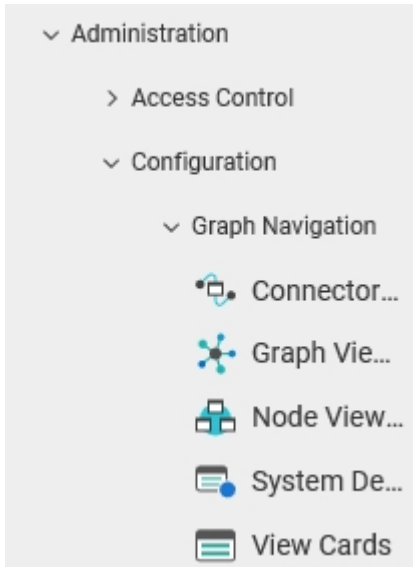


Creating Graph Views

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



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:

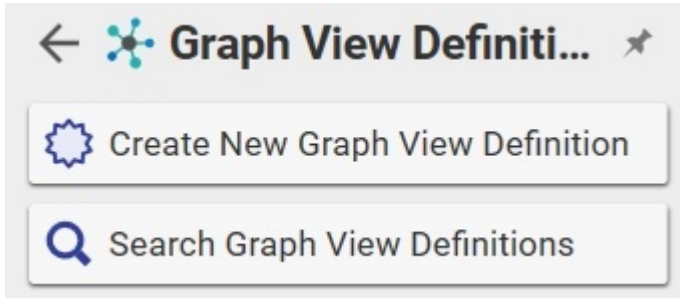
- **Node View Definition(NVD)**
- **Node Type(NT)**
- **Connector View Definition(CVD)**
- **Connector Types(CT)**
- **View Card(VC)**
- **System Default Card**

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.



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

Graph View Defi... x

Graph View Definition 1

Save

Done

Delete

^
 Graph View Definition

Name

Query Definition
 ...

Description

^
 Item Types
 Default View Cards
 Nodes
 Connectors

ItemTypes v ☆

Hidden v

v
 v
 v

Name ↑

Description [...]

Entry Graph V...

3. Enter a unique **Name** for the GVD.
4. Save the Graph View Definition.
5. Click the Add ItemTypes icon null 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.

©2026 Aras Corporation All Copyrights Reserved

3

Sample Ad Hoc

Sample Ad Hoc

Save
 Done
 Discard

^ Graph View Definition

Name

Query Definition

Description

^ Item Types Default View Cards Nodes Connectors

ItemTypes

	Name ↑	Description [...]	Entry Graph V...
	Part		

Important

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.

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



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



Sample Ad Hoc



Graph View Definition

Name
Sample Ad Hoc

Query Definition
[Part Query](#)

Description

Item Types Default View Cards Nodes Connectors

 **View Cards** 



	Name	Description [...]	Template De...	ItemType Id [...]
	part_Sample		{"container":{"...	Part
	Document_Sample		{"container":{"...	Document
	CAD_Sample Ad...		{"container":{"...	CAD
	User_Sample Ad...		{"container":{"...	User
	Simple MCO_Sa...		{"Container":{"...	Simple MCO...
	Simple MCO Part...		{"container":{"...	Simple MCO

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.



Sample Ad Hoc



Graph View Definition

Name **Query Definition**

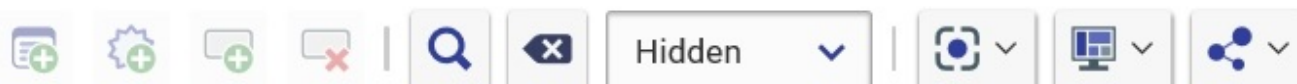
Sample Ad Hoc

[Part Query](#)

Description

Item Types **Default View Cards** Nodes Connectors

 **View Cards** 



	Name	Description [...]	Template De...	ItemType Id [...]
	part_Sample		{"container":{"...	Part
	Document_Sample		{"container":{"...	Document
	CAD_Sample Ad...		{"container":{"...	CAD
	User_Sample Ad...		{"container":{"...	User
	Simple MCO_Sa...		{"Container":{"...	Simple MCO ...

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.

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.

Important

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.

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**.

Sample Ad Hoc

Sample Ad Hoc

Edit

^

Graph View Definition

Name

Sample Ad Hoc

Query Definition

[Part Query](#)

Description

^

Item Types

Default View Cards

Nodes

Connectors

View Cards

^

Hidden

^

	Name	Description [...]	Template De...	ItemType Id [...]

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



Do you want to generate GVD based on selected Query Definition?

- ☒ **Treat Relationship As Connector**
- ☒ **Generate View Cards For Each Query Item**

Generate

Cancel

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..
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.



Sample Ad Hoc

Sample Ad Hoc

Edit

Graph View Definition

Name

Sample Ad Hoc

Query Definition

Part Query

Description

Item Types

Default View Cards

Nodes

Connectors

gn_GraphViewNode

Hidden

Node View [...]	Node Type [...]	Start Query It...	End Query It...	External Quer...	Internal Query...
	Part_Sample Ad Hoc	Part	Part		
	Document_Sample...	Document	Document		
	CAD_Sample Ad Hoc	CAD	CAD		
	Part Created By_S...	Part Created By	Part Created By		
	Simple MCO Part r...	Simple MCO Part r...	Simple MCO Part r...		
	Simple MCO Samp...	Simple MCO	Simple MCO		

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.

Sample Ad Hoc

Sample Ad Hoc

Save

Done

Discard

Graph View Definition

Name

Sample Ad Hoc

Query Definition

Part Query

Description

Item Types

Default View Cards

Nodes

Connectors

gn_GraphViewNode

Hidden

Node View [...]	Node Type [...]	Start Query It...	End Query It...	External Quer...	Internal Quer...

Important

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.



Important

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.

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.

The screenshot shows a web application window titled 'View Card 3'. At the top, there is a header bar with a hamburger menu icon and the text 'View Card 3'. Below the header, there is a toolbar with three buttons: 'Save' (with a download icon), 'Done' (with a checkmark icon), and 'Delete'. Below the toolbar, there is a tab labeled 'View Card' with an upward arrow icon. The main content area contains three form fields: 'Name' (a single-line text input), 'Description' (a multi-line text area), and 'Template Definition' (a large multi-line text area).

Figure 12

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

View Card 4 x

View Card 4

Save

Done

Delete

View Card

Name

Part Card

Description

Template Definition

```
{
  "container": {
    "type": "grid",
    "style": {
      "width": 74,

```

Creating Node View and Connector View Definitions

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

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

←

Node View Definitions

★

⚙️ Create New Node View Definition

🔍 Search Node View Definitions

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

Node View Defi... x

 **Node View Definition 1**

Save

Done

Delete

^

Node View Definition

Name

^

View Cards

 **View Cards** v 





Hidden v

 v v v

	Name	Description [...]	Template De...
---	------	-------------------	----------------

- Specify a unique **Name**.
- Click the Add View Cards icon null on the View Cards tab. The Search dialog box appears.
- Enter p* in the Name column and click the Search icon:

©2026 Aras Corporation All Copyrights Reserved



16

Select Items



View Cards ▾

Simple ▾

	Name	Description ...	Template De...
▼			
	CAD_Sampl...		{"container":{"..."
	Document_S...		{"container":{"..."
	Document_S...		Document S...
	part_Sample		{"container":{"..."
	Simple MCO ...		{"container":{"..."
	Simple MCO...		{"Container":....
	User_Sampl...		{"container":{"..."

< Prev Next > Page: 1 of 1 | 7 Results |

OK Cancel

6. Select **Part Card**.





Part NVD

 View Cards



Hidden



	Name	Description [...]	Template De...
	Part Card		{"container":{"...

Attaching the Open Graph Action to an ItemType

©2026 Aras Corporation All Copyrights Reserved



Part

Part

Edit

Item Type

Name
Part

Singular Label
Part

Show Parameters Tab
Never

History Template
Default

Plural Label
Parts

Default Structure View
Tabs On

Versioning
Versionable ☒
Discipline
Automatic
Revisions
Default

Search
Auto Search ☐
Default Page Size
25
Max Records

Unlock On Logout ☐
Dependent ☐
Is Relationship ☐
Enforce Discovery ☒
Use Src Access ☐
Allow Private Permissions ☒

Implementation Type
Single Item ☒
Poly Item ☐
Federated Item ☐

Enable for Secure Social

Class Structure

Small Icon

Large Icon

Properties RelationshipTypes Views Server Events Actions Life Cycles Workflows TOC Access TOC View Client Events Can Add Permissions

Actions

Hidden

Name	Type	Location	Method [...]	Target	Body [...]	Can Execute [...]	S...
PE: rollup all part...	ItemType	Server	PE_RollupAllParts...	None			768
PE_ManualRelease	Item	Server	PE_ManualRelease	None			896

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
- 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.

