view and is the initial node displayed in the graph view.
Graph Navigation	The ability to dynamically access item data and related items by navigating relationships through a series of Nodes and Connectors that are displayed in a graphical format.
Graph View	The view that displays Nodes and Connectors data using the configured Graph View Definition. It also provides access to the supported Graph Navigation functionality.
Graph View Definition (GVD)	A definition containing all the information and mappings required to generate a Graph View.
Node	A node represents a point of focus for data and a point within a network of inter-related data.
Node Type (NT)	A definition for the dynamic aggregation of data to present in a Node.
Node View Definition (NVD)	A definition for the presentation styling of a Node that uses View Cards.
View Card	A View Card is a reusable Item that specifies data binding, styling, and layout for both Nodes and Connectors.

©2025 Aras Corporation All Copyrights Reserved.

2 Overview

The Graph Navigation application enables users to visualize and explore complex data structures. Administrators define and customize Graph Views that users can open for any enabled ItemType. Users can quickly find related items from within a Graph View by expanding underlying data and opening item forms for selected nodes in new tabs.

Use one of the following methods to create a Graph View Definition:

- Create a Query Definition using the Query Builder application that can then be used to restrict the information displayed within a Graph View.
- Create an Ad Hoc Query. This type of query is created dynamically at runtime as needed. All the properties and items related to a context item will be displayed to the user.

This guide describes the procedures used to create a Graph View similar to the one depicted below. For information regarding end user interaction, see the *Graph Navigation User Guide*.

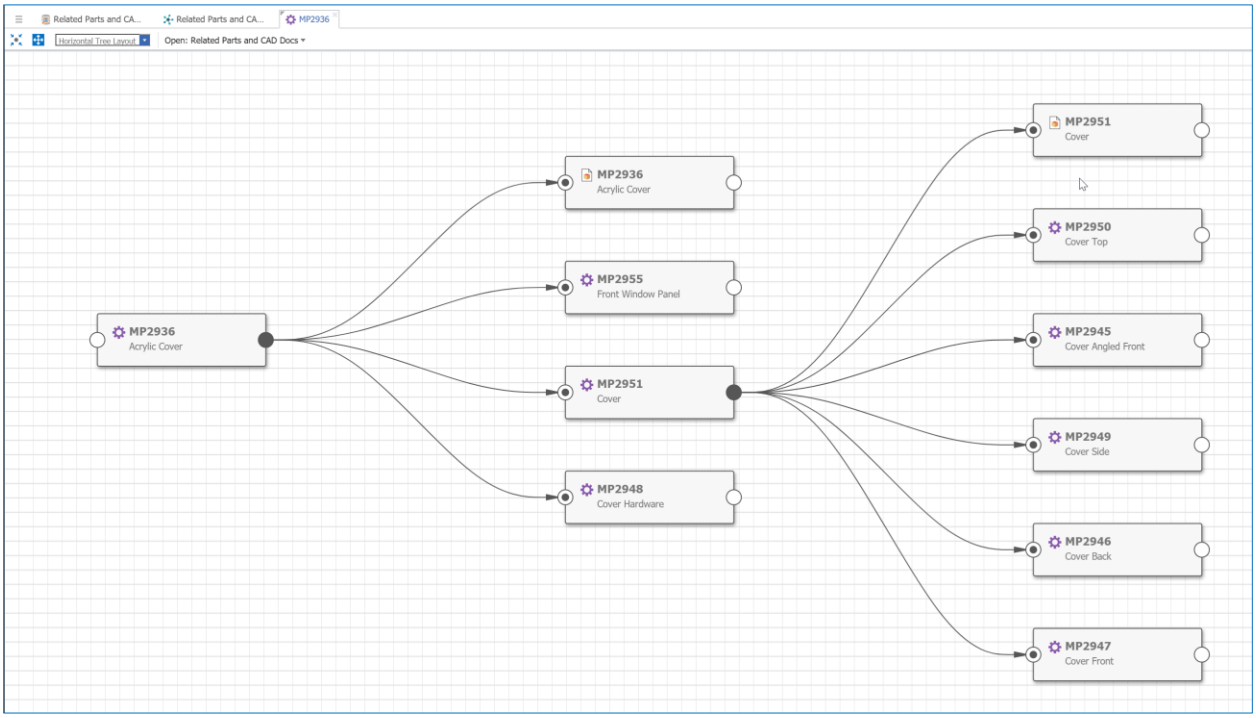

Figure 1.

©2025 Aras Corporation All Copyrights Reserved.

3 Creating Graph Views

Graph Views are defined using **Graph View Definition** Items, found under **Administration** --> **Configuration** --> **Graph Navigation** --> **Graph View Definitions** in the TOC.

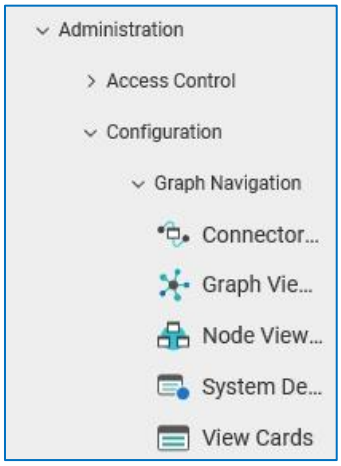

Figure 2.

A Graph View can display data constrained by a Query Definition (Query Based) or can display all data related to a root item if you create an Ad Hoc Query.

Graph View Definitions (GVD) are supported by the following ItemTypes:

- **Node View Definition (NVD)**
- **Node Type (NT)**
- **Connector View Definition (CVD)**
- **Connector Types (CT)**
- **View Card (VC)**
- **System Default Card**

3.1 Creating an Ad Hoc Graph View Definition

Ad-hoc GVDs allow users to freely navigate items and item relationships based on the selected root item.

Use the following procedure:

1. In the TOC go to **Administration>Configuration>Graph Navigation>Graph View Definitions**. The following menu appears.

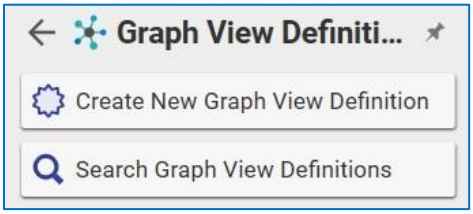

Figure 3.

©2025 Aras Corporation All Copyrights Reserved.

2. Click **Create New Graph View Definition**. The Graph View Definition screen appears.

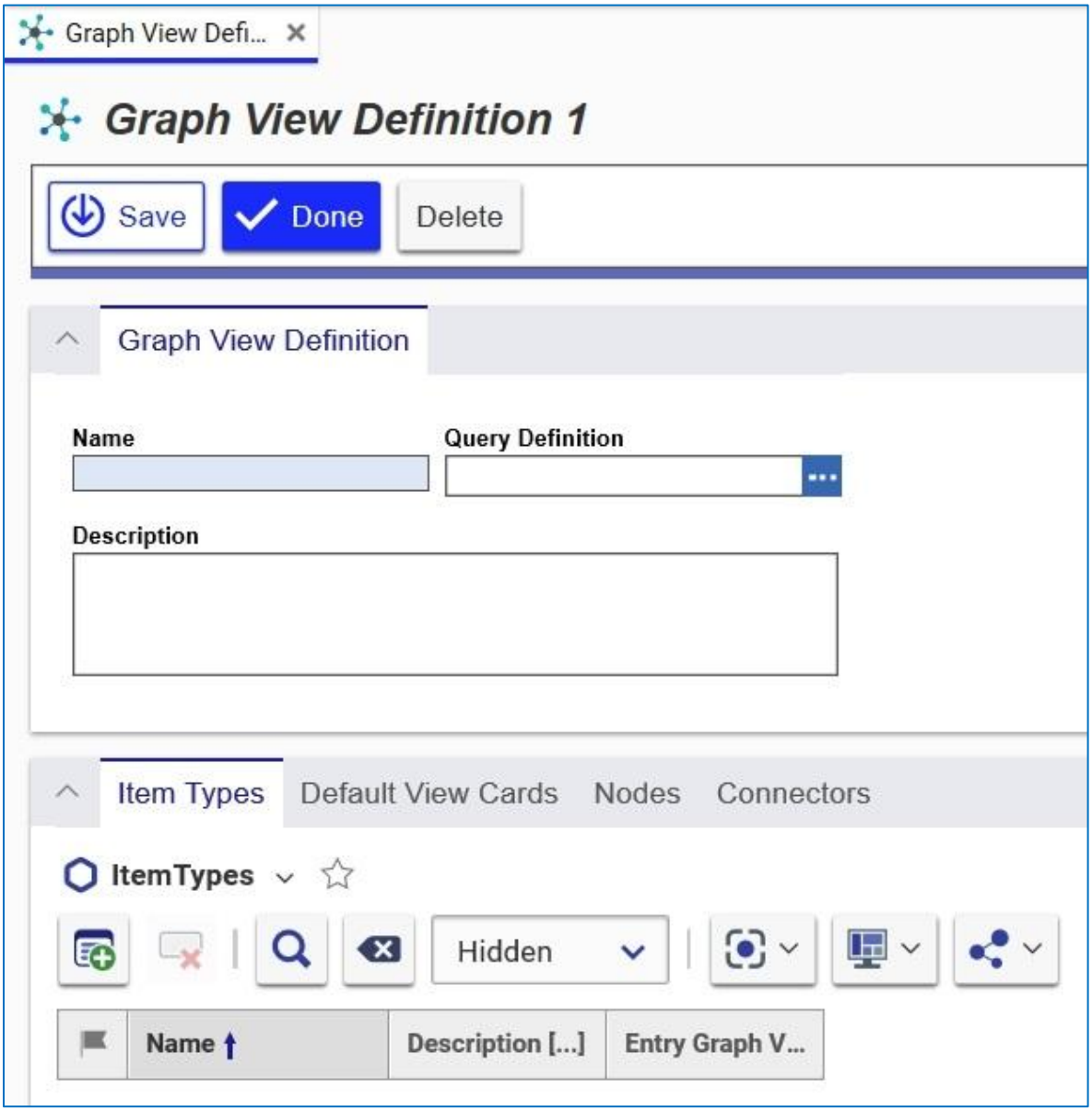

Figure 4.

3. Enter a unique **Name** for the GVD.
4. Save the Graph View Definition.

©2025 Aras Corporation All Copyrights Reserved.

5. Click the **Add ItemTypes** icon  in the Item Types Relationship Tab and pick the ItemTypes that will be permitted as root ItemTypes for the GVD. The following example uses the Part ItemType from which you can generate the example Graph View.

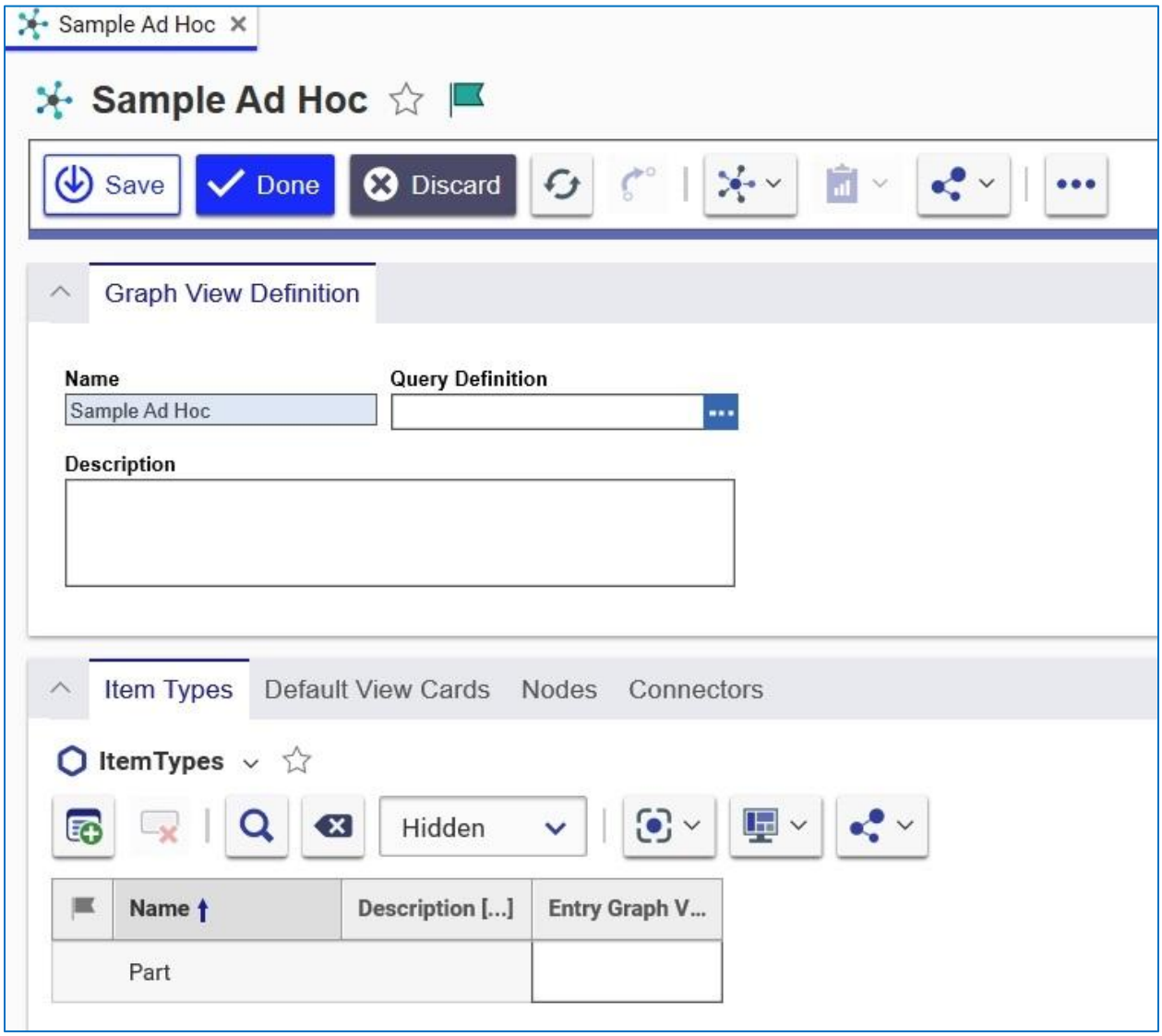

Figure 5.

Note: ItemType entries are not required for Ad-hoc GVDs. When they are not provided, the GVD can be used with ANY ItemType. The ItemType entries provide a way to restrict the root node. However, it does not restrict which ItemTypes can be reached from the root node.

6. Click the **Default View Cards** relationship tab to associate ItemTypes in the Graph View with specific View Cards.

©2025 Aras Corporation All Copyrights Reserved.

7. Click the **Add View Cards** icon  to add an existing view card to the Graph View.
8. Click the **New View Card** icon  to add a new view card to the Graph View.
9. Click the **New gn_GVDefaultCard**  icon to create a View Card. A new row appears.

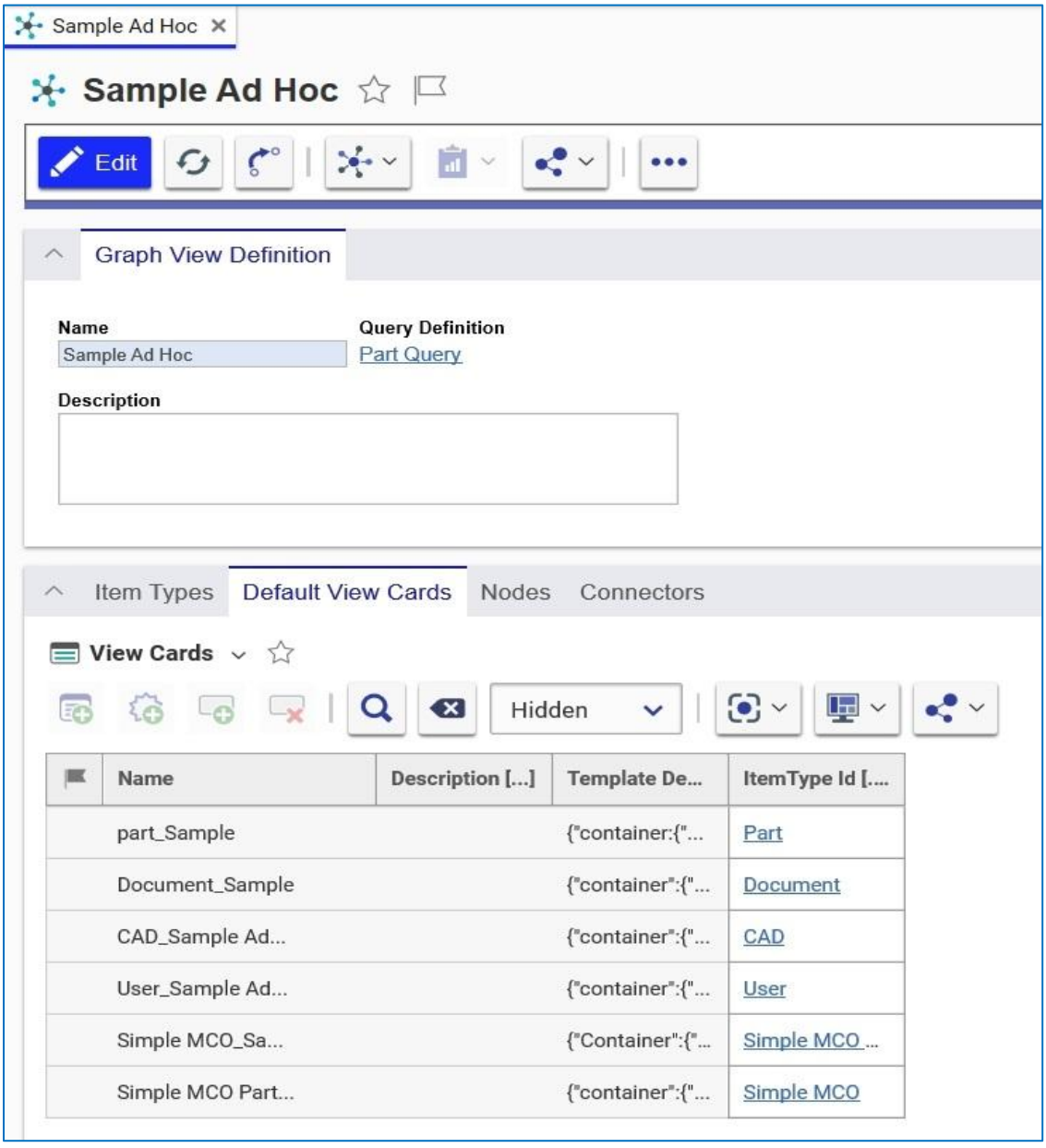

Figure 6.

©2025 Aras Corporation All Copyrights Reserved.

10. Double-click the ItemType ID field and click the ellipses to select the ItemType to associate with the Part default view card. The ItemType Search dialog box appears.

    Parts in the example Graph View use the Part Card. Graph View Definition Default View Cards take precedent over System Default Cards.

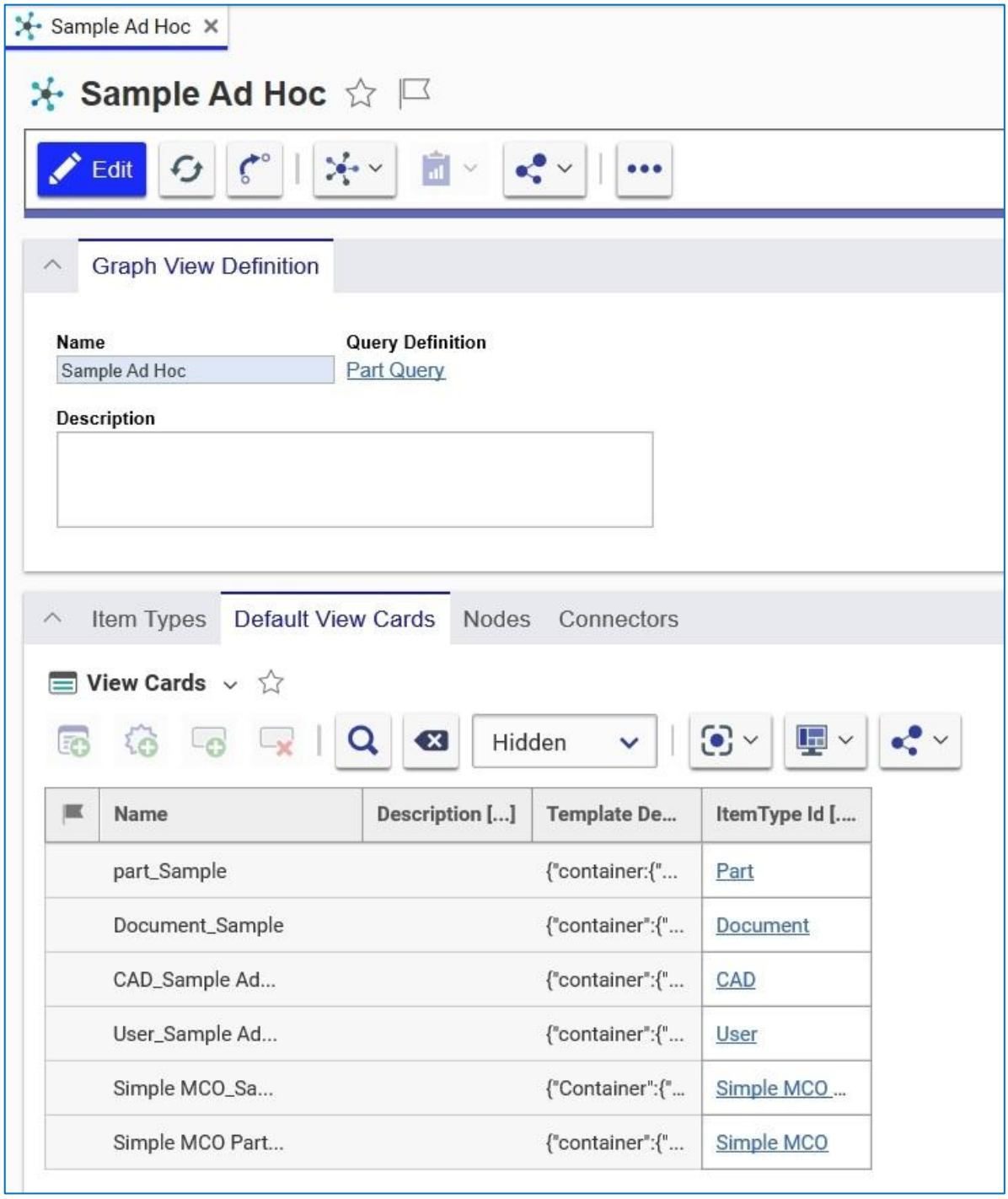

Figure 7.

©2025 Aras Corporation All Copyrights Reserved.

3.2 Creating a Query-Based Graph View Definition

Query based GVDs enable users to view items, relationships, and data that are explicitly defined within a Query Definition. This can be useful when there is a need to filter or focus the user's interaction on specific items and details.

3.2.1 Creating a Query Definition

Before creating a Query-based Graph View Definition, 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. Use a carefully configured Query Definition to automatically generate the necessary underlying items for the Graph View Definition.

Note: Only item properties selected in the Query Definition can be mapped to node and connector view cards. Mapping of Query Item properties to View Card content bindings is handled by the Node Type and Connector Type ItemTypes.

3.2.2 Creating a Query-Based Graph View Definition

Use the following procedure to create a Query Based Graph View Definition:

1. Create a **Query Definition** using the Query Builder Application.
2. Create a new Graph View Definition item, specifying a unique **Name** and selecting the desired **Query Definition**.

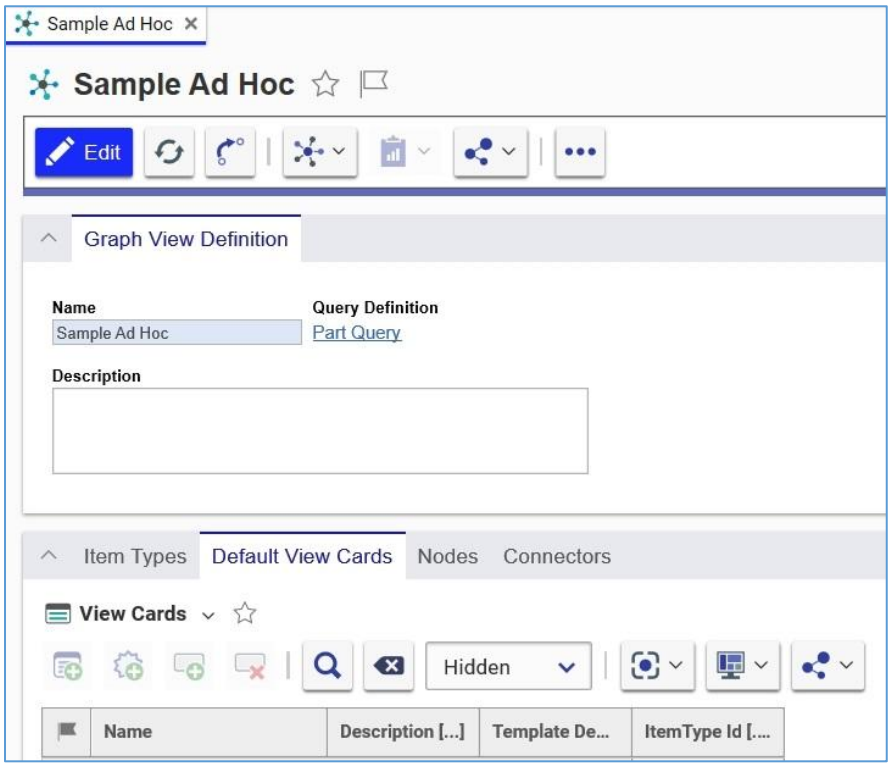

Figure 8.

©2025 Aras Corporation All Copyrights Reserved.

3. Save the Graph View Definition. The following popup dialog appears:

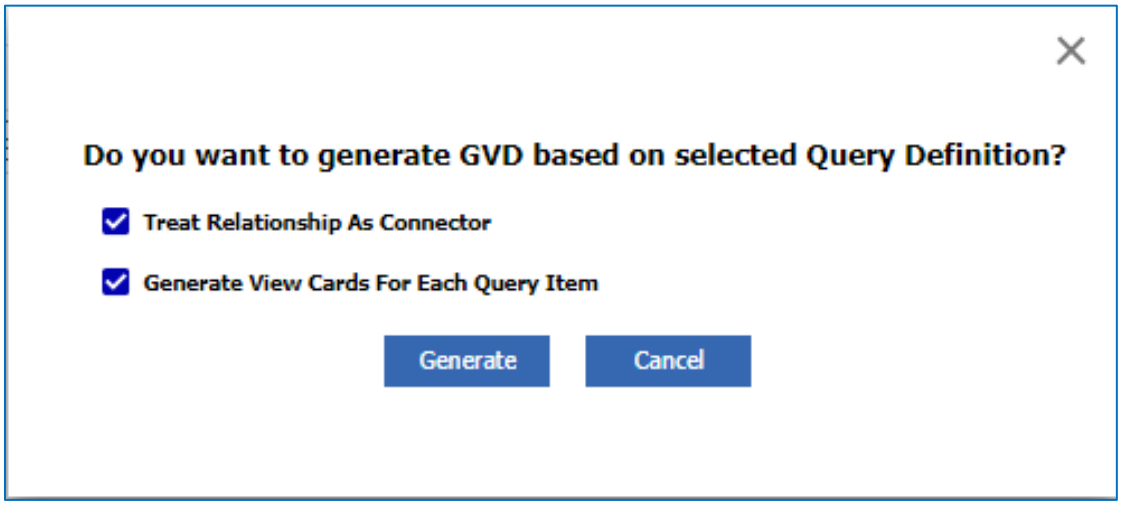

Figure 9.

4. Clicking the generate button runs a method that generates all underlying ItemTypes and Relationships and maps them to the query items of the Query Definition. If you select the **Treat Relationship As Connector** option, all relationships in the Query Definition are treated as connectors. Otherwise, each relationship is drawn as a separate node and connectors will be implicit. If you select the **Generate View Cards For Each Query Item** a unique view card is generated for each Query Item and added as a GVD Default View Card. The generated view cards will have a row for each property selected for the Query Item in the Query Definition.
5. In the **Default View Cards** Relationship Tab, you may associate ItemTypes in the Graph View with specific View Cards. Click on the **New Relationship** icon to pick a View Card. Then select the ItemType to associate the View Card with in the **ItemType ID** cell..

©2025 Aras Corporation All Copyrights Reserved.

6. In the **Nodes** Relationship Tab you can view and edit generated Node Types which are associated with the respective Query Item. Click on a Node Type to be directed to the Node Type Form. You may also associate an existing **Node View Definition** to a specific Node within the relationship tab.

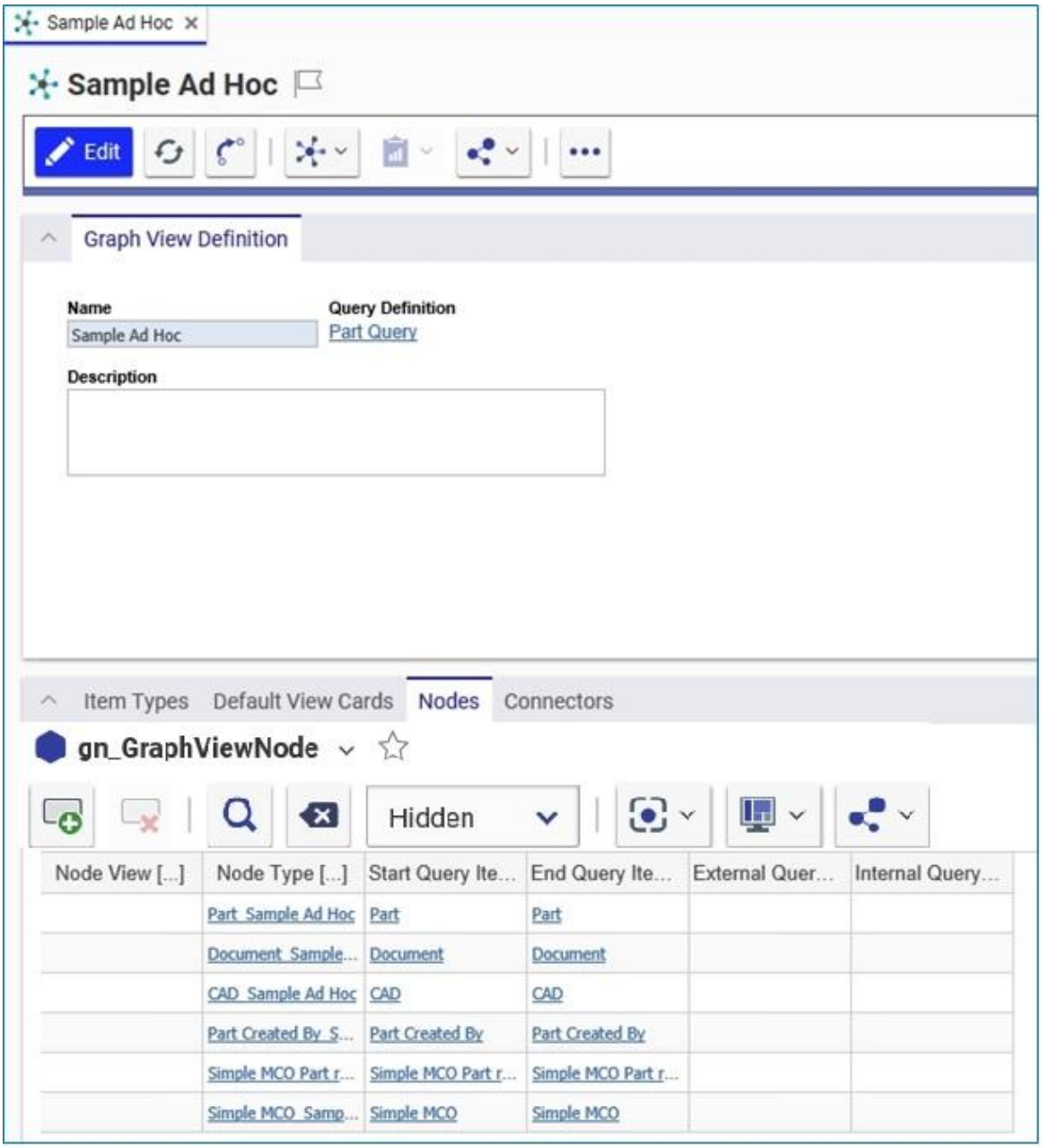

Figure 10.

©2025 Aras Corporation All Copyrights Reserved.

7. From a Node Type Form, you may add, remove, or edit Node Type Properties. The **Name** of a property is used to bind data to a View Card. See Section 4.1 for an example of a View Card that uses the **content_binding** attribute to refer to Node Type content. The **Value Template** is used to define the content to display. The Value Template supports both static and generated content. In Figure 11, observe that both name and state use Value Templates which refer to static and generated data.

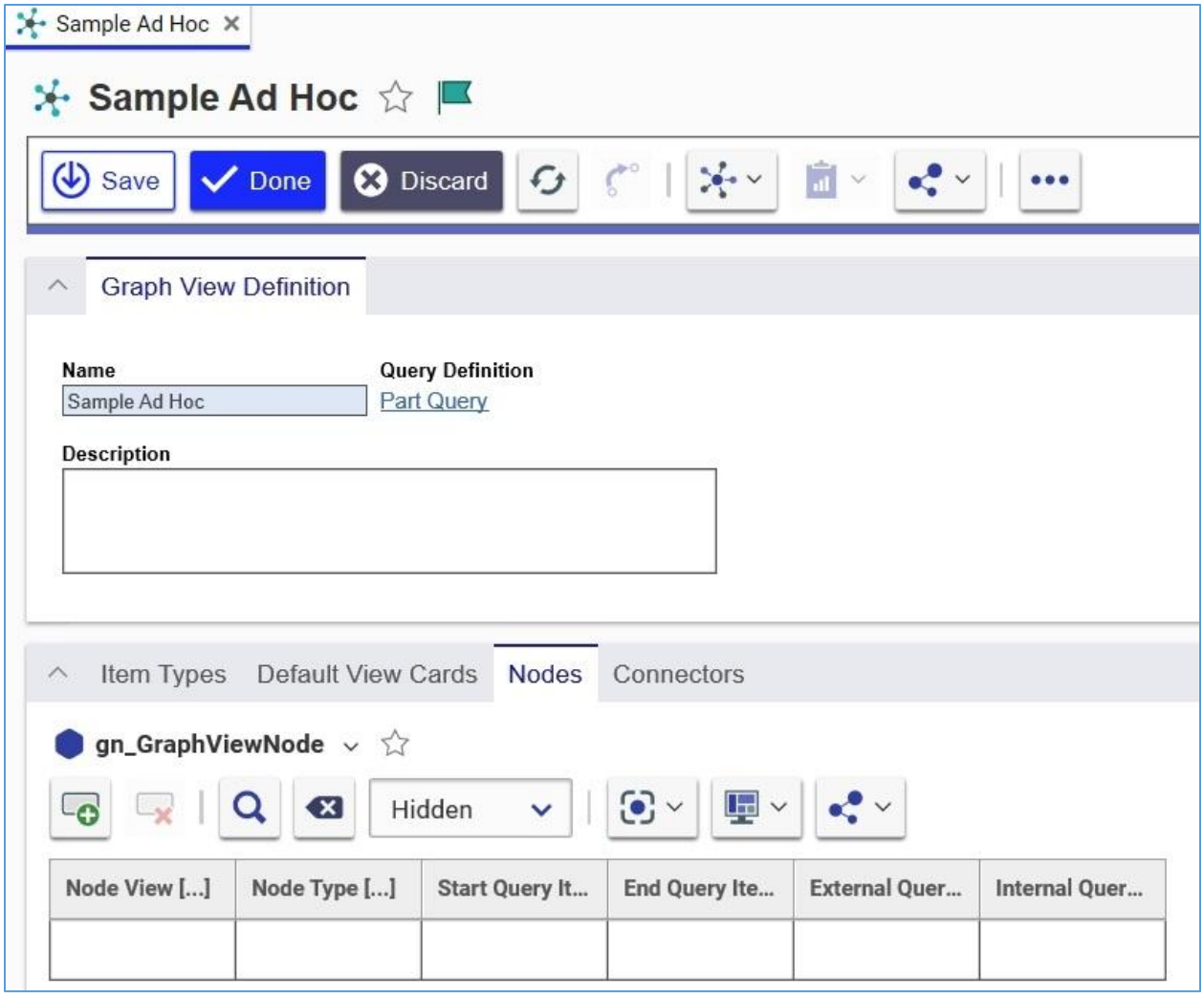

Figure 11.

Note: Append '@' for System ItemType properties. i.e. {@type}

8. If desired, repeat Steps 6 and 7 for the generated connectors and Connector Types found in the **Connectors** Relationship Tab.

Note: The entries required for the ItemTypes relationship of the GVD are generated automatically for Query-based GVDs. The requirements for modifying these entries for Query-based GVDs is currently outside the scope of this guide.

©2025 Aras Corporation All Copyrights Reserved.

3.3 Creating a View Card

Use the following procedure:

1. Navigate to **Administration→Configuration →Graph Navigation → View Cards** in the TOC. The View Cards menu appears.
2. Click **Create New View Card**. The View Card window appears.

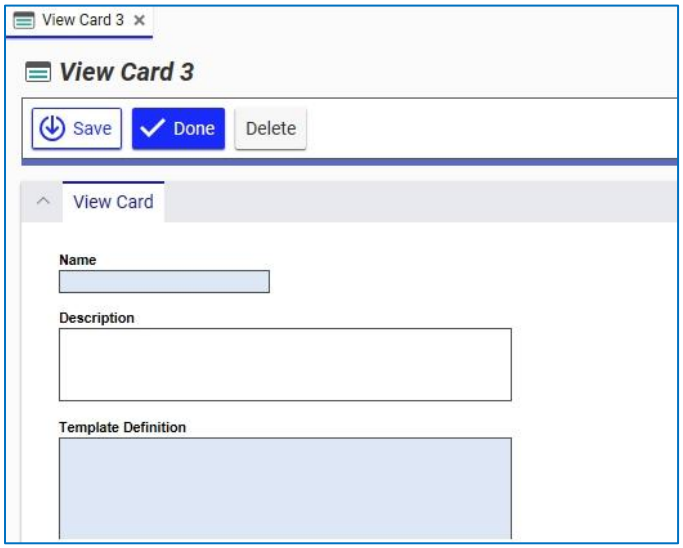

Figure 12

3. Specify a unique **Name.**
4. Specify the View Card Template Definition. See Section 4.1 for supported attributes and sample Templates.

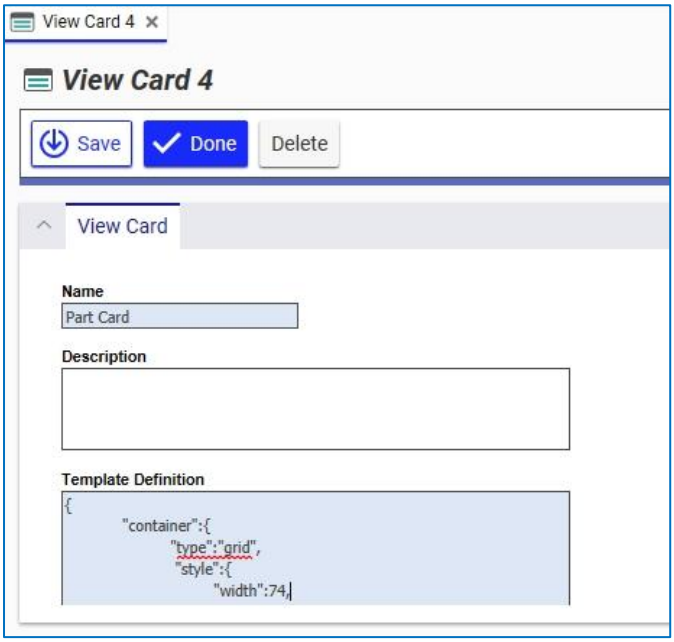

Figure 12.

©2025 Aras Corporation All Copyrights Reserved.

3.4 Creating Node View and Connector View Definitions

Use the following procedure to create either a Node View or a Connector View Definition:

1. Click **Administration→Configuration →Graph Navigation->Node View Definitions** in the TOC. The following menu appears:

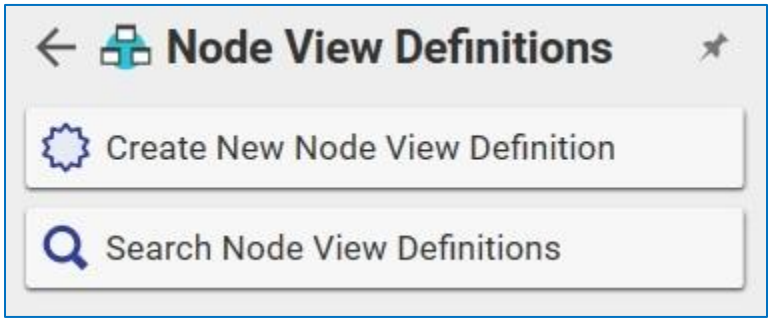

Figure 13.

2. Select **Create New Node View Definition.** The Node View Definition window appears:

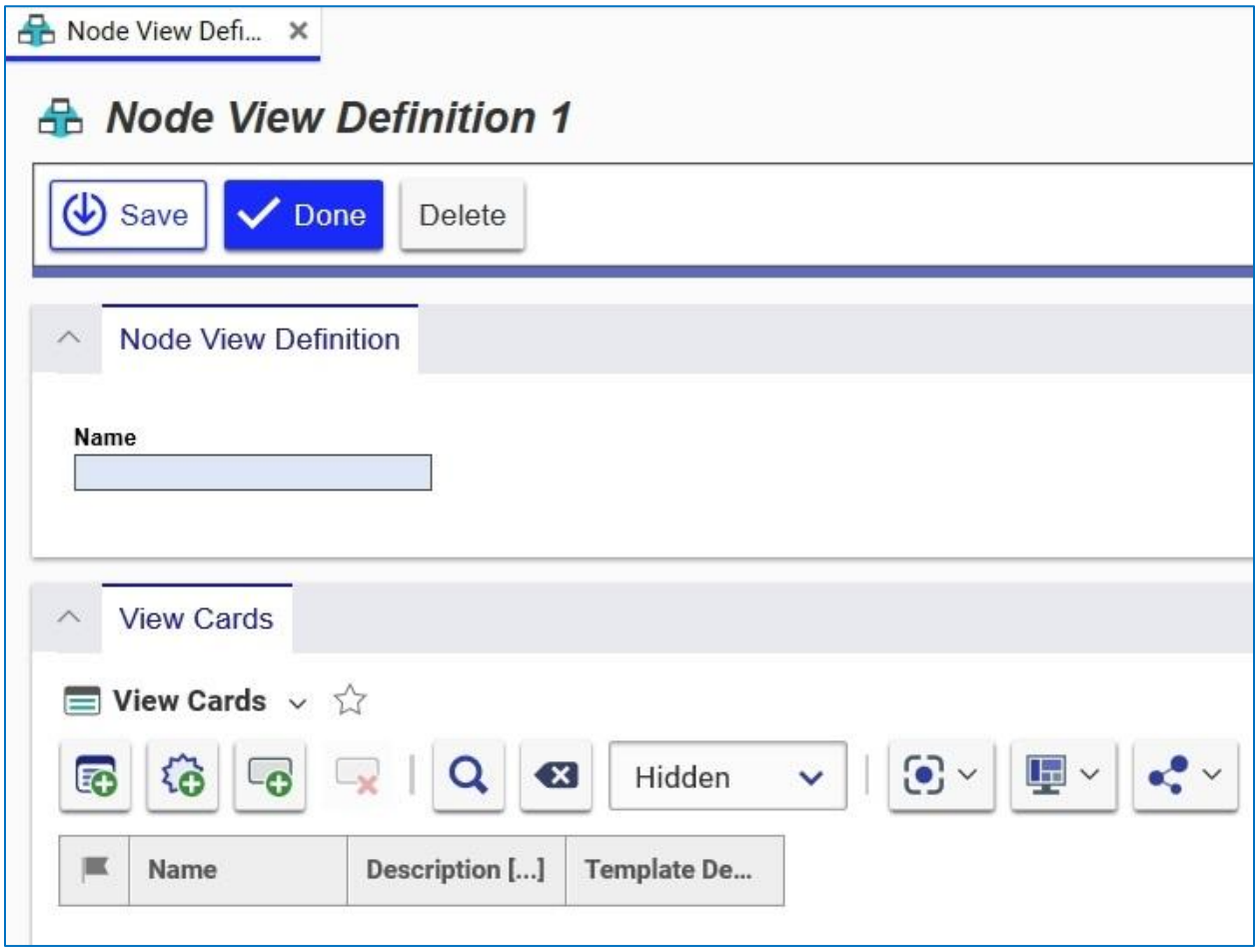

Figure 14.

©2025 Aras Corporation All Copyrights Reserved.

3. Specify a unique **Name.**

4. Click the **Add View Cards** icon  on the View Cards tab. The Search dialog box appears.

5. Enter p* in the Name column and click the Search icon:

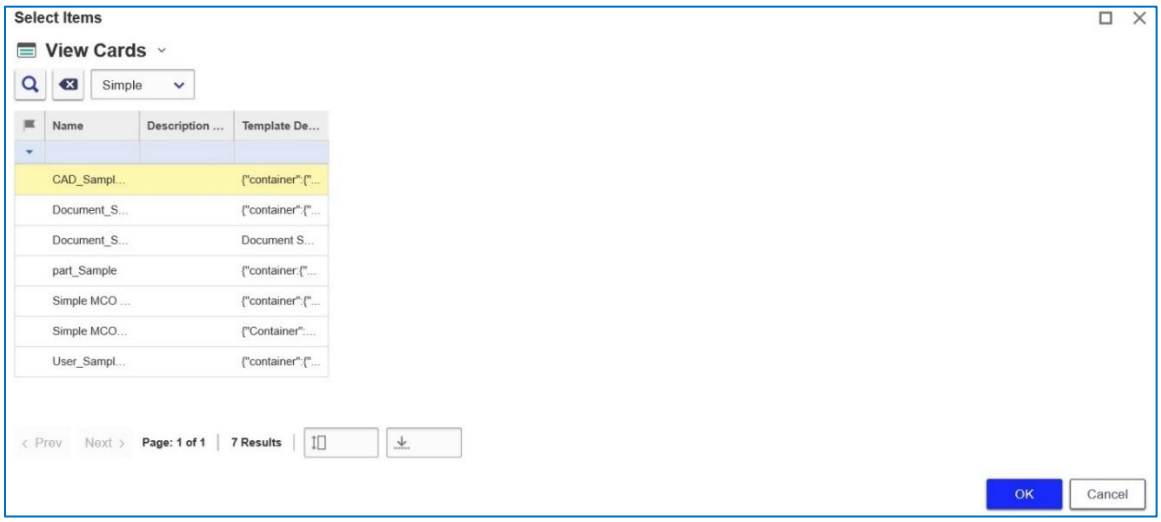

Figure 15.

6. Select **Part Card**.

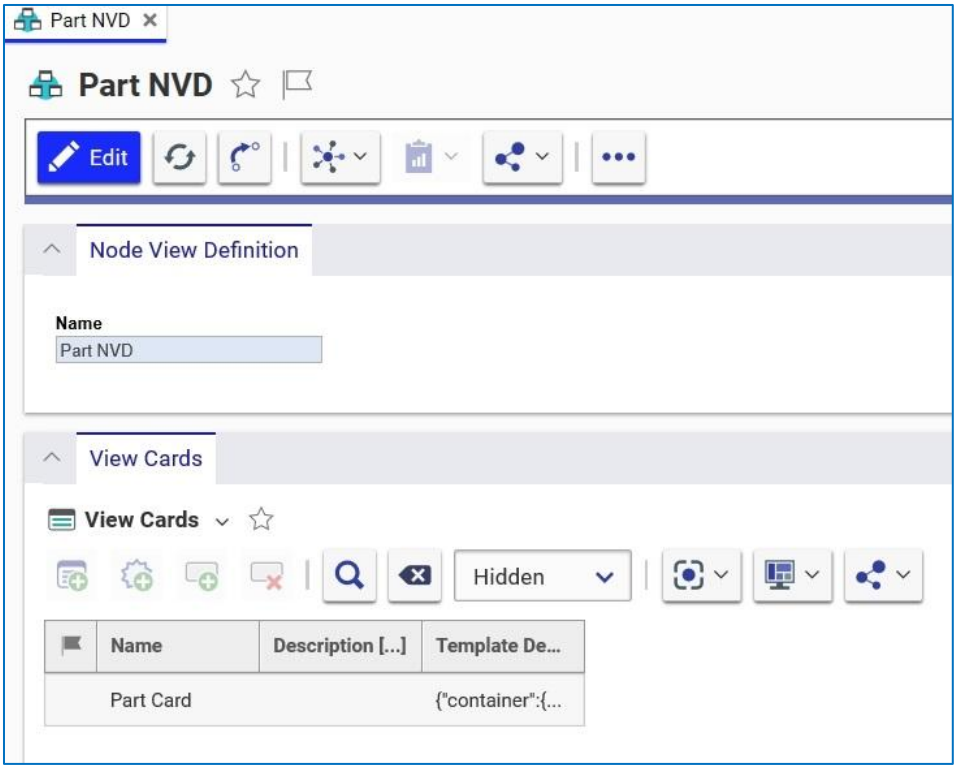

Figure 16.

©2025 Aras Corporation All Copyrights Reserved.

7. Click Save and Done to save and unclaim the NVD.

3.5 Attaching the Open Graph Action to an ItemType

To enable a user to open a Graph View on a specific ItemType, navigate to the desired ItemType and on the Actions tab, add an Action that runs the **gn_ShowGraph** Method. When this action is applied to an item instance, it will generate a Graph View in a new tab. If no available Graph View Definitions have been created by an administrator, a system default Ad Hoc Graph View will always be available.

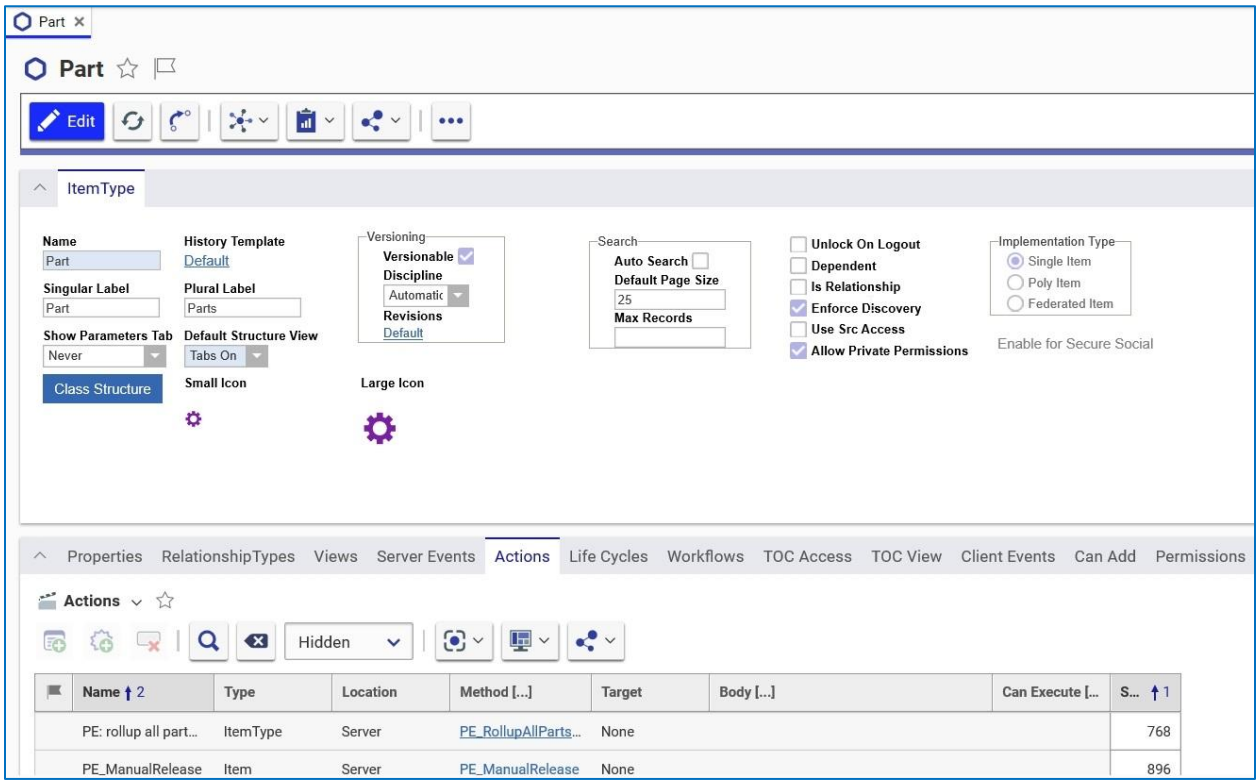

Figure 17.

3.6 Exporting Graph Views

When you create and export packages that include graph views, it is important to include any associated items with the top-level Graph View definition. You may need to expose some of these items in the TOC to package them:

- gn_GraphViewDefinition
- gn_ViewCard
- gn_ConnectorType
- gn_NodeType
- gn_ConnectorViewDefinition

©2025 Aras Corporation All Copyrights Reserved.

- gn_NodeViewDefinition
- gn_systemDefaultCard

You may also need to package any associated Query Definitions (qry_QueryDefinition), Actions, and Methods, depending on your graph view configuration.

©2025 Aras Corporation All Copyrights Reserved.

4 Appendix

4.1 View Card Template

A View Card template definition is a JSON structure containing node or connector rendering settings. If the graph rendering engine is not able to parse the template JSON or it has valid syntax but an invalid structure, the rendering engine will use the system default view card.

4.1.1 Node Template Structure

The general structure of a node template is as follows:

```
{
   "container": {
      "type": "grid",
      "style": {
         "rows": [{"height": 20}, … , {"height": 25}],
         "cols": [{"width": 50}, … , {"width": 75}],
         "cells": [{
                     "content": "image"|"text",
                     "content_binding": "binding_prop_name",
                     "col": column_index,
                     "row": row_index,
                     "style": {
                        // Text styling properties. Optional.
                     }
                  }
               ],
         // Set of styling settings described below
      }
   }
}
```

The root element of the template JSON is the “container” object. It contains **type** (currently only *grid* is supported) and **style**, the object containing the node structure and style settings.

- **rows** and **cols**: The arrays of row and column definitions required for grid content and used to determine the node content grid structure.
- **cells:** The array of items to be rendered inside the node.
    - **content:** Determines if the content of the cell is *text* or if it is an *image*.

©2025 Aras Corporation All Copyrights Reserved.

- **content_binding:** Contains the item name that must correspond to the name of the property specified in the Node Type (Figure 11).
- **col** and **row:** Determine the place in the grid where to render item.

View Cards support two levels of Styling Control:

1. Global (Applied to all cells within a node).
2. Individual cell styling.

4.1.2 Node Global Styling

The following controls are supported for the node:

- **Height**: The height of the node in pixels. Example:

  ```
  "height": 100
  ```
  Note: Minimal allowed height is equal to 45 pixels.

- **Width**: The width of the node in pixels. Example:

  ```
  "width": 100
  ```
  Note: Minimal allowed width is equal to 45 pixels.

- **Background Color**: The color of the node background. Examples:

  ```
  "color": "#F7F7F7",
  "color": "black"
  ```

- **Padding** (left, right, top, bottom): The padding of content inside of the node. Example:

  `"padding": 12` – same padding for all sides will be applied.
  `"padding": {"left": 10}` – specified padding is set to desired value, other values default to 12.
  `"padding": {"left": 5, "right": 5, "top": 10, "bottom": 10}`

- **Border Width**: The node border width in pixels. Example:

  ```
  "border": {"width": 1}
  ```

- **Border Color:** The color of the node border. Example:

  ```
  "border": {"color": "#555555"}
  ```

- **Border Corner Radius**: The radius in pixels for corner smoothing. Example:

  ```
  "border": {"cornerRadius": 2}
  ```

- **Border Shadow**: The border shadow with properties **horizontal** (horizontal shadow size in pixels), **vertical** (vertical shadow size in pixels), **color** and **opacity**. If any value is missed –the default one is used. Example:

  ```
  "border": {
     "shadow": {
        "horizontal": 2,
  ```

©2025 Aras Corporation All Copyrights Reserved.

```
        "vertical": 3,
        "color": "#000000",
        "opacity": 0.5
    }
}
```

The following is an example containing multiple border properties:

```
"border": {
    "width": 1,
    "color": "#555555",
    "cornerRadius": 2,
    "shadow": {
        "horizontal": 2,
        "vertical": 3,
        "color": "#000000",
        "opacity": 0.5
    }
}
```

4.1.3 Node Cell Styling

The following controls are supported for cells containing text content:

- **Font Family**: The font family. Example:

```
"font": {
    "family": "Tahoma"
}
```

- **Font Size:** The font size in pixels. Example:

```
"font": {
    "size": 10
}
```

- **Font Weight**: The font weight supports the following values:

```
"font": {
    "weight": "normal"|"bold"|"bolder"|"lighter"|"100"|"200"|
              "300"|"400"|"500"|"600"|"700"|"800"|"900"|"inherit"
}
```

- **Font Color:** The font color. Example:

```
"font": {
    "color": "#777777"
}
```

- **Text Decoration:** The text decoration supports the following values:

```
"textDecoration": "none"|"underline"|"overline"|"line-
through"|"blink"|"inherit"
```

- **Horizontal Alignment:** The horizontal alignment supports the following values:

©2025 Aras Corporation All Copyrights Reserved.

```
"horizontalAlignment": "start"|"middle"|"end"
```

- **Vertical alignment**: The horizontal alignment supports the following values:

```
"verticalAlignment": "hanging"|"middle"|"baseline"
```

- **Background Color:** The background color for the text. Example:

```
"backgroundColor": "#F7F7F7"
```

Example of a text cell definition:

```
{
   "content": "text",
   "content_binding": "part_number",
   "col": 2,
   "row": 1,
   "style": {
      "verticalAlignment": "top",
      "horizontalAlignment": "end",
      "textDecoration": "underline",
      "backgroundColor": "gray",
      "font": {
         "size": 10,
         "family": "Tahoma",
         "weight": "normal",
         "color": "#777777"
      }
   }
}
```

4.1.4 Simple Connector Styling

The following controls are supported for a simple connector:

- **Color**: The color of the connector line. Example:

```
"color": "black"
```

- **Weight**: The weight of the connector line in pixels. Example:

```
"weight": 2
```

- **Arrowhead**: The size and color of arrowhead which will be located near target node. Example:

```
"arrowhead": {
   "color": "black",
   "height": 17,
   "width": 7
}
```

Example of a simple connector template:

```
{
```

©2025 Aras Corporation All Copyrights Reserved.

```
      "style": {
         "color": "black",
         "weight": 2,
         "arrowhead": {
            "color": "black",
            "height": 17,
            "width": 7
         }
      }
   }
```

4.1.5 Connector with Label Styling

Use the node template format to define a template for a connector that includes labels. See Section 4.1.1 for more information. The following is an example of a connector view card template:

```
{
   "container": {
      "type": "grid",
      "style": {
         "width":20,
         "height":90,
         "color": "#F7F7F7",
         "padding": 12,
         "rows": [{"height": 20}],
         "cols": [{"width": 120}],
         "border": {
            "width": 1,
            "color": "#555555",
            "cornerRadius": 2
         },
         "cells": [
            {
               "content": "text",
               "content_binding": "keyed_name",
               "col": 0,
               "row": 0
            }

         ]
      }
   },
   "style": {
      "color": "black",
      "weight": 2,
      "arrowhead": {
         "color": "black",
         "height": 17,
         "width": 7
      }
   }
}
```

©2025 Aras Corporation All Copyrights Reserved.

4.1.6 Sample View Card

The following example shows a template definition that can be used in view cards to style both nodes and connectors.

```
{
   "container": {
      "type": "grid",
      "style": {
         "width":200,
         "height":90,
         "color": "#F7F7F7",
         "padding": {
            "left": 12,
            "right": 12,
            "top": 12,
            "bottom": 12
         },
         "rows": [{"height": 20}, {"height": 16}, {"height": 16},
            {"height": 16}],
         "cols": [{"width": 22}, {"width": 120}, {"width": 35}],
         "border": {
            "width": 1,
            "color": "#555555",
            "cornerRadius": 2,
            "shadow": {
               "horizontal": 2,
               "vertical": 3,
               "color": "#000000",
               "opacity": 0.5
            }
         },
         "cells": [
            {
               "content": "image",
               "content_binding": "icon",
               "col": 0,
               "row": 0
            },
            {
               "content": "text",
               "content_binding": "keyed_name",
               "col": 1,
               "row": 0
            },
            {
               "content": "text",
               "content_binding": "generation",
               "col": 2,
               "row": 0,
               "style": {
                  "verticalAlignment": "top",
                  "horizontalAlignment": "end",
                  "font": {
```

©2025 Aras Corporation All Copyrights Reserved.

```
                    "size": 10,
                    "family": "Tahoma",
                    "weight": "normal",
                    "color": "#777777"
                }
            }
        },
        {
            "content": "text",
            "content_binding": "name",
            "col": 1,
            "row": 1,
            "style": {
                "font": {
                    "size": 12,
                    "family": "Tahoma",
                    "weight": "normal",
                    "color": "#333333"
                }
            }
        },
        {
            "content": "text",
            "content_binding": "classification",
            "col": 1,
            "row": 2,
            "style": {
                "font": {
                    "size": 10,
                    "family": "Tahoma",
                    "weight": "normal",
                    "color": "#333333"
                }
            }
        },
        {
            "content": "text",
            "content_binding": "state",
            "col": 1,
            "row": 3,
            "style": {
                "font": {
                    "size": 10,
                    "family": "Tahoma",
                    "weight": "normal",
                    "color": "#777777"
                }
            }
        }
    ]
  }
},
"style": {
  "arrowhead": {
    "height": 17,
```

©2025 Aras Corporation All Copyrights Reserved.

```
            "width": 7
        }
    }
}
```

©2025 Aras Corporation All Copyrights Reserved.