

# Anzo<sup>®</sup> 5.3 User Guide

**Last Updated:** 6/21/2023

---

Online documentation is available at [docs.cambridgesemantics.com](https://docs.cambridgesemantics.com)

# Table of Contents

Onboarding & Virtualization .....	18
Onboarding Structured Data .....	19
Adding Data Sources .....	21
Connecting to a Database .....	21
Defining a Database Schema .....	24
Adding a CSV Data Source .....	59
Adding a JSON Data Source .....	67
Adding an XML Data Source .....	77
Adding a SAS Data Source .....	83
Adding a Parquet Data Source .....	92
Configuring a CSV or Parquet Data Source for Incremental Processing .....	96
Assigning Primary Keys in an Onboarded Schema .....	101
Creating or Changing Foreign Keys .....	104
Ingesting Data Sources via ETL Pipelines .....	111
Ingesting a New Data Source .....	111
Re-Ingesting an Updated Data Source .....	118
Ingesting a Data Source with a Metadata Dictionary .....	123
Managing Data Source Metadata .....	131
Working with Mappings .....	143
Working with Pipelines .....	191
Directly Loading Data Sources via Graphmarts .....	201
Creating a Graphmart from a Data Source .....	201
Adding a Data Source to an Existing Graphmart .....	205
Direct Load Advanced Settings Reference .....	208
Onboarding or Virtualizing Data with the Graph Data Interface .....	213

Introduction to the GDI .....	213
General GDI Concepts and Usage .....	215
GDI Usage by Data Source Type .....	299
GDI Property Reference .....	403
<b>Onboarding Unstructured Data .....</b>	<b>417</b>
Creating an Unstructured Pipeline .....	418
Running an Unstructured Pipeline .....	441
Pipeline Settings Reference .....	444
Annotator Settings Reference .....	450
External Service Annotator .....	450
Keyword and Phrase Annotator .....	453
Knowledgebase Annotator .....	455
Regex Annotator .....	459
<b>Versioning and Migrating Artifacts .....</b>	<b>463</b>
Creating and Restoring Versions of Artifacts .....	464
Creating a Backup Version of an Artifact .....	464
Restoring a Backup Version of an Artifact .....	467
Exporting an Artifact .....	469
Making Values Replaceable on Export .....	475
Importing Exported Versions of Artifacts .....	476
<b>Model .....</b>	<b>480</b>
Introduction to Model Concepts .....	481
Introduction to Managed Models .....	484
Model Requirements and Recommendations .....	486
Uploading a Model to Anzo .....	491

Creating a Model .....	494
Editing a Model .....	497
Opening Models in the Editor .....	498
Changing Model Components .....	501
Class Editor Reference .....	504
Property Editor Reference .....	505
Downloading a Model .....	507
Defining Resource Templates .....	512
Blend .....	516
Working with Datasets .....	517
Adding an Empty Dataset for an Export Step .....	518
Importing an Existing Dataset (FLDS) .....	522
Creating a Dataset from RDF Files .....	528
Creating a Graphmart from a Dataset .....	534
Adding a Dataset to an Existing Graphmart .....	540
Managing Dataset Editions .....	545
Introduction to Editions .....	545
Creating an Edition .....	548
Modifying an Edition .....	553
Deleting a Saved Edition .....	555
Working with Graphmarts .....	558
Creating a Graphmart .....	559
Copying a Graphmart .....	560
Graphmart Settings Reference .....	562
Creating an Elasticsearch Index from Graphmart Data .....	566



Adding Data Layers to Graphmarts .....	572
Introduction to Data Layers .....	572
Creating a New Data Layer .....	574
Cloning an Existing Data Layer .....	577
Using Query Contexts in Queries .....	580
Defining Execution Conditions .....	584
Advanced Data Access Settings .....	586
Adding Steps to Data Layers .....	589
Directly Loading a Data Source (Direct Load Step) .....	589
Creating an Elasticsearch Index (Elasticsearch Indexing Step) .....	594
Taking a Snapshot of an Elasticsearch Index (Elasticsearch Snapshot Step) .....	598
Exporting Data to an FLDS (Export Step) .....	602
Loading a Dataset from the Catalog (Load Dataset Step) .....	609
Pre-Compiling a Query (Pre-Compile Query Step) .....	613
Creating a Reusable Query Template .....	617
Running a Transformation Query (Query Step) .....	628
Inferring New Data (RDFS+ Inference Step) .....	632
Validating the Data (Validation Step) .....	641
Constructing a View of the Data (View Step) .....	646
Creating Data on Demand Endpoints .....	654
Creating an Auto-Generated Endpoint .....	655
Creating a Custom Endpoint (Preview) .....	662
Sharing Access to Graphmarts .....	675
Introduction to Graphmart and Data Layer Sharing .....	675
Configuring Graphmart and Data Layer Permissions .....	679
Profiling Data Sources, Datasets, and Graphmarts .....	686

Generating a Source Data Profile .....	687
Generating a Dataset Data Profile .....	692
Generating a Graphmart Data Profile .....	697
Data Profiling Metrics .....	702
<b>Access .....</b>	<b>714</b>
<b>Accessing Data with Hi-Res Analytics Dashboards .....</b>	<b>715</b>
Introduction to Hi-Res Analytics .....	716
Getting Started: Exploring and Visualizing Data .....	723
Create a New Dashboard .....	723
Explore the Data .....	726
Create Visualizations of the Data .....	730
Creating Dashboards .....	735
Creating a Graphmart Dashboard .....	735
Creating a Network Navigator Dashboard (Preview) .....	739
Creating Lenses .....	769
Creating an AnzoKO Web Page Lens .....	769
Creating Chart Lenses .....	771
Creating a Drill Down Lens .....	791
Creating a Form Lens .....	795
Creating a List Lens .....	799
Creating a Query Lens .....	802
Creating a Resource Tree Navigator Lens .....	803
Creating a Table Lens .....	807
Creating a Web Page Lens .....	812
Creating Dashboard Filters .....	817
Creating a Cloud Filter .....	817

Creating a Date Range Filter .....	822
Creating a Hierarchy Filter .....	827
Creating a Limit Filter .....	831
Creating a List Filter .....	835
Creating a Numeric Range Filter .....	839
Creating a Presence Filter .....	844
Creating a Quartile Filter .....	848
Creating a Range Slider Filter .....	851
Creating a Relative Time Filter .....	854
Creating a Search Filter .....	858
Creating a Single Select List Filter .....	862
Creating a Types Filter .....	866
Combining Data from Multiple Classes .....	871
Calculating Values in Lenses and Filters .....	878
Searching for Text in Unstructured Documents .....	884
Supported Search Syntax .....	891
Sharing Access to a Dashboard or Lens .....	893
Configuring Dashboard or Lens Permissions .....	896
Cloning a Lens .....	899
Exporting a Lens .....	901
Deleting a Lens .....	904
Accessing Data with the Query Builder .....	906
Running SPARQL Queries in the Query Builder .....	907
Searching for Quads in the Query Builder .....	916
Accessing Data on Demand Endpoints .....	922
Accessing an Endpoint Programmatically .....	923

Accessing an Endpoint from an Application .....	927
Accessing Data via the OData API .....	928
Downloading the Anzo ODBC and JDBC Drivers .....	930
JDBC Driver Quick Reference .....	938
OData Reference .....	944
OData URL Conventions .....	944
Supported Query Operators .....	944
Accessing Data from the SPARQL Endpoint .....	954
Accessing Data from the HTTP Client Interface .....	965
Sharing Access to Artifacts .....	973
SPARQL Best Practices and Query Templates .....	980
SPARQL Best Practices .....	981
SPARQL Query Templates .....	988
Anzo Function and Formula Reference .....	994
String Functions .....	995
BUSINESS_ENTITY_EXCLUDER .....	997
CONCATENATE .....	997
CONCATURL .....	997
CONTAINS .....	998
ENCODE_FOR_URI .....	998
ESCAPEHTML .....	999
FIND .....	999
FINDREVERSE .....	1000
GROUP_CONCAT .....	1001
GROUPCONCAT .....	1001

LANG .....	1003
LANGMATCHES .....	1003
LCASE .....	1004
LEFT .....	1004
LEN .....	1005
LEVENSHTEIN_DIST .....	1005
LOWER .....	1006
MD5 .....	1006
MID .....	1007
REGEX .....	1007
REGEXP_SUBSTR .....	1008
REPLACE .....	1009
RIGHT .....	1010
SEARCH .....	1010
SHA1 .....	1011
SHA224 .....	1012
SHA256 .....	1012
SHA384 .....	1012
SHA512 .....	1013
STRAFTER .....	1013
STRBEFORE .....	1014
STRDT .....	1014
STRENDIS .....	1015
STRLANG .....	1016
STRLEN .....	1016
STRSTARTS .....	1016

STRUUID .....	1017
SUBSTITUTE .....	1017
SUBSTR .....	1018
TOURI .....	1019
TRIM .....	1019
UCASE .....	1020
UPPER .....	1020
Math Functions .....	1021
ABS .....	1023
ADD .....	1023
AVERAGEIF .....	1023
AVERAGEIFS .....	1024
AVG .....	1025
CEILING .....	1025
COS .....	1026
DIVIDE .....	1026
EQUAL .....	1027
EXP .....	1027
FACT .....	1028
FLOOR .....	1028
GE .....	1029
GT .....	1029
HAMMING_DIST .....	1030
HAVERSINE_DIST .....	1031
LE .....	1031
LN .....	1032

LOG .....	1032
LOG2 .....	1033
LT .....	1033
MAXVAL .....	1034
MINVAL .....	1034
MOD .....	1035
MULTIPLY .....	1035
NOT_EQUAL .....	1036
NPV .....	1036
NUMERIC-ADD .....	1037
NUMERIC-SUBTRACT .....	1038
PI .....	1038
POWER .....	1038
PRODUCT .....	1039
QUOTIENT .....	1039
RAD .....	1040
RAND .....	1040
RANDBETWEEN .....	1041
ROUND .....	1041
ROUNDDOWN .....	1042
ROUNDUP .....	1042
SIN .....	1043
SQRT .....	1043
SUBTRACT .....	1044
SUM .....	1044
SUMIF .....	1045

SUMIFS .....	1045
SUMPRODUCT .....	1046
SUMSQ .....	1047
TAN .....	1047
Aggregate Functions .....	1049
AVERAGEIF .....	1050
AVERAGEIFS .....	1051
AVG .....	1052
CHOOSE_BY_MAX .....	1052
CHOOSE_BY_MIN .....	1053
COUNT .....	1053
COUNT_DISTINCT .....	1054
COUNTIF .....	1054
COUNTIFS .....	1055
GROUP_CONCAT .....	1056
GROUPCONCAT .....	1056
MAX .....	1057
MEDIAN .....	1058
MIN .....	1058
MODE .....	1059
MODEPERCENT .....	1059
PERCENTILE_CONT .....	1060
PERCENTILE_DISC .....	1061
PRODUCT .....	1061
SAMPLE .....	1062
STDEV .....	1062



STDEVP .....	1062
SUM .....	1063
SUMIF .....	1063
SUMIFS .....	1064
SUMPRODUCT .....	1065
SUMSQ .....	1065
VAR .....	1066
VARP .....	1066
WEIGHTEDAVERAGE .....	1067
Date and Time Functions .....	1068
DATE .....	1069
DATEPART .....	1070
DATETIME (or xsd:dateTime) .....	1070
DAY .....	1071
DAYSFROMDURATION .....	1071
DUR_TO_MILLIS .....	1072
DURATION .....	1072
DURATIONFORMAT .....	1072
DURATIONPERIODFORMAT .....	1073
FORMATDATE .....	1074
HOUR .....	1075
MASKEDDATETIME .....	1076
MILLIS .....	1077
MINUTE .....	1077
MONTH .....	1077
NOW .....	1078

NOWMILLIS .....	1078
PARSEDATETIME .....	1079
SECOND .....	1079
TIME .....	1080
TIMEPART .....	1080
TIMEVALUE .....	1081
TIMEZONE .....	1081
TODAY .....	1082
TZ .....	1082
WEEKDAY .....	1082
WEEKNUM .....	1083
xsd:date .....	1084
YEAR .....	1084
YEARMONTH .....	1085
Casting Functions .....	1086
BNODE .....	1087
BOOLEAN .....	1087
BYTE .....	1088
CONCATURL .....	1088
DATATYPE .....	1089
DATETIME (or xsd:dateTime) .....	1089
DATEVALUE .....	1090
DECIMAL .....	1090
DOUBLE .....	1091
DURATION .....	1091
DURATIONFORMAT .....	1091

ENCODE_FOR_URI .....	1092
FLOAT .....	1093
FORMATDATE .....	1093
FORMATFRACTION .....	1094
FORMATNUMBER .....	1095
INT .....	1095
INTEGER .....	1096
LONG .....	1096
PARSEDATETIME .....	1097
RAD .....	1097
SERIALIZE .....	1098
SHORT .....	1098
STR .....	1099
TEXT .....	1099
TIMEVALUE .....	1100
TOURI .....	1100
UUID .....	1101
xsd:date .....	1101
Logical Functions .....	1102
AND .....	1103
BOUND .....	1103
CASE .....	1104
COALESCE .....	1104
EQUAL .....	1105
IF .....	1105
IFERROR .....	1106

IN .....	1106
NOT .....	1107
NOT_EQUAL .....	1107
NOT_IN .....	1108
OR .....	1108
PARTITIONINDEX .....	1109
SAMETERM .....	1109
UNBOUND .....	1110
Informational or Testing Functions .....	1111
CONTAINS .....	1112
GE .....	1112
GT .....	1113
ISBLANK .....	1114
ISDATATYPE .....	1114
ISERROR .....	1115
ISIRI .....	1115
ISLITERAL .....	1116
ISNUMERIC .....	1116
ISURI .....	1117
LANG .....	1117
LANGMATCHES .....	1118
LE .....	1118
LOCALNAME .....	1119
LT .....	1119
METADATAGRAPHURI .....	1120
NAMESPACE .....	1121

SAMETERM .....	1121
Hash Functions .....	1123
MD5 .....	1123
SHA1 .....	1124
SHA224 .....	1124
SHA256 .....	1124
SHA384 .....	1125
SHA512 .....	1125
Window Aggregate and Ranking Functions .....	1127
WINDOW_AVG .....	1128
WINDOW_COUNT .....	1129
WINDOW_MAX .....	1131
WINDOW_MIN .....	1132
WINDOW_NTILE .....	1134
WINDOW_PERCENTILE .....	1135
WINDOW_PERCENTILE_CONT .....	1136
WINDOW_PERCENTILE_DISC .....	1136
WINDOW_PRODUCT .....	1137
WINDOW_QUARTILE .....	1139
WINDOW_SUM .....	1139
FAQ .....	1143

# Onboarding & Virtualization

The topics in this section provide instructions for onboarding unstructured data and onboarding or virtualizing structured or semi-structured data. In addition to the information about onboarding new sources, this section includes topics on migrating artifacts from one Anzo instance to another. For instructions on importing files that are in RDF format (Turtle or N-Triple), see [Creating a Dataset from RDF Files](#).

- [Onboarding Structured Data](#) ..... 19
- [Onboarding Unstructured Data](#) ..... 417
- [Versioning and Migrating Artifacts](#) ..... 463

# Onboarding Structured Data

There are three ways to onboard structured and semi-structured data to Anzo:

- [Automated ETL Pipeline Workflow](#)
- [Automated Direct Data Load Workflow](#)
- [Manual Ingestion with the Graph Data Interface](#)

## Automated ETL Pipeline Workflow

If the Spark component is installed, you can onboard data using Anzo's built-in pipelines that use an automated workflow and follow the traditional extract, transform, and load (ETL) process. When a pipeline is generated, Anzo automatically creates a data model, mappings, and the ETL jobs required to ingest the source. These ingestion pipelines natively support CSV, JSON, XML, SAS, and Parquet files, along with all common database connections, including SQL, Oracle, MySQL, HIVE, and others.

### How to get started with onboarding data via ETL pipelines

1. The first step in onboarding data using the automated ETL workflow is to connect Anzo to data sources. See [Adding Data Sources](#).
2. Then see [Ingesting Data Sources via ETL Pipelines](#) for next steps.

## Automated Direct Data Load Workflow

If Spark is not installed or you do not want to use the ETL pipeline workflow, you can use another automated workflow that follows an extract, load, and transform (ELT) process. In the ELT workflow, data sources are onboarded directly to graphmarts. Data layers with SPARQL queries are automatically generated to transform and blend the data to an analytics-ready knowledge graph. The AnzoGraph Graph Data Interface (GDI) Java plugin (sometimes called the Data Toolkit) is used to connect to the sources, create a model, and generate the data layer queries. The automated direct data load workflow supports all of the data sources that the automated ETL ingestion process supports.

### How to get started with the direct data load workflow

1. The first step in onboarding data using the automated direct data load workflow is to connect Anzo to data sources. See [Adding Data Sources](#).
2. Then see [Directly Loading Data Sources via Graphmarts](#) for next steps.

## Manual Ingestion with the Graph Data Interface

For advanced users who are familiar with SPARQL, the GDI can also be invoked by writing queries from scratch. The GDI is extremely flexible, allowing you to connect directly to sources in queries and control all aspects of the extract, load, and transform process. In addition to the data sources that the two automated workflows support, you can also onboard raw data and data from HTTP/REST endpoints with manually written GDI queries.

### How to get started onboarding data manually with GDI queries

To get started writing GDI queries for manual data onboarding, see [Onboarding or Virtualizing Data with the Graph Data Interface](#).

#### Note

For instructions on importing files that are in RDF format (Turtle or N-Triple), see [Creating a Dataset from RDF Files](#).



## Adding Data Sources

This topics in this section provide instructions for connecting to data sources, importing data, and working with schemas.

[Connecting to a Database](#)

[Defining a Database Schema](#)

[Adding a CSV Data Source](#)

[Adding a JSON Data Source](#)

[Adding an XML Data Source](#)

[Adding a SAS Data Source](#)

[Adding a Parquet Data Source](#)

[Configuring a CSV or Parquet Data Source for Incremental Processing](#)

[Assigning Primary Keys in an Onboarded Schema](#)

[Creating or Changing Foreign Keys](#)

### Connecting to a Database

Follow the instructions below to connect to a database data source, such as a Microsoft, Oracle, Hadoop, Teradata, PostgreSQL, or Google database.

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:

Data Sources   Schemas   Mappings   Pipelines										
<div> <input type="text" value="Search"/> <div> </div> </div> <div>Add Data Source</div>										
<input type="checkbox"/>	Title	↑	Description	Type	Schema	Creator	Created Date	Updated Date	Actions	
	DB			Database Data...	emrddb, new_...	System Admin...	Dec 8, 2022	Dec 8, 2022		
	Movies			CSV Data Sour...	Movies	System Admin...	Dec 8, 2022	Dec 8, 2022		
	Tickets			CSV Data Sour...	Tickets	System Admin...	Dec 9, 2022	Dec 12, 2022		
<div>Rows per page: 25 1-3 of 3</div>										

- Click the **Add Data Source** button, select **Database Data Source**, and then choose the type of database to connect to. Anzo opens the Create Database Data Source screen for the type of database that you chose. For example:

Create Oracle Database Data Source

Title \*

Description

User \*

Password \*

Server

CANCEL

SAVE

- At the top of the screen, specify a **Title** for the source, and enter an optional **Description**.
- Enter any additional details and the credentials that are required for making the source connection. The options that appear depend on the type of database connection:
  - User**: Type the user name used to log in to the database.
  - Password**: Type the password for the user.
  - Server**: Type the server name or IP address for the source. Include the port if necessary.
  - Database**: If necessary, type the partition that contains the data.
  - Extended Properties**: For Hadoop Hive or Impala databases, enter the extended attributes that you use.
- Click **Save** to save the data source connection. Anzo displays the **Tables** tab. Before proceeding to select or create a Schema, you may want to test connectivity to the database. To do so, click the **Overview** tab and then click the **Test Connectivity** button. If the connection fails, adjust the data source details until the connection is successful.

#### Note

Only the connection between Anzo and the database is tested. Connectivity is not tested between AnzoGraph and the database. If you use the direct data load workflow to onboard data directly to a graphmart, Cambridge Semantics recommends that you perform a manual

test to ensure that AnzoGraph can also connect to the database.

After connecting to the source, the next step is to add one or more schemas. See [Defining a Database Schema](#) for instructions.

## Defining a Database Schema

The schema defines the source data to onboard. There are multiple options available for defining a database schema. You can import a predefined schema from the database, you can write a static SQL query that defines the data to onboard, or, if you want to import data incrementally, you can write an incremental SQL query that ingests a subset of the data.

Click an option from the list below for instructions on creating that type of schema:

[Importing a Predefined Schema](#)

[Creating a Schema from an SQL Query](#)

[Creating an Incremental Schema](#)

## Related Topics

[Connecting to a Database](#)

## Importing a Predefined Schema

Follow the steps below to import a predefined schema from the source database to Anzo.

### Note

You can import or create up to 5 schemas per database data source. To include more than 5 schemas, create another data source for the additional schemas.

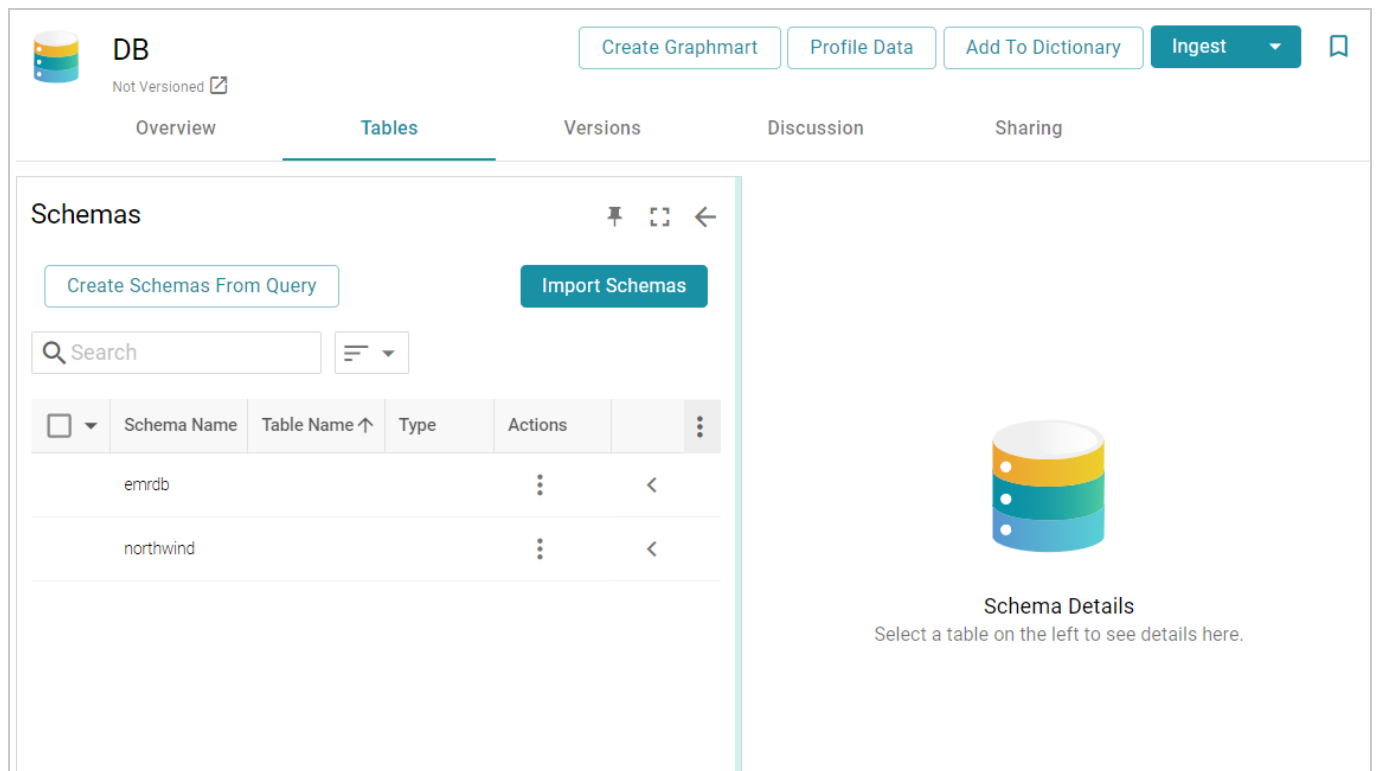
### Tip

By default, Anzo is configured to exclude Views from the list of available Schemas to import. For information about including Views as tables that can be imported, see [Including Views as Schemas for Database Data Sources](#) in the Administration Guide.

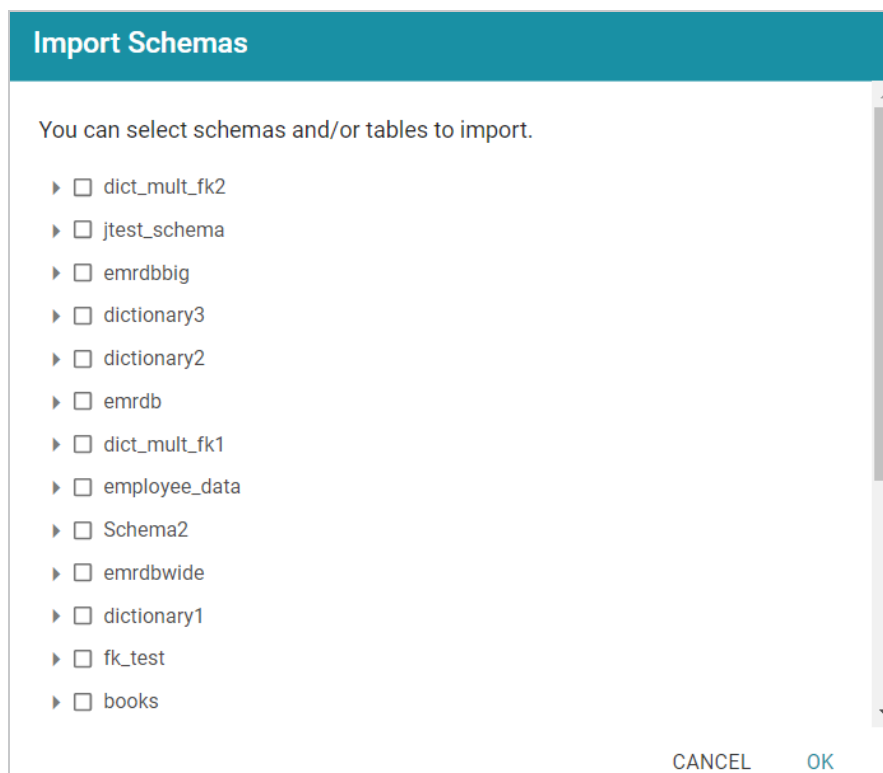
1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:

Data Sources   Schemas   Mappings   Pipelines										
	<input type="text" value="Search"/>				<button>Add Data Source</button>					
<input type="checkbox"/>	Title	↑	Description	Type	Schema	Creator	Created Date	Updated Date	Actions	
	DB			Database Data...	emrddb, new_...	System Admin...	Dec 8, 2022	Dec 8, 2022		
	Movies			CSV Data Sour...	Movies	System Admin...	Dec 8, 2022	Dec 8, 2022		
	Tickets			CSV Data Sour...	Tickets	System Admin...	Dec 9, 2022	Dec 12, 2022		
<div>Rows per page: 25 1-3 of 3 &lt; &gt;</div>										

- Click the database for which you want to create a schema. Anzo displays the Tables tab for the source. For example:



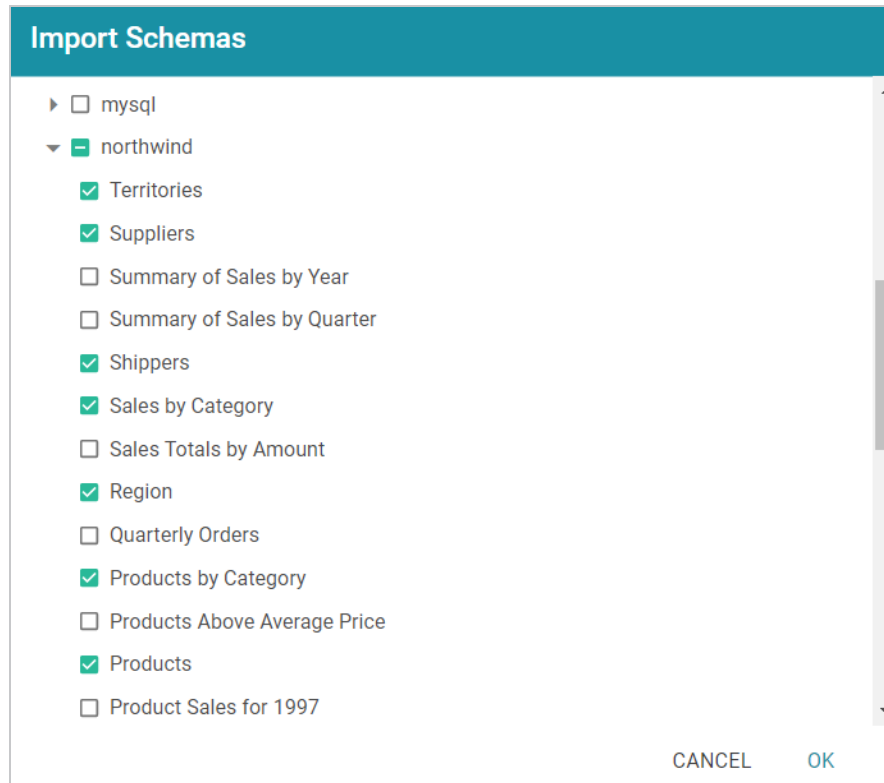
3. Click the **Import Schemas** button. Anzo displays the Import Schemas dialog box, which lists any predefined schemas in the database. For example:



### Note

If you do not see a schema that you expect to see, make sure that you have the necessary access to the data source.

- To import an entire schema, select the checkbox next to each schema that you want to import. If you want to import a subset of the tables in a schema, expand the schema and select the checkbox next to each table that you want to include. For example:



- When you have finished selecting schemas, click **OK**. Anzo imports the selected schemas and lists the imported schemas on the Tables tab. For example:

The screenshot shows the Anzo database interface. At the top, there's a 'DB' header with a 'Not Versioned' status and buttons for 'Create Graphmart', 'Profile Data', 'Add To Dictionary', and 'Ingest'. Below this are tabs for 'Overview', 'Tables' (selected), 'Versions', 'Discussion', and 'Sharing'. The 'Schemas' panel on the left contains a search bar, 'Create Schemas From Query', and 'Import Schemas' buttons. It lists four schemas: 'emrdb', 'mysql', 'new\_hires', and 'northwind'. Each schema has a table icon, a name, a 'Table Name' column with an upward arrow, and an 'Actions' column with a vertical ellipsis and a left arrow. The right panel, titled 'Schema Details', features a database icon and a message: 'Select a table on the left to see details here.'

Once the schemas are imported, they are listed on the left side of the screen. You can expand a schema to view its tables. Selecting a row in the schema displays the sample data on the right side of the screen. Now that a schema has been defined, the source data can be onboarded to Anzo. If you use the Spark ETL pipeline workflow, see [Ingesting Data Sources via ETL Pipelines](#). If you do not use Spark, see [Directly Loading Data Sources via Graphmarts](#) for next steps.

## Related Topics

[Connecting to a Database](#)

## Creating a Schema from an SQL Query

Follow the instructions below to create a schema by writing an SQL query that defines the data to onboard.

### Note

You can import or create up to 5 schemas per database data source. To include more than 5 schemas, create another data source for the additional schemas.





### Tip

For better ETL pipeline performance, it is beneficial to include joins and/or filters in schema queries rather than configuring those operations at the mapping level. For more information, see [Performance Considerations for Database ETL Pipelines](#).

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:


<



**DB**  
 Not Versioned 

Create Graphmart

Profile Data

Add To Dictionary

Ingest 






Overview

**Tables**

Versions


Discussion


Sharing





Schemas   


Create Schemas From Query

Import Schemas

 Search




<input type="checkbox"/>	Schema Name	Table Name ↑	Type	Actions	
<input type="checkbox"/>	emrdb				
<input type="checkbox"/>	northwind				



**Schema Details**  
Select a table on the left to see details here.

- Click the **Create Schemas From Query** button. Anzo displays the Create Schemas dialog box:

Create Schemas 

Schema Name \*

Table Name \*

The name of the schema table

1 Database Query \*

CANCEL

SAVE

- In the Create Schemas dialog box, specify a name for this schema in the **Schema Name** field.

5. In the **Table Name** field, specify a name for the table in the schema that the query will create.
6. Type the SQL statement in the text box. The statement can include any functionality that the source database supports. Anzo does not validate the SQL.

### Note

If the SQL query requires quotes around values, such as '2010-01-01' or 'TestValue', make sure that you use single quotes ('). For example:

```
SELECT * FROM Movies WHERE production_day='2021-08-01'
```

Including double quotes (") in a schema query results in an error when the query is run.

The following example creates a schema named employees. A table named all\_employees will be created in the schema, and the table will be created from the SQL query:

```
SELECT EmployeeID, FirstName, LastName, Title, Salary, BirthDate,
HireDate, Region, Country
FROM northwind.Employees
WHERE EmployeeID
```

**Create Schemas**

Schema Name\*  
employees

Table Name\*  
all\_employees

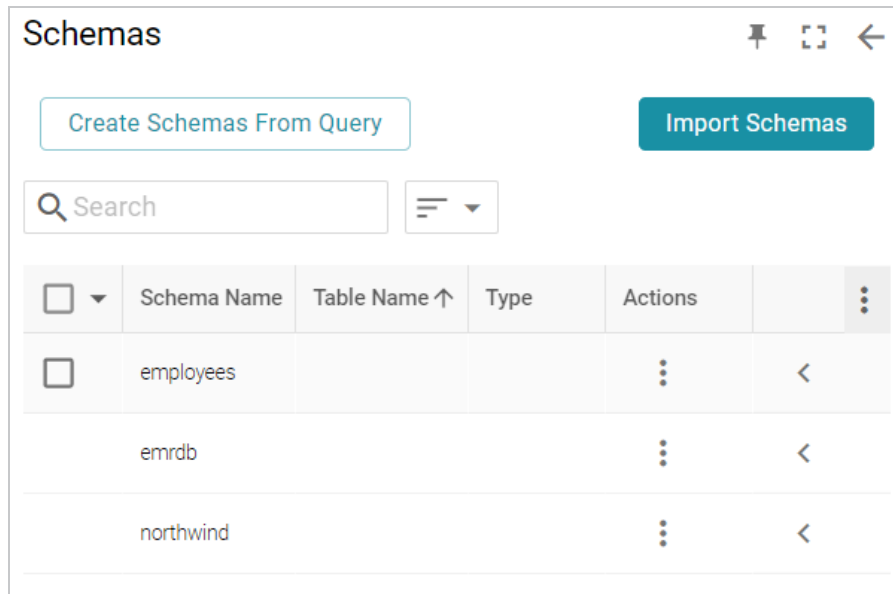
The name of the schema table

Database Query\*

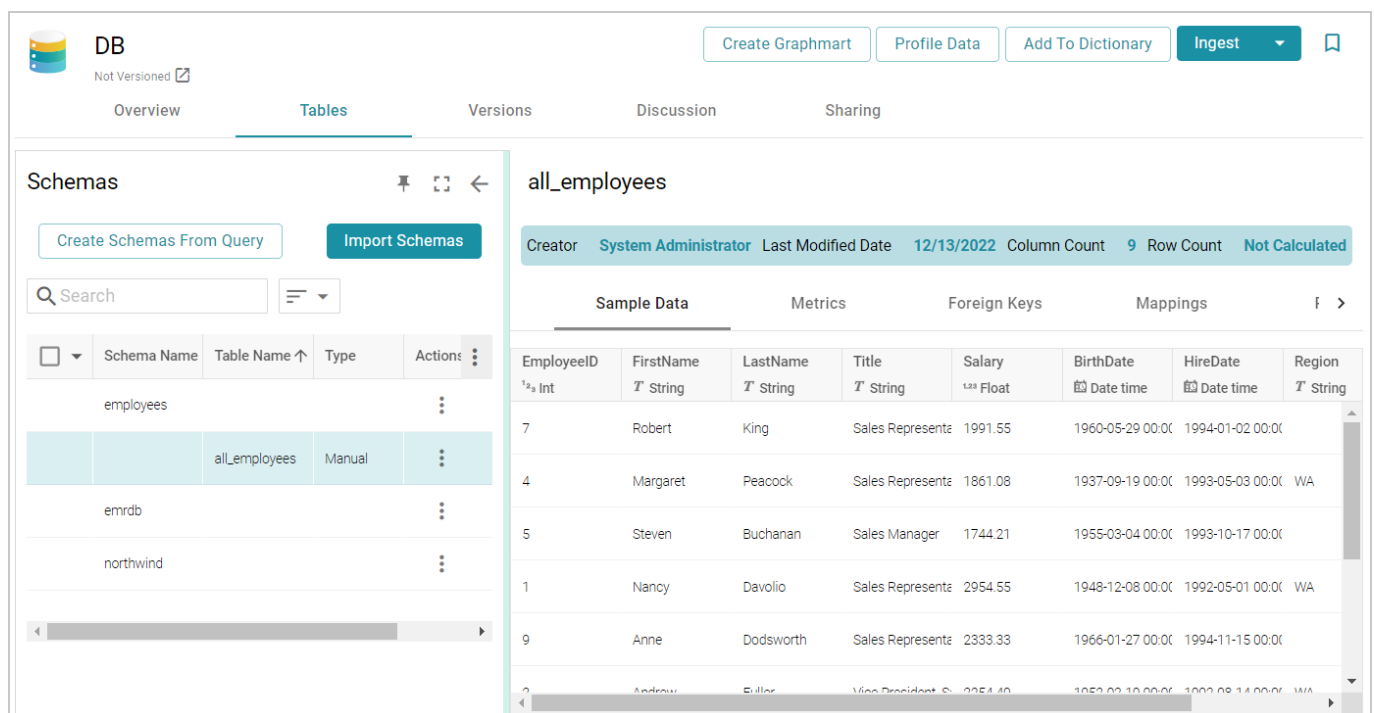
```
1 SELECT EmployeeID, FirstName, LastName, Title, Salary, BirthDate, HireDate, Region, Country
2 FROM northwind.Employees
3 WHERE EmployeeID
```

CANCEL SAVE

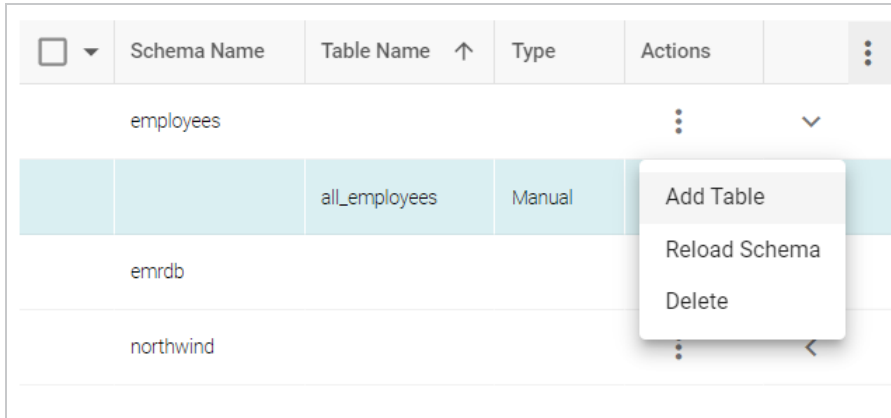
- Click **Save** to save the query. Anzo creates the new schema and adds it to the list of schemas on the Tables screen. For example:



You can expand the schema to view its tables. Selecting a row in the schema displays the sample data on the right side of the screen. For example:



8. If you want to create additional tables in the schema, follow these steps:
- Click the menu icon (⋮) in the Actions column for the schema name and select **Add Table**.



The Create New Table dialog box is displayed.

The 'Create New Table' dialog box is shown. It features a title bar with the text 'Create New Table' and a maximize button. The main area contains a 'Table Name \*' field with a placeholder 'The name of the schema table'. Below this is a large text area labeled 'Enter Query Here'. At the bottom left, there is a toggle switch for 'Include increment data', which is currently turned off. At the bottom right, there are 'CANCEL' and 'SAVE' buttons.

- In the Create New Table dialog box, specify a name for the new table in the **Table Name** field.
- In the **Schema Query** field, write the SQL query that defines the data for the new table.
- Click **Save** to add the table to the schema and return to the Tables screen.

When you are finished creating schemas, the source data can be onboarded to Anzo. If you use the Spark ETL pipeline workflow, see [Ingesting Data Sources via ETL Pipelines](#). If you do not use Spark, see [Directly Loading Data Sources via Graphmarts](#) for next steps.

## Creating an Incremental Schema

If you onboard data via Spark ETL pipelines, follow the instructions below to create a Schema by writing an SQL query that defines a subset of the data to onboard in increments. If you do not use Spark, see [Incremental Onboarding with the GDI](#) for instructions.

### Note

You can import or create up to 5 schemas per database data source. To include more than 5 schemas, create another data source for the additional schemas.

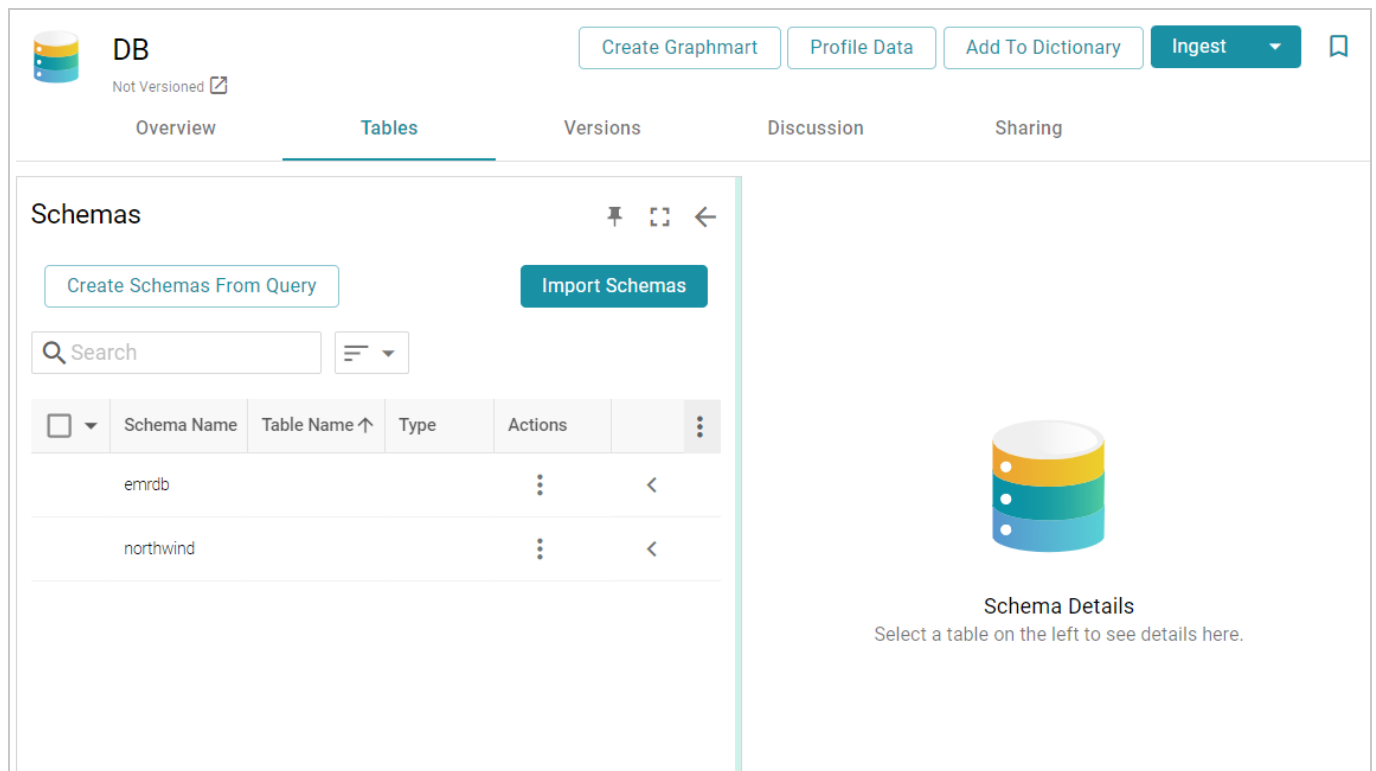
### Tip

For better ETL pipeline performance, it is beneficial to include joins and/or filters in schema queries rather than configuring those operations at the mapping level. For more information, see [Performance Considerations for Database ETL Pipelines](#).

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:

Data Sources   Schemas   Mappings   Pipelines										
<div> <div> <div> <div></div> <div>Search</div> </div> <div> <div></div> <div></div> </div> </div> <div>Add Data Source</div> </div>										
<input type="checkbox"/>	Title		Description	Type	Schema	Creator	Created Date	Updated Date	Actions	
<input type="checkbox"/>	DB			Database Data...	emrddb, new_...	System Admin...	Dec 8, 2022	Dec 8, 2022		
<input type="checkbox"/>	Movies			CSV Data Sour...	Movies	System Admin...	Dec 8, 2022	Dec 8, 2022		
<input type="checkbox"/>	Tickets			CSV Data Sour...	Tickets	System Admin...	Dec 9, 2022	Dec 12, 2022		
<div> <div>Rows per page: 25</div> <div>1-3 of 3</div> <div> <div></div> <div></div> </div> </div>										

- Click the database for which you want to create a schema. Anzo displays the Tables tab for the source. For example:



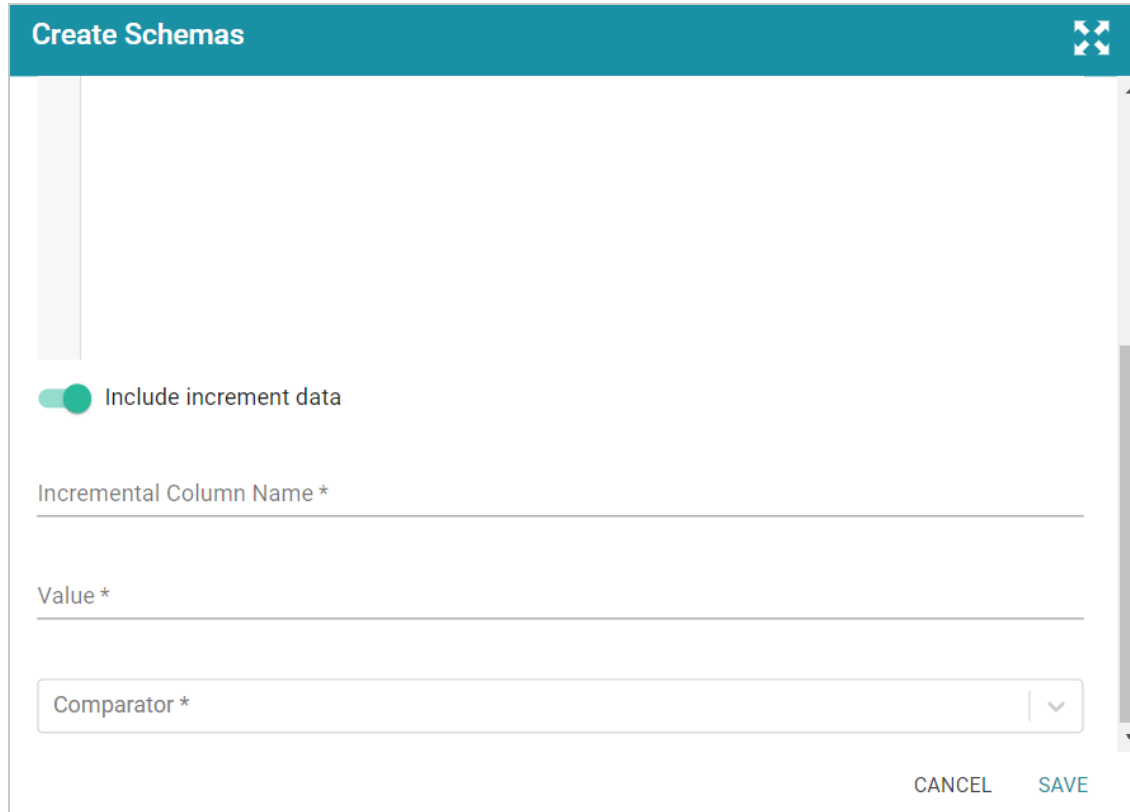
- Click the **Create Schemas From Query** button. Anzo displays the Create Schemas dialog box:

The 'Create Schemas' dialog box has a teal header with the title 'Create Schemas' and a maximize button. Below the header, there are three input fields: 'Schema Name \*', 'Table Name \*', and 'Database Query \*'. The 'Table Name \*' field has a placeholder text 'The name of the schema table'. At the bottom right, there are 'CANCEL' and 'SAVE' buttons.

- In the Create Schemas dialog box, specify a name for this schema in the **Schema Name** field.



5. In the **Table Name** field, specify a name for the table in the schema that the query will create.
6. At the bottom of the screen, enable the **Include increment data** option by sliding the slider to the right. Anzo displays additional settings. For example:



**Create Schemas**

☒ Include increment data

Incremental Column Name \*

Value \*

Comparator \*

CANCEL SAVE

7. Populate the following fields so that you can use the values as a guide for writing the schema query:
  - **Incremental Column Name:** The source column whose value will be used to increment the data.
  - **Value:** The value in the column to use as the stopping point for the first import process and the starting point for the next import.

#### Note

Do not include quote characters in the **Value** field. If the SQL query requires quotes around values, such as '2010-01-01' or 'TestValue', include the quotes around the {INCREMENTVALUE} parameter in the query and not in the Value field. For example, if the value to increment on is '2010-01-01', specify **2010-01-01** in the Value field and add the quotes to the query like the following example:

```
SELECT * FROM Orders WHERE OrderData > '{INCREMENTVALUE}'
```

In addition, make sure that you use single quotes (') in schema queries. Including double quotes (") in a schema query results in an error when the query is run.

- **Comparator:** The operator to use for comparing source values against the value above.
8. In the query text field, type the SQL statement that will target the appropriate source data. The WHERE clause must include the incremental column name, the comparison operator, and an INCREMENTVALUE parameter. This parameter is substituted with the **Value** at runtime. For example, in the query below, the incremental column name is **EmployeeID**, the comparator is **>** (greater than), and the { INCREMENTVALUE } parameter is specified after the comparator. {INCREMENTVALUE} is replaced with the value in the **Value** field at runtime:

```
SELECT EmployeeID, FirstName, LastName, Title, Salary, BirthDate,
HireDate, Region, Country
FROM northwind.Employees
WHERE EmployeeID > {INCREMENTVALUE}
```

Make sure that the query includes the INCREMENTVALUE parameter and uses the same Incremental Column Name and Comparator values as the fields below the query. For example:

Create Schemas

Table Name \*

new\_employees

The name of the schema table

Database Query \*

```
1 SELECT EmployeeID, FirstName, LastName, Title, Salary, BirthDate, HireDate, Region, Country
2 FROM northwind.Employees
3 WHERE EmployeeID > {INCREMENTVALUE}
```

☒ Include increment data

Incremental Column Name \*

EmployeeID

Value \*

5

Comparator

Greater Than

CANCEL

SAVE

- Click **Save** to save the query. Anzo creates the new schema and adds it to the list of schemas on the Tables screen. For example:

# Schemas

Create Schemas From Query

Import Schemas

<input type="checkbox"/>	Schema Name	Table Name ↑	Type	Actions	
	emrdb				<
<input type="checkbox"/>	new_hires				▼
		new_employees	Manual		
	northwind				<

You can expand the schema to view its tables. Selecting a row in the schema displays the sample data on the right side of the screen.

DB

Not Versioned

Create Graphmart

Profile Data

Add To Dictionary

Ingest

Overview

Tables

Versions

Discussion

Sharing

Schemas

Create Schemas From Query

Import Schemas

<input type="checkbox"/>	Schema Name	Table Name ↑	Type	Actions	
	emrdb				<
	new_hires				▼
		new_employees	Manual		
	northwind				<

new\_employees

Creator

System Administrator

Last Modified Date

12/13/2022

Column Count

9

Row Count

Not Calculated

Sample Data

Metrics

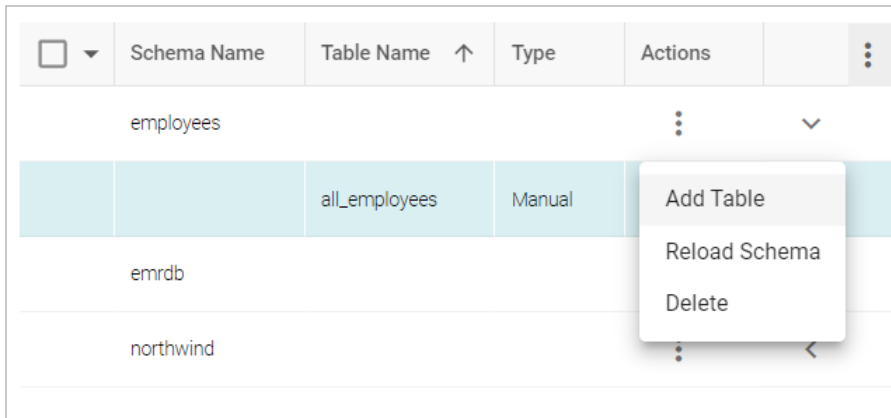
Foreign Keys

Mappings

Pipe

EmployeeID	FirstName	LastName	Title	Salary	BirthDate	HireDate	Region
<sup>Int</sup>	<sup>T</sup> String	<sup>T</sup> String	<sup>T</sup> String	<sup>Float</sup>	<sup>Date time</sup>	<sup>Date time</sup>	<sup>String</sup>
13	FIRST NAME	Test	HR		1955-05-07 00:00	1993-10-17 00:00	WA
8	Laura	Callahan	Inside Sales Coor	2100.5	1958-01-09 00:00	1994-03-05 00:00	WA
11	sebastian	ken	Inside Sales Coor	2100.5	1958-01-09 00:00	1994-01-02 00:00	WA
12	Mathew	John	Sales	2100.5	1955-05-07 00:00	1993-10-17 00:00	
6	Michael	Suyama	Sales Represents	2004.07	1963-07-02 00:00	1993-10-17 00:00	
7	Robert	King	Sales Represents	1991.55	1960-05-29 00:00	1994-01-02 00:00	
14	First Test	Test2	HR2		2022-07-28 00:00	2022-05-28 00:00	

10. If you want to create additional tables in the schema, follow these steps:
- Click the menu icon (⋮) in the Actions column for the schema name and select **Add Table**.



The Create New Table dialog box is displayed.

A screenshot of the 'Create New Table' dialog box. It has a teal header with the title 'Create New Table' and a maximize button. The main area contains a 'Table Name \*' field with a placeholder 'The name of the schema table'. Below it is a large text area labeled 'Enter Query Here'. At the bottom left, there is a toggle switch for 'Include increment data'. At the bottom right, there are 'CANCEL' and 'SAVE' buttons.

- In the Create New Table dialog box, specify a name for the new table in the **Table Name** field.
- In the **Schema Query** field, write the SQL query that defines the data for the new table.
- Click **Save** to add the table to the schema and return to the Tables screen.

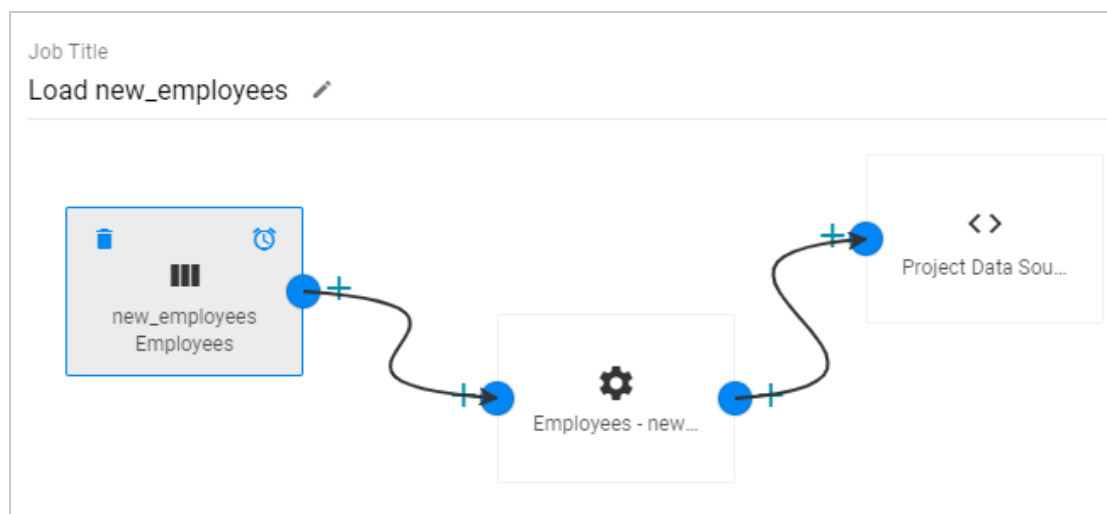
When you are finished creating schemas, the source data can be onboarded to Anzo. If you use the Spark ETL pipeline workflow, see [Ingesting Data Sources via ETL Pipelines](#). If you do not use Spark, see [Directly Loading Data Sources via Graphmarts](#) for next steps.

#### Note

See [Incremental Pipeline Reference](#) for important information about running a pipeline that includes an incremental schema.

## Incremental Pipeline Reference

When an incremental schema is added to an ETL job, a clock icon (🕒) is displayed when hovering over the component in the job. For example:



Clicking the clock icon opens the Incremental Load dialog box, which lists the Incremental Column Name, Value, and Comparator from the schema query. For example:

**Incremental Load**

Incremental Column Name

EmployeeID

Value \*

5

Comparator

Greater Than

CANCEL

SAVE

Publishing the job for this example will onboard only the records for which the **EmployeeID** is greater than **5**. When the job is finished, Anzo adjusts the incremental load value to list the last value that was onboarded for the incremental column. Every time the pipeline is published, Anzo changes the incremental load value parameter to the highest or lowest value for the column, depending on the Comparator.

For example, viewing the Incremental Load dialog box after running the job above shows that the last EmployeeID value that was onboarded was 9:

**Incremental Load**

Incremental Column Name

EmployeeID

Value \*

9

Comparator

Greater Than

CANCEL

SAVE

The next time this job is run, Anzo will onboard only the records where **EmployeeID** is greater than **9**. To view the number of rows processed after running a job, you can search the System Datasource for the following predicate on the Find tab in the Query Builder:

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#rowsProcessed>
```

The Object column shows the number of rows processed each time the pipeline was run. For example:

Query

Find

Select a datasource:

System Datasource

Subject

Predicate

Object

Graph

<http://cambridgesemantics.com/or

Clear

Add Statements

Find

Results (1)

☒ Show native

☒ Subject

☒ Predicate

☒ Object

☒ Named Graph

EDIT

DELETE

<input type="checkbox"/>	Subject ↓	Predicate	Object	Named Graph
<input type="checkbox"/>	« http://cambridgesemantics.com/Project/0a5a4d73-8606-5c6c-6ac1-9c303768c7e1/a322404f-2f1b-fa9f-83b7-57f1ca9355ce/8cd892b7-b97e-f948-9ae4-479d3f4ef0fc/Job/d725475d-0a8b-ab69-5a8a-7690dd714163/ff74f93b-5a48-4c60-9eca-226202ef2f23 »	« http://cambridgesemantics.com/ontologies/2015/08/SDIService#rowsProcessed »	« 9 »	http://cambridgesemantics.com/Project/0a5a4d73-8606-5c6c-6ac1-9c303768c7e1/a322404f-2f1b-fa9f-83b7-57f1ca9355ce/8cd892b7-b97e-f948-9ae4-479d3f4ef0fc/Job/d725475d-0a8b-ab69-5a8a-7690dd714163/JobExecData

Rows per page: 50

## Related Topics

[Creating an Incremental Schema](#)

## Performance Considerations for Database ETL Pipelines

This topic highlights performance-related information that is helpful to consider when setting up an onboarding pipeline for a database data source.

### Take Advantage of the Source Database

Onboarding data from a database involves two systems, the source database and the Spark infrastructure. The way that you configure the pipeline's schema and mappings controls which system performs some of the time-consuming operations such as joining and filtering the data. In short, schema operations are processed by the source database, and mapping transformations are processed by Spark. Maximizing the use of the source database to join and filter data can have a significant impact on the overall performance of the ETL pipeline.



## Use Schema Queries to Join and Filter Data

When defining the schema for a database source, you have the option to write SQL queries to create the schema tables. If join and/or filter operations are required, consider writing schema queries that perform those operations (see [Creating a Schema from an SQL Query](#) for more information). Since the source database runs the schema queries and then sends the filtered result set to Spark, Spark has fewer operations to perform when publishing the ETL pipeline.

Alternatively, if the schema selects all of the source data and joins or filters are configured at the mapping level, the source database sends the entire result set to Spark and Spark performs the join and filter operations when publishing the pipeline.

In general, databases perform join and filter operations much faster than Spark. And Cambridge Semantics recommends that you incorporate joins and filters in schema queries when possible, rather than transforming the data downstream in the mappings that Spark processes.

### Related Topics

[Connecting to a Database](#)

[Defining a Database Schema](#)

[Partitioning a Database Table for Parallel Ingestion](#)

### Partitioning a Database Table for Parallel Ingestion

When you ingest data from a database, Anzo creates one ETL job for each table in the schema. When there are multiple jobs in a pipeline, Spark processes the jobs in parallel, one job per executor. If the source has a very large table, however, and one job ingests all of the data for that table, overall pipeline performance can slow down because one Spark executor processes all of the data from that table. To take advantage of parallel ingestion if a data source has one or more large tables, you can use Anzo Semantic Service calls to partition the tables. The resulting ETL job for a partitioned table has smaller sections that can be ingested in parallel by multiple executors.

This topic provides instructions on using the Anzo command line interface to compute a partition and assign the partition to a table so that Anzo can leverage the information during ingestion.

#### Tip

When a pipeline is configured to use the Sparkler ETL engine to compile jobs, Sparkler automatically attempts to partition RDBMS tables if the table has a primary column that is an integer data type and a data source profile has been generated (as described in [Generating a Source Data Profile](#)). Sparkler can also be configured to attempt to partition tables without requiring a data profile. For

more information, see [Configuring a Sparkler Engine](#) in the Administration Guide.

## Computing and Assigning Partitions to a Table

When creating a partition for a table, choose a column with an integer data type to partition on. You add metadata to that column to define the size and number of partitions, and then you call an Anzo service that computes the predicates for the partition. Once the predicates are computed, you call another service to assign the partitions to the table so that Anzo can apply the partitions when generating the ETL job. The steps below guide you through computing and assigning partitions.

### Note

When you supply the metadata for computing partitions, you will need to know the row count for the table that will be partitioned. Calculating the row count in Anzo requires generating statistics on the schema. You might want to generate statistics in advance before starting the steps below. For instructions, see [Generating a Source Data Profile](#).

1. First, view the metadata for the data source so that you can retrieve the URI for the schema that contains the table to partition. Run the following command to return the data source metadata:

```
anzo get data_source_uri
```

The data source URI can be found on the Overview tab for the data source.

The screenshot shows the Anzo DB interface. At the top, there's a header with a database icon, the name 'DB', and a 'Not Versioned' status. Action buttons include 'Create Graphmart', 'Profile Data', 'Add To Dictionary', and 'Ingest'. Below the header is a navigation bar with tabs: 'Overview' (selected), 'Tables', 'Versions', 'Discussion', and 'Sharing'. The main content area is divided into two columns. The left column contains metadata: Description (None), User (sysadmin), Password (masked with asterisks), Server (10.115.0.23), and Database (None). A 'Test Connection' button is at the bottom of this column. The right column is titled 'General' and contains a table with the following data:

General	
Type	Database
Creator	System Administrator
Updated	Dec 8, 2022 4:50 PM
Released	Dec 8, 2022 4:50 PM

Below the table, there is a highlighted box containing the URL: <http://cambridgesemantics.com/MySQL> with a copy icon.

For example:

```
anzo get
http://cambridgesemantics.com/MySQLDatabaseDataSource/690c96c79b0c4383bc9
08310e6fdb3f
```

Anzo returns the metadata for the data source.

2. In the data source metadata results, look for the schema URI for which you want to create a partition. The schema URI is the object of a triple that follows the pattern below:

```
data_source_uri
<http://cambridgesemantics.com/ontologies/DataSources#dbSchema> schema_
uri
```

For example, the URI below identifies the **northwind** schema:

```
data_source_uri
<http://cambridgesemantics.com/ontologies/DataSources#dbSchema>
<
```

```
http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763d  
ffefaab4/Schema/northwind>
```

3. Using the schema URI from the previous step, run the following command to view the metadata for the schema:

```
anzo get schema_uri
```

For example:

```
anzo get  
http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763d  
ffefaab4/Schema/northwind
```

Anzo returns the metadata for the schema.

4. In the schema metadata results, find the URI for the table that you want to partition. The table URI is the object of a triple that follows the pattern below:

```
schema_uri  
<http://cambridgesemantics.com/ontologies/DataSources#schemaTable> table_  
uri
```

For example, the URI below identifies the **ORDERS** table in the northwind schema:

```
schema_uri  
<http://cambridgesemantics.com/ontologies/DataSources#schemaTable>  
<  
http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763d  
ffefaab4/Schema/northwind/ORDERS>
```

5. Next, identify the URI for the column that you want to use for computing the partitions. The column that you choose should have an integer data type. You can view the column URIs as well as metadata for the columns in the output of the previous step, or you can run the following command to narrow the results to the list of columns for the table. This command finds all of the results for which the table URI is the subject:

```
anzo find -sub table_uri
```

For example:

```
anzo find -sub
http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763d
ffefaab4/Schema/northwind/ORDERS
```

The column URIs are the object of a triple that follows the pattern below:

```
table_uri
<http://cambridgesemantics.com/ontologies/DataSources#tableColumn>
column_uri
```

For example, the URI below identifies the ORDERID column in the ORDERS table:

```
table_uri
<http://cambridgesemantics.com/ontologies/DataSources#tableColumn>
<
http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763d
ffefaab4/Schema/northwind/ORDERS/ORDERID>
```

6. Once you retrieve the column URI, create a .trig file that includes the metadata for the column. You will add new partition properties to the file. Run the following command to output a .trig file that contains the column metadata:

```
anzo find -sub column_uri --output-file /path/filename.trig
```

For example, the following command retrieves all of the results for which ORDERID is the subject. It outputs the results to a file called ComputePartitions.trig in the current directory:

```
anzo find -sub
http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763d
ffefaab4/Schema/northwind/ORDERS/ORDERID
--output-file ComputePartitions.trig
```

The output below shows the contents of the resulting ComputePartitions.trig file.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```

<http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763
dffefaab4/Schema/northwind> {

<http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763
dffefaab4/Schema/northwind/ORDERS/ORDERID>
    a <http://cambridgesemantics.com/ontologies/DataSources#Column>
    <http://cambridgesemantics.com/ontologies/DataSources#DataField> ;

<http://cambridgesemantics.com/ontologies/DataSources#columnAutoIncrement>
t> "false"^^xsd:boolean ;

<http://cambridgesemantics.com/ontologies/DataSources#columnCaseSensitive>
e> "false"^^xsd:boolean ;

<http://cambridgesemantics.com/ontologies/DataSources#columnDerivedOwlProperty>
    <http://cambridgesemantics.com/ont/autogen/Fu/DB/northwind#Orders
OrderID> ;
    <http://cambridgesemantics.com/ontologies/DataSources#columnIndex>
"1"^^xsd:int ;
    <http://cambridgesemantics.com/ontologies/DataSources#columnJdbcType>
"integer" ;
    <http://cambridgesemantics.com/ontologies/DataSources#columnName>
"OrderID" ;
    <http://cambridgesemantics.com/ontologies/DataSources#columnNullable>
"false"^^xsd:boolean ;
    <http://cambridgesemantics.com/ontologies/DataSources#columnPrimaryKey>
"true"^^xsd:boolean ;
    <http://cambridgesemantics.com/ontologies/DataSources#columnRemarks> ""
;
    <http://cambridgesemantics.com/ontologies/DataSources#columnSize>
"10"^^xsd:int ;
    <http://cambridgesemantics.com/ontologies/DataSources#columnType>
xsd:int ;
    <http://cambridgesemantics.com/ontologies/DataSources#columnTypeName>
"INT" .
}

```

7. Modify the .trig file from the previous step to specify the partitioning metadata. The metadata to add includes the number of partitions to create as well as the total number of rows in the data source table. To provide the required metadata, edit the file as follows:
- At the top of the file, replace the schema URI with the following service URI:

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#ComputePartitioningPredicatesRequest>
```

In the example above,

```
<http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763dffefaab4/Schema/northwind> is replaced by  
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#ComputePartitioningPredicatesRequest>
```

- Towards the bottom of the file, at the end of the column metadata and inside the ending brace ( } ), add the following contents:

```
# PARTITIONING METADATA
```

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#ComputePartitioningPredicatesRequest>
```

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#numberOfPartitions> "number_of_partitions"^^<http://www.w3.org/2001/XMLSchema#int> ;
```

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#numberOfRows> "number_of_rows"^^<http://www.w3.org/2001/XMLSchema#long> ;
```

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#tableColumn> column_uri ;
```

a

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#BaseComputePartitioningPredicatesRequest> ,
```

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#ComputeColumnBasedPartitioningRequest> .
```

- c. In the new triples, replace the placeholders with the appropriate values for your environment:
- **number\_of\_partitions**: Specify the number of partitions to create for the table. Choose the value based on the number of Spark nodes or executors that are available. If you do not know the number, **12** is recommended.
  - **number\_of\_rows**: Specify the total number of rows for the table. After generating source data metrics, you can view the row count by viewing the Tables tab for the schema and clicking the table to show the metrics for that table. For example:

The screenshot shows the Databricks UI with the 'Tables' tab selected. On the left, a list of schemas is shown, including 'Employees', 'EmployeeTer...', 'Invoices', 'Order Details', and 'Order Details ...'. The 'Invoices' table is selected. On the right, the 'Invoices' table details are shown, including a table with sample data and metrics. The 'Row Count' is highlighted as 2155.

Creator	System Administrator	Last Modified Date	12/08/2022	Column Count	26	Row Count	2155
Sample Data							
ShipName	ShipAddress	ShipCity	ShipRegion	ShipPostal...	ShipCountry	CustomerID	Custor
T String	T String	T String	T String	T String	T String	T String	T Str
Alfred-s Futterkis	Obere Str. 57	Berlin		12209	Germany	ALFKI	Alfr
Alfred-s Futterkis	Obere Str. 57	Berlin		12209	Germany	ALFKI	Alfr
Ana Trujillo Empe	Avda. de la Const	Mxico D.F.		5021	Mexico	ANATR	Ana
Alfreds Futterkist	Obere Str. 57	Berlin		12209	Germany	ALFKI	Alfr
Alfreds Futterkist	Obere Str. 57	Berlin		12209	Germany	ALFKI	Alfr
Alfred-s Futterkis	Obere Str. 57	Berlin		12209	Germany	ALFKI	Alfr

- **column\_uri**: The URI for the partition column from step 5. You can copy the URI from the top of the file.

The example below shows the complete ComputePartitions.trig file after completing steps a, b, and c.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#Compute
PartitioningPredicatesRequest> {
```

```
<http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a135414087
1b763dffefaab4/Schema/northwind/ORDERS/ORDERID>
```

```
    a <http://cambridgesemantics.com/ontologies/DataSources#Column
    <http://cambridgesemantics.com/ontologies/DataSources#DataField> ;
```



```

<http://cambridgesemantics.com/ontologies/DataSources#columnAutoIncrement> "false"^^xsd:boolean ;

<http://cambridgesemantics.com/ontologies/DataSources#columnCaseSensitive> "false"^^xsd:boolean ;

<http://cambridgesemantics.com/ontologies/DataSources#columnDerivedOWLProperty>
    <http://cambridgesemantics.com/ont/autogen/Fu/DB/northwind#OrderID> ;
    <http://cambridgesemantics.com/ontologies/DataSources#columnIndex>
        "1"^^xsd:int ;

<http://cambridgesemantics.com/ontologies/DataSources#columnJdbcType>
    "integer" ;
    <http://cambridgesemantics.com/ontologies/DataSources#columnName>
        "OrderID" ;

<http://cambridgesemantics.com/ontologies/DataSources#columnNullable>
    "false"^^xsd:boolean ;

<http://cambridgesemantics.com/ontologies/DataSources#columnPrimaryKey>
    "true"^^xsd:boolean ;

<http://cambridgesemantics.com/ontologies/DataSources#columnRemarks>
    "" ;
    <http://cambridgesemantics.com/ontologies/DataSources#columnSize>
        "10"^^xsd:int ;
    <http://cambridgesemantics.com/ontologies/DataSources#columnType>
        xsd:int ;

<http://cambridgesemantics.com/ontologies/DataSources#columnTypeName>
    "INT" .
# PARTITIONING METADATA

<http://cambridgesemantics.com/ontologies/2015/08/SDIService#ComputePartitioningPredicatesRequest>

```

```

<http://cambridgesemantics.com/ontologies/2015/08/SDIService#numberOfPartitions> "12"^^<http://www.w3.org/2001/XMLSchema#int> ;

<http://cambridgesemantics.com/ontologies/2015/08/SDIService#numberOfRows> "830"^^<http://www.w3.org/2001/XMLSchema#long> ;

<http://cambridgesemantics.com/ontologies/2015/08/SDIService#tableColumn>

<http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763dffefaab4/Schema/northwind/ORDERS/ORDERID> ;
a
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#BaseComputePartitioningPredicatesRequest> ,

<http://cambridgesemantics.com/ontologies/2015/08/SDIService#ComputeColumnBasedPartitioningRequest> .}

```

8. When the .trig file is complete, save and close the file. It becomes input to the Anzo Compute Column Based Table Partitioning Predicates service. The service returns the response to use to assign the partitions that can be used during ingestion. Run the following command to call the partitioning service:

```

anzo call
http://cambridgesemantics.com/semanticServices/SDIService#computeColumnBasedTablePartitioningPredicates /path/filename.trig > /path/output_file.trig

```

Where **filename.trig** is the file from step 7 and **output\_file.trig** is the new file to create. For example, the following command calls the partitioning service and saves the response in a file called **AssignPartitions.trig** in the current directory.

```

anzo call
http://cambridgesemantics.com/semanticServices/SDIService#computeColumnBasedTablePartitioningPredicates ComputePartitions.trig >
AssignPartitions.trig

```

The service returns the list of partition predicates. The number of predicates depends on the number of partitions that were specified in the compute file. For example, a portion of the resulting AssignPartitions.trig file is shown below. You can see the complete file by clicking [here](#).

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://cambridgesemantics.com/ontologies/2015/08/SDIService#ComputePartitioningPredicatesResponse> {
  _:u0d4a3eab-713f-4dbe-b5ad-da5676d6b721 a
  <http://cambridgesemantics.com/ontologies/DataSources#PartitionPredicate>
  ;
    <http://cambridgesemantics.com/ontologies/DataSources#value>
    "<http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763dffefaab4/Schema/northwind/ORDERS/ORDERID> >= 69
      &&
    <http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763dffefaab4/Schema/northwind/ORDERS/ORDERID> < 138" .

  _:u255f046c-73a1-44ef-b446-c5a6126c9cc1 a
  <http://cambridgesemantics.com/ontologies/DataSources#PartitionPredicate>
  ;
    <http://cambridgesemantics.com/ontologies/DataSources#value>
    "<http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763dffefaab4/Schema/northwind/ORDERS/ORDERID> >= 414
      &&
    <http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763dffefaab4/Schema/northwind/ORDERS/ORDERID> < 483" .
  ...
}
```

9. Modify the .trig file from the previous step to specify the metadata that the Anzo Assign Table Partitioning Predicates service will use to assign the partitions. To provide the required metadata, edit the file as follows:
  - a. At the top of the file, replace the ComputePartitioningPredicatesResponse URI with the following Assign service URI:

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#AssignTablePartitioningPredicatesRequest>
```

In the example above,

`<http://cambridgesemantics.com/ontologies/2015/08/SDIService#ComputePartitioningPredicatesResponse>` is replaced by

`<http://cambridgesemantics.com/ontologies/2015/08/SDIService#AssignTablePartitioningPredicatesRequest>`

- b. At the bottom of the file inside the ending brace ( `}` ), locate the following triple pattern:

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#ComputePartitioningPredicatesResponse>
```

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#partitioningPredicates>
```

```
list_of_predicate_uris .
```

Where `list_of_predicate_uris` is a comma-separated list of all of the predicate URIs from the file. For example, this is the relevant statement from the `AssignPartitions.trig` file shown above:

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#ComputePartitioningPredicatesResponse>
```

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#partitioningPredicates> _:u0d4a3eab-713f-4dbe-b5ad-da5676d6b721 ,
      _:u255f046c-73a1-44ef-b446-c5a6126c9cc1 , _:u2aa40e92-e60d-4b11-9d75-24e84c91ec06 ,
      _:u3472d783-ce53-4aa3-acc2-a9e57d3f318b ,
      _:u3bdcalac-be4e-4374-b8b9-ce22c18c179a , _:u3eea38d2-8f40-4cb4-b297-8378d68d90e6 ,
      _:u5752b0a1-4262-403a-a029-8a2f54a18f2f ,
      _:u60d4dea2-f06f-45d5-87bf-242e295494ff , _:u702c4eed-2531-4578-a7e7-5ea4120e86ce ,
      _:ub35b741b-6020-4f7d-9a71-7d6c17adf9c9 ,
      _:ub4a0d80b-7e17-4d88-8648-e5c05cea2069 , _:ub8b7ef03-3bbc-41c4-ba43-360bc69620a0 .
```

- c. Like the substep a above, replace the `ComputePartitioningPredicatesResponse` URI with the `Assign` service URI.

In the example above,

`<http://cambridgesemantics.com/ontologies/2015/08/SDIService#ComputePartitioningPredicatesResponse>` is replaced by

`<http://cambridgesemantics.com/ontologies/2015/08/SDIService#AssignTablePartitioningPredicatesRequest>`

- d. At the end of the list of predicate URIs, change the period (.) to a semicolon (;), and then add the following new statements after the semicolon :

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#tableURI> table_uri ;  
a  
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#AssignTablePartitioningPredicatesRequest> .
```

Where **table\_uri** is the table URI from step 4. For example:

```
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#tableURI>  
  
<http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763dffefaab4/Schema/northwind/ORDERS> ;  
a  
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#AssignTablePartitioningPredicatesRequest> .
```

For example, the end of the AssignPartitions.trig file now looks like this:

```
...  
  
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#AssignTablePartitioningPredicatesRequest>  
  
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#partitioningPredicates> _:u0d4a3eab-713f-4dbe-b5ad-da5676d6b721 ,  
    _:u255f046c-73a1-44ef-b446-c5a6126c9cc1 , _:u2aa40e92-e60d-4b11-9d75-24e84c91ec06 ,  
    _:u3472d783-ce53-4aa3-acc2-a9e57d3f318b ,  
    _:u3bdca1ac-be4e-4374-b8b9-ce22c18c179a , _:u3eea38d2-8f40-4cb4-b297-8378d68d90e6 ,  
    _:u5752b0a1-4262-403a-a029-8a2f54a18f2f ,
```

```

    _:u60d4dea2-f06f-45d5-87bf-242e295494ff , _:u702c4eed-2531-4578-
a7e7-5ea4120e86ce , _:ub35b741b-6020-4f7d-9a71-7d6c17adf9c9 ,
    _:ub4a0d80b-7e17-4d88-8648-e5c05cea2069 , _:ub8b7ef03-3bbc-41c4-
ba43-360bc69620a0 ;

<http://cambridgesemantics.com/ontologies/2015/08/SDIService#tableUR
I>

<http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a135414087
1b763dffefaab4/Schema/northwind/ORDERS> ;

    a
<http://cambridgesemantics.com/ontologies/2015/08/SDIService#AssignT
ablePartitioningPredicatesRequest> .
}

```

If you would like to view the complete sample file, click [here](#).

10. When the .trig file is complete, save and close the file. It becomes input to the Assign Table Partitioning Predicates service. The service assigns the partitions to the data source to inform the ingestion process. Run the following command to call the assigning service:

```

anzo call
http://cambridgesemantics.com/semanticServices/SDIService#assignTablePart
itioningPredicates filename.trig

```

Where **filename.trig** is the file you edited in the previous step. For example:

```

anzo call
http://cambridgesemantics.com/semanticServices/SDIService#assignTablePart
itioningPredicates AssignPartitions.trig

```

When the prompt returns, the process is complete. If you view the metadata for the table that was partitioned (e.g., run `anzo find -sub table_uri`), the metadata contains a new `<http://cambridgesemantics.com/ontologies/DataSources#tablePredicates>` URI that lists the partition predicates.

Once the partitioning is complete, the source data can be onboarded to Anzo. For instructions on onboarding the data, see [Ingesting Data Sources via ETL Pipelines](#).

## Related Topics

[Connecting to a Database](#)

[Defining a Database Schema](#)

[Generating a Source Data Profile](#)

## Adding a CSV Data Source

Follow the instructions below to add a CSV data source and import the data from the files.















### Note

If your CSV Data Source is consistently updated with new or changed files, you can configure the Data Source to process the data incrementally. For details, see [Configuring a CSV or Parquet Data Source for Incremental Processing](#).

### Tip

For information about updating a CSV Data Source if a file changes, see [How do I update Anzo if a file in my CSV data source changes?](#)

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:

Data Sources   Schemas   Mappings   Pipelines										
	<input type="text" value="Search"/>				<button>Add Data Source</button>					
	Title	↑	Description	Type	Schema	Creator	Created Date	Updated Date	Actions	
	DB			Database Data...	emrdb, new_...	System Admin...	Dec 8, 2022	Dec 8, 2022		
	Movies			CSV Data Sour...	Movies	System Admin...	Dec 8, 2022	Dec 8, 2022		
	Tickets			CSV Data Sour...	Tickets	System Admin...	Dec 9, 2022	Dec 12, 2022		
Rows per page: 25   1-3 of 3   <   >										

2. Click the **Add Data Source** button and select **File Data Source > CSV Data Source**. Anzo opens the Create CSV Data Source screen.

### Create CSV Data Source

Title \*

Description

CANCEL

SAVE

3. Specify a name for the source in the **Title** field, and type an optional description in the **Description** field. Then click **Save**. Anzo saves the source and displays the Tables tab.

CSV

Movies

Not Versioned

Create Graphmart

Profile Data

Add To Dictionary

Ingest

Overview

Tables

Versions

Discussion

Sharing

CSV

Please add new files to get started.

Add New File



4. Click the **Add New File** button. Anzo displays the Add New File dialog box.

Add New File

Source


☒ From Your Computer ☐ From File Store

Upload to

sysadmin User Folder

Change

The location where your uploaded files will be put on the Anzo Server. Defaults to your user folder.



Drag and drop files or [browse](#) from computer.

Please select the .csv files to import

Selected: None

CLEAR ALL

CANCEL

NEXT

5. Follow the appropriate steps below depending on whether the CSV files are on your computer or the shared File Store:

**If the files are on your computer:**

**Note**

The **From Your Computer** option is a convenient way to do a one-time ingestion so you can quickly get started with your data. It should not be relied upon as part of a regular ingestion workflow unless the server is configured to store uploaded files on the shared File Store. For more information, see [Setting the Default Base File Store Path for File Uploads](#) in the Administration Guide. Data source files that are routinely updated and re-ingested should be hosted on a configured File Store.

- a. As a best practice, check the upload location that is listed in the **Upload To** field by hovering your pointer over the value to view the tooltip. Make sure the upload location is a directory on the shared file store and not in the server installation path. If the file is not uploaded to the shared file

store it is not accessible by applications like AnzoGraph. In addition, other users cannot create graphmarts from the data source because they typically do not have access to the file location. For example, viewing the Upload To location for the screen above shows that the file will be uploaded to the server installation path, `/opt/Anzo/Server/data...`

The screenshot shows a configuration window with two radio buttons under the 'Source' heading: 'From Your Computer' (selected) and 'From File Store'. Below this is the 'Upload to' section, which contains a text box with the value 'sysadmin User Folder' and a 'Change' button to its right. A tooltip is visible over the text box, displaying the path `/opt/Anzo/Server/data/userUploads/sysadmin_20235` and the text '... on the Anzo Server. Defaults to your user folder.'

If your Upload To location is configured to upload the file to the server installation path, click **Change** and select an upload location that is on the shared file store. For example, the image below shows the Upload Folder Location dialog box that is presented after clicking **Change**. A folder called **fileUploads** is selected on the shared store.

The 'Upload Folder Location' dialog box features a teal header. Below the header, there is a 'Current Folder' field with a back arrow icon, a forward slash, and a 'GO' button. The main area is divided into two panes. The left pane shows 'Server Shared Filesystem' (highlighted in light blue) and 'sysadmin User Folder'. The right pane is a list of folders: 'docs', 'fileUploads' (highlighted in teal), 'flights', 'fmcsa', and 'ghih10'. At the top of the right pane, there is a 'Selected:' field showing '/fileUploads/' with a close icon and a 'CLEAR ALL' button. At the bottom of the dialog, there are three buttons: 'CREATE NEW FOLDER', 'CANCEL', and 'OK'.

- b. Drag and drop the files onto the screen or click **browse** to navigate to the files and select them. Anzo attaches the files and the **Next** button becomes active.

- c. Click **Next**. Anzo lists the uploaded files on the left side of the screen with a status of Pending. For example:

Overview

Tables

Versions

Discussion

Sharing

Schemas

Add New File

Process Pending Files

Search

<div></div>	Title	Status	Actions
	MovieActors	<div>Pending</div>	<div></div>
	MovieComposers	<div>Pending</div>	<div></div>
	MovieDirectors	<div>Pending</div>	<div></div>
	MovieEditors	<div>Pending</div>	<div></div>
	MovieProducers	<div>Pending</div>	<div></div>
	Movies	<div>Pending</div>	<div></div>

Rows per page: 1-6 of 6

CSV

Schema Details

Please process the files to view them.

If the files are on the File Store:

- a. Click the **From File Store** radio button. Anzo displays the file selection dialog box.

**Add New File**

Source

☐ From Your Computer ☒ From File Store

Current Folder  
↑ /movies/ GO ☐ Show All File Types

☒ Insert Wildcard ☐ Use Incremental Processing  
Ex: \*.json, \*north\*.json APPLY

Selected: None CLEAR ALL

Server Shared Filesystem

sysadmin User Folder

- MovieActors.csv
- MovieActors2.csv
- MovieComposers.csv
- MovieDirectors.csv
- MovieEditors.csv
- MovieProducers.csv

CANCEL NEXT

- b. On the left side of the screen, select the File Store that hosts the CSV files. On the right side of the screen, navigate to the directory that contains the files to import. The screen displays the list of files in the directory.
- c. Select each file that you want to import. If you have multiple files with the same schema—the files contain the same columns listed in the same order—and you want the files to be imported as if they are a **single file**, you can select the **Insert Wildcard** option. Then type a string using asterisks as wildcard characters to find the files with similar names. Files that match the specified string will be imported as **one file** and will result in one job being created in the pipeline to ingest all of the files that are selected by the specified string. After typing a string, click **Apply** to include that string in the Selected list.

## Example

The image below shows a directory with several CSV files. For this example, **part.csv** and **partsupp.csv** have the same schema and can be imported as one file. The **Insert Wildcard** option is selected, and **part\*.csv** is specified to identify the two files.

☒ Insert Wildcard ☐ Use Incremental Processing

Ex: \*.json, \*north\*.json APPLY

Selected: part\*.csv supplier.csv region.csv CLEAR ALL

Server Filesystem

sysadmin User Folder

part.csv

partsupp.csv

region.csv

supplier.csv

- d. When you finish selecting files, click **Next** to close the dialog box. Anzo lists the uploaded files on the left side of the screen with a status of Pending. For example:

Overview

Tables

Versions

Discussion

Sharing

Schemas

Add New File

Process Pending Files

Search

<input type="checkbox"/>	Title	Status	Actions
<input type="checkbox"/>	MovieActors	Pending	
<input type="checkbox"/>	MovieComposers	Pending	
<input type="checkbox"/>	MovieDirectors	Pending	
<input type="checkbox"/>	MovieEditors	Pending	
<input type="checkbox"/>	MovieProducers	Pending	
<input type="checkbox"/>	Movies	Pending	

Rows per page: 1-6 of 6

CSV

Schema Details

Please process the files to view them.

6. If you do not need to change CSV file options, click the **Process Pending Files** button to import all of the pending files. Anzo imports the data and updates the status to Processed.

If you do need to change CSV file options, click the menu icon (⋮) for that file and select **Edit**. To change the options for multiple files, select the checkbox next to each file, and then click the **Edit** button at the bottom of the table. Anzo displays the Edit CSV File screen. For example, the image below shows the Edit screen for a single file:

**Edit CSV File**

Title  
MovieActors

File Path  
/movies/MovieActors.csv [BROWSE](#)

☒ Contains Header ☐ Use Extended Sample ☐ Infer All Columns as Strings

Delimiter

Row Delimiter

Csv Escape Character


Char Set

Quote Character

CANCEL SAVE & IMPORT

Change the options as needed and then click **Save & Import** to import the file or files. Anzo imports the data and updates the status to Processed.

7. Once the files are processed, you can click a table row on the left side of the screen to display the schema on the right side of the screen. For example:



## Movies

Not Versioned

[Create Graphmart](#)
[Profile Data](#)
[Add To Dictionary](#)
[Ingest](#)

Overview
**Tables**
Versions
Discussion
Sharing

### Schemas

[Add New File](#)

Title	Status	Actions
MovieActors	Processed	⋮
MovieComposers	Processed	⋮
MovieDirectors	Processed	⋮
MovieEditors	Processed	⋮
MovieProducers	Processed	⋮
Movies	Processed	⋮

Rows per page: 1-6 of 6

### MovieActors

**Creator** System Administrator **Last Modified Date** 12/13/2022 **Column Count** 4 **Row Count** Not Calculated

Sample Data
Metrics
Foreign Keys
Mappings >

MovieID <small>I Int</small>	MovieTitle <small>T String</small>	ActorID <small>I Int</small>	ActorName <small>T String</small>
30271424	Alvin and the Chipmunks (film se	4995125	Matthew Gray Gubler
30271424	Alvin and the Chipmunks (film se	876916	Kaley Cuoco
30271424	Alvin and the Chipmunks (film se	1582246	Kimberly Williams-Paisley
15400287	Mission: Impossible (film series)	31460	Tom Cruise
30271424	Alvin and the Chipmunks (film se	921466	Tony Hale
15400287	Mission: Impossible (film series)	689567	Ving Rhames
30271424	Alvin and the Chipmunks (film se	2647835	Zachary Levi

**Tip**

For information about assigning primary keys and creating foreign keys, see [Assigning Primary Keys in an Onboarded Schema](#) and [Creating or Changing Foreign Keys](#).

The source data can now be onboarded to Anzo. If you use the Spark ETL pipeline workflow, see [Ingesting Data Sources via ETL Pipelines](#). If you do not use Spark, see [Directly Loading Data Sources via Graphmarts](#) for next steps.

## Related Topics

## Configuring a CSV or Parquet Data Source for Incremental Processing

## Adding a JSON Data Source

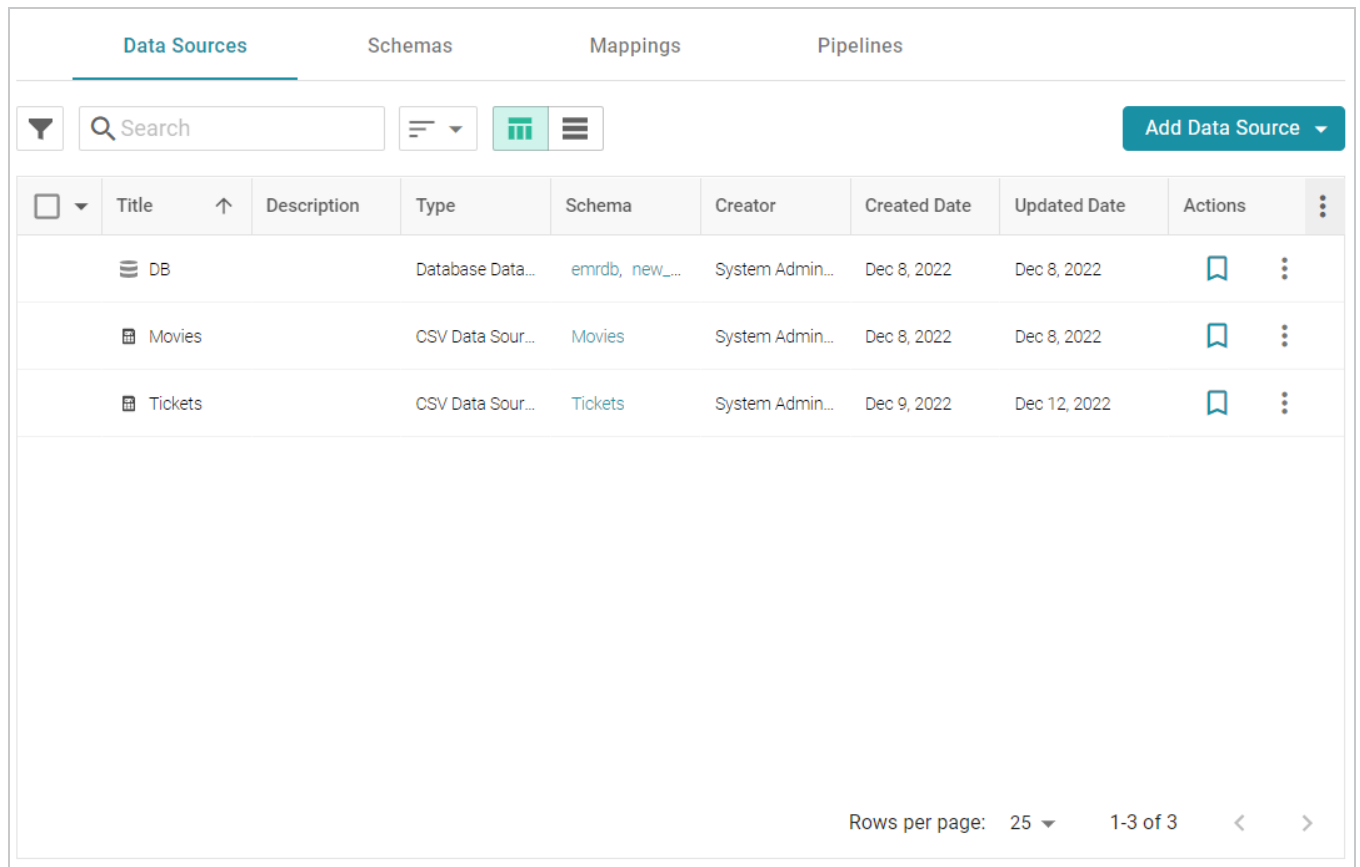
This topic provides instructions for adding a JSON data source, scanning the file, and generating the schema.

### Note

When a large amount of data is ingested from a single JSON file, the resulting ETL pipeline can take an extremely long time to complete because a single job is created. Since it is a single job, a single ETL engine node processes the data while other resources remain idle. The best approach to loading a large data set in JSON format is to divide the data into several smaller files and then import

the batch of files. The resulting pipeline has several smaller jobs that can be processed in parallel.

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:



The screenshot shows the 'Data Sources' tab in the Anzo application. At the top, there are tabs for 'Data Sources', 'Schemas', 'Mappings', and 'Pipelines'. Below the tabs is a search bar with a magnifying glass icon and the text 'Search'. To the right of the search bar are two icons: a green bar chart and a grey list icon. Further right is a blue button labeled 'Add Data Source' with a dropdown arrow. Below these elements is a table with the following columns: Title, Description, Type, Schema, Creator, Created Date, Updated Date, and Actions. The table contains three rows of data sources. At the bottom right of the table, there is a pagination control showing 'Rows per page: 25' and '1-3 of 3' with navigation arrows.

	Title	Description	Type	Schema	Creator	Created Date	Updated Date	Actions
	DB		Database Data...	emrdb, new_...	System Admin...	Dec 8, 2022	Dec 8, 2022	
	Movies		CSV Data Sour...	Movies	System Admin...	Dec 8, 2022	Dec 8, 2022	
	Tickets		CSV Data Sour...	Tickets	System Admin...	Dec 9, 2022	Dec 12, 2022	

2. Click the **Add Data Source** button and select **File Data Source** > **JSON Data Source**. Anzo opens the Create JSON Data Source screen.



### Create JSON Data Source

Title \*

Description

JSON File Location \*

BROWSE

Schema Type

Flatx▼

Root Element Name

Advanced▼

CANCEL

SAVE & EXTRACT SCHEMA

3. Specify a name for the data source in the **Title** field, and type an optional description in the **Description** field.

4. Click the **JSON File Location** field to open the File Location dialog box.

### File Location

Source


☒ From Your Computer ☐ From File Store

Upload to

sysadmin User Folder

Change

The location where your uploaded files will be put on the Anzo Server. Defaults to your user folder.



Drag and drop files or [browse](#) from computer.

Please select the .json file to import

Selected: None

CLEAR ALL

CREATE NEW FOLDER

CANCEL

OK

5. Follow the appropriate steps below depending on whether the file is on your computer or the shared File Store:

**If the file is on your computer:**

#### Note

The **From Your Computer** option is a convenient way to do a one-time ingestion so you can quickly get started with your data. It should not be relied upon as part of a regular ingestion workflow unless the server is configured to store uploaded files on the shared File Store. For more information, see [Setting the Default Base File Store Path for File Uploads](#) in the Administration Guide. Data source files that are routinely updated and re-ingested should be hosted on a configured File Store.

- a. As a best practice, check the upload location that is listed in the **Upload To** field by hovering your pointer over the value to view the tooltip. Make sure the upload location is a directory on the

shared file store and not in the server installation path. If the file is not uploaded to the shared file store it is not accessible by applications like AnzoGraph. In addition, other users cannot create graphmarts from the data source because they typically do not have access to the file location.

For example, viewing the Upload To location for the screen above shows that the file will be uploaded to the server installation path, `/opt/Anzo/Server/data...`

Source

☒ From Your Computer ☐ From File Store

Upload to

sysadmin User Folder Change

`/opt/Anzo/Server/data/userUploads/sysadmin_20235`

If your Upload To location is configured to upload the file to the server installation path, click **Change** and select an upload location that is on the shared file store. For example, the image below shows the Upload Folder Location dialog box that is presented after clicking **Change**. A folder called **fileUploads** is selected on the shared store.

Upload Folder Location

Current Folder: `/` GO

Selected: `/fileUploads/` CLEAR ALL

Server Shared Filesystem

- docs
- fileUploads**
- flights
- fmcsa
- rh1h10

sysadmin User Folder

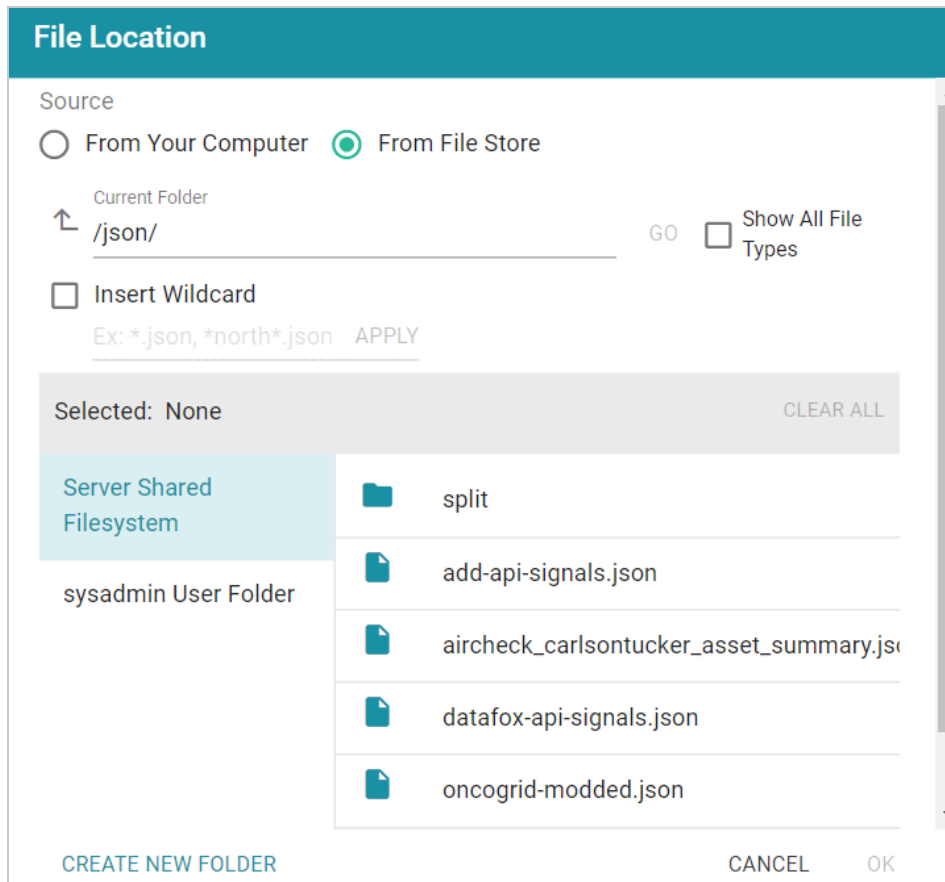
CREATE NEW FOLDER CANCEL OK

- Drag and drop the file onto the screen or click **browse** to navigate to the file and select it. Anzo attaches the file and the **OK** button becomes active.

- c. Click **OK**. Anzo lists the path to the file in the JSON File Location field.

**If the file is on the File Store:**

- a. Click the **From File Store** radio button.
- b. In the File Location dialog box, on the left side of the screen, select the appropriate File Store. On the right side of the screen, navigate to the directory that contains the file to import. The screen displays the list of files in the directory. For example:



- c. Select the file that you want to import and then click **OK** to close the dialog box. If you have multiple files with the same schema—the files contain the same arrays in the same order—you can select the **Insert Wildcard** option. Then type a string using asterisks as wildcard characters to find the files with similar names. Files that match the specified string will be imported as **one file** and will result in one job being created in the pipeline to ingest all of the files that are selected by the specified string. You can specify up to 16,000 files using a wildcard. After typing a string, click **Apply** to include that string in the Selected list.

### Important

If you have a batch of files that were generated to split a large data set, do NOT specify the batch of files using the wildcard syntax. Select one file from the batch. You will select the rest of the files in a later step. Selecting all files with a wildcard essentially merges the data into one large file, resulting in one ETL job that would be processed by limited ETL engine resources rather than multiple jobs that could be processed in parallel.

### Example

The image below shows a directory with multiple JSON files. For this example, **add-api-signals.json** and **datafox-api-signals.json** have the same schema and can be imported as one file. The **Insert Wildcard** option is selected, and **\*signals.json** is specified to identify the two files.

The screenshot shows the 'Source' configuration window in Anzo. The 'From File Store' radio button is selected. The 'Current Folder' is '/json/'. The 'Insert Wildcard' checkbox is checked, and the example text 'Ex: \*.json, \*north\*.json' is shown. Below this, a list of files is displayed under the 'Server Shared Filesystem' section. The files are: 'split' (folder), 'add-api-signals.json', 'aircheck\_carlsontucker\_asset\_summary.json', and 'datafox-api-signals.json'. The 'Selected' field shows '\*signals.json' with a 'CLEAR ALL' button next to it.

6. Specify the type of schema that Anzo should create. Click the **Schema Type** field and select one of the following types from the drop-down list:
  - **Flat:** By default, the Schema Type is set to **Flat**. A flat schema type results in a single schema table with a single mapping file and ETL job. Generating a flat schema is ideal for files with many different objects with nested relationships where there are many one-to-one relationships. If the file contains a large number of arrays or a number of arrays that are large in size, however, generating a flat schema is not recommended. The import can require extensive server resources and take a long time to process.

### Note

In Flat mode, Anzo creates relationships that go from the parent node to the child node. For example: Person → Address.

- **Relational:** A relational schema type results in multiple schema tables, mappings, and jobs. Generating a relational schema is ideal for files that include many arrays or a number of very large arrays. Creating a relational schema from a file that contains many different objects with one-to-one relationships can result in poor import performance and a very large number of small tables, mappings, and ETL jobs.

### Note

In Relational mode, Anzo creates relationships that go from the child node to the parent node. For example: Address → Person.

Anzo performs pre-processing before creating the schema. If the specified Schema Type would result in poor performance or require extensive resources, Anzo displays a warning and prompts you to change the schema type before proceeding with the schema creation.

7. When data is onboarded, Anzo sets the root object name to "json." If you want to specify an alternate name for this source, type the new name in the **Root Element Name** field.
8. If you are importing a batch of files or want to configure other advanced options, expand the **Advanced** section of the screen and proceed to the next steps.

Advanced

Schema File Location [BROWSE](#)

Scan Depth (-1 means scan complete file) \*

100

Repeating Element Paths

Add Part JSON File locations [BROWSE](#)

9. The **Schema File Location** field defines where Anzo saves the generated schema. Cambridge Semantics recommends that you leave the field blank. If you want to designate a custom location, click **Browse** and choose a file location.

10. The value in the **Scan Depth** field indicates the number of entities in the file that Anzo should scan to find all of the unique objects to include as classes and properties in the generated model. The scan process follows nested objects, counting one object array as one row. Edit the value as needed. A value of **-1** instructs Anzo to scan the entire file.
11. If the JSON file contains lists of objects that are not defined in arrays, the file scan cannot determine if any of the objects are the same type, and Anzo treats each object as a new type. To ensure that repeating object paths are treated as the same type if the file does not include arrays, use standard JSON path syntax to define repeating element types in the **Repeating Element Paths** field. Separate paths with semicolons (;). If the file includes arrays, leave this field blank.

For example, when Anzo scans the following sample JSON markup, people, vehicles, and maintenance would become object types without a defined relationship:

```
{
  "people":
  {
    "personal": {
      "age": 20,
      "gender": "M",
      "name": {
        "first": "John",
        "last": "Doe"
      }
    },
    "vehicles": {
      "type": "car",
      "model": "Honda Civic",
      "insurance": {
        "company": "ABC Insurance",
        "policy_num": 12345
      },
      "maintenance":
      {
        "date": "07-17-2017",
        "desc": "oil change"
      },
      {
        "date": "01-03-2018",
```

```

        "desc": "new tires"
    }
}
}
}

```

By defining the following paths in the Repeating Element Paths field, the scan knows that "people" is an object type, "vehicles" map to people, and "maintenance" is related to vehicles, which map to people.

```
$.people;$.people.vehicles;$.people.vehicles.maintenance
```

12. If you are importing a batch of files, click the **Add Part JSON File Locations** field to open the File Location dialog box. Select each of the files included in the batch and then click **OK** to close the dialog box.
13. Click **Save & Extract Schema** to scan the file and generate the schema. Anzo saves the data source, creates the schema, and displays the Tables tab. You can click a table row on the left side of the screen to display the Schema on the right side of the screen. For example:

The screenshot shows the Datafox interface with the 'article' schema selected. The left sidebar lists schemas: article, company, json, mentions, participant\_lists, and snippets. The main area displays the 'article' schema details, including a table of sample data.

Creator	System Administrator	Last Modified Date	12/13/2022	Column Count	8	Row Count	Not Calculated
Sample Data							
id	object_type	published	source	sources_id	title	url	
1	news	2018-01-03 18:00	Business Wire - C	6	Prosper Marketpl	http	
7	news	2017-09-22 16:00	Business Wire - T	51	Prosper Announc	http	
3	news	2017-09-22 16:00	Business Wire - T	45	Prosper Announc	http	
8	news	2017-08-14 15:00	Business Wire - F	66	Prosper Reports	http	
5	news	2017-09-22 16:00	Business Wire - T	49	Prosper Announc	http	

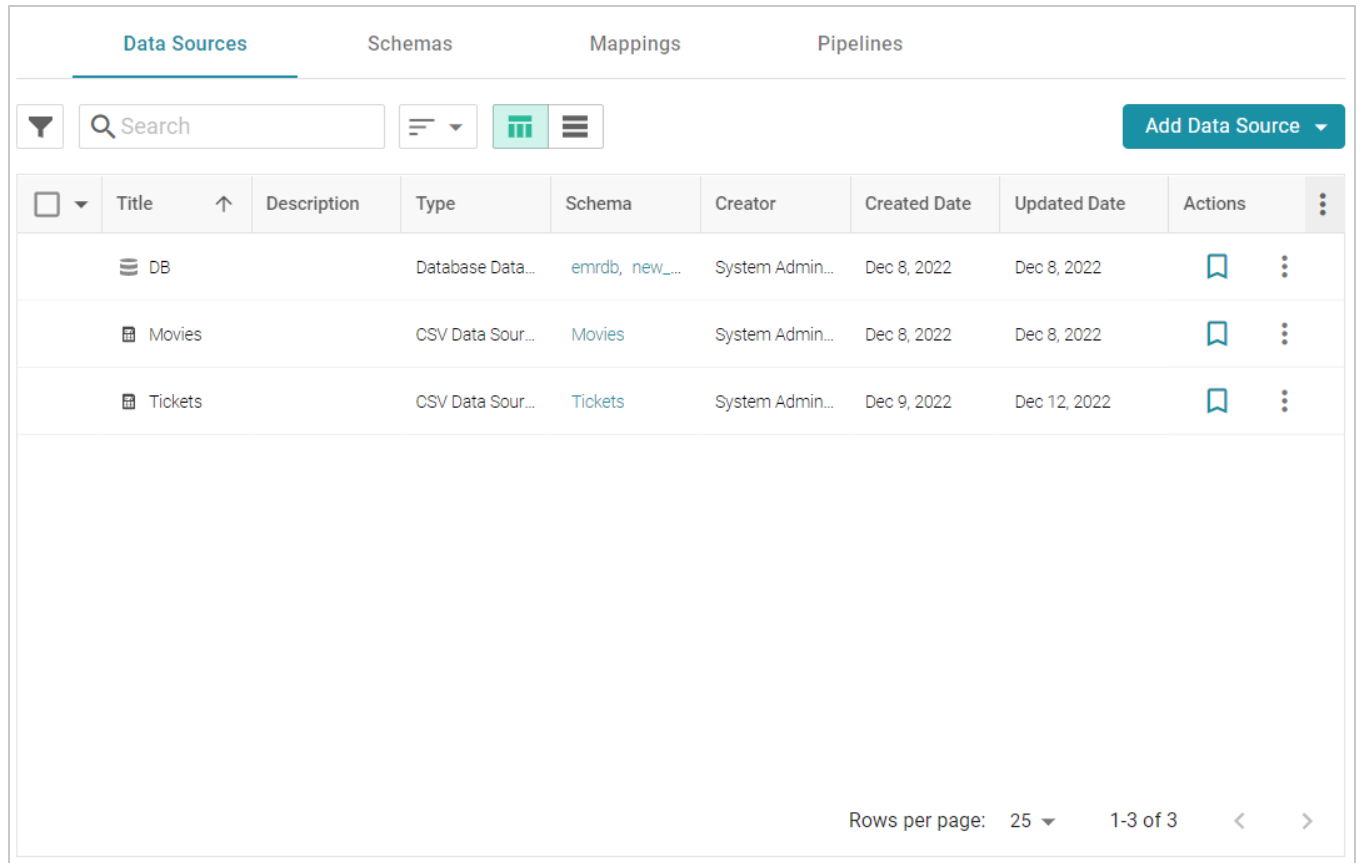


The source data can now be onboarded to Anzo. If you use the Spark ETL pipeline workflow, see [Ingesting Data Sources via ETL Pipelines](#). If you do not use Spark, see [Directly Loading Data Sources via Graphmarts](#) for next steps.

## Adding an XML Data Source

This topic provides instructions for adding an XML Data Source, scanning the file, and generating the Schema.

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:



The screenshot shows the 'Data Sources' tab in the Anzo application. At the top, there are tabs for 'Data Sources', 'Schemas', 'Mappings', and 'Pipelines'. Below the tabs is a search bar with a magnifying glass icon and the text 'Search'. To the right of the search bar are two icons: a green grid icon and a black list icon. Further right is a blue button labeled 'Add Data Source' with a dropdown arrow. Below these elements is a table with the following columns: 'Title', 'Description', 'Type', 'Schema', 'Creator', 'Created Date', 'Updated Date', 'Actions', and a dropdown arrow. The table contains three rows of data:

Title	Description	Type	Schema	Creator	Created Date	Updated Date	Actions
DB		Database Data...	emrdb, new_...	System Admin...	Dec 8, 2022	Dec 8, 2022	
Movies		CSV Data Sour...	Movies	System Admin...	Dec 8, 2022	Dec 8, 2022	
Tickets		CSV Data Sour...	Tickets	System Admin...	Dec 9, 2022	Dec 12, 2022	

At the bottom right of the table, there is a pagination bar that says 'Rows per page: 25' with a dropdown arrow, followed by '1-3 of 3' and navigation arrows.

2. Click the **Add Data Source** button and select **File Data Source > XML Data Source**. Anzo opens the Create XML Data Source screen.

### Create XML Data Source

Title \*

Description

XML File Location \*

BROWSE

Schema Type

Flat

×

▼

Schema File Location

BROWSE

Scan Depth (-1 means scan complete file) \*

100

Repeating Element Paths

CANCEL

SAVE & EXTRACT SCHEMA

3. Specify a name for the data source in the **Title** field, and type an optional description in the **Description** field.

4. Click the **XML File Location** field to open the File Location dialog box.

File Location

Source


☒ From Your Computer ☐ From File Store

Upload to

sysadmin User Folder

Change

The location where your uploaded files will be put on the Anzo Server. Defaults to your user folder.



Drag and drop files or [browse](#) from computer.  
Please select the .xml file to import

Selected: None

CLEAR ALL

CREATE NEW FOLDER

CANCEL

OK

5. Follow the appropriate steps below depending on whether the file is on your computer or the shared File Store:

**If the file is on your computer:**

#### Note

The **From Your Computer** option is a convenient way to do a one-time ingestion so you can quickly get started with your data. It should not be relied upon as part of a regular ingestion workflow unless the server is configured to store uploaded files on the shared File Store. For more information, see [Setting the Default Base File Store Path for File Uploads](#) in the Administration Guide. Data source files that are routinely updated and re-ingested should be hosted on a configured File Store.

- a. As a best practice, check the upload location that is listed in the **Upload To** field by hovering your pointer over the value to view the tooltip. Make sure the upload location is a directory on the shared file store and not in the server installation path. If the file is not uploaded to the shared file

store it is not accessible by applications like AnzoGraph. In addition, other users cannot create graphmarts from the data source because they typically do not have access to the file location. For example, viewing the Upload To location for the screen above shows that the file will be uploaded to the server installation path, `/opt/Anzo/Server/data...`

The screenshot shows a configuration window with two radio buttons under the 'Source' label: 'From Your Computer' (selected) and 'From File Store'. Below this is the 'Upload to' section, which contains a text box with 'sysadmin User Folder' and a 'Change' button to its right. A tooltip is visible over the text box, displaying the path `/opt/Anzo/Server/data/userUploads/sysadmin_20235` and the text '... on the Anzo Server. Defaults to your user folder.'

If your Upload To location is configured to upload the file to the server installation path, click **Change** and select an upload location that is on the shared file store. For example, the image below shows the Upload Folder Location dialog box that is presented after clicking **Change**. A folder called **fileUploads** is selected on the shared store.

The 'Upload Folder Location' dialog box has a teal header. Below the header, there is a 'Current Folder' field with a back arrow icon, a forward slash '/', and a 'GO' button. A 'Selected:' field shows `/fileUploads/` with a close icon and a 'CLEAR ALL' button. The main area is divided into two panes. The left pane, titled 'Server Shared Filesystem', lists 'sysadmin User Folder'. The right pane shows a list of folders: 'docs', 'fileUploads' (highlighted in teal), 'flights', 'fmcsa', and 'nhih10'. At the bottom, there are three buttons: 'CREATE NEW FOLDER', 'CANCEL', and 'OK'.

- Drag and drop the file onto the screen or click **browse** to navigate to the file and select it. Anzo attaches the file and the **OK** button becomes active.
- Click **OK**. Anzo lists the path to the file in the XML File Location field.

### If the file is on the File Store:

- a. Click the **From File Store** radio button.
- b. In the File Location dialog box, on the left side of the screen, select the appropriate File Store. On the right side of the screen, navigate to the directory that contains the file to import. The screen displays the list of files in the directory. For example:

**File Location**

Source

☐ From Your Computer ☒ From File Store

Current Folder  
↑ /xml/ GO ☐ Show All File Types

☐ Insert Wildcard  
Ex: \*.json, \*north\*.json APPLY

Selected: None CLEAR ALL

Server Shared Filesystem

sysadmin User Folder

people.xml

sample.xml

test.xml

CREATE NEW FOLDER CANCEL OK

- c. Select the file that you want to import and then click **OK** to close the dialog box. Anzo lists the path to the file in the XML File Location field.

If you have multiple files with the same schema—the files contain the same elements in the same order—and you want the files to be imported as if they are a **single file**, you can select the **Insert Wildcard** option. Then type a string using asterisks as wildcard characters to find the files with similar names. Files that match the specified string will be imported as **one file** and will result in one job being created in the pipeline to ingest all of the files that are selected by the specified string. After typing a string, click **Apply** to include that string in the Selected list.

6. Specify the type of schema that Anzo should create. Click the **Schema Type** field and select one of the following types from the drop-down list:

- **Flat:** By default, the Schema Type is set to **Flat**. A flat schema type results in a single schema table with a single mapping file and ETL job. Generating a flat schema is ideal for files with many different objects with nested relationships where there are many one-to-one relationships. If the file contains a large number of arrays or a number of arrays that are large in size, however, generating a flat schema is not recommended. The import can require extensive server resources and take a long time to process.

**Note**

In Flat mode, Anzo creates relationships that go from the parent node to the child node. For example: Person → Address.

- **Relational:** A relational schema type results in multiple schema tables, mappings, and jobs. Generating a relational schema is ideal for files that include many arrays or a number of very large arrays. Creating a relational schema from a file that contains many different objects with one-to-one relationships can result in poor import performance and a very large number of small tables, mappings, and ETL jobs.

**Note**

In Relational mode, Anzo creates relationships that go from the child node to the parent node. For example: Address → Person.

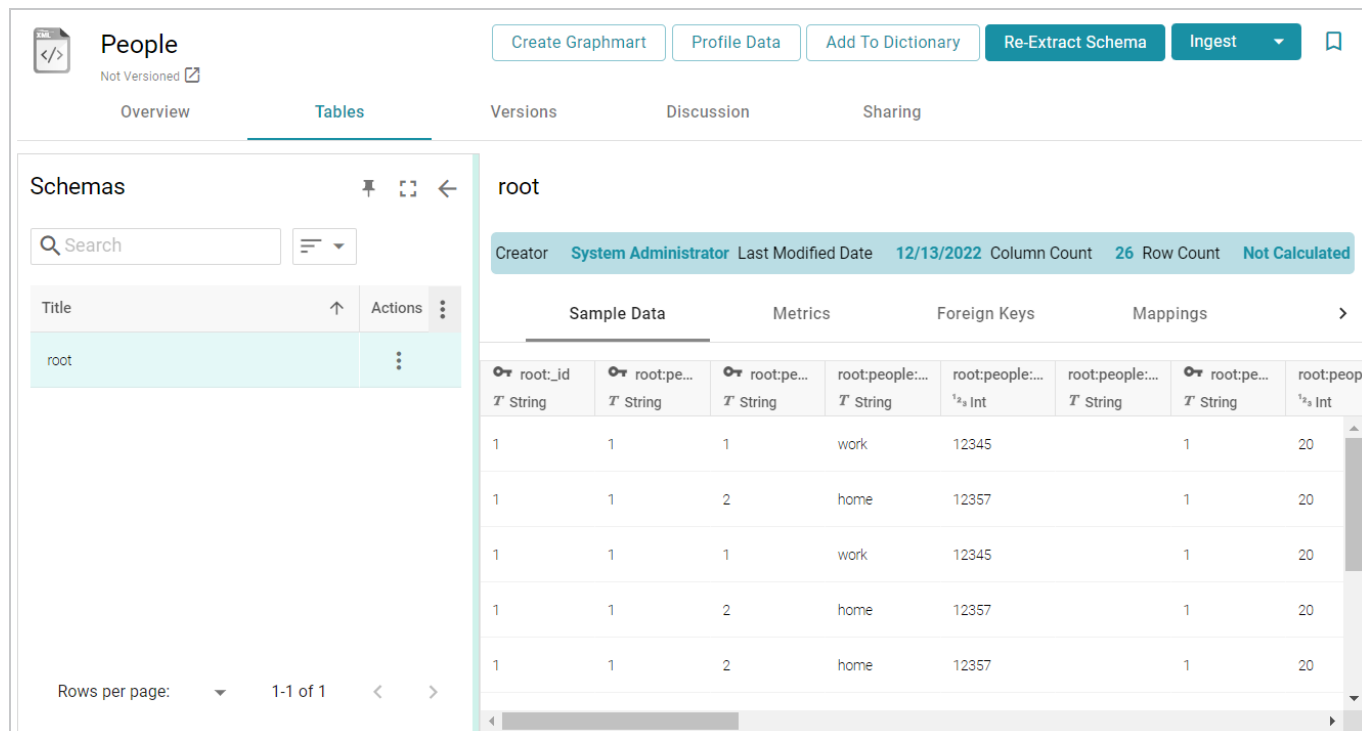
Anzo performs pre-processing before creating the schema. If the specified Schema Type would result in poor performance or require extensive resources, Anzo displays a warning and prompts you to change the schema type before proceeding with the schema creation.

7. The **Schema File Location** field defines where Anzo saves the generated schema. Cambridge Semantics recommends that you leave the field blank. If you want to designate a custom location, click **Browse** and choose a file location.
8. The value in the **Scan Depth** field indicates the number of entities in the file that Anzo should scan to find all of the unique objects to include as classes and properties in the generated model. The scan process follows nested objects, counting one object array as one row. Edit the value as needed. A value of **-1** instructs Anzo to scan the entire file.
9. If the XML file contains lists of objects that are not nested, the file scan cannot determine if any of the objects are the same type, and Anzo treats each object as a new type. To ensure that repeating object paths are treated as the same type if the XML elements are all at the same level, use standard XML

path (XPath) syntax to define the repeating element types in the **Repeating Element Paths** field. If the file nests elements, leave this field blank. Separate paths with semicolons (;). For example:

```
/root/people;/root/people/vehicles;/root/people/vehicles/maintenance
```

10. Click **Save & Extract Schema** to scan the file and generate the schema. Anzo saves the data source, creates the schema, and displays the Tables screen. You can click a table row on the left side of the screen to display the schema on the right side of the screen. For example:



The screenshot shows the Anzo interface for a data source named 'People'. The 'Tables' tab is selected, showing a list of schemas on the left and a detailed view of the 'root' schema on the right. The 'root' schema view includes a table with columns: root\_id, root:pe..., root:pe..., root:people..., root:people..., root:people..., root:pe..., and root:peopl. The table contains 6 rows of data.

root_id	root:pe...	root:pe...	root:people...	root:people...	root:people...	root:pe...	root:peopl
T String	T String	T String	T String	Int	T String	T String	Int
1	1	1	work	12345		1	20
1	1	2	home	12357		1	20
1	1	1	work	12345		1	20
1	1	2	home	12357		1	20
1	1	2	home	12357		1	20

The source data can now be onboarded to Anzo. If you use the Spark ETL pipeline workflow, see [Ingesting Data Sources via ETL Pipelines](#). If you do not use Spark, see [Directly Loading Data Sources via Graphmarts](#) for next steps.

## Adding a SAS Data Source

Follow the instructions below to add a SAS data source and import data from SAS7BDAT files.

### Note

When importing data from SAS files, Anzo imports any metadata that is defined in the files. The metadata only becomes visible, however, when a Metadata Dictionary is created for the source. For more information, see [Creating a Metadata Dictionary \(Preview\)](#).

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:

2. Click the **Add Data Source** button and select **File Data Source** > **SAS Data Source**. Anzo opens the Create SAS Data Source screen.

Create SAS Data Source

Title \*

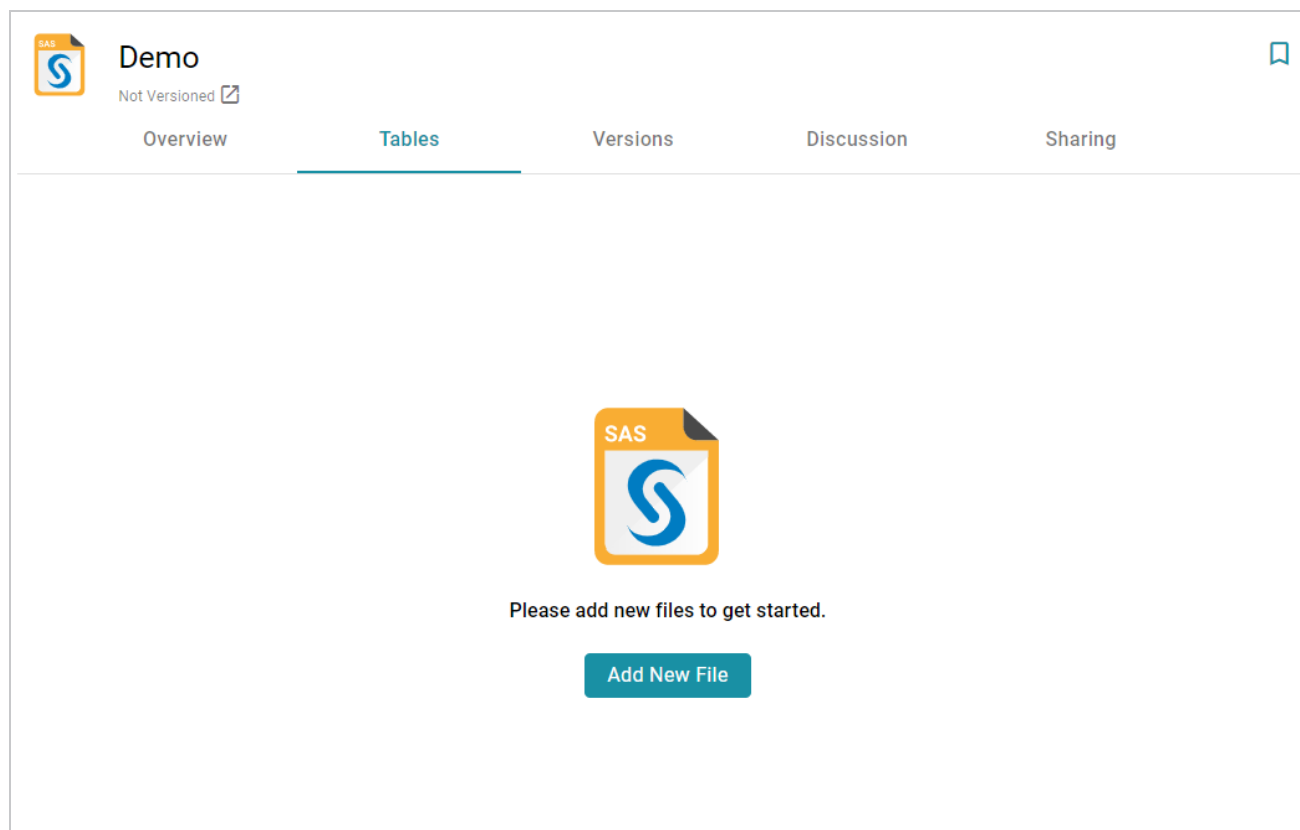
Description

CANCEL

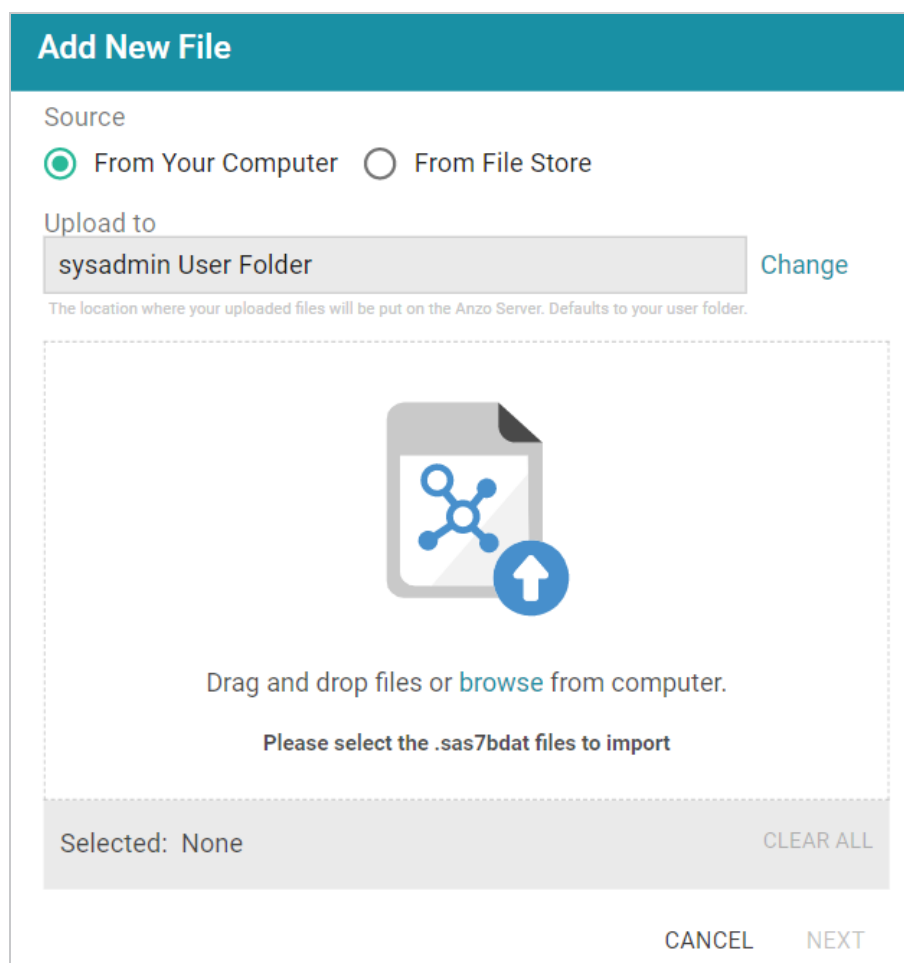
SAVE



3. Specify a name for the source in the **Title** field, and type an optional description in the **Description** field. Then click **Save**. Anzo saves the source and displays the Tables tab.



4. Click the **Add New File** button. Anzo displays the Add New File dialog box.



The dialog box is titled "Add New File" in a teal header. Below the header, there are two radio buttons under the label "Source": "From Your Computer" (selected) and "From File Store". Below this is a text field labeled "Upload to" containing "sysadmin User Folder", with a "Change" link to its right. A tooltip below the text field reads: "The location where your uploaded files will be put on the Anzo Server. Defaults to your user folder." The main area of the dialog is a large dashed box containing a file icon with a blue upload arrow and the text: "Drag and drop files or [browse](#) from computer." Below this is the instruction: "Please select the .sas7bdat files to import". At the bottom left, it says "Selected: None", and at the bottom right is a "CLEAR ALL" link. At the very bottom are "CANCEL" and "NEXT" buttons.

5. Follow the appropriate steps below depending on whether the SAS files are on your computer or the shared File Store:

**If the files are on your computer:**

**Note**

The **From Your Computer** option is a convenient way to do a one-time ingestion so you can quickly get started with your data. It should not be relied upon as part of a regular ingestion workflow unless the server is configured to store uploaded files on the shared File Store. For more information, see [Setting the Default Base File Store Path for File Uploads](#) in the Administration Guide. Data source files that are routinely updated and re-ingested should be hosted on a configured File Store.

- a. As a best practice, check the upload location that is listed in the **Upload To** field by hovering your pointer over the value to view the tooltip. Make sure the upload location is a directory on the

shared file store and not in the server installation path. If the file is not uploaded to the shared file store it is not accessible by applications like AnzoGraph. In addition, other users cannot create graphmarts from the data source because they typically do not have access to the file location.

For example, viewing the Upload To location for the screen above shows that the file will be uploaded to the server installation path, `/opt/Anzo/Server/data...`

The screenshot shows a configuration window with two radio buttons under the heading "Source": "From Your Computer" (selected) and "From File Store". Below this is the "Upload to" section, which contains a text field with the value "sysadmin User Folder" and a "Change" button to its right. A tooltip is visible over the text field, displaying the path `/opt/Anzo/Server/data/userUploads/sysadmin_20235` and the text "on the Anzo Server. Defaults to your user folder."

If your Upload To location is configured to upload the file to the server installation path, click **Change** and select an upload location that is on the shared file store. For example, the image below shows the Upload Folder Location dialog box that is presented after clicking **Change**. A folder called **fileUploads** is selected on the shared store.

The screenshot shows the "Upload Folder Location" dialog box. At the top, there is a "Current Folder" field with a back arrow icon and a "GO" button. Below this is a "Selected:" field showing `/fileUploads/` with a close icon and a "CLEAR ALL" button. The main area is a list of folders under the heading "Server Shared Filesystem". The folders listed are "docs", "fileUploads" (which is highlighted in blue), "flights", "fmcsa", and "nhih10". At the bottom of the dialog, there are three buttons: "CREATE NEW FOLDER", "CANCEL", and "OK".

- b. Drag and drop the files onto the screen or click **browse** to navigate to the files and select them. Anzo attaches the files and the Next button becomes active.

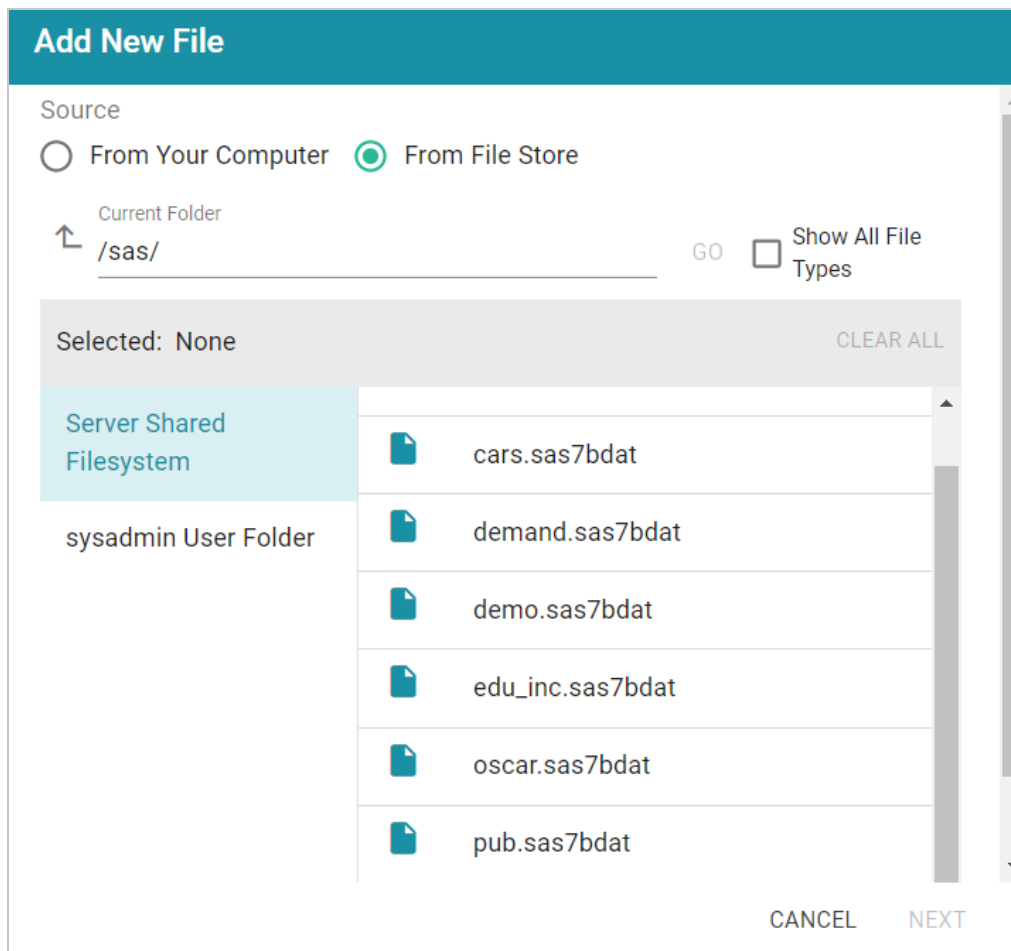
- c. Click **Next**. Anzo lists the uploaded files on the left side of the screen with a status of Pending. For example:

The screenshot shows the Anzo 'Demo' interface. At the top, there's a header with the SAS logo, the title 'Demo', and a 'Not Versioned' status. Below the header are tabs: Overview, Tables (selected), Versions, Discussion, and Sharing. The main content area is titled 'Schemas' and contains a table with four rows. Each row has a checkbox, a title, an upward arrow, a status of 'Pending', and an actions menu. To the right of the table is a large SAS logo and the text 'Schema Details' followed by 'Please process the files to view them.'.



<input type="checkbox"/>	Title	↑	Status	Actions
<input type="checkbox"/>	airline	↑	Pending	⋮
<input type="checkbox"/>	cars	↑	Pending	⋮
<input type="checkbox"/>	demo	↑	Pending	⋮
<input type="checkbox"/>	pub	↑	Pending	⋮

**If the files are on the File Store:**

- Click the **From File Store** radio button.
- In the File Location dialog box, on the left side of the screen, select the appropriate File Store. On the right side of the screen, navigate to the directory that contains the file to import. The screen displays the list of files in the directory. For example:






- c. Select each file that you want to import. When you finish selecting files, click **Next** to close the dialog box. Anzo lists the uploaded files on the left side of the screen with a status of Pending. For example:



**Demo**  
 Not Versioned 









Overview
 **Tables**
 Versions
 Discussion
 Sharing

### Schemas






Add New File
Process Pending Files



<input type="checkbox"/>	Title	↑	Status	Actions
<input type="checkbox"/>	airline		 Pending	
<input type="checkbox"/>	cars		 Pending	
<input type="checkbox"/>	demo		 Pending	
<input type="checkbox"/>	pub		 Pending	

Rows per page:
1-4 of 4



### Schema Details

Please process the files to view them.

- If you do not need to change SAS file options, click the **Process Pending Files** button to import all of the pending files. Anzo imports the data and updates the status to Processed.

If you do need to change SAS file options, click the menu icon (⋮) for that file and select **Edit**. To change the options for multiple files, select the checkbox next to each of the files, and then click the **Edit** button at the bottom of the table. Anzo displays the Edit SAS File screen. For example, the image below shows the Edit screen for a single file:

Edit SAS Files

Title

airline

File Path

/sas/airline.sas7bdat

BROWSE

☐ Force column names to lower case
 ☐ Infer numeric column as decimal

Scale of inferred decimal

☐ Infer numeric column as Float
 ☐ Infer numeric column as Int
 ☐ Infer numeric column as Long
 ☐ Infer numeric column as Short
 ☐ Extract column labels

Number of seconds to allow reading of file metadata

Minimum byte length of input splits

CANCEL

SAVE & IMPORT

Change the options as needed and then click **Save & Import** to import the SAS file or files. Anzo imports the data and updates the status to Processed.

- Once the files are processed, the **Create Graphmart**, **Profile Data**, **Add To Dictionary** and **Ingest** options become available. You can click a table row on the left side of the screen to display the schema on the right side of the screen. For example:

Demo

Not Versioned

Create Graphmart

Profile Data

Add To Dictionary

Ingest

Overview

Tables

Versions

Discussion

Sharing

Schemas

Add New File

Process Pending Files

Search

<input type="checkbox"/>	Title	Status	Actions
<input checked="" type="checkbox"/>	airline	Processed	
<input type="checkbox"/>	cars	Processed	
<input type="checkbox"/>	demo	Processed	
<input type="checkbox"/>	pub	Processed	

Rows per page:

1-4 of 4

airline

Creator

System Administrator

Last Modified Date

12/13/2022

Column Count

6

Row Count

Not Calculated

Sample Data

Metrics

Foreign Keys

Mappings

F

YEAR	Y	W	R	L	K
1,23 Double	1,23 Double	1,23 Double	1,23 Double	1,23 Double	1,23 Double
1952	2.26500010490417	0.310000002384186	0.35589998960495	1.80200004577637	0.574000000953674
1951	1.94799995422363	0.296999990940094	0.393999993801117	1.54999995231628	0.564000010490417
1948	1.21399998664856	0.243000000715256	0.145400002598763	1.41499996185303	0.611999988555908
1949	1.35399997234344	0.259999990463257	0.218099996447563	1.38399994373322	0.559000015258789
1957	4.42000007629395	0.379000008106232	0.304500013589859	2.70700001716614	0.921000003814697
1950	1.56900000572205	0.277999997138977	0.315699994564056	1.38800001144409	0.573000013828278

The source data can now be onboarded to Anzo. If you use the Spark ETL pipeline workflow, see [Ingesting Data Sources via ETL Pipelines](#). If you do not use Spark, see [Directly Loading Data Sources via Graphmarts](#) for next steps.

## Adding a Parquet Data Source

Follow the instructions below to create a Parquet data source. You can onboard one file or multiple files with the identical format (schema) per source.

### Note

If your Parquet Data Source is consistently updated with new or changed files, you can configure the Data Source to process the data incrementally. For details, see [Configuring a CSV or Parquet Data Source for Incremental Processing](#).

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:



Create Parquet Data Source


Title \*

Description


CANCEL

SAVE

- Specify a name for the source in the **Title** field, and type an optional description in the **Description** field. Then click **Save**. Anzo saves the source and displays the Overview tab. For example:



Events

Not Versioned 

Overview

Versions

Discussion

Sharing

Description

None

Parquet File

None

General

Type

Parquet

Creator

System Administrator


Updated

Dec 13, 2022 4:09 PM

Released

Dec 13, 2022 4:09 PM

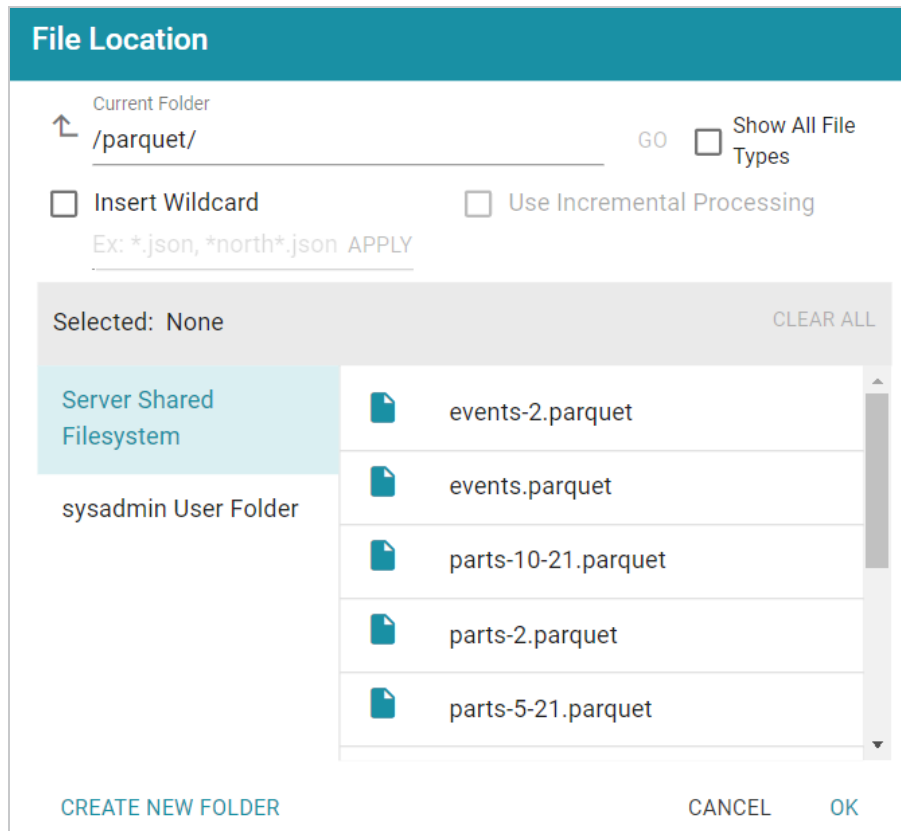
http://cambridgesemantics.com/ParquetData



Tags

None

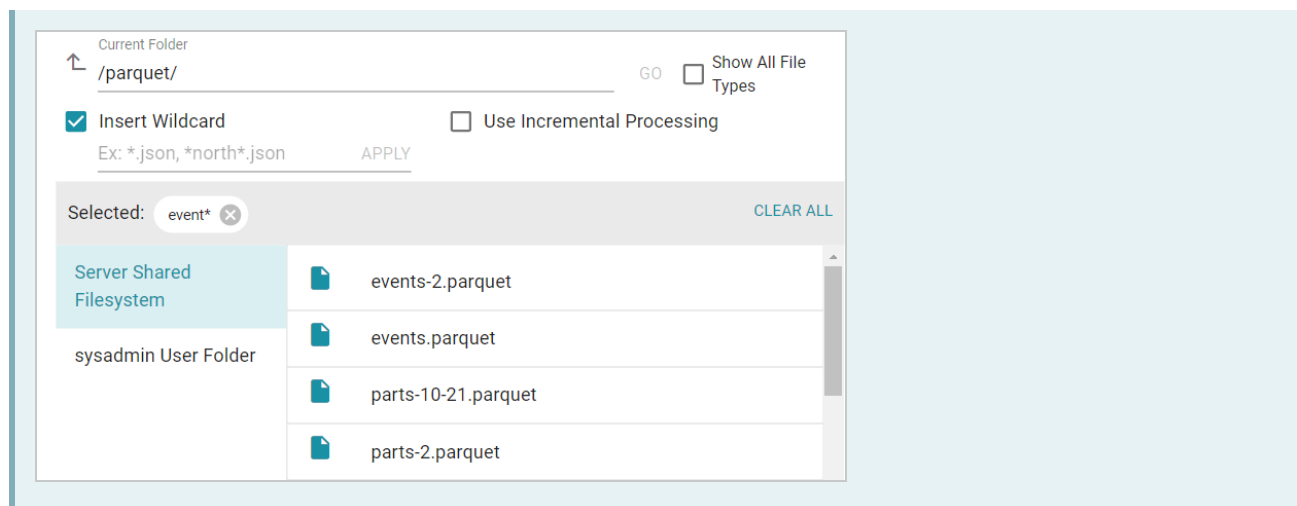
- On the Overview tab, click in the **Parquet File** field to make the value editable. Then click **Browse** to open the File Location dialog box and select the file to import.
- In the File Location dialog box on the left side of the screen, select the file store for the Parquet file. On the right side of the screen, navigate to the directory that contains the file to import. The screen displays the list of files in the directory. For example:



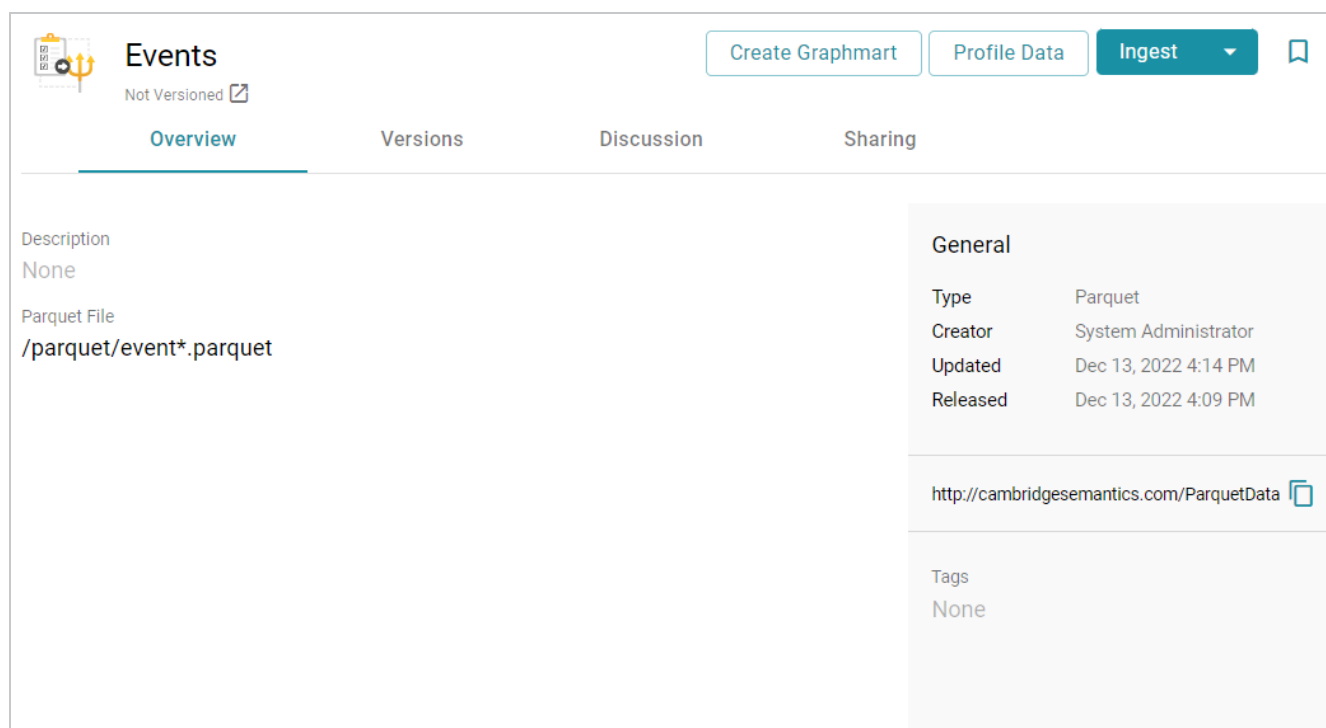
6. Select the file that you want to import. If you have multiple files with the identical format you can select the **Insert Wildcard** option. Then type a string using asterisks as wildcard characters to find the files with similar names. Files that match the specified string will be imported as **one file** and will result in one job being created in the pipeline to ingest all of the files that are selected by the specified string. You can specify up to 16,000 files using a wildcard. After typing a string, click **Apply** to include that string in the Selected list.

### Example

The image below shows a directory with multiple Parquet files. The **events.parquet** and **events-2.parquet** file have the identical format and can be imported as one file. The **Insert Wildcard** option is selected, and **event\*.parquet** is specified to identify the two files.



- After selecting the file, click **OK** to close the File Location dialog box. Then click the checkmark icon (✓) to save the change. The **Create Graphmart**, **Profile Data**, and **Ingest** buttons become available.



The source data can now be onboarded to Anzo. If you use the Spark ETL pipeline workflow, see [Ingesting Data Sources via ETL Pipelines](#). If you do not use Spark, see [Directly Loading Data Sources via Graphmarts](#) for next steps.

## Related Topics

[Configuring a CSV or Parquet Data Source for Incremental Processing](#)

## Configuring a CSV or Parquet Data Source for Incremental Processing

If you have a CSV or Parquet data source that is consistently updated with new or changed files, you can configure the source to process the data incrementally. When a source is set up to onboard data incrementally and new files are added to the data source directory, those new files are ingested and added to the existing dataset. This topic describes how incremental processing works, gives recommendations for organizing the source files, and provides instructions for configuring incremental processing.

- [How Incremental Processing Works](#)
- [Organizing the Data Source Directory](#)
- [Configuring Incremental Processing](#)

### How Incremental Processing Works

When you configure file-based incremental processing, there are two methods to choose from:

#### File Name Strategy

The first method is called the **File Name Strategy**. When this strategy is chosen, Anzo saves the names of the files that are onboarded. Subsequent workflow runs ingest any files whose names are not saved. This method is useful when you know that new files will be added to the data source directory but the existing files will not be edited. If the contents of a previously ingested file changes but the file name does not, that file will not be reprocessed during the next run.

#### Last Modified Strategy

The second method is called the **Last Modified Strategy**. When this strategy is chosen, Anzo saves the last modified timestamps for the files that are ingested when the workflow is run. Subsequent runs ingest any files whose last modified timestamps are greater than the saved timestamps. Each time an incremental workflow run, the job-specific last modified date will be updated to the latest modified date of the files processed. This method is useful when you know that the contents of previously ingested files may change in addition to adding new files. Since the last modified date is updated when a file is changed, the changes will be processed during the next run.

### Organizing the Data Source Directory

Regardless of the strategy you use to process data incrementally, it is important to consider the schema when organizing the data source files on the file store. In order to enable incremental processing, the source files to import must be specified using wildcard characters (\*). That means the list of files that are targeted by the wildcard need to have the same schema.

## CSV Data Sources

For CSV data sources, you have two options. You can create one subdirectory per schema and then add files multiple times, once for each schema. With this structure, you would import the files in each directory separately and could specify the wildcard like `*.csv` to import all of the files in a directory. You can also place all of the files into a single directory and use more detailed text when specifying wildcards. For example, when you add files you apply multiple wildcard values such as `patients_*.csv` and `medication_*.csv`.

## Parquet Data Sources

For Parquet data sources, you can only choose one schema per source. You must create a separate data source for each schema type. You may want to create one directory per schema. Then each Parquet source can target one directory and specify a wildcard value such as `*.parquet`.

## Configuring Incremental Processing

When adding or modifying a CSV or Parquet data source, you configure incremental processing when you are adding files to the source from the file store. This section provides instructions for adding files to a source and configuring incremental processing. For instructions on adding a new data source, see [Adding a CSV Data Source](#) or [Adding a Parquet Data Source](#).

1. When selecting the source files on the file store, select the **Insert Wildcard** checkbox. Enabling the wildcard option activates the Use Incremental Processing option.

File Location

Current Folder

↑

/nfs/data/parquet/

GO

☐ Show All File Types

☒ Insert Wildcard

Ex: \*.json, \*north\*.json

APPLY

☐ Use Incremental Processing

Selected: None

CLEAR ALL

Server Filesystem

parts-5-21.parquet

parts-6-21.parquet

parts-7-21.parquet

parts-8-21.parquet

parts-9-21.parquet

parts.parquet

sysadmin User Folder

CANCEL

OK

- Below the checkbox, type a string using asterisks (\*) as wildcard characters to find the files to be processed. Then click **Apply** to apply the string. If you are configuring a CSV data source, you can apply multiple wildcard strings to target files with different schemas. The image below shows an example for a Parquet source. The string `parts*` is applied to select all of the files with names that start with "parts."

File Location

Current Folder

↑

/nfs/data/parquet/

GO

☐ Show All File Types

☒ Insert Wildcard

Ex: \*.json, \*north\*.json

APPLY

☐ Use Incremental Processing

Selected: parts\*

×

CLEAR ALL

Server Filesystem

sysadmin User Folder

parts-5-21.parquet

parts-6-21.parquet

parts-7-21.parquet

parts-8-21.parquet

parts-9-21.parquet

parts.parquet

CANCEL

OK

- Next, select the **Use Incremental Processing** checkbox. The Configure Incremental Processing dialog box is displayed:

Configure Incremental Processing

Strategy

▼

CANCEL

APPLY

- Click the **Strategy** drop-down list and select the strategy to use for incremental processing. The following list describes the options. For more details about the strategies, see [How Incremental Processing Works](#) above.

- **File Name Strategy:** Select this option if file names should be used to target the new source data to process each time the graphmart that contains this source is reloaded or refreshed.
  - **Last Modified Strategy:** Select this option if the last modified date should be used to target the new source data to process each time the graphmart that contains this source is reloaded or refreshed.
5. If you chose **File Name Strategy**, click **Apply** and then click **OK** or **Next** to proceed. The source is now configured to process data incrementally. If you chose **Last Modified Strategy**, proceed to the next step.
  6. If you chose **Last Modified Strategy**, the Baseline options are displayed. The Baseline determines when the last modified date begins.

**Configure Incremental Processing**

Strategy  
Last Modified Strategy

Baseline

☒ All matching files

☐ All matching files since 11/16/2021 3:29pm

CANCEL APPLY

7. By default **All matching files** is selected as the Baseline. This means all files in the directory that are matched by the wildcard string will be ingested when the graphmart is reloaded or refreshed. If you have older files that you do not want to be ingested, you can select **All matching files since**. Then click the timestamp field and specify the date and time to use as the Baseline.
8. When you have finished configuring the Strategy, click **Apply**. Then click **OK** or **Next** to proceed and finish configuring the data source if necessary.

When the graphmart is created, all of the data that matches the wildcard string and meets the baseline requirements will be onboarded. When new files are added to the data source directory, reloading or refreshing the graphmart will update the data.

The source data can now be onboarded to Anzo. If you use the Spark ETL pipeline workflow, see [Ingesting Data Sources via ETL Pipelines](#). If you do not use Spark, see [Directly Loading Data Sources via Graphmarts](#) for next steps.



## Assigning Primary Keys in an Onboarded Schema

If you have a Data Source without Primary Keys and you want to be able to create Foreign Key relationships, you can assign Primary Keys in the Schema. Follow the instructions below to edit a Schema and assign Primary Keys. For instructions on creating Foreign Keys, see [Creating or Changing Foreign Keys](#).

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:

Data Sources

Schemas

Mappings

Pipelines

Search

Import Schemas

<div><div></div></div>	Title	<div>↑</div>	Data Source	Updated Date	Actions	<div></div>
<div><div></div></div>	<div><div></div>Datafox</div>		Datafox	Dec 13, 2022	<div><div></div><div></div></div>	<div></div>
<div><div></div></div>	<div><div></div>Demo</div>		Demo	Dec 13, 2022	<div><div></div><div></div></div>	<div></div>
<div><div></div></div>	<div><div></div>emrdb</div>		DB	Dec 13, 2022	<div><div></div><div></div></div>	<div></div>
<div><div></div></div>	<div><div></div>Events</div>		Events	Dec 13, 2022	<div><div></div><div></div></div>	<div></div>
<div><div></div></div>	<div><div></div>Movies</div>		Movies	Dec 8, 2022	<div><div></div><div></div></div>	<div></div>
<div><div></div></div>	<div><div></div>mysql</div>		DB	Dec 13, 2022	<div><div></div><div></div></div>	<div></div>
<div><div></div></div>	<div><div></div>new_hires</div>		DB	Dec 13, 2022	<div><div></div><div></div></div>	<div></div>
<div><div></div></div>	<div><div></div>northwind</div>		DB	Dec 8, 2022	<div><div></div><div></div></div>	<div></div>

Rows per page:

25

1-10 of 10

- Click the schema for which you want to assign primary keys. Anzo displays the Tables tab for the data source that contains the schema. For example:



5. Click the edit icon (✎). The Edit dialog box is displayed. For example:

Edit

Name \*

catid

Name of the column

Semantic Type

Int

X | < > v

Type of the column

☐ Primary Key

CANCEL

SAVE

6. On the Edit screen, select the **Primary Key** checkbox. Then click **Save** to save the change. The column is now the Primary Key for the table, and a key icon is displayed next to the column name. For example:

Sample Data		Metrics
<div><div><div><div>Key</div></div></div><div>catid</div><div><div><div>123</div></div>Int</div></div>		<div>catgroup</div> <div>T String</div>
7		Shows
10		Concerts
2		Sports

Repeat the steps above to assign primary keys for additional tables. For instructions on creating foreign keys, see [Creating or Changing Foreign Keys](#).

## Creating or Changing Foreign Keys

Follow the instructions below to edit a schema to add foreign keys. For instructions on designating primary keys, see [Assigning Primary Keys in an Onboarded Schema](#).

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:

Data Sources									
Schemas									
Mappings									
Pipelines									
<div><div><div></div><div>Search</div></div><div><div></div><div></div></div><div><div></div><div></div></div></div> <div>Add Data Source</div>									
<input type="checkbox"/>	Title	Description	Type	Schema	Creator	Created Date	Updated Date	Actions	
<input checked="" type="checkbox"/>	DB		Database Data...	emrdb, new_...	System Admin...	Dec 8, 2022	Dec 8, 2022	<a href="#">Bookmark</a>	<a href="#">More</a>
<input type="checkbox"/>	Movies		CSV Data Sour...	Movies	System Admin...	Dec 8, 2022	Dec 8, 2022	<a href="#">Bookmark</a>	<a href="#">More</a>
<input type="checkbox"/>	Tickets		CSV Data Sour...	Tickets	System Admin...	Dec 9, 2022	Dec 12, 2022	<a href="#">Bookmark</a>	<a href="#">More</a>
Rows per page: 25 1-3 of 3 < >									

2. Click the **Schemas** tab to view the list of all schemas. (You can also access a schema by selecting the data source that contains the schema.) The image below shows a view of the Schemas tab:

Data Sources

Schemas

Mappings

Pipelines

Search

Import Schemas

<div><div></div></div>	Title	<div>↑</div>	Data Source	Updated Date	Actions	<div></div>
<div><div></div></div>	<div><div></div> Datafox</div>		Datafox	Dec 13, 2022	<div><div></div><div></div></div>	<div></div>
<div><div></div></div>	<div><div></div> Demo</div>		Demo	Dec 13, 2022	<div><div></div><div></div></div>	<div></div>
<div><div></div></div>	<div><div></div> emrdb</div>		DB	Dec 13, 2022	<div><div></div><div></div></div>	<div></div>
<div><div></div></div>	<div><div></div> Events</div>		Events	Dec 13, 2022	<div><div></div><div></div></div>	<div></div>
<div><div></div></div>	<div><div></div> Movies</div>		Movies	Dec 8, 2022	<div><div></div><div></div></div>	<div></div>
<div><div></div></div>	<div><div></div> mysql</div>		DB	Dec 13, 2022	<div><div></div><div></div></div>	<div></div>
<div><div></div></div>	<div><div></div> new_hires</div>		DB	Dec 13, 2022	<div><div></div><div></div></div>	<div></div>
<div><div></div></div>	<div><div></div> northwind</div>		DB	Dec 8, 2022	<div><div></div><div></div></div>	<div></div>

Rows per page: 25

1-10 of 10

- Click the schema for which you want to assign foreign keys. Anzo displays the Tables tab for the data source that contains the schema. For example:



## Note

The Suggested Keys tab is populated with foreign key suggestions after a data profile has been generated for the schema. For information about data profiling, see [Generating a Source Data Profile](#).

5. To change an existing key, click a row in the table to open the Edit Foreign Key dialog box. To create a new key, click the **Create** button to open the Create Foreign Key dialog box.

**Create Foreign Key**

Your key will not be saved unless you select columns from two different tables.

Key Type: **Manual**

Name  
Name of the foreign key

Source Table

Select source table

Select source columns

Only View Selected Columns ☒

Target Table

Select target table

Select target columns

Only View Selected Columns ☒

CANCEL SAVE

6. On the Create Foreign Key screen, specify a name for the key in the **Name** field.
7. Specify the source and target tables for this key.
  - **Source Table:** The source table is the table where the new foreign key is created. This table refers to the primary key from the Target Table. Click the **Source Table** drop-down list and select the schema table where the foreign key should be created.
  - **Target Table:** The target table is the table that contains the primary key to be referenced by the Source Table. Click the **Target Table** drop-down list and select the schema table that will pass values to the source table.
8. Specify the source and target columns for this key:
  - **Source Columns:** The source column is the column that becomes the foreign key to the target table's primary key. Click the **Select Source Columns** drop-down list and select the source column. To create a composite key by selecting an additional column, click the Select Source Columns drop-down list again and select a column.



### Tip

By default the screen shows sample values from the selected source column. If you want to view sample values from all columns in the source table, you can disable the **Only View Selected Columns** option by sliding the slider to the left.

- **Target Columns:** The target column is the primary key column in the target table. Click the **Select Target Columns** drop-down list and select the target column. To create a composite key by selecting an additional column, click the Select Target Columns drop-down list again and select a column.

### Tip

By default the screen shows sample values from the selected target column. If you want to view sample values from all columns in the target table, you can disable the **Only View Selected Columns** option by sliding the slider to the left.

For example, the image below creates a relationship called **eventDate** where the **dateid** column in the **tickit\_events** table becomes the foreign key and references the values from the primary key column, **dateid**, in the **tickit\_dates** table.

### Create Foreign Key

Your key will not be saved unless you select columns from two different tables.

Key Type: **Manual**

Name  
eventDate

Name of the foreign key

#### Source Table

tickit\_events

dateid x

Only View Selected Columns ☒

dateid

2114

1851

2090

#### Target Table

tickit\_dates

dateid x

Only View Selected Columns ☒

dateid

1828

1830

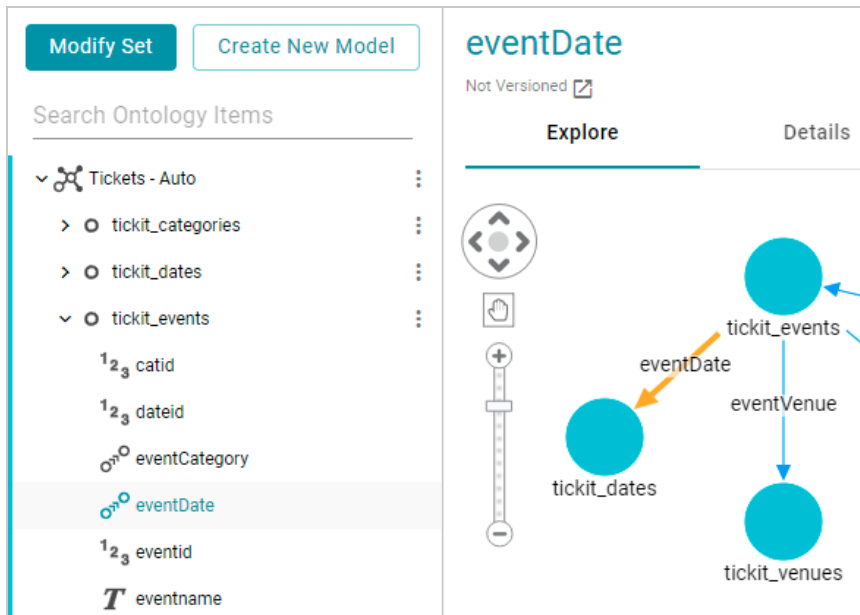
1829

CANCEL

SAVE

- When you have finished supplying values, click **Save** to create the new key and return to the Foreign Key list. To create additional keys. Repeat this process to create additional keys.

When you ingest the data using this schema, the foreign keys become RDF OWL object properties in the data model. For example, the image below shows a portion of the model that was generated after ingesting the schema that has the foreign key in the example above. In the model, **eventDate** is an object property in the **ticket\_events** class:



## Ingesting Data Sources via ETL Pipelines

The topics in this section provide instructions for ingesting data from structured data sources using the **Ingest** ETL pipeline process. The Ingest workflow automatically generates a Model, Mappings, and an ETL Pipeline when you ingest a data source for the first time. If the schema changes and the pipeline components need to be updated, you can configure subsequent Ingest workflows to reuse and update the existing components or regenerate them.

### Note

If the source data is updated but the schema does not change, or if the model or mappings are modified and the schema is not affected, you do not need to re-ingest the source using the Ingest workflow. You can simply republish the pipeline or the affected jobs in the pipeline. See [Publishing a Pipeline or Subset of Jobs](#) for more information.

The way you configure the Ingest workflow depends on whether you are ingesting a Data Source for the first time, are re-ingesting a Data Source because the Schema changed, or whether the source has an associated Metadata Dictionary. Select the appropriate instructions below for guidance on configuring the initial Ingest workflow, a subsequent workflow, or a workflow with a Metadata Dictionary:

[Ingesting a New Data Source](#)

[Re-Ingesting an Updated Data Source](#)

[Ingesting a Data Source with a Metadata Dictionary](#)

[Managing Data Source Metadata](#)

[Working with Mappings](#)

[Working with Pipelines](#)

### Ingesting a New Data Source

Follow the instructions below to set up the Ingest workflow for a new source that has not been previously ingested. The procedure below focuses on configuring the workflow to generate a new model in addition to the mappings, ETL jobs, and dataset pipeline that is needed to ingest the data into Anzo, convert it to the graph data model, and make it available for inclusion in a graphmart.

### Tip

For instructions on ingesting an updated source, see [Re-Ingesting an Updated Data Source](#). If the data source has an associated Metadata Dictionary that you want to apply to the workflow, see [Ingesting a Data Source with a Metadata Dictionary](#).

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:

<

2. On the Data Sources screen, click the name of the source for which you want to ingest data. Anzo displays the Tables screen for the source. For example:

The screenshot shows the Anzo interface with the 'Tables' tab selected. On the left, the 'Schemas' panel lists 'emrdb' and 'northwind'. The main area displays an 'Alphabetical list of products' table. The table has the following columns: ProductID (Int), ProductName (String), SupplierID (Int), CategoryID (Int), QuantityPerUnit (String), UnitPrice (Decimal), UnitsInStock (Short), and UnitsOnOrder (Short). The table shows sample data for various products.

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder
43	Ipoh Coffee	20	1	16 - 500 g tins	46.0000	17	10
34	Sasquatch Ale	16	1	24 - 12 oz bottles	14.0000	111	0
70	Outback Lager	7	1	24 - 355 ml bottle	15.0000	15	10
39	Chartreuse verte	18	1	750 cc per bottle	18.0000	69	0
67	Laughing Lumberjack Lager	16	1	24 - 12 oz bottles	14.0000	52	0
2	Chang	1	1	24 - 12 oz bottles	19.0000	17	40

- Click the **Ingest** button. If the source has more than one schema, Anzo displays the select schema dialog box. In the drop-down list, select the schema to use, and then click **OK**. For example:

The screenshot shows a dialog box titled 'Select the schema you want to use'. It contains a text input field with the value 'emrdb' and a dropdown arrow. Below the input field are two buttons: 'CANCEL' and 'OK'.

Anzo opens the Ingest dialog box and automatically populates the Data Source Connection value. If there is only one configured Data Store, the Anzo Data Store value is also auto-populated. In addition, if the default ETL Engine is configured for the system, the Auto Map Engine Config field is also populated (see [Configure the Default ETL Engine](#) in the Administration Guide). For example, in the image below the Anzo Data Store field is not populated because there are multiple available choices. The ETL Engine field is populated because the Local Sparkler Engine is configured as the default ETL Engine:

**Ingest**

Source Data Connection

DB

Anzo Data Store

Anzo Data Store

Auto Map Engine Config

Local Sparkler Engine

☒ Select all tables ☐ Custom select

Advanced

CANCEL SAVE

4. If necessary, click the **Anzo Data Store** field and select the Data Store for this pipeline.
5. If necessary, click the **Auto Map Engine Config** field and select the ETL engine to use for this pipeline.
6. By default, the **Select all tables** radio button is enabled to ingest the data for all tables in the Schema. If you do not want to add all tables, click the **Custom select** radio button and then select each of the tables to add.
7. By default, the Ingest workflow is configured to generate a new model in addition to the mappings and jobs that are needed to onboard the data. You can click **Save** to save the configuration and proceed with the model and pipeline generation. If you want customize the URI that is generated for the new model or the class and property URIs in the model, you can click **Advanced** to expand the screen and view the following options:

Advanced ^

☒ Create New Model ☐ Use Existing Model

Schema Ontology URI

Schema Class Prefix

Schema Property Prefix

☐ Transform Property Names

The list below describes the options:

- **Schema Ontology URI:** The URI for the model. When this field is blank, Anzo generates the model URI with the following format:

```
http://cambridgesemantics.com/ont/autogen/xx/<schema_name>
```

Where `xx` is a hash snippet based on the model's globally unique identifier (GUID). If you want to specify a different format, you can type that URI into the Schema Ontology URI field. For example, a URI such as `http://mycompany.com.ontology/movies` results in a model URI of `http://mycompany.com.ontology/movies`.

### Important

Make sure that Schema Ontology URI is unique. If the URI is not unique, this model will overwrite any existing model that uses this URI.

- **Schema Class Prefix:** The URI prefix format to use for classes in the model. When this field is blank, Anzo generates class URIs using the following format:

```
http://cambridgesemantics.com/ont/autogen/xx/<schema_name>#<class_name>
```

Where `xx` is a hash snippet based on the model's GUID. If you want to specify a different format for class URIs, type the prefix to use in this field. For example, a prefix such as `http://mycompany.com.ontology/class` results in class URIs like `http://mycompany.com.ontology/class#<class_name>`.

### Tip

Since you are specifying a prefix format, and the class name will be appended to the prefix, it is permissible to set Schema Class Prefix to the same value across schemas.

- **Schema Property Prefix:** The URI prefix format to use for properties in the model. When this field is blank, Anzo generates property URIs using the following format:

```
http://cambridgesemantics.com/ont/autogen/xx/<schema_name>#<class_name>_<property_name>
```

Where xx is a hash snippet based on the model's GUID. If you want to specify a different format for property URIs, type the prefix to use in this field. you can type that URI into the Schema Property Prefix field. For example, a prefix such as


```
http://mycompany.com.ontology/property results in property URIs like  
http://mycompany.com.ontology/property#<class_name>_<property_name>.
```

### Tip


Since you are specifying a prefix format, and the property name will be appended to the prefix, it is permissible to set Schema Property Prefix to the same value across schemas.


- **Transform Property Names:** Transforms property names to upper or lower case letters. To transform names, select the **Transform Property Names** checkbox. Then select the **To lowercase** radio button if you want to convert property names to lowercase or select the **To UPPERCASE** radio button if you want to convert property names to uppercase.
8. Click **Save** if you changed advanced options. Anzo creates a Pipeline and generates the Model and Mappings according to the options you specified.
  9. In the main navigation menu under **Onboard**, click **Structured Data**. Then click the **Pipelines** tab.
  10. Click the name of the pipeline that you created. Anzo displays the Pipeline Overview screen. For example:






# Load DB - mysql to store

Not Versioned 

Publish All 



Overview

Jobs

History

Versions

Discussion

>

Description

None

Engine Configuration

Local Sparkler Engine

Graph datasource

store

General

Type

Dataset Pipeline

Creator


System Administrator

Updated

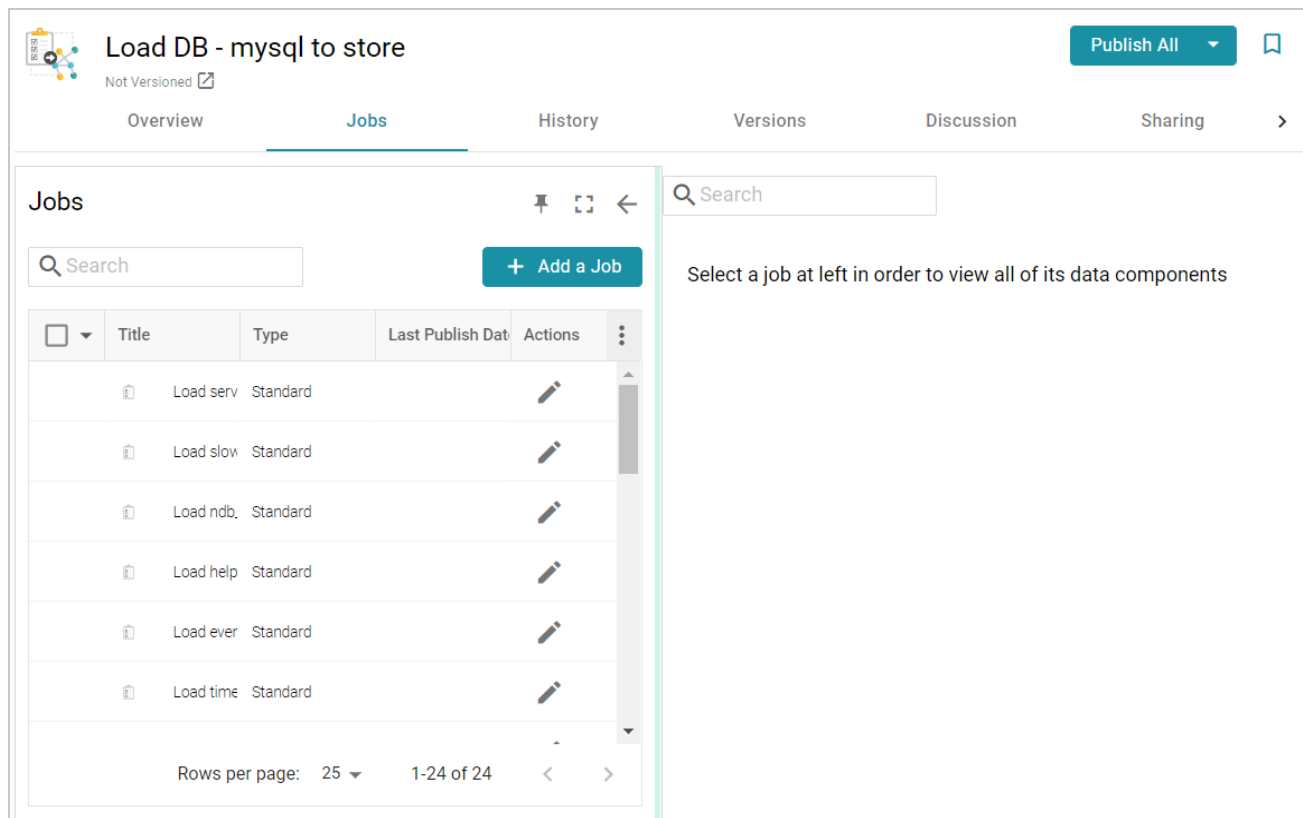
a few seconds ago

Released

a few seconds ago

<http://cambridgesemantics.com/Project> 

11. If you would like to see the jobs that Anzo created for this pipeline, click the **Jobs** tab. The jobs are listed on the left side of the screen. A job exists for each of the tables that were imported. If this pipeline has not been published previously, the right side of the screen remains blank. After the jobs are run, selecting a job from the list displays its history on the right. For example, the image below shows a new pipeline that has not been published:



- To run all of the jobs, click the **Publish All** button at the top of the screen. To publish a subset of the jobs, select the checkbox next to each job that you want to run and then click the **Publish** button above the list of jobs. Anzo runs the pipeline and generates the resulting file-based linked data set in a new subdirectory under the specified Anzo data store.

When the pipeline finishes, this run of the pipeline becomes the **Managed Edition**. The Managed Edition always contains the latest successfully published data for all of the jobs in the pipeline. If one or more of the jobs failed, those jobs are excluded from the edition. If you publish the failed jobs at a later date or you create and publish additional jobs in the pipeline, the data from those jobs is also added to the Managed Edition. For more information about editions, see [Managing Dataset Editions](#).

The new dataset also becomes available in the Datasets catalog. From the catalog, you can create a new graphmart from the dataset or add the dataset to an existing graphmart. See [Creating a Graphmart from a Dataset](#) or [Adding a Dataset to an Existing Graphmart](#) for instructions.

## Re-Ingesting an Updated Data Source

Follow the instructions below to re-ingest the data for a data source whose schema has been updated. The procedure below focuses on configuring the workflow to reuse the existing model and update the mappings and ETL jobs for the existing pipeline. For instructions on ingesting a new data source, see [Ingesting a New](#)

### Note

If the source data is updated but the schema does not change, or if the model or mappings are modified and the schema is not affected, you do not need to re-ingest the source using the Ingest workflow. You can simply republish the pipeline or the affected jobs in the pipeline. See [Publishing a Pipeline or Subset of Jobs](#) for more information.

### Tip

For information about updating a CSV Data Source if a file changes, see [How do I update Anzo if a file in my CSV data source changes?](#)

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:

Search

Add Data Source

<div></div>	Title	Description	Type	Schema	Creator	Created Date	Updated Date	Actions
<div></div>	DB	Database Data...	emrdb, new...	System Admin...	Dec 8, 2022	Dec 8, 2022	<div></div> <div></div>	
<div></div>	Movies	CSV Data Sour...	Movies	System Admin...	Dec 8, 2022	Dec 8, 2022	<div></div> <div></div>	
<div></div>	Tickets	CSV Data Sour...	Tickets	System Admin...	Dec 9, 2022	Dec 12, 2022	<div></div> <div></div>	

Rows per page: 25

1-3 of 3

The screenshot shows the Anzo interface with the 'Schemas' panel on the left and a table of product data on the right. The 'Schemas' panel lists 'emrdb' and 'northwind'. The table on the right is titled 'Alphabetical list of products' and shows columns for ProductID, ProductName, SupplierID, CategoryID, QuantityPerUnit, UnitPrice, UnitsInStock, and UnitsOnOrder.

ProductID	ProductNa...	SupplierID	CategoryID	QuantityPer...	UnitPrice	UnitsInStock	UnitsOnOrder
<sup>1</sup> / <sub>2</sub> Int	T String	<sup>1</sup> / <sub>2</sub> Int	<sup>1</sup> / <sub>2</sub> Int	T String	<sup>1</sup> / <sub>2</sub> Decimal	<sup>1</sup> / <sub>2</sub> Short	<sup>1</sup> / <sub>2</sub> Short
43	Iphoh Coffee	20	1	16 - 500 g tins	46.0000	17	10
34	Sasquatch Ale	16	1	24 - 12 oz bottles	14.0000	111	0
70	Outback Lager	7	1	24 - 355 ml bottle	15.0000	15	10
39	Chartreuse verte	18	1	750 cc per bottle	18.0000	69	0
67	Laughing Lumbe	16	1	24 - 12 oz bottles	14.0000	52	0
2	Chang	1	1	24 - 12 oz bottles	19.0000	17	40

3. Reload any changed schemas into Anzo by clicking the menu icon (⋮) in the **Actions** column for the schema and selecting **Reload Schema**. For example:

The screenshot shows the 'Schemas' panel with a context menu open for the 'emrdb' schema. The menu options are 'Add Table', 'Reload Schema', and 'Delete'.

4. Click the **Ingest** button. If the source has more than one schema, Anzo displays the select schema dialog box. In the drop-down list, select the schema to use, and then click **OK**. For example:

Select the schema you want to use

emrdb

CANCEL

OK

Anzo opens the Ingest dialog box. The options are populated with the values from the previous workflow configuration. For example:

Ingest

Source Data Connection

DB

Anzo Data Store

Store

Auto Map Engine Config

Local Sparkler Engine

☒ Select all tables

☐ Custom select

Advanced

CANCEL

SAVE

- Click **Advanced** to view additional configuration options. By default, the Ingest workflow is configured to use the Existing Model, and additional options are presented for controlling the regeneration of artifacts and the handling of property type mismatches. For example:

Advanced

☐ Create New Model

☒ Use Existing Model

Model

northwind - Auto

☐ Regenerate Entire Model

☒ Regenerate Mappings and Jobs

In case of property type mismatch

☐ Merge types using most permissive

☒ Add a new property with a different type

This list below describes the advanced options:

- **Regenerate Entire Model:** Selecting this option means that Anzo deletes all entities from the existing model and recreates them. The model that results from the current ingestion process will contain only the data from the current process.

#### Example

If a previous run generated a model that contains classes A, B, and C, and the current data contains Classes C, D, and E, selecting **Regenerate Entire Model** results in a model that contains only classes C, D, and E. If **Regenerate Entire Model** is NOT selected, the resulting model will contain classes A, B, C, D, and E.

- **Regenerate Mappings and Jobs:** Selecting this option means that Anzo deletes all entities from the existing mappings and jobs and recreates them. The artifacts that result from the current ingestion process will contain only the data from the current process.

#### Example

If a previous run generated mappings and jobs that contain tables A and B and the current run is ingesting tables C and D, selecting **Regenerate Mappings and Jobs** results in artifacts that contain only tables C and D. If **Regenerate Mappings and Jobs** is NOT selected, the resulting artifacts contain tables A, B, C, and D.

- **Merge types using most permissive:** Anzo looks at the inferred types in both versions of the Schema (the old and new versions) and chooses the type that covers all inputs. In most cases Anzo sets the type to **String**.
- **Add a new property with a different type:** If Anzo encounters a type mismatch, it adds a new property with the new type to the existing Model.

#### Note

When associating column names in the new schema with the existing model, the match is case-insensitive. Anzo matches the names based on the spelling. For example, "myInt" matches "MYint."

6. Click **Save**. Anzo updates the pipeline and regenerates or updates the model and mappings according to the options you specified.
7. In the main navigation menu under **Onboard**, click **Structured Data**. Then click the **Pipelines** tab.

8. Click the name of the pipeline you configured. Anzo displays the Pipeline Overview screen. For example:

The screenshot shows the 'Load DB - northwind to store' pipeline overview. At the top, there's a title bar with a 'Publish All' button and a bookmark icon. Below the title bar are tabs for 'Overview', 'Jobs', 'History', 'Versions', 'Discussion', and 'Sharing'. The 'Overview' tab is selected. The main content area is divided into two columns. The left column contains metadata: 'Description' (None), 'Engine Configuration' (Local Sparkler Engine), 'Graph datasource' (store), 'Dataset' (DB - northwind to store with a link to a CSI file), and 'Managed Editions'. The 'Managed Editions' section shows a table with one entry: 'Default Edition' with a description 'Contains the latest successfull...', a date '12/13/2022 10:17AM', and an actions menu. The right column contains a 'General' section with fields for 'Type' (Dataset Pipeline), 'Creator' (System Administrator), 'Updated' (Dec 13, 2022 10:02 AM), and 'Released' (Dec 13, 2022 10:02 AM). Below this is a link to 'http://cambridgesemantics.com/Project' with a copy icon.

**Load DB - northwind to store** Not Versioned Publish All Bookmark

**Overview** Jobs History Versions Discussion Sharing

Description  
None

Engine Configuration  
Local Sparkler Engine

Graph datasource  
store

Dataset  
DB - northwind to store <http://csi.com/FileBasedLinkedDataSet/45f4e4197688bbcccb46bd076c79ac12>

Managed Editions

Title	↑	Description	Most Recent Published Date	Actions
Default Edition		Contains the latest successfull...	12/13/2022 10:17AM	⋮

**General**

Type Dataset Pipeline

Creator System Administrator

Updated Dec 13, 2022 10:02 AM

Released Dec 13, 2022 10:02 AM

<http://cambridgesemantics.com/Project>

9. To run all of the jobs, click the **Publish All** button at the top of the screen. To publish a subset of the jobs, select the checkbox next to each job that you want to run and then click the **Publish** button above the list of jobs. Anzo runs the pipeline and generates the resulting file-based linked data set in a new subdirectory under the specified Anzo data store.

When the pipeline finishes, this run of the pipeline becomes the **Managed Edition**. The Managed Edition always contains the latest successfully published data for all of the jobs in the pipeline. For more information about editions, see [Managing Dataset Editions](#). If an existing graphmart contains the resulting dataset, refresh or reload the graphmart to load the modified data. See [Working with Graphmarts](#) for information about working with graphmarts.

## Ingesting a Data Source with a Metadata Dictionary

Follow the instructions below to ingest data for a schema that has an associated metadata dictionary.

### Tip

For instructions on ingesting data without a metadata dictionary, see [Ingesting a New Data Source](#) or [Re-Ingesting an Updated Data Source](#).

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:

2. On the Data Sources screen, click the name of the source for which you want to ingest data. Anzo displays the Tables screen for the source. For example:



The screenshot shows the Anzo DB interface. The top navigation bar includes 'Create Graphmart', 'Profile Data', 'Add To Dictionary', and 'Ingest'. The 'Tables' tab is selected. On the left, the 'Schemas' panel shows a list of schemas: 'emrdb' and 'northwind'. The main panel displays an 'Alphabetical list of products' table. The table has the following columns: ProductID (Int), ProductName (String), SupplierID (Int), CategoryID (Int), QuantityPerUnit (String), UnitPrice (Decimal), UnitsInStock (Short), and UnitsOnOrder (Short). The table contains 69 rows of data.

- Click the **Ingest** drop-down button and select **Ingest With Dictionary**. If the source has more than one schema, Anzo displays the select schema dialog box. In the drop-down list, select the schema to use, and then click **OK**. For example:

The screenshot shows a dialog box titled 'Select the schema you want to use'. It features a dropdown menu with 'emrdb' selected. Below the dropdown are two buttons: 'CANCEL' and 'OK'.

Anzo opens the Ingest With Dictionary dialog box, which lists the dictionary to use. If there is only one configured data store, the Anzo Data Store value is also auto-populated. In addition, if the default ETL Engine is configured for the system, the Auto Map Engine Config field is also populated (see [Configure the Default ETL Engine](#) in the Administration Guide). For example:

Ingest With Dictionary

Dictionary: emrdb
 

Anzo Data Store

ETL Engine

☒ Select all tables
 ☐ Custom select

Advanced

CANCEL

SAVE

- If necessary, click the **Anzo Data Store** field and select the data store for this pipeline.
- If necessary, click the **ETL Engine** field and select the ETL engine to use for this pipeline.
- By default, Anzo enables the **Select all tables** radio button to ingest the data for all tables in the schema. If you do not want to add all tables, click the **Custom select** radio button and then select each of the tables to add.
- To view model and dictionary options for this pipeline, expand the **Advanced** section of the dialog box:

**If this source has not been previously ingested and no model exists**, Anzo displays the following options:

Advanced

Create New Model:

Schema Ontology URI

Schema Class Prefix

Schema Property Prefix

☒ Include Unmatched Concepts
 ☒ Create single model property for shared property concepts

The list below describes the options:

- **Schema Ontology URI:** The URI for the model. When this field is blank, Anzo generates the model URI with the following format:

```
http://cambridgesemantics.com/ont/autogen/xx/<schema_name>
```

Where xx is a hash snippet based on the model's globally unique identifier (GUID). If you want to specify a different format, you can type that URI into the Schema Ontology URI field. For example, a URI such as `http://mycompany.com.ontology/movies` results in a model URI of `http://mycompany.com.ontology/movies`.

### Important

Make sure that Schema Ontology URI is unique. If the URI is not unique, this model will overwrite any existing model that uses this URI.

- **Schema Class Prefix:** The URI prefix format to use for classes in the model. When this field is blank, Anzo generates class URIs using the following format:

```
http://cambridgesemantics.com/ont/autogen/xx/<schema_name>#<class_name>
```

Where xx is a hash snippet based on the model's GUID. If you want to specify a different format for class URIs, type the prefix to use in this field. For example, a prefix such as

`http://mycompany.com.ontology/class` results in class URIs like

`http://mycompany.com.ontology/class#<class_name>`.

### Tip

Since you are specifying a prefix format, and the class name will be appended to the prefix, it is permissible to set Schema Class Prefix to the same value across schemas.

- **Schema Property Prefix:** The URI prefix format to use for properties in the model. When this field is blank, Anzo generates property URIs using the following format:

```
http://cambridgesemantics.com/ont/autogen/xx/<schema_name>#<class_name>_<property_name>
```

Where `xx` is a hash snippet based on the model's GUID. If you want to specify a different format for property URIs, type the prefix to use in this field. you can type that URI into the Schema Property Prefix field. For example, a prefix such as

`http://mycompany.com.ontology/property` results in property URIs like

`http://mycompany.com.ontology/property#<class_name>_<property_name>`.

#### Tip

Since you are specifying a prefix format, and the property name will be appended to the prefix, it is permissible to set Schema Property Prefix to the same value across schemas.

- **Include Unmatched Concepts:** This option specifies whether to ingest new data that does not map to the concepts that are defined in the dictionary. For example, imagine that the dictionary includes a class concept with 10 properties that map to columns in the schema. However, the new data contains 15 columns, 5 of which are not mapped to properties in the dictionary. If **Include Unmatched Concepts** is enabled, Anzo ingests the data for all 15 columns and updates the model to include the 5 unmatched properties. The 5 new properties are added to a new subclass of the class that is defined in the dictionary. If **Include Unmatched Concepts** is disabled, Anzo ingests only the 10 columns that match the concepts in the dictionary. New properties will not be added to the model.
- **Create single model property for shared property concepts:** If you have a shared property that exists in two or more classes, selecting this option means that Anzo will create one multi-domained property in the model instead multiple individual properties with a single domain. When this option is disabled, Anzo creates a separate property for each instance of the shared property.

**If this source has been previously ingested and a model exists**, Anzo displays the following options:

Advanced ^

☐ Create New Model ☒ Use Existing Model

Model: emrdb - emrdb - Auto

☒ Include Unmatched Concepts

☒ Create single model property for shared property concepts

☐ Replace Entire Ontology ☒ Regenerate Mappings and Jobs

The list below describes the options:

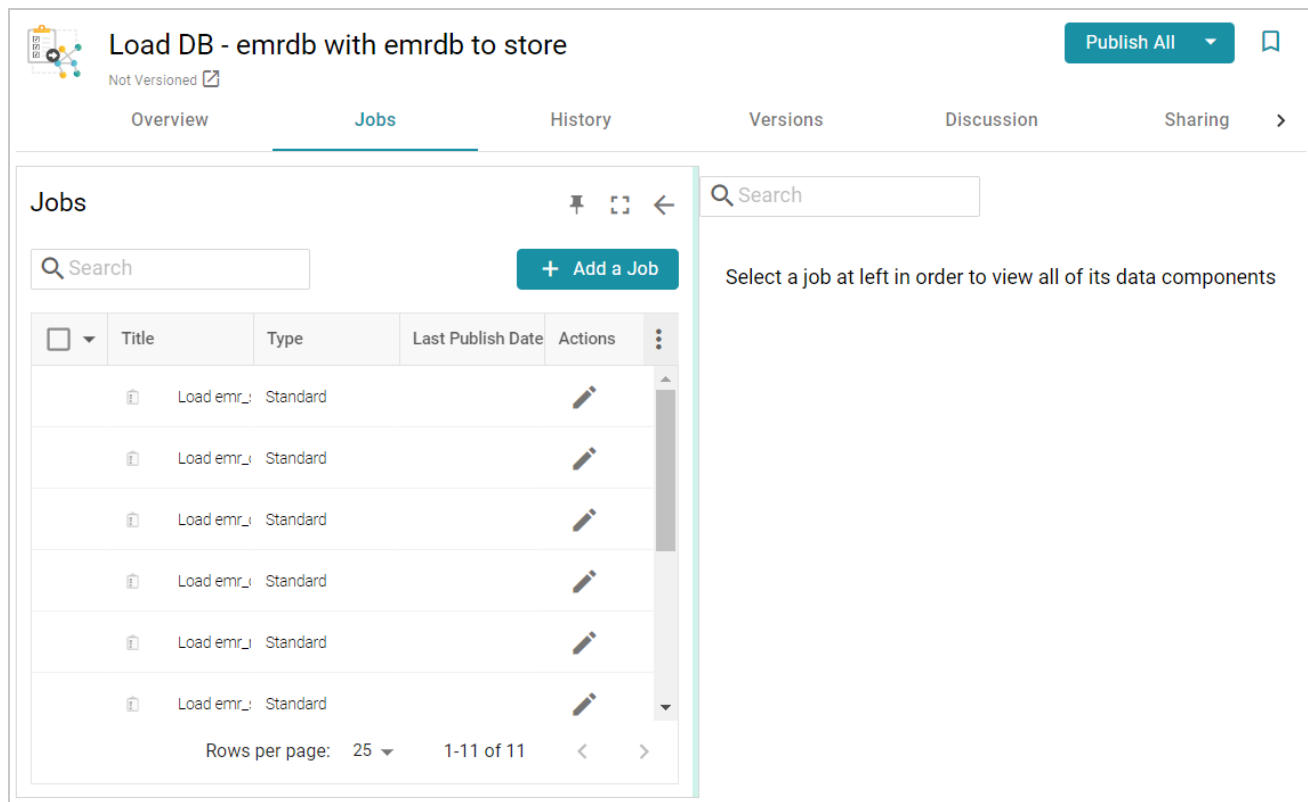
- **Use Existing Model:** Select this option if you want to reuse the existing model. Anzo displays the name of the model that will be used and presents additional model-related options at the bottom of the screen.
- **Include Unmatched Concepts:** This option specifies whether to ingest new data that does not map to the concepts that are defined in the dictionary. For example, imagine that the dictionary includes a class concept with 10 properties that map to columns in the schema. However, the new data contains 15 columns, 5 of which are not mapped to properties in the dictionary. If **Include Unmatched Concepts** is enabled, Anzo ingests the data for all 15 columns and updates the model to include the 5 unmatched properties. The 5 new properties are added to a new subclass of the class that is defined in the dictionary. If **Include Unmatched Concepts** is disabled, Anzo ingests only the 10 columns that match the concepts in the dictionary. New properties will not be added to the model.
- **Create single model property for shared property concepts:** If you have a shared property that exists in two or more classes, selecting this option means that Anzo will create one multi-domained property in the model instead multiple individual properties with a single domain. When this option is disabled, Anzo creates a separate property for each instance of the shared property.
- **Replace Entire Ontology:** Selecting this option means that Anzo deletes all entities from the existing model and recreates them. The model that results from the current ingestion workflow will contain only the data from the current process. For example, if a previous run generated a model that contains classes A, B, and C, and the current data contains Classes C, D, and E, selecting **Replace Entire Ontology** results in a model that contains only classes C, D, and E. If **Replace Entire Ontology** is NOT selected, the resulting model will contain classes A, B, C, D, and E.

- **Regenerate Mappings and Jobs:** Selecting this option means that Anzo deletes all entities from the existing mappings and jobs and recreates them. The artifacts that result from the current ingestion workflow will contain only the data from the current process. For example, if a previous run generated mappings and jobs that contain tables A and B and the current run is ingesting tables C and D, selecting **Regenerate Mappings and Jobs** results in artifacts that contain only tables C and D. If **Regenerate Mappings and Jobs** is NOT selected, the resulting artifacts contain tables A, B, C, and D.

8. Click **Save**. Anzo creates a pipeline (or updates the existing one) and generates or updates the model and mappings according to the options you specified.
9. In the main navigation menu under **Onboard**, click **Structured Data**. Then click the **Pipelines** tab.
10. Click the name of the pipeline to run. Anzo displays the pipeline overview screen. For example:

The screenshot shows the 'Overview' tab of a pipeline named 'Load DB - emrdb with emrdb to Store'. The pipeline is marked as 'Not Versioned'. The left sidebar lists the pipeline's components: Description (None), Engine Configuration (Local Sparkler En...), and Graph datasource (Store). The right panel displays 'General' information: Type (Dataset Pipeline), Creator (System Administrator), Updated (8 minutes ago), and Released (8 minutes ago). A URL 'http://cambridgesemantics.com/Proje...' is also visible, along with a 'Tags' section containing 'Auto-Gen'. A 'Publish All' button is located in the top right corner.

11. If you would like to see the jobs that Anzo created for this data source, click the **Jobs** tab. The jobs are listed on the left side of the screen. A job exists for each of the tables that were imported. If this pipeline has not been published previously, the right side of the screen remains blank. After the jobs are run, selecting a job from the list displays its history on the right. For example, the image below shows a new pipeline that has not been published:



12. To run all of the jobs, click the **Publish All** button at the top of the screen. To publish a subset of the jobs, select the checkbox next to each job that you want to run and then click the **Publish** button above the list of jobs. Anzo runs the pipeline and generates the resulting file-based linked data set in a new subdirectory under the specified Anzo data store.

When the pipeline finishes, this run of the pipeline becomes the **Managed Edition**. The Managed Edition always contains the latest successfully published data for all of the jobs in the pipeline. If one or more of the jobs failed, those jobs are excluded from the edition. If you publish the failed jobs at a later date or you create and publish additional jobs in the pipeline, the data from those jobs is also added to the Managed Edition. For more information about editions, see [Managing Dataset Editions](#).

The new dataset also becomes available in the Datasets catalog. From the catalog, you can create a new graphmart from the dataset or add the dataset to an existing graphmart. See [Creating a Graphmart from a Dataset](#) or [Adding a Dataset to an Existing Graphmart](#) for instructions.

## Managing Data Source Metadata

The topics in this section provide information about working with data source metadata.

[Creating a Metadata Dictionary \(Preview\)](#)

## Related Topics

### Creating a Metadata Dictionary (Preview)

Metadata dictionaries are similar to models in that they define the desired business meaning and structure of the data after it is onboarded to Anzo and converted to the graph model. Unlike models, though, Metadata dictionaries offer maximum flexibility for normalizing the data that comes from various sources and structures. A single dictionary can be used to link conceptually identical elements (columns) from many different schemas, independent of any models and mappings. The metadata dictionary structure becomes the basis for creating and reusing models and mappings. As model and mapping artifacts are generated, deleted, and recreated over time, the growing body of information about business meaning and the concepts that link source schema elements to properties in the model remain available in the metadata dictionaries.

#### Note

The Metadata Dictionary functionality is a **Preview** release. Features are considered "Preview" when the implementation has recently been incorporated into the product and significant development may still be underway. Cambridge Semantics recommends that you do not plan to use metadata dictionaries in a production environment.

This topic provides instructions for creating and managing metadata dictionaries.

- [Creating a Metadata Dictionary from a Schema](#)
- [Creating a Metadata Dictionary from Scratch](#)
- [Defining Concepts in a Metadata Dictionary](#)

### Creating a Metadata Dictionary from a Schema

Follow the instructions below to create a new metadata dictionary from a schema.

#### Tip

The steps below start with viewing a Schema and then adding that Schema to a new dictionary. That method allows for flexibility in choosing which Schema tables are added to the dictionary. However, you can also create a dictionary first and then add an entire Schema to it. To do so, select **Metadata Hub** from the **Onboard** menu. On the Dictionaries screen, click the **Create** button and select **From Schema**. In the Create Metadata Dictionary dialog box, select the Schema to add to the new dictionary.

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:

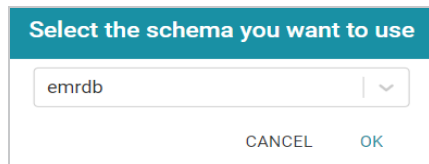


Data Sources   Schemas   Mappings   Pipelines										
<div> <div> <div></div> <div>Search</div> </div> <div> <div></div> <div></div> </div> </div> <div>Add Data Source</div>										
	Title	Description	Type	Schema	Creator	Created Date	Updated Date	Actions		
	DB		Database Data...	emrddb, new_...	System Admin...	Dec 8, 2022	Dec 8, 2022			
	Movies		CSV Data Sour...	Movies	System Admin...	Dec 8, 2022	Dec 8, 2022			
	Tickets		CSV Data Sour...	Tickets	System Admin...	Dec 9, 2022	Dec 12, 2022			
<div>Rows per page: 25 1-3 of 3</div>										

- On the Data Sources screen, click the name of the Data Source for which you want to create a dictionary. Anzo displays the Tables screen for the source. For example:

<div> <div>MySQL DB</div> <div>Not Versioned</div> <div>Profile Data</div> <div>Add To Dictionary</div> <div>Ingest</div> <div></div> </div>										
<div> <div>Overview</div> <div>Tables</div> <div>Versions</div> <div>Discussion</div> <div>Sharing</div> </div>										
<div> <div>Schema Metrics</div> <div> <div>Create Schemas From Query</div> <div>Import Schemas</div> </div> <div> <div>Search</div> <div></div> </div> <div> <div>Schema Name</div> <div>Table Name</div> <div>Type</div> <div>Actions</div> </div> <div> <div>northwind</div> <div></div> </div> </div>										
<div> <div>Schema Details</div> <div>Select a table on the left to see details here.</div> </div>										

- Click the **Add To Dictionary** button. If the source has more than one schema, Anzo displays the select schema dialog box. In the drop-down list, select the schema to add to the dictionary, and then click **OK**. For example:

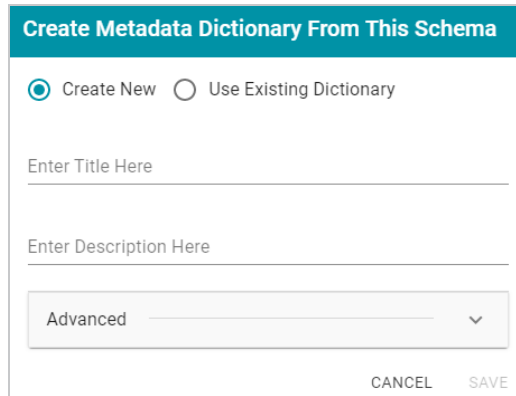


Select the schema you want to use

emrdb

CANCEL OK

Anzo opens the Create Metadata Dictionary From This Schema dialog box.



Create Metadata Dictionary From This Schema

☒ Create New ☐ Use Existing Dictionary

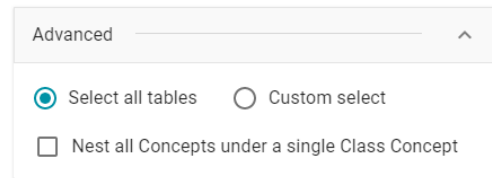
Enter Title Here

Enter Description Here

Advanced

CANCEL SAVE

4. In the dialog box, leave the **Create New** radio button selected.
5. Enter a name for the dictionary in the **Title** field and specify an optional description in the **Description** field.
6. To configure additional options, such as limiting the schema tables that are added to the dictionary, click **Advanced** to display the advanced options. The list below describes each option.



Advanced

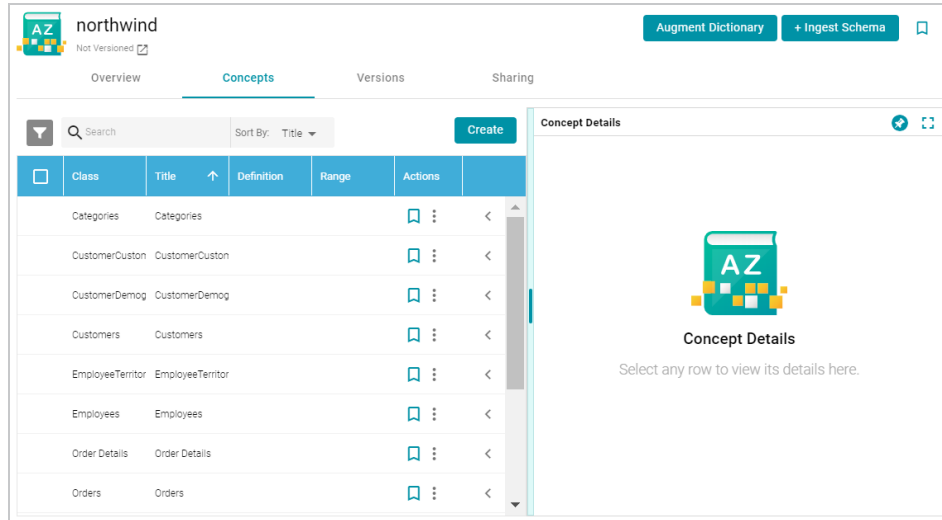
☒ Select all tables ☐ Custom select

☐ Nest all Concepts under a single Class Concept

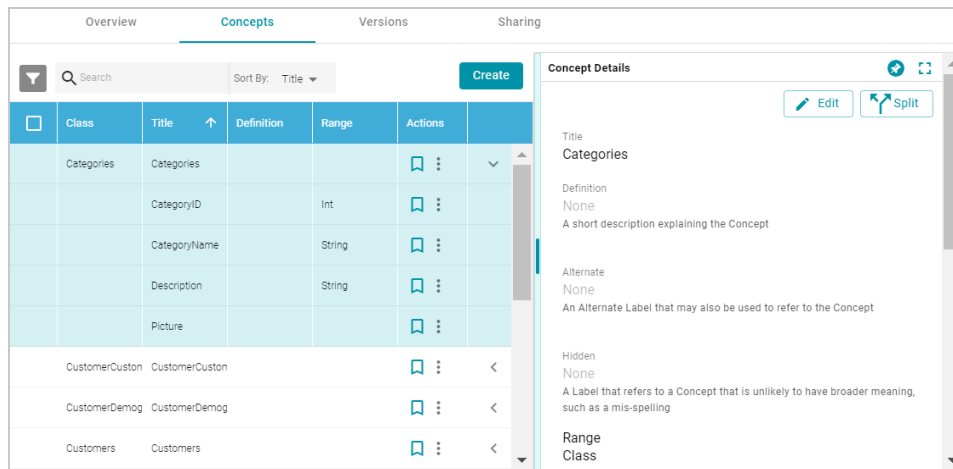
- **Select all tables:** Select this option if you want to include all of the schema tables in the dictionary. Each table becomes a class concept, and each column in the table becomes a property concept under the class.
- **Custom select:** Select this option if you want to include a subset of the schema tables in the dictionary. Clicking **Custom select** displays the list of schema tables. Select the checkbox for each table that you want to add to the dictionary.
- **Nest all Concepts under a single Class Concept:** If all of the schema tables contain the same type of properties and could belong in the same class, you can select this option to merge all of the properties from all of the tables into a single class concept. For example, if the source is multiple CSV files where each file (table in the schema) contains the data for a single study in a

group of studies, enabling this option would merge all of the properties from each file into one class. Anzo uses one of table names as the name for the class concept in the dictionary.

7. Click **Save**. Anzo creates the dictionary and displays a message that asks if you want to view the new dictionary. Click **Go to Dictionary** to open the dictionary in the Metadata Hub. The Concept tab is displayed. For example:



8. Click a row in the list of concepts on the left to view the concept details on the right side of the screen. Click the < character in the table to expand a class concept and view its property concepts. For example:

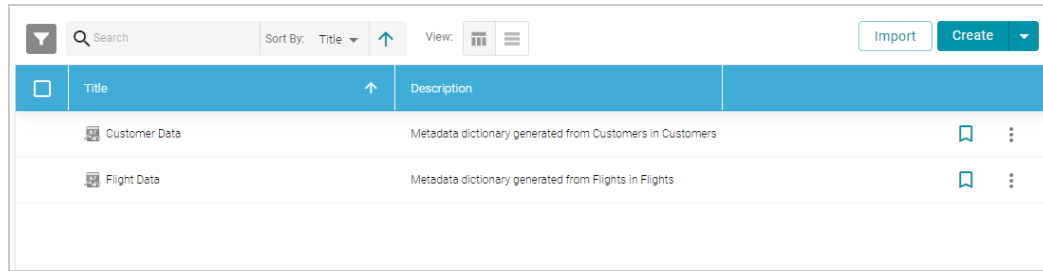


Create and edit concepts as needed. See [Defining Concepts in a Metadata Dictionary](#) below for information about working with concepts.

## Creating a Metadata Dictionary from Scratch

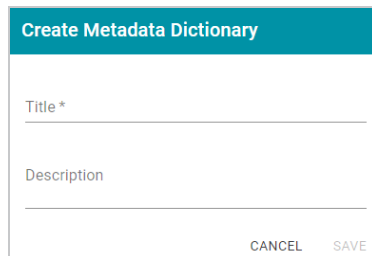
Follow the instructions below to create a metadata dictionary from scratch.

1. In the Anzo application, expand the **Onboard** menu and click **Metadata Hub**. Anzo displays the Dictionaries screen. For example:



	Title	Description	
	Customer Data	Metadata dictionary generated from Customers in Customers	
	Flight Data	Metadata dictionary generated from Flights in Flights	

2. Click the **Create** button at the top of the screen, and select **Manual**. Anzo displays the Create Metadata Dictionary dialog box.



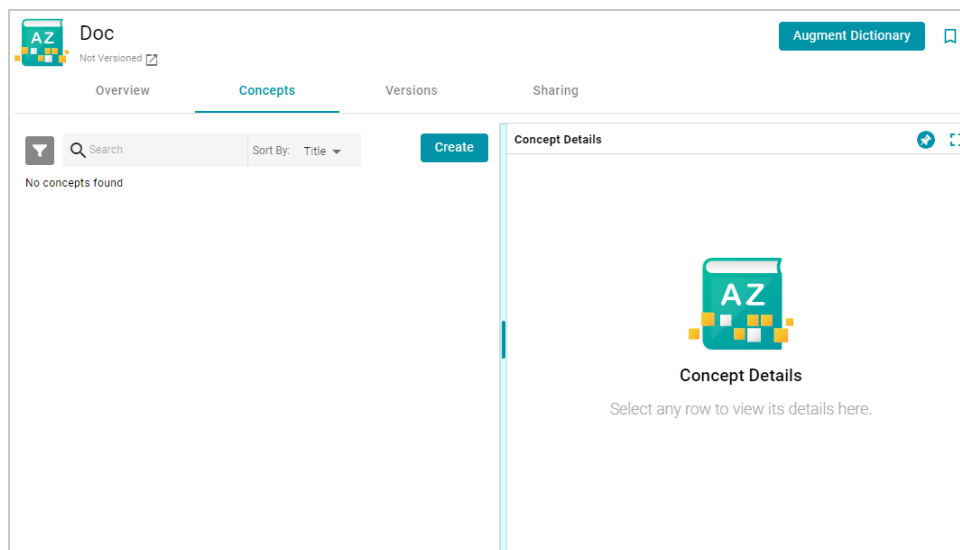
**Create Metadata Dictionary**

Title \*

Description

CANCEL SAVE

3. Type a name for the dictionary in the **Title** field and supply an optional description in the **Description** field.
4. Click **Save** to create the new dictionary. Anzo saves the dictionary and displays the empty Concepts tab. For example:



**Doc** Not Versioned **Augment Dictionary**

Overview **Concepts** Versions Sharing

**Search** Sort By: Title **Create**

No concepts found

**Concept Details**

**AZ**

**Concept Details**

Select any row to view its details here.

Create and edit concepts as needed. See [Defining Concepts in a Metadata Dictionary](#) below for information about working with concepts.

















## Defining Concepts in a Metadata Dictionary

This section provides examples and instructions for defining the concepts in a data dictionary.

- [Merging Concepts](#)
- [Creating a Concept](#)
- [Splitting a Concept](#)

### Merging Concepts

It is common for schemas, especially relational database schemas, to have multiple tables with foreign key relationships. When the schema is added to a dictionary, each table becomes a class concept, resulting in a dictionary that includes multiple concepts with different names but the same meaning. To simplify the data model, similar concepts can be consolidated into one concept. For example, the concept list below has a "CustomerCustomerDemo" class and a class called "CustomerDemographics."

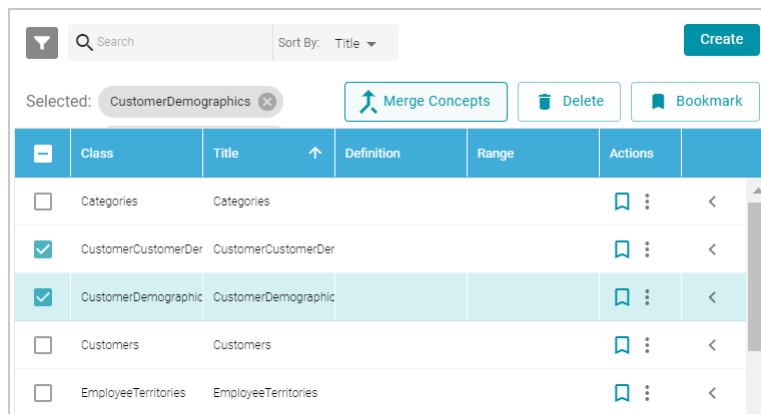
	Class	Title		Definition	Range	Actions	
<input type="checkbox"/>	Categories	Categories				 	<
<input checked="" type="checkbox"/>	CustomerCustomerDemo	CustomerCustomerDemo				 	<
<input checked="" type="checkbox"/>	CustomerDemographics	CustomerDemographics				 	<
<input type="checkbox"/>	Customers	Customers				 	<
<input type="checkbox"/>	EmployeeTerritories	EmployeeTerritories				 	<
<input type="checkbox"/>	Employees	Employees				 	<
<input type="checkbox"/>	Order Details	Order Details				 	<

The two customer demo concepts share properties such as CustomerID and CustomerTypeID, which are foreign key relationships across the tables/classes. The classes can be merged into a single concept, creating one class in the model that contains all of the customer demographics-related properties.

**Note** Modifications that you make to a data dictionary do not change the source schema.

#### To merge concepts

1. Select the checkbox next to each concept that you want to merge, and then click the **Merge Concepts** button above the table. For example:



Anzo displays the Merge Concepts dialog box, which lists the classes to merge and enables you to specify the title and description of the new, merged class. For example:

Merge Concepts - Select or Add Preferences

	Concept 1 - CustomerDemographics	Concept 2 - CustomerCustomerDemo	Merged Concepts
Title	<input type="checkbox"/> CustomerDemographics	<input type="checkbox"/> CustomerCustomerDemo	Title
Definition			Definition

CANCEL
SAVE

- On the Merge Concepts screen, if you want to name the merged class with one of the existing class names, select the checkbox next to that class. The **Title** field on the right is populated with that name, and you have the option to edit it. If you do not want to use any existing titles, type a new title in the **Title** field.
- In the **Definition** field, type an optional description for the class. For example:

Merge Concepts - Select or Add Preferences

	Concept 1 - CustomerDemographics	Concept 2 - CustomerCustomerDemo	Merged Concepts
Title	<input checked="" type="checkbox"/> CustomerDemographics	<input type="checkbox"/> CustomerCustomerDemo	Title CustomerDemographics
Definition			Definition Customer demographics concept

CANCEL
SAVE

- Click **Save** to merge the concepts. Anzo displays a confirmation dialog box that lists the concepts that will be merged and asks if you want to proceed. Click **OK** to complete the merge.

- When the merge is complete, the concept list is displayed with the changes. You can select the merged class to view and modify concept details on the right side of the screen. For example, the image below shows the details for the merged CustomerDemographics concept. The names of the concepts that were merged to CustomerDemographics are listed in the **Alternate** field. Sources that include those labels, "CustomerDemographics" and "CustomerCustomerDemo," will be mapped to "CustomerDemographics" in the model. You can edit the Alternate field to add other labels that might come from future source schemas.

The screenshot shows the Anzo interface. On the left is a table of concepts, and on the right is a 'Concept Details' panel for 'CustomerDemographics'.

Class	Title	Definition	Range	Actions
Categories	Categories			
CustomerDemog	CustomerDemog	Customer demog		
	CustomerDesc		String	
	CustomerID		String	
	CustomerTypel		String	
	FK_CustomerC		Customers	
Customers	Customers			
EmployeeTerritor	EmployeeTerritor			

**Concept Details**

**Title**  
CustomerDemographics

**Definition**  
Customer demographics concept  
A short description explaining the Concept

**Alternate**  
CustomerCustomerDemo  
An Alternate Label that may also be used to refer to the Concept

**Hidden**  
None  
A Label that refers to a Concept that is unlikely to have broader meaning, such as a mis-spelling

**Range**  
Class

**Primary Keys:** (2) Create

## Creating a Concept

Follow the instructions below to create a new class or property concept in a data dictionary.

- To add a new concept, click the **Create** button on the right side of the screen. Anzo displays the Create New Concept screen.

2. Under **New Concept Type**, select the radio button for the type of concept to create:
  - **Data Property:** A data property has an object that is a literal value. For example, a property like `FirstName` is a data property. Its object has a value such as "Jane."
  - **Object Property:** An object property has an object that relates a class to another class. These types of relationships are usually foreign keys in the source. For example, a property like `CustomerID` might relate the `Customers` class to the `Orders` class.
  - **Class:** A class concept contains a group of related properties, such as a table name from a source schema.
3. Depending on the type of concept you are creating, specify the appropriate required and optional details:
  - **Title:** The name for this class or property concept.
  - **Definition:** An optional description for the new concept.
  - **Alternate:** An optional list of labels that should map to this new class or property concept.
  - **Hidden:** An optional list of labels that should be hidden in the data model that is generated from this dictionary.
  - **Range:** For property concepts, this required field specifies the data type for the property.
  - **Class:** For property concepts, this required field lists the class or classes the property belongs to.



For example, the image below creates a data property for reviews of orders. The new property is named ReviewText and "Comment," "Comments," and "Review" are included as Alternate labels so that those properties in source schemas are mapped to ReviewText in the model when the data is onboarded.

**Create New Concept**

New Concept Type:  
☒ Data Property ☐ Object Property ☐ Class

Title \*  
ReviewText

Definition

Alternate  
Comments Comment Review

Alternate ADD

Hidden

Range  
String

Class  
Reviews

CANCEL SAVE

4. Click **Save** to add the new concept to the dictionary.

## Splitting a Concept

If you determine that one concept should be separated into multiple concepts, you can quickly split the concept and create an additional one by moving any of the original concept's elements to a new concept. Follow the instructions below to split a concept.

1. In the list of concepts, select the row for concept that you want to split and then click the **Split** button in the Concept Details. Anzo displays the Split Concept screen, which lists the original concept on the left and the new concept on the right. For example:

Split Concept: flights10k

	Original Concept - flights10k (Split 1)	Split Concept												
Title	flights10k	Enter Title Here												
Alternate		You can select and drag the labels from the left panel (from either hidden or alternate) Click on multiple chips to drag more than one at a time before dragging.												
Hidden		You can select and drag the labels from the left panel (from either hidden or alternate) Click on multiple chips to drag more than one at a time before dragging.												
Schema Locations	Schema Locations: (1) <table> <tr> <th>Schema</th> <th>Table</th> </tr> <tr> <td>Flights</td> <td>flights10k</td> </tr> </table>	Schema	Table	Flights	flights10k	Schema Locations: (none) <table> <tr> <th>Schema</th> <th>Table</th> </tr> <tr> <td colspan="2">You can select and drag locations from the original concept to here</td> </tr> </table>	Schema	Table	You can select and drag locations from the original concept to here					
Schema	Table													
Flights	flights10k													
Schema	Table													
You can select and drag locations from the original concept to here														
Property Concepts	Property Concepts: (31) <table> <tr> <th>Title</th> <th>Definition</th> </tr> <tr> <td>SCHEDULED_ARRIVAL</td> <td></td> </tr> <tr> <td>SECURITY_DELAY</td> <td></td> </tr> <tr> <td>SCHEDULED_TIME</td> <td></td> </tr> </table>	Title	Definition	SCHEDULED_ARRIVAL		SECURITY_DELAY		SCHEDULED_TIME		Property Concepts: (none) <table> <tr> <th>Title</th> <th>Definition</th> </tr> <tr> <td colspan="2">You can select and drag property concepts from the original class concept to here</td> </tr> </table>	Title	Definition	You can select and drag property concepts from the original class concept to here	
Title	Definition													
SCHEDULED_ARRIVAL														
SECURITY_DELAY														
SCHEDULED_TIME														
Title	Definition													
You can select and drag property concepts from the original class concept to here														

CANCEL SAVE

- Under **Split Concept**, type a name for the new concept in the **Title** field.
- For the rest of the fields, you can drag elements from the Original Concept to the Split Concept. For example, the image below creates a new Delays class concept and moves the delay-related properties from the original concept to the new concept.

Split Concept: flights10k

	Original Concept - flights10k (Split 1)	Split Concept																
Title	flights10k	<div>Enter Title Here</div> <div>delays</div>																
Alternate		<div>You can select and drag the labels from the left panel (from either hidden or alternate)</div> <div>Click on multiple chips to drag more than one at a time before dragging.</div>																
Hidden		<div>You can select and drag the labels from the left panel (from either hidden or alternate)</div> <div>Click on multiple chips to drag more than one at a time before dragging.</div>																
Schema Locations	<div>Schema Locations: (1)</div> <table> <tr> <th>Schema</th><th>Table</th></tr> <tr> <td>Flights</td><td>flights10k</td></tr> </table>	Schema	Table	Flights	flights10k	<div>Schema Locations: (none)</div> <table> <tr> <th>Schema</th><th>Table</th></tr> <tr> <td colspan="2"> <div>You can select and drag locations from the original concept to here</div> </td></tr> </table>	Schema	Table	<div>You can select and drag locations from the original concept to here</div>									
Schema	Table																	
Flights	flights10k																	
Schema	Table																	
<div>You can select and drag locations from the original concept to here</div>																		
Property Concepts	<div>Property Concepts: (24)</div> <table> <tr> <th>Title</th><th>Definition</th></tr> <tr><td>SCHEDULED_ARRIVAL</td><td></td></tr> <tr><td>SCHEDULED_TIME</td><td></td></tr> <tr><td>TAXI_OUT</td><td></td></tr> </table>	Title	Definition	SCHEDULED_ARRIVAL		SCHEDULED_TIME		TAXI_OUT		<div>Property Concepts: (7)</div> <table> <tr> <th>Title</th><th>Definition</th></tr> <tr><td>AIRLINE_DELAY</td><td></td></tr> <tr><td>ARRIVAL_DELAY</td><td></td></tr> <tr><td>WEATHER_DELAY</td><td></td></tr> </table>	Title	Definition	AIRLINE_DELAY		ARRIVAL_DELAY		WEATHER_DELAY	
Title	Definition																	
SCHEDULED_ARRIVAL																		
SCHEDULED_TIME																		
TAXI_OUT																		
Title	Definition																	
AIRLINE_DELAY																		
ARRIVAL_DELAY																		
WEATHER_DELAY																		

CANCEL SAVE

- When you are finished configuring the new concept, click **Save**. Anzo displays a confirmation dialog box that lists the concepts that will be split and asks if you want to proceed. Click **OK** to complete the split and return to the Concepts screen.

For instructions on onboarding data using a data dictionary, see [Ingesting a Data Source with a Metadata Dictionary](#).

### Note

If you make changes to a dictionary after the schema has been ingested, you must re-ingest the schema to incorporate the dictionary changes. You can click the **Ingest Schema** button at the top of the dictionary screen. Or you can follow the instructions in [Ingesting a Data Source with a Metadata Dictionary](#) to re-ingest the data with the modified dictionary.

## Working with Mappings

Anzo enables you to map and transform your data with the Anzo for Office plugin for Microsoft Excel. The topics in this section provide information about creating and editing the mappings that describe the relationships between your schemas and models.

### Tip

For instructions on installing Anzo for Office, see [Installing the Anzo for Office Plugin](#) in the Deployment Guide.

[Creating a New Mapping](#)

[Configuring Mappings to Ingest a Subset of the Source Data](#)

[Transforming Data in Mappings](#)

[Supported Mapping Functions](#)

## Creating a New Mapping

This topic provides instructions for using the Anzo for Office Excel plugin to create a new basic mapping. Typically users create one mapping for each target and source pair. For example, if you have a project that ingests data from 10 tables in a source or 10 CSV files, the project will likely include 10 mappings. You can create mappings where multiple sources map to one target, but one mapping cannot include multiple targets.

Follow these steps to create a new mapping:

1. [Create a Mapping and Select References](#)
2. [Define the Source for the Mapping](#)
3. [Define the Target for the Mapping](#)
4. [Map the Source Elements to Target Elements](#)

## Create a Mapping and Select References

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Then click the **Mappings** tab. Anzo displays the Mappings screen, which lists any existing mappings. For example:

Data Sources

Schemas

Mappings

Pipelines

Sort By: Title

View:

Add Mappings

<input type="checkbox"/>	Title	Description	Updated Date	Tags	Actions
<input type="checkbox"/>	DB - emrddb - emr_acti...		Jun 18, 2020	Auto-Gen	
<input type="checkbox"/>	DB - emrddb - emr_com...		Jun 18, 2020	Auto-Gen	
<input type="checkbox"/>	DB - emrddb - emr_com...		Jun 18, 2020	Auto-Gen	
<input type="checkbox"/>	DB - emrddb - emr_med...		Jun 18, 2020	Auto-Gen	
<input type="checkbox"/>	DB - emrddb - emr_med...		Jun 18, 2020	Auto-Gen	
<input type="checkbox"/>	DB - emrddb - emr_obs...		Jun 18, 2020	Auto-Gen	
<input type="checkbox"/>	DB - emrddb - emr_obs...		Jun 18, 2020	Auto-Gen	
<input type="checkbox"/>	DB - emrddb - emr_pat...		Jun 18, 2020	Auto-Gen	

Rows per page: 20

1-20 of 34

- Click the **Add Mappings** button at the top of the screen and select **Add Mappings**. Anzo opens the Anzo for Office plugin in Microsoft Excel and prompts you to enter the Anzo server connection information.
- Provide your server connection and login information and then click the arrow icon () to connect to the server and open the mapping tool. Anzo displays the New Mapping dialog box.

New Mapping

Title

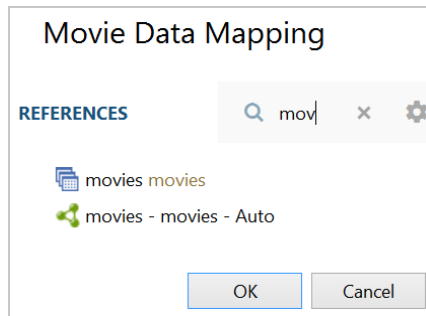
REFERENCES

Add references to schemas and ontologies

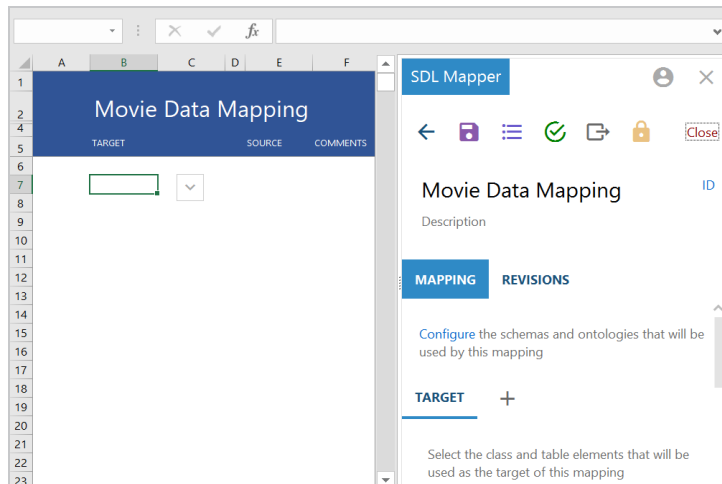
OK

Cancel

- In the **Title** field, type a name for the mapping.
- In the **References** field, select the schemas and models that the new mapping should have access to. Type a value to search for available sources. If you want to include system items in the search, you can click the cog icon () to select it. Anzo lists the sources that match the search text.



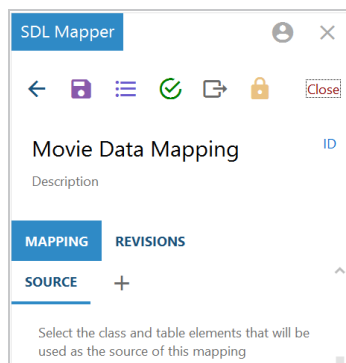
6. Hover over a source in the list of references to display the plus icon (+). Click the icon for each source that you want to add as a reference, and then click **OK**. The new mapping opens. The mapping workbook is in the center pane and the mapping menu and configuration details are displayed in the right pane. For example:



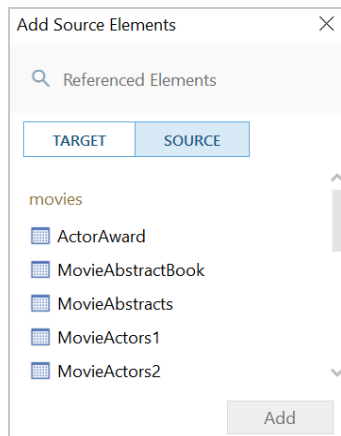
## Define the Source for the Mapping

Complete these steps to define the source to use for the mapping:

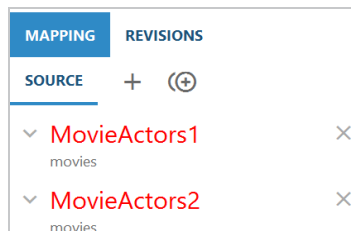
1. In the right pane, scroll to the **Source** section in the **Mapping** tab.



2. Click the plus icon (+) next to **Source**. Anzo opens the Add Source Elements dialog box.



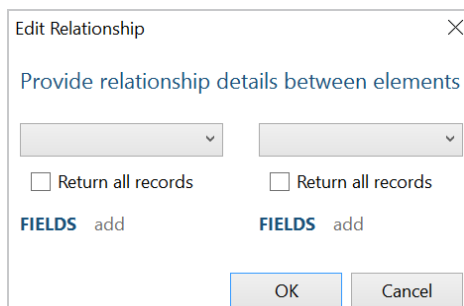
3. Select a source to add to the mapping and click **Add**. If you want to add additional sources, select another source and click **Add** again. Anzo adds the source to the list under Source. You can click the X icon to the right of a source name to delete that source from the list.



4. When you are finished adding sources, close the Add Source dialog box.
5. If you added one source, proceed to the next step. If you added multiple sources, Anzo displays an Add Relationship icon ((+)) next to the plus icon so that you can create a relationship between the sources by specifying the join criteria.

Follow these steps to specify relationships:

- a. Click the Add Relationship icon ((+)). Anzo displays the Edit Relationship dialog box.



- b. In the drop-down lists at the top of the dialog box, select one source on the left and the other source on the right. Anzo displays the fields under each source.

Provide relationship details between elements

MovieActors1 ▼ MovieActors2 ▼

☐ Return all records ☐ Return all records

**FIELDS** add **FIELDS** add

MovieID MovieID  
MovieTitle MovieTitle  
ActorID ActorID  
ActorName ActorName  
ActorCategory

OK Cancel

- c. Determine which field to join on from each source, and then select those fields. Double-click a field on the left to select it, then double-click the join field on the right. You can use the **Return all records** check boxes above the field lists to specify whether to return all records from either the right or the left side if no matching field is found.

Provide relationship details between elements

MovieActors1 ▼ MovieActors2 ▼

☐ Return all records ☒ Return all records

**FIELDS** add remove **FIELDS** add remove

ActorID ActorID

OK Cancel

- d. Click **OK** to create the relationship and close the Edit Relationship dialog box. Anzo lists the relationship definition at the bottom of the right pane.

SOURCE + (+)

▼ **MovieActors1**  
movies

▼ **MovieActors2**  
movies

SOURCE RELATIONSHIPS

**MovieActors1** ↔ **MovieActors2**  
ActorID = ActorID



If you have additional sources to join, click the Add Relationship icon again and repeat these steps to relate each source.

6. Click the save icon (💾) at the top of the mapping configuration pane to save the mapping changes.

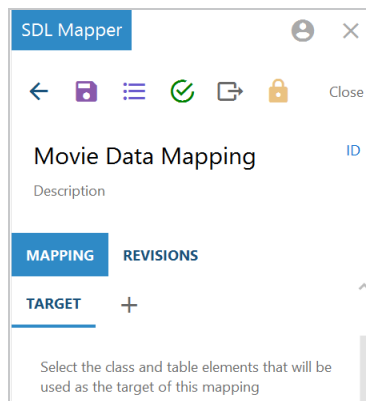
## Define the Target for the Mapping

Complete the steps below to define the class and table elements that the mapping should target.

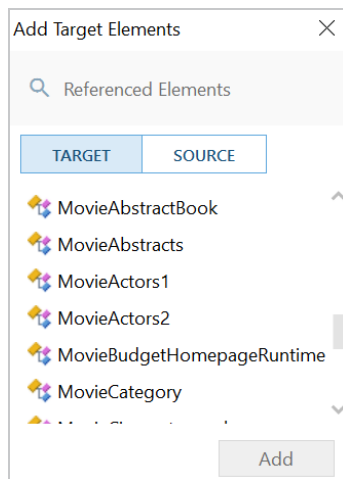
### Tip

For information about setting up parameters to ingest a subset of the source data, see [Configuring Mappings to Ingest a Subset of the Source Data](#).

1. In the right pane, scroll to the **Target** section in the **Mapping** tab.




2. Click the plus icon (+) next to **Target**. Anzo opens the Add Target Elements dialog box.



3. Select the model or table that you want to map to and click **Add**. Anzo adds the target to the Target list in the right pane and populates the workbook with the target elements. For example:

Movie Data Mapping	
TARGET	SOURCE
MovieActors	MovieActors1, MovieActors2
ActorID	Int
ActorName	String
MovieID	Int
MovieTitle	String

4. Close the Add Target dialog box and click the save icon (  ) at the top of the mapping configuration pane to save the mapping changes.

## Map the Source Elements to Target Elements

In the workbook, follow these steps to complete the mapping by specifying which source element maps to each target element:

1. Click in a source cell next to a target field. Anzo displays the **Source Mapping** text box below the cell.

SOURCE	
	MovieActors2, MovieActors1
String =	
Int =	
String =	Source Mapping ^
Int =	
String =	

2. In the Source Mapping text box, start typing the source field name. Anzo displays a list of fields that match the text.

SOURCE	
	MovieActors2, MovieActors1
String =	
Int =	
String =	Actor ^
Int =	
String =	<div> ActorID  ActorID  ActorName  ActorName  ActorCategory </div>

### Important

Type all values in the Source Mapping text box. Do not type in any cells. Any text typed in a cell directly is invalid and can cause issues with the mapping.

3. In the list of results, click the source field that maps to the target. Anzo adds the field name to the cell and opens the Source Mapping text box for the next source.

SOURCE

MovieActors2, MovieActors1

String = ActorCategory

Int =

String =

Int =

String =

Source Mapping ^

- If you chose a model or ontology as the target, some of the classes in the target likely include additional properties or related classes. In the workbook, the properties for related classes in are indented the target list. For example:

TARGET
emr_medication_ibfk_5
SPECIALTYID
emr_medication_ibfk_6
STUDYID
MEDICATIONACTIVEFLAG
MEDICATIONACTIVITYID
MEDICATIONAGE
MEDICATIONCREATEDDATE

In the example, SPECIALTYID and STUDYID are object properties in the related emr\_medication\_ibfk\_5 and emr\_medication\_ibfk\_6 classes.

To map the source for these properties, first map the source for the class and then the source for the object properties becomes available in the mapping. For example:

TARGET	SOURCE
emr_medication_ibfk_5	emr_specialty → emr_specialty
SPECIALTYID	Int = MEDICATIONSPECIALTYID
emr_medication_ibfk_6	emr_study → emr_study
STUDYID	Int = MEDICATIONSTUDYID
MEDICATIONACTIVEFLAG	String = MEDICATIONACTIVEFLAG
MEDICATIONACTIVITYID	Int = MEDICATIONACTIVITYID
MEDICATIONAGE	Int = MEDICATIONAGE
MEDICATIONCREATEDDATE	Date = MEDICATIONCREATEDDATE

In the example, once emr\_medication\_ibfk\_5 is mapped to the emr\_specialty source class or table and emr\_medication\_ibfk\_6 is mapped to emr\_study, the source for the SPECIALTYID and STUDYID object properties are available to map.

- Complete the mapping by entering the appropriate source for each target that you want to map. You do not have to enter source for all targets. For information about using functions to transform the source data, see [Transforming Data in Mappings](#).

### Note

To enter a literal value in the source mapping, type the value in the Source Mapping text box and then press **Enter**. The literal value is added to the cell with green text to distinguish it from fields.

6. If you want to add a new target and source pair to the mapping, click the target field that is above the cell where you want to add the new target. Then click the lines icon (≡) that appears to the left of the target name. Click the Add icon (+) that becomes available and choose the target and source elements by following the same process that you used when you mapped the source elements.
7. When you are finished mapping fields, click the save icon (💾) at the top of the mapping configuration pane to save the mapping changes. For instructions on performing other common editing tasks, see [Editing Mappings](#) below.

## Editing Mappings

The table below provides instructions for working with mapping components. When changing mappings, click Save (💾) periodically to save your changes.

What do you want to do?	Instructions
Add a target and source pair	<ol style="list-style-type: none"><li>1. Click the target element above the row where you want to add a new pair.</li><li>2. Click the lines icon (≡) that appears to the left of the target name.</li><li>3. Click the Add icon (+) that becomes available under the lines icon. Anzo selects the target cell and opens the <b>Target Mapping</b> text box.</li><li>4. In the Target Mapping text box, start typing the target field name. Anzo displays a list of fields that match the text. Select the target element in the results list.</li><li>5. In the Source column, click the cell that corresponds to the target you added. Anzo opens the Source Mapping text box.</li></ol>

What do you want to do?	Instructions
	6. In the Source Mapping text box, start typing the source field name. Anzo displays a list of fields that match the text. Select the source element in the results list.
Delete a target and source pair	Click the target element in the row that you want to delete. Then click the lines icon (≡) that appears to the left of the target name. Click the Trashcan icon (🗑) that becomes available.
Modify the mapping data references	To change selected data that the mapping can access, click the Configuration icon (⚙) in the right pane. In the References dialog box, add or remove elements as needed.
Validate changes to a mapping	Click the Validate icon (✅) to validate the mapping. Anzo displays any errors in the Validation Results screen.

For more advanced information about working with mappings, see [Transforming Data in Mappings](#).

## Related Topics

[Configuring Mappings to Ingest a Subset of the Source Data](#)

[Transforming Data in Mappings](#)

[Supported Mapping Functions](#)

## Configuring Mappings to Ingest a Subset of the Source Data

Anzo mappings include an option to set up parameters or criteria for ingesting source data so that you can create a graph data set that contains a subset of the data rather than all values. For example, if you want to import data that has decades worth of historical information but you are only interested in ingesting data from certain years, you can set criteria to filter out data that does not fall between those years.

### Note

If the data source is a database, you can typically achieve better overall ETL pipeline performance by using schema queries to join and/or filter data rather than configuring mappings to perform those types of operations. For more information, see [Performance Considerations for Database ETL Pipelines](#).

Follow the instructions below to set up the parameters to use as criteria and add the criteria to filters:

1. [Open the Mapping](#)
2. [Create Parameters to Use as Filter Criteria](#)
3. [Add Filters to the Mapping to Apply the Criteria](#)

## Open the Mapping

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Then click the **Mappings** tab. Anzo displays the Mappings screen, which lists any existing mappings. For example:

Data Sources

Schemas

Mappings

Pipelines

Search

Sort By: Title




View:

Add Mappings

<input type="checkbox"/>	Title	Description	Updated Date	Tags	Actions
	DB - emrdb - emr_acti...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
	DB - emrdb - emr_com...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
	DB - emrdb - emr_com...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
	DB - emrdb - emr_med...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
	DB - emrdb - emr_med...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
	DB - emrdb - emr_obs...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
	DB - emrdb - emr_obs...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
	DB - emrdb - emr_pati...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>

Rows per page: 201-20 of 34

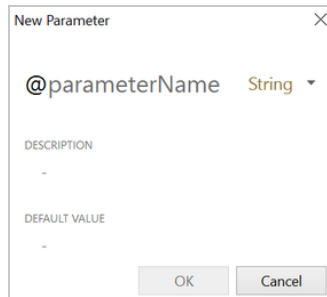
2. Click the name of the mapping that you want to edit. Anzo displays the mapping details. For example:

DB - emrdb - emr_observation		<a href="#">Edit</a> 
Not Versioned 		
Overview	Pipeline	Ontologies
Target	Source	General
PATIENTID	PATIENTID	Type Mapping
emr_observation_ibfk_2 → emr_observationdescription_OBSERVATIONID	OBSERVATIONDESCRIPTIONID	Creator System Administrator
OBSERVATIONLOWNORMAL	OBSERVATIONLOWNORMAL	Updated Jun 18, 2020 10:15 AM
emr_observation_ibfk_4 → emr_study_STUDYID	STUDYID	Released : Jun 18, 2020 10:15 AM
emr_observation_ibfk_1 → emr_patient_PATIENTID	PATIENTID	Details
OBSERVATIONID	OBSERVATIONID	Pipeline <a href="#">Load DB emrdb</a>
		<a href="http://cambridgesemantics.com/automap/...">http://cambridgesemantics.com/automap/...</a> 
		Tags
		Auto-Gen

3. Click **Edit** at the top of the screen. Provide your server connection and login information and then click the arrow icon (➔) to connect to the server and open the mapping.

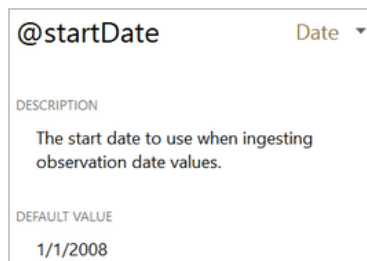
## Create Parameters to Use as Filter Criteria

1. In the mapping configuration pane, click the plus icon (+) next to **Parameters**. Anzo opens the New Parameter dialog box.



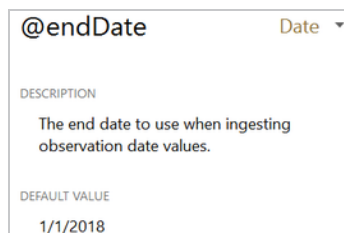
The 'New Parameter' dialog box is shown. It has a title bar with a close button (X). The main area contains three fields: a parameter name field with the text '@parameterName' and a data type dropdown menu set to 'String'; a 'DESCRIPTION' field with a hyphen '-' as a placeholder; and a 'DEFAULT VALUE' field with a hyphen '-' as a placeholder. At the bottom are 'OK' and 'Cancel' buttons.

2. Type a name for the new parameter in the **parameterName** field.
3. Next to the parameter name, click the data type drop-down list and select the data type of the source field whose value you want to use as filter criteria.
4. Type an optional description for the parameter in the **Description** field.
5. In the **Default Value** field, type the literal value to use as criteria for the source data. The value that you type must match the format for the data type that you chose. Do not include functions or formulas that transform the value in the Default Value field. You can transform the values when you create the filter that applies this parameter. For example:




An example of the 'New Parameter' dialog box. The parameter name is '@startDate' and the data type is 'Date'. The description is 'The start date to use when ingesting observation date values.' The default value is '1/1/2008'.

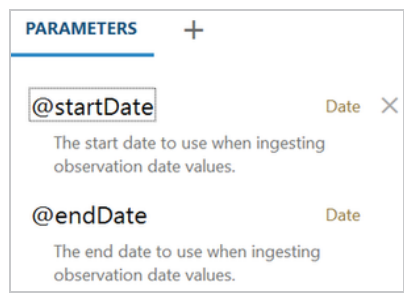
6. Click **OK** to save the new parameter and add it to the Parameters list. Repeat the steps in this section to create any additional parameters, for example, if you are filtering on dates and need to set the beginning and end dates to filter on. For example:



Another example of the 'New Parameter' dialog box. The parameter name is '@endDate' and the data type is 'Date'. The description is 'The end date to use when ingesting observation date values.' The default value is '1/1/2018'.

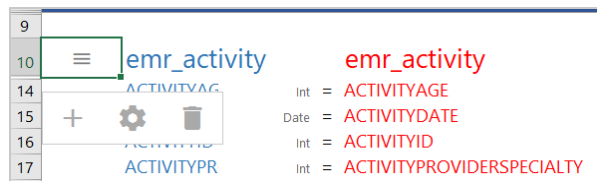
7. Click the save icon (  ) at the top of the mapping configuration pane to save the mapping changes.


To edit parameters in the Parameters list in the mapping configuration pane, click a parameter name to open the Edit Parameter dialog box. To delete parameters, hover the pointer over the parameter name and click the X that appears to the right of the parameter. For example:



## Add Filters to the Mapping to Apply the Criteria

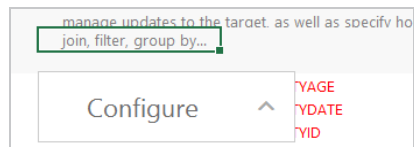
1. Click the cell to the left of the target table name to open the menu. For example:



2. Click the Cog icon (  ) in the menu to open the configuration section of the mapping.



3. In the configuration section, click the cell that contains the **join, filter, group by...** text in the Target column. If necessary, click the drop-down arrow next to the cell to open the Configure text box.



4. In the Configure text box, start typing **filter**. When "filter" appears below the text box, click it to add the filter keyword to the cell.



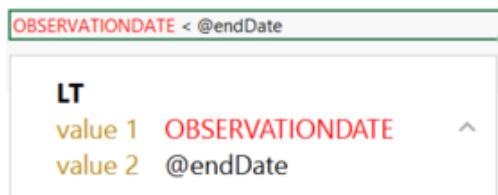
5. Click the cell in the Source column that corresponds to the filter you entered in the Target column. Anzo displays the **Source Mapping** text box below the cell. If necessary, click the drop-down arrow (▼) next to the cell to open the Source Mapping text box.
6. In the Source Mapping text box enter the expression to use for the filter. Type @ to display the list of parameters to select from. For instructions on using functions in the filter, see [Using Functions to Transform Source Data](#).

For example, the following expression filters on values whose OBSERVATIONDATE is greater than the date in the @startDate parameter:



7. Create any additional filters by repeating steps 2 – 5.

For example, the following expression filters on values whose OBSERVATIONDATE is less than the date in the @endDate parameter:



By adding filters for the @startDate and @endDate parameters, the example mapping is configured to ingest only the source records with observation dates that fall between the start and end dates.

8. Click the save icon (💾) at the top of the mapping configuration pane to save the mapping changes.

## Related Topics

[Creating a New Mapping](#)

[Transforming Data in Mappings](#)

[Supported Mapping Functions](#)

## Transforming Data in Mappings

Using the Anzo for Office plugin for Microsoft Excel, you can transform data to further define relationships between data elements, perform lightweight data preparation, or create sophisticated transformations. This topic provides information about creating advanced mappings and using Excel-like functions to transform

data during the ETL process. This topic also describes the most commonly used mapping functions.

## Note

If the data source is a database, you can typically achieve better overall ETL pipeline performance by using schema queries to join and/or filter data rather than configuring mappings to perform those types of operations. For more information, see [Performance Considerations for Database ETL Pipelines](#).

For instructions on creating a new mapping, see [Creating a New Mapping](#). For instructions on setting up parameters to ingest a subset of the source data, see [Configuring Mappings to Ingest a Subset of the Source Data](#).

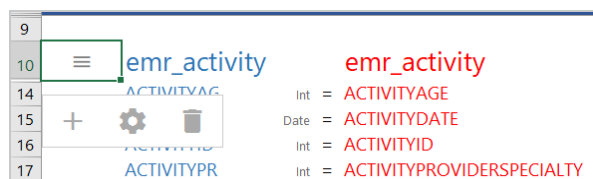
- Configuring Groups, Filters, Joins, Updates, and Merges
- Using Functions to Transform Source Data
- Commonly Used Functions

## Configuring Groups, Filters, Joins, Updates, and Merges

This section provides information about configuring groups, joins, filters, references, and merges at the mapping level so that they can be used by any functions that you use to transform the source data. The table below the steps describes each of the mapping level configuration options.

To implement a mapping level configuration:

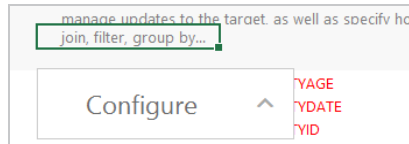
1. Click the cell to the left of the target table name to open the menu. For example:



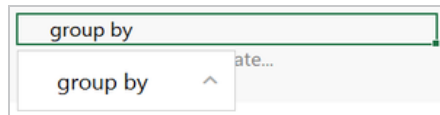
- 2 Click the Cog icon () in the menu to open the configuration section of the mapping.



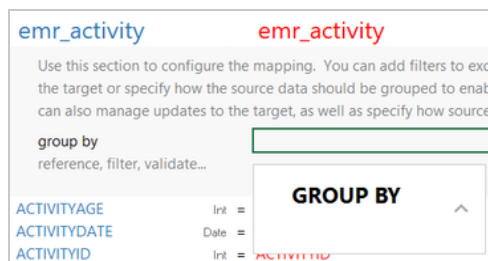
3. In the configuration section, click the cell that contains the **join, filter, group by...** text in the Target column. If necessary, click the drop-down arrow next to the cell to open the Configure text box.



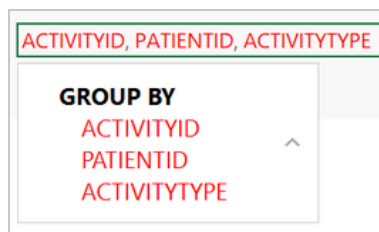
4. In the Configure text box, start typing the option that you want to configure. Anzo displays the options that match the text. Select an option to add it to the cell. For example:



5. Click the cell in the Source column that corresponds to the option you entered in the Target column. Anzo populates the Source text box with the appropriate keywords and arguments. For example:



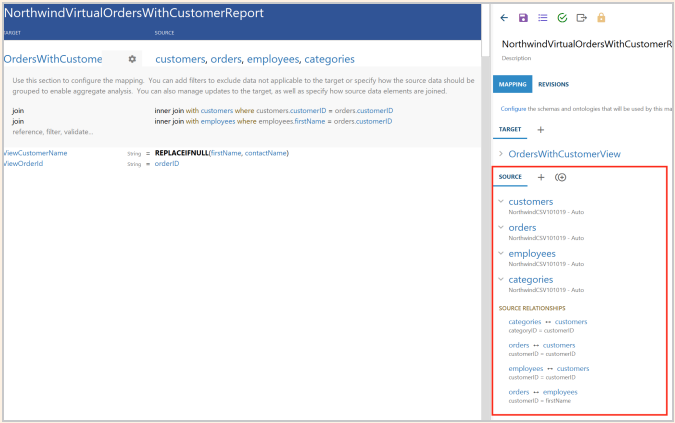
6. In the Source text box, click next to an argument or under a keyword and start typing column names for the columns that you want to add. Press **Ctrl +** to enter multiple columns. Then click the up arrow to enter the columns in the cell. For example:



The table below describes the mapping configuration options:

Option & Arguments	Description
FILTER	The FILTER keyword restricts the results that the mapping functions return. FILTER supports a single expression, and the expression must return a boolean value.

Option & Arguments	Description
GROUP BY	<p>The GROUP BY clause designates data groups and is required for aggregate functions. When an aggregate function is used, the solution is first divided into the groups defined by the GROUP BY clause, and then the aggregate value is calculated for each group.</p>
JOIN  join element condition	<p>A JOIN combines rows from two tables based on related columns. You can specify joins when you map two sources to one target.</p> <ul style="list-style-type: none"><li>• <b>join</b>: The kind of join to use. Type one of the following options:<ul style="list-style-type: none"><li>▪ <b>inner join</b>: Returns only the records that have matching values in both tables.</li><li>▪ <b>outer join</b>: Returns all records from both tables when there is a match in either the left or right table.</li><li>▪ <b>left join</b>: Returns all records from the left table and joins only the records from the right table that match the condition.</li><li>▪ <b>right join</b>: Returns all records from the right table and joins only the records from the left table that match the condition.</li></ul></li><li>• <b>element</b>: One of the tables to join. The table that you specify depends on the type of join you are creating. For right joins, choose the left table. For left joins, choose the right table.</li><li>• <b>condition</b>: The condition to use to join the two tables.</li></ul> <p>For example, the following join uses an inner join to join all of the records from the MovieActors1 and MovieActors2 tables when the MovieID is the same in both tables.</p> <div><pre>inner join with MovieActors1 where MovieActors1.MovieID, condition, MovieActors2.MovieID</pre><div><p><b>JOIN</b></p><p>join inner join</p><p>element MovieActors1</p><p>condition MovieActors1.MovieID</p><p>=</p><p>MovieActors2.MovieID</p></div></div> <div><p><b>Important</b></p><p>When including joins in mappings, do not create joins that result in multiple</p></div>

Option & Arguments	Description
	<p>primary tables. Mappings with two or more primary tables are invalid. For example, the following mapping is invalid because it has two primary tables, orders and categories.</p> 
MERGE BY	The MERGE BY clause enables you to merge multiple source rows into a single target row.
REFERENCE element condition	The REFERENCE keyword enables you to create a referential join between two tables.
UPDATE action key	

### Using Functions to Transform Source Data

This section provides information about how to add functions to perform operations on source data.

The Source Mapping text box that you use to select source fields also includes a list of functions that you can apply to transform the source data. When you type in the Source Mapping box, in addition to available source fields, the mapping tool also displays a list of the functions that match the text you type. For example:

Int	=	MovieID
Long	=	
Int	=	a
String	=	<div> <div>AVG</div> <div>AND</div> <div>MASKLASTNCHARS</div> <div>MAP</div> <div>MAX</div> <div>DATEPARSE</div> </div>

To add a function, select it from the list. The mapping tool adds the function to the cell and the Source Mapping box shows the arguments for the function that you chose. For example:

=	MovieID
=	DATEPARSE()
=	<div> <div>DATEPARSE</div> <div>date text</div> <div>date format</div> </div>

In the Source Mapping text box, enter the source field name, text, or additional functions that you want to include for the arguments. Enter arguments by typing to the right of the argument name. When entering literal values, press **Enter** to submit the value.

For example, the DATEPARSE function below converts MovieReleaseDate values from string types to dates in dd-MM-yyyy format:

Int	=	MovieID
String	=	DATEPARSE(MovieReleaseDate, "dd-MM-yyyy")
String	=	<div> <div>DATEPARSE</div> <div>date text MovieReleaseDate</div> <div>date format "dd-MM-yyyy"</div> </div>

### Tip

The format that you specify for dates is flexible. For example, typing the format "dd-MMM-yy" displays values such as "01-JAN-19."

## Commonly Used Functions

The table below describes the functions that are commonly used in mappings. For a complete list of the supported functions, see [Supported Mapping Functions](#).

## Note

Any time you type a literal value into a function argument, press **Enter** to submit the value.

Function & Arguments	Description
<b>IF</b>  test value if true value if false value if error	<p>This function evaluates the condition in the <b>test</b> argument and assigns the value in <b>value if true</b> or <b>value if false</b> based on the results.</p> <ul style="list-style-type: none"><li>• <b>test</b>: Use boolean columns or functions that return boolean: LE, LT, GE, GT, EQUALS, NOT_EQUAL, ISNULL, NOT, IN.</li><li>• <b>value if true</b>: The value to output if <b>test</b> returns true.</li><li>• <b>value if false</b>: The value to output if <b>test</b> returns false.</li><li>• <b>value if error</b>: Cambridge Semantics recommends that you leave this argument blank.</li></ul>
<b>DATEPARSE</b>  date text date format	<p>This function converts a string that contains a date value (<b>date text</b>) to the specified <b>date format</b>.</p> <ul style="list-style-type: none"><li>• <b>date text</b>: The property that contains the date value in string format.</li><li>• <b>date format</b>: The format that you want the date to follow. Specify days as "d," months as "M," and years as "y." For example, "yyyy-MM-dd."</li></ul> <p>For example, the source mapping below converts the MovieReleaseDate values from strings to dates in the format "dd-MM-yyyy":</p> <p><b>DATEPARSE</b> date text <b>MovieReleaseDate</b> date format "dd-MM-yyyy"</p> <p>The format that you specify for dates is flexible. For example, typing the format "dd-MMM-yy" displays values such as "01-JAN-19."</p>
<b>DATETIMEPARSE</b>  date text date format	<p>This function converts a string that contains a datetime value (<b>date text</b>) to the specified <b>date format</b>.</p> <ul style="list-style-type: none"><li>• <b>date text</b>: The property that contains the datetime value in string format.</li></ul>

Function & Arguments	Description
	<ul style="list-style-type: none"> <li>• <b>date format:</b> The format that you want the datetime to follow. For the date, specify days as "d," months as "M," and years as "y." For the time, specify "H" for hours, "m" for minutes, and "s" for seconds. For example, "yyyy-MM-dd HH:mm:ss."</li> </ul> <p>For example, the source mapping below converts the PATIENTLASTPMODATE from a string value to a datetime value in the format "MM-dd-yyyy HH:mm:ss":</p> <p><b>DATETIMEPARSE</b>  date text PATIENTLASTPMODATE  date format "MM-dd-yyyy HH:mm:ss"</p>
UPPER  value	This function converts a string <b>value</b> to upper case letters.
LOWER  value	This function converts a string <b>value</b> to lower case letters.
REPLACEIFNULL  expression if null expression	<p>This function evaluates the <b>expression</b>. If the result is null, Anzo replaces the null with the value in <b>if null expression</b>.</p> <ul style="list-style-type: none"> <li>• <b>expression:</b> The source column or expression to evaluate.</li> <li>• <b>if null expression:</b> The expression to replace null values with. The resulting value must be the same data type as the target. For example, if mapping to a target with a double data type, "10.01" is valid but the string "missing" is not.</li> </ul> <p>For example, the source mapping below replaces any null values in the PATIENTID integer column with the integer 999:</p> <p><b>REPLACEIFNULL</b>  expression PATIENTID  if null expression 999</p>
REPLACEIFNULLLOREMPY	This function evaluates the <b>string expression</b> . If the result is null or empty



Function & Arguments	Description
<b>Y</b>  string expression if null or empty expression	<p>(""), Anzo replaces the empty or null with the value in <b>if null or empty expression</b>.</p> <ul style="list-style-type: none"> <li>• <b>string expression</b>: The source column or expression that evaluates to string.</li> <li>• <b>if null or empty expression</b>: The expression to replace null or empty values with. The resulting value must be a string.</li> </ul> <p>For example, the source mapping below replaces any null or empty values in the GENDER column with "Not Specified":</p> <p><b>REPLACEIFNULLLOREEMPTY</b>            string expression <b>GENDER</b>            if null or empty string expression "Not Specified"</p>
<b>ISNULL</b>  expression	<p>This function evaluates the source column values in <b>expression</b> and returns "true" if the value is null and "false" if it is not null. You must choose a column in the <b>expression</b> argument; do not type a literal value or a function.</p>
<b>SPLIT</b>  string delimiter	<p>This function splits a <b>string</b> value into multiple values based on the specified <b>delimiter</b>.</p> <ul style="list-style-type: none"> <li>• <b>string</b>: The source column or function that evaluates to a string.</li> <li>• <b>delimiter</b>: The character to use to delimit the <b>string</b>.</li> </ul>
<b>SPLITARRAY</b>  string delimiter index	<p>This function splits a <b>string</b> value into an array based on the <b>delimiter</b>. From the array, the function retrieves only the portion of the value that you specify in the <b>index</b>.</p> <ul style="list-style-type: none"> <li>• <b>string</b>: The source column or function that evaluates to a string.</li> <li>• <b>delimiter</b>: The character to use to delimit the string.</li> <li>• <b>index</b>: An integer that specifies the portion of the array to retrieve. Indexes start at zero. The first portion of the array is 0, the second is 1, and so on. Choose an index that you know exists or the mapping becomes invalid.</li> </ul> <p>For example, the following source mapping retrieves only the last four</p>

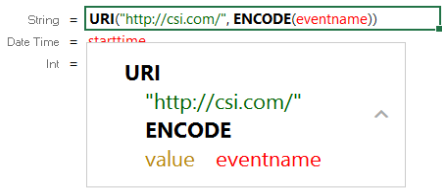
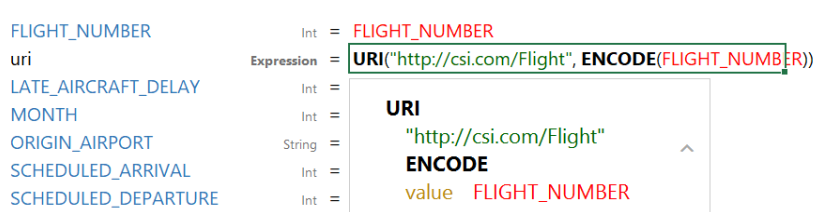
Function & Arguments	Description
	<p>digits of social security numbers:</p> <p><b>SPLITARRAY</b></p> <p>string    SSN</p> <p>delimiter    "-"</p> <p>index    2</p>
<p>IN</p> <p>value</p> <p>set to check</p>	<p>This function checks whether a given <b>value</b> exists in a set of values (<b>set to check</b>). If the value exists in the set, IN returns "true." If the value does not exist in the set, IN returns "false." IN does not do comparisons on string values.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The value to look for in the set.</li> <li>• <b>set to check</b>: The set of values to compare the value against. After typing a character, press <b>Enter</b> to submit the value, then press <b>Ctrl +</b> to add the next value. All items in the set must be the same data type.</li> </ul> <p>For example, the following source mapping checks to see if PATIENTID falls in the set of 1, 100, 1000:</p> <p><b>IN</b></p> <p>value    PATIENTID</p> <p>set to check    1</p> <p>                  100</p> <p>                  1000</p>
<p>MAKELIST</p> <p>expression</p>	<p>This function maps multiple source columns to a single target property. The function does not create a list; it creates new rows, one for each column that is mapped to the target.</p> <ul style="list-style-type: none"> <li>• <b>expression</b>: The list of columns that you want to map to the target. After adding a source column press <b>Ctrl +</b> to select the next column.</li> </ul>
<p>REGEX</p> <p>input</p>	<p>This function finds all patterns in the <b>input</b> string that match the specified regular expression (<b>regex</b>). It replaces the input patterns with the value in</p>

Function & Arguments	Description
regex replace	<p><b>replace</b> and returns the resulting string.</p> <ul style="list-style-type: none"> <li>• <b>input</b>: The source column or expression that evaluates to a string.</li> <li>• <b>regex</b>: The regular expression to use to find matches in the input string. For information about REGEX syntax, see the W3C <a href="#">Regular Expression Syntax</a> specification.</li> <li>• <b>replace</b>: The string that should replace the input patterns that match regex.</li> </ul> <p>For example, the source mapping below uses the REGEX function to search for the pattern "PS" in the COMPLAINTSTRING values and replaces each PS with a hyphen (-):</p> <p><b>REGEX</b></p> <pre>input COMPLAINTSTRING regex "PS" replace "-"</pre>
CONCATENATE text	<p>This function concatenates multiple string values (<b>text</b>) and returns a single string.</p> <ul style="list-style-type: none"> <li>• <b>text</b>: The string values to concatenate, including any delimiters that you want to use. Press <b>Ctrl +</b> to enter multiple values.</li> </ul> <p>For example, the source mapping below concatenates PATIENTHOMESTATE and PATIENTHOMEZIP:</p> <p><b>CONCATENATE</b></p> <pre>text PATIENTHOMESTATE PATIENTHOMEZIP</pre>
EQUAL value1 value2	<p>This function compares numeric values and returns "true" if <b>value1</b> is equal to <b>value2</b> and "false" if the values are not equal (value1 = value2).</p> <ul style="list-style-type: none"> <li>• <b>value1</b>: The numeric value to compare to value2.</li> <li>• <b>value2</b>: The numeric value to compare to value1.</li> </ul>
NOT_EQUAL	<p>This function compares numeric values and returns "true" if <b>value1</b> does not equal <b>value2</b> and "false" if the values are equal (value1 != value2).</p>

Function & Arguments	Description
value1 value2	<ul style="list-style-type: none"> <li>• <b>value1</b>: The numeric value to compare to value2.</li> <li>• <b>value2</b>: The numeric value to compare to value1.</li> </ul>
GE  value1 value2	<p>This function compares numeric values and returns "true" if <b>value1</b> is greater than or equal to <b>value2</b> and "false" if value1 is less than value2 (value1 &gt;= value2).</p> <ul style="list-style-type: none"> <li>• <b>value1</b>: The numeric value to compare to value2.</li> <li>• <b>value2</b>: The numeric value to compare to value1.</li> </ul>
GT  value1 value2	<p>This function compares numeric values and returns "true" if <b>value1</b> is greater than <b>value2</b> and "false" if value1 is less than or equal to value2 (value1 &gt; value2).</p> <ul style="list-style-type: none"> <li>• <b>value1</b>: The numeric value to compare to value2.</li> <li>• <b>value2</b>: The numeric value to compare to value1.</li> </ul>
LE  value1 value2	<p>This function compares numeric values and returns "true" if <b>value1</b> is less than or equal to <b>value2</b> and "false" if value1 is greater than value2 (value1 &lt;= value2).</p> <ul style="list-style-type: none"> <li>• <b>value1</b>: The numeric value to compare to value2.</li> <li>• <b>value2</b>: The numeric value to compare to value1.</li> </ul>
LT  value1 value2	<p>This function compares numeric values and returns "true" if <b>value1</b> is less than <b>value2</b> and "false" if value1 is greater than or equal to value2 (value1 &lt; value2).</p> <ul style="list-style-type: none"> <li>• <b>value1</b>: The numeric value to compare to value2.</li> <li>• <b>value2</b>: The numeric value to compare to value1.</li> </ul>
AND  logical1 logical2	<p>This logical function evaluates two or more logical statements (<b>logical1</b>, <b>logical2</b>) and returns "true" if all conditions are met or "false" if any condition is not met. All logical statements must evaluate to the same data type.</p> <ul style="list-style-type: none"> <li>• <b>logical1</b>: The first logical condition to evaluate. This argument needs to include a logical function that returns a boolean value,</li> </ul>

Function & Arguments	Description
	<p>such as AND, OR, GT, GE, LE, LT, EQUAL, NOT_EQUAL, ISNULL, NOT, IN.</p> <ul style="list-style-type: none"> <li>• <b>logical2</b>: The second logical condition to evaluate. This argument also needs to include a logical function that returns a boolean value.</li> </ul>
<p>OR</p> <p>logical1</p> <p>logical2</p>	<p>This logical function evaluates two or more logical statements (<b>logical1</b>, <b>logical2</b>) and returns "true" if any of the conditions are met or "false" if none of the conditions are met. All logical statements must evaluate to the same data type.</p> <ul style="list-style-type: none"> <li>• <b>logical1</b>: The first logical condition to evaluate. This argument needs to include a logical function that returns a boolean value, such as AND, OR, GT, GE, LE, LT, EQUAL, NOT_EQUAL, ISNULL, NOT, IN.</li> <li>• <b>logical2</b>: The second logical condition to evaluate. This argument also needs to include a logical function that returns a boolean value.</li> </ul>
<p>NOT</p> <p>logical</p>	<p>This logical function evaluates whether data does not meet the condition (<b>logical</b>) that you specify.</p> <ul style="list-style-type: none"> <li>• <b>logical</b>: The logical condition to evaluate. This argument needs to include a logical function that returns a boolean value, such as AND, OR, GT, GE, LE, LT, EQUAL, NOT_EQUAL, ISNULL, NOT, IN.</li> </ul>
<p>NUMERIC_ADD</p> <p>v1</p> <p>v2</p>	<p>This function adds the values of the numeric expressions that you specify (<b>v1 + v2</b>).</p> <ul style="list-style-type: none"> <li>• <b>v1</b>: The first addend in the equation.</li> <li>• <b>v2</b>: The second addend in the equation.</li> </ul>
<p>DIVIDE</p> <p>v1</p>	<p>This function divides the values of the numeric expressions that you specify (<b>v1/v2</b>).</p>

Function & Arguments	Description
<div>v2</div>	<ul style="list-style-type: none"> <li>• <b>v1</b>: The divisor in the equation.</li> <li>• <b>v2</b>: The dividend in the equation.</li> </ul>
<div>MULTIPLY</div> <div>v1</div> <div>v2</div>	<p>This function multiplies the values of the numeric expressions that you specify (<b>v1</b> x <b>v2</b>).</p> <ul style="list-style-type: none"> <li>• <b>v1</b>: The multiplicand in the equation.</li> <li>• <b>v2</b>: The multiplier in the equation.</li> </ul>
<div>NUMERIC_SUBTRACT</div> <div>v1</div> <div>v2</div>	<p>This function subtracts the values of the numeric expressions that you specify (<b>v1</b> - <b>v2</b>).</p> <ul style="list-style-type: none"> <li>• <b>v1</b>: The minuend in the equation.</li> <li>• <b>v2</b>: The subtrahend in the equation.</li> </ul>
<div>LOOKUP</div> <div>from</div> <div>get</div> <div>fields</div> <div>values</div>	<p>This function enables you to look up values in a supplemental table. LOOKUP joins the <b>from</b> lookup table to a source table on the columns specified in the <b>fields</b> and <b>values</b> arguments. The function returns the value from <b>get</b> in the lookup table that corresponds to each row's value in the <b>values</b> argument.</p> <ul style="list-style-type: none"> <li>• <b>from</b>: The lookup table to perform the join against.</li> <li>• <b>get</b>: The field or property to retrieve from the lookup table in the <b>from</b> argument.</li> <li>• <b>fields</b>: The field or fields from the lookup table to compare with the values from the primary table.</li> <li>• <b>values</b>: The values from the primary table to compare with the <b>fields</b> from the lookup table.</li> </ul>
<div>MAP</div> <div>value</div> <div>map</div>	<p>This function retrieves values from a map that you define. The map is a collection of key/value pairs. The function uses the specified <b>value</b> as a key in the <b>map</b> and returns the value associated with the key.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The key or keys to use to look up the value from the map.</li> <li>• <b>map</b>: The map to use to look up the values. You can click the <b>map</b></li> </ul>

Function & Arguments	Description
	argument name to open the Edit Map dialog box and define or change a map.
URI()	<p>When specified in the Source column in mappings, this function transforms the property values to URI format by concatenating each of the components that you specify in the function. To ensure that values with spaces and other characters are encoded as valid URIs, the URI function is often used with the ENCODE function.</p> <p>For example, the following mapping for a "ticket_events" table transforms an "eventname" string to URI format by prepending "http://csi.com/" to the encoded event names:</p>  <p>This example mapping results in triples such as:</p> <pre>&lt;ticket_events&gt; &lt;eventname&gt; "http://csi.com/Rolling+Stones"</pre> <p>You can also enter URI() in the Target column to specify that the Expression in the Source column should be the URI of the entity that is being created. For example, the mapping below generates an entity URI by prepending "http://csi.com/Flight" to the flight number value:</p>  <p>The example URI specification results in triples such as:</p> <pre>&lt;http://csi.com/Flight1234&gt; &lt;FLIGHT_NUMBER&gt; 1234</pre>

Function & Arguments	Description
	<pre>&lt;http://csi.com/Flight1234&gt; &lt;ORIGIN_AIRPORT&gt; "BOS"</pre>

**Related Topics**

- [Creating a New Mapping](#)
- [Configuring Mappings to Ingest a Subset of the Source Data](#)
- [Supported Mapping Functions](#)

**Supported Mapping Functions**

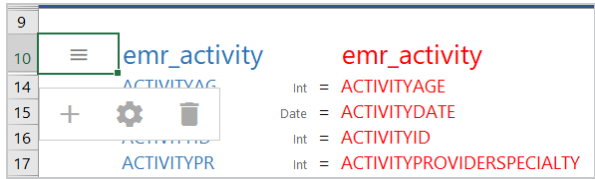
This topic describes the mapping functions that Anzo supports. For information about adding functions to mappings, see [Transforming Data in Mappings](#).

- [Aggregate Functions](#)
- [Boolean Operators](#)
- [Conditional Expressions](#)
- [Data Type Conversion Functions](#)
- [Lookup and Mapping Functions](#)
- [Numeric Functions](#)
- [String Functions](#)

**Aggregate Functions**

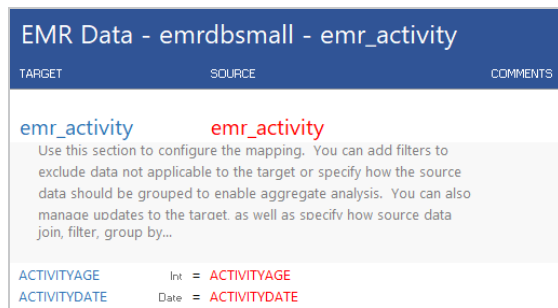
Aggregate functions rely on the groups that you define by configuring a GROUP BY statement for the mapping. All aggregate functions use the GROUP BY that you specify. Follow these instructions to configure a GROUP BY statement:

1. Click the cell to the left of the target table name to open the menu. For example:

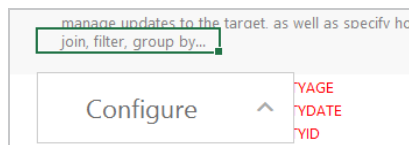




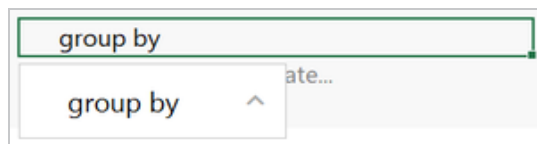
- Click the Cog icon (  ) in the menu to open the configuration section of the mapping.



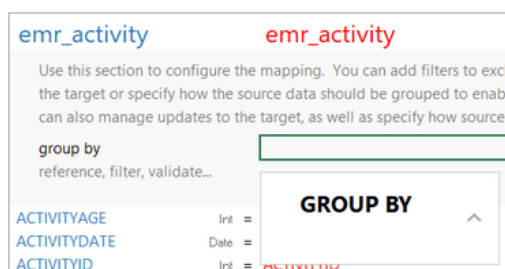
- In the configuration section, click the cell that contains the **join, filter, group by...** text in the Target column. If necessary, click the drop-down arrow next to the cell to open the Configure text box.



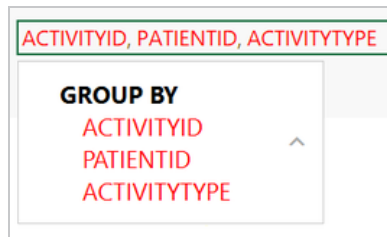
- In the Configure text box, start typing "group by." Anzo completes the text and displays **group by** in the box. Click the up arrow to enter group by in the cell.



- Click the cell in the Source column that corresponds to the group by you entered in the Target column. Anzo enters GROUP BY in the Source text box.



- In the Source text box, click under GROUP BY and start typing column names for the columns that you want to group on. Press **Ctrl +** to enter multiple columns. Then click the up arrow to enter the columns in the cell. For example:



When you finish configuring the GROUP BY, save the mapping. When you use aggregate functions in the mapping, the functions group data according to the configured GROUP BY.

The table below describes the supported aggregate functions.

Function & Arguments	Description
AVG number	<p>This function calculates the arithmetic mean for the group of numeric values that you specify in the <b>number</b> argument.</p> <ul style="list-style-type: none"><li>• <b>number</b>: The column or expression that evaluates to a numeric value. The average is computed for the group or groups in the mapping's GROUP BY statement.</li></ul> <p>For example, the following source mapping calculates the average NUMBER_OF_BYTES for each event. The GROUP BY statement for the mapping includes EVENTID.</p> <p><b>AVG</b> number NUMBER_OF_BYTES</p>
COUNT value	<p>This function counts the number of instances for a grouped <b>value</b>. This function does not perform COUNT DISTINCT.</p>
MAX value	<p>This function calculates the maximum value for the group of numeric values that you specify in the <b>value</b> argument.</p>
MIN value	<p>This function calculates the minimum value for the group of numeric values that you specify in the <b>value</b> argument.</p>
SUM number	<p>This function calculates the sum of the group of numeric values that you specify in the <b>number</b> argument.</p>

## Boolean Operators

This section describes the boolean operators that you can use to target specific data and expand or reduce the number of records that are returned.

Function & Arguments	Description
<b>EQUAL</b>  value1 value2	<p>This function compares numeric values and returns "true" if <b>value1</b> is equal to <b>value2</b> and "false" if the values are not equal (value1 = value2).</p> <ul style="list-style-type: none"><li>• <b>value1</b>: The numeric value to compare to value2.</li><li>• <b>value2</b>: The numeric value to compare to value1.</li></ul>
<b>GE</b>  value1 value2	<p>This function compares numeric values and returns "true" if <b>value1</b> is greater than or equal to <b>value2</b> and "false" if value1 is less than value2 (value1 &gt;= value2).</p> <ul style="list-style-type: none"><li>• <b>value1</b>: The numeric value to compare to value2.</li><li>• <b>value2</b>: The numeric value to compare to value1.</li></ul>
<b>GT</b>  value1 value2	<p>This function compares numeric values and returns "true" if <b>value1</b> is greater than <b>value2</b> and "false" if value1 is less than or equal to value2 (value1 &gt; value2).</p> <ul style="list-style-type: none"><li>• <b>value1</b>: The numeric value to compare to value2.</li><li>• <b>value2</b>: The numeric value to compare to value1.</li></ul>
<b>IN</b>  value set to check	<p>This function checks whether a given <b>value</b> exists in a set of values (<b>set to check</b>). If the value exists in the set, IN returns "true." If the value does not exist in the set, IN returns "false." IN does not do comparisons on string values.</p> <ul style="list-style-type: none"><li>• <b>value</b>: The value to look for in the set.</li><li>• <b>set to check</b>: The set of values to compare the value against. After typing a character, press <b>Enter</b> to submit the value, then press <b>Ctrl +</b> to add the next value. All items in the set must be the same data type.</li></ul> <p>For example, the following source mapping checks to see if PATIENTID falls in the set of 1, 100, 1000:</p>

Function & Arguments	Description
	<b>IN</b> value PATIENTID set to check 1 100 1000
ISNULL expression	This function evaluates the source column values in <b>expression</b> and returns "true" if the value is null and "false" if it is not null. You must choose a column in the <b>expression</b> argument; do not type a literal value or a function.
LE value1 value2	This function compares numeric values and returns "true" if <b>value1</b> is less than or equal to <b>value2</b> and "false" if value1 is greater than value2 (value1 <= value2). <ul style="list-style-type: none"> <li>• <b>value1</b>: The numeric value to compare to value2.</li> <li>• <b>value2</b>: The numeric value to compare to value1.</li> </ul>
LT value1 value2	This function compares numeric values and returns "true" if <b>value1</b> is less than <b>value2</b> and "false" if value1 is greater than or equal to value2 (value1 < value2). <ul style="list-style-type: none"> <li>• <b>value1</b>: The numeric value to compare to value2.</li> <li>• <b>value2</b>: The numeric value to compare to value1.</li> </ul>
NOT_EQUAL value1 value2	This function compares numeric values and returns "true" if <b>value1</b> does not equal <b>value2</b> and "false" if the values are equal (value1 != value2). <ul style="list-style-type: none"> <li>• <b>value1</b>: The numeric value to compare to value2.</li> <li>• <b>value2</b>: The numeric value to compare to value1.</li> </ul>

## Conditional Expressions

This section describes the functions that you can use to perform different computations based on whether a conditional expression evaluates to true or false.

Function & Arguments	Description
AND	This logical function evaluates two or more logical statements ( <b>logical1</b> , <b>logical2</b> ) and returns "true" if all conditions are met or "false" if any

Function & Arguments	Description
logical1 logical2	<p>condition is not met. All logical statements must evaluate to the same data type.</p> <ul style="list-style-type: none"> <li>• <b>logical1</b>: The first logical condition to evaluate. This argument needs to include a logical function that returns a boolean value, such as AND, OR, GT, GE, LE, LT, EQUAL, NOT_EQUAL, ISNULL, NOT, IN.</li> <li>• <b>logical2</b>: The second logical condition to evaluate. This argument also needs to include a logical function that returns a boolean value.</li> </ul>
IF  test value if true value if false value if error	<p>This function evaluates the condition in the <b>test</b> argument and assigns the value in <b>value if true</b> or <b>value if false</b> based on the results.</p> <ul style="list-style-type: none"> <li>• <b>test</b>: Use boolean columns or functions that return boolean: LE, LT, GE, GT, EQUALS, NOT_EQUAL, ISNULL, NOT, IN.</li> <li>• <b>value if true</b>: The value to output if <b>test</b> returns true.</li> <li>• <b>value if false</b>: The value to output if <b>test</b> returns false.</li> <li>• <b>value if error</b>: Cambridge Semantics recommends that you leave this argument blank.</li> </ul>
OR  logical1 logical2	<p>This logical function evaluates two or more logical statements (<b>logical1</b>, <b>logical2</b>) and returns "true" if any of the conditions are met or "false" if none of the conditions are met. All logical statements must evaluate to the same data type.</p> <ul style="list-style-type: none"> <li>• <b>logical1</b>: The first logical condition to evaluate. This argument needs to include a logical function that returns a boolean value, such as AND, OR, GT, GE, LE, LT, EQUAL, NOT_EQUAL, ISNULL, NOT, IN.</li> <li>• <b>logical2</b>: The second logical condition to evaluate. This argument also needs to include a logical function that returns a boolean value.</li> </ul>
NOT	This logical function evaluates whether data does not meet the

Function & Arguments	Description
logical	<p>condition (<b>logical</b>) that you specify.</p> <ul style="list-style-type: none"> <li>• <b>logical</b>: The logical condition to evaluate. This argument needs to include a logical function that returns a boolean value, such as AND, OR, GT, GE, LE, LT, EQUAL, NOT_EQUAL, ISNULL, NOT, IN.</li> </ul>
REPLACEIFNULL  expression  if null expression	<p>This function evaluates the <b>expression</b>. If the result is null, Anzo replaces the null with the value in <b>if null expression</b>.</p> <ul style="list-style-type: none"> <li>• <b>expression</b>: The source column or expression to evaluate.</li> <li>• <b>if null expression</b>: The expression to replace null values with. The resulting value must be the same data type as the target. For example, if mapping to a target with a double data type, "10.01" is valid but the string "missing" is not.</li> </ul> <p>For example, the source mapping below replaces any null values in the PATIENTID integer column with the integer 999:</p> <p><b>REPLACEIFNULL</b>  expression PATIENTID  if null expression 999</p>
REPLACEIFNULLLOREEMPTY  string expression  if null or empty expression	<p>This function evaluates the <b>string expression</b>. If the result is null or empty (""), Anzo replaces the empty or null with the value in <b>if null or empty expression</b>.</p> <ul style="list-style-type: none"> <li>• <b>string expression</b>: The source column or expression that evaluates to string.</li> <li>• <b>if null or empty expression</b>: The expression to replace null or empty values with. The resulting value must be a string.</li> </ul> <p>For example, the source mapping below replaces any null or empty values in the GENDER column with "Not Specified":</p> <p><b>REPLACEIFNULLLOREEMPTY</b>  string expression GENDER  if null or empty string expression "Not Specified"</p>

## Data Type Conversion Functions

This section describes functions that you can use to convert values from one data type to another.

Function & Arguments	Description
<b>BOOLEANPARSE</b>  value	<p>This function converts a string (<b>value</b>) that contains "true" and "false" values to boolean format.</p> <div><b>Note</b> Specifying a source column for which some instances do not contain "true" or "false" values can cause the ETL job to fail. Cambridge Semantics recommends using TRYPARSEBOOLEAN unless you are certain that all instances of <b>value</b> contain the words "true" or "false."</div>
<b>DATEPARSE</b>  date text date format	<p>This function converts a string that contains a date value (<b>date text</b>) to the specified <b>date format</b>.</p> <div><b>Note</b> Specifying a source column for which some instances do not contain date values can cause the ETL job to fail. Cambridge Semantics recommends using TRYPARSEDATE unless you are certain that all instances of <b>date text</b> contain a date.</div> <ul style="list-style-type: none"><li>• <b>date text</b>: The property that contains the date value in string format.</li><li>• <b>date format</b>: The format that you want the date to follow. Specify days as "d," months as "M," and years as "y." For example, "yyyy-MM-dd."</li></ul> <p>For example, the source mapping below converts the MovieReleaseDate values from strings to dates in the format "dd-MM-yyyy":</p> <p><b>DATEPARSE</b> date text <b>MovieReleaseDate</b> date format <b>"dd-MM-yyyy"</b></p> <p>The format that you specify for dates is flexible. For example, typing the format "dd-MMM-yy" displays values such as "01-JAN-19."</p>


Function & Arguments	Description
DATETIMEPARSE  date text date format	<p>This function converts a string that contains a datetime value (<b>date text</b>) to the specified <b>date format</b>.</p> <div> <p><b>Note</b></p> <p>Specifying a source column for which some instances do not contain datetime values can cause the ETL job to fail. Cambridge Semantics recommends using TRYPARSEDATETIME unless you are certain that all instances of <b>date text</b> contain a datetime.</p> </div> <ul style="list-style-type: none"> <li>• <b>date text</b>: The property that contains the datetime value in string format.</li> <li>• <b>date format</b>: The format that you want the datetime to follow. For the date, specify days as "d," months as "M," and years as "y." For the time, specify "H" for hours, "m" for minutes, and "s" for seconds. For example, "yyyy-MM-dd HH:mm:ss."</li> </ul> <p>For example, the source mapping below converts the PATIENTLASTPMODATE from a string value to a datetime value in the format "MM-dd-yyyy HH:mm:ss":</p> <p><b>DATETIMEPARSE</b>  date text PATIENTLASTPMODATE  date format "MM-dd-yyyy HH:mm:ss"</p>
DECIMALPARSE  value	<p>This function converts a string (<b>value</b>) that contains a decimal value to decimal format.</p> <div> <p><b>Note</b></p> <p>Specifying a source column for which some instances do not contain decimal values can cause the ETL job to fail. Cambridge Semantics recommends using TRYPARSEDECIMAL unless you are certain that all instances of <b>value</b> contain a decimal.</p> </div>
DOUBLEPARSE	<p>This function converts a string (<b>value</b>) that contains a double value to double format.</p>

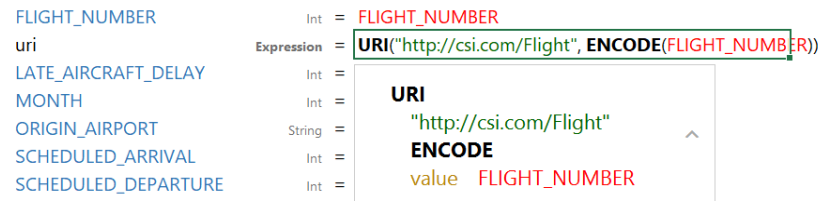


Function & Arguments	Description
value	<p><b>Note</b></p> <p>Specifying a source column for which some instances do not contain double values can cause the ETL job to fail. Cambridge Semantics recommends using TRYPARSEDOUBLE unless you are certain that all instances of <b>value</b> contain a double.</p>
FLOATPARSE value	<p>This function converts a string (<b>value</b>) that contains float values to float format.</p> <p><b>Note</b></p> <p>Specifying a source column for which some instances do not contain float values can cause the ETL job to fail. Cambridge Semantics recommends using TRYPARSEFLOAT unless you are certain that all instances of <b>value</b> contain floats.</p>
INTPARSE value	<p>This function converts a string (<b>value</b>) that contains integer values to integer format.</p> <p><b>Note</b></p> <p>Specifying a source column for which some instances do not contain integer values can cause the ETL job to fail. Cambridge Semantics recommends using TRYPARSEINT unless you are certain that all instances of <b>value</b> contain integers.</p>
LONGPARSE value	<p>This function converts a string (<b>value</b>) that contains a long integer value (from -2,147,483,648 to 2,147,483,647) to long format.</p> <p><b>Note</b></p> <p>Specifying a source column for which some instances do not contain long values can cause the ETL job to fail. Cambridge Semantics recommends using TRYPARSELONG unless you are certain that all instances of <b>value</b> contain long data.</p>

Function & Arguments	Description
SHORTPARSE  value	<p>This function converts a string (<b>value</b>) that contains a short integer value (from -32,678 to 32,767) to short format.</p> <div> <b>Note</b>            Specifying a source column for which some instances do not contain short values can cause the ETL job to fail. Cambridge Semantics recommends using TRYPARSESHORT unless you are certain that all instances of <b>value</b> contain short data.         </div>
TIMEPARSE  time text time format	<p>This function converts a string that contains <b>time text</b> to a time value in the <b>time format</b> that you specify.</p> <div> <b>Note</b>            Specifying a source column for which some instances do not contain time values can cause the ETL job to fail. Use this function only when all instances of <b>time text</b> contain a time value.         </div> <ul style="list-style-type: none"> <li>• <b>time text</b>: The property that contains the time value in string format.</li> <li>• <b>time format</b>: The format that you want the time value to follow. Specify "H" for hours, "m" for minutes, and "s" for seconds. For example, "HH:mm:ss."</li> </ul>
TOSTRING  value format	<p>This function converts a <b>value</b> that is a double data type to string <b>format</b>.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The double type values that you want to convert to string format.</li> <li>• <b>format</b>: The format code for the new string value. For example, "%.0f".</li> </ul>
TRYPARSEBOOLEAN  value if error	<p>This function attempts to convert a string <b>value</b> to a boolean data type. If an instance cannot be converted, Anzo replaces the string with the value in <b>if error</b>.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The string value that contains "true" or "false" values.</li> <li>• <b>if error</b>: The boolean value to replace the string with if an error occurs with the conversion.</li> </ul>
TRYPARSEDATE	<p>This function attempts to convert a date string <b>value</b> to a date data type in the</p>

Function & Arguments	Description
value date format if error	<p><b>date format</b> that you specify. If an instance cannot be converted, Anzo replaces the string with the value in <b>if error</b>.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The string value that contains date data.</li> <li>• <b>date format</b>: The format that you want the date to follow. Specify days as "d," months as "M," and years as "y." For example, "yyyy-MM-dd." Or if your data has values such as 09APR2020, specify the date format "ddMMMyyyy."</li> <li>• <b>if error</b>: The date value to replace the string with if an error occurs with the conversion.</li> </ul>
TRYPARSEDATETIME  value date format if error	<p>This function attempts to convert a datetime string <b>value</b> to a date data type in SQL date format (YYYY-MM-DD). If an instance cannot be converted, Anzo replaces the string with the value in <b>if error</b>.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The string value that contains datetime data.</li> <li>• <b>date format</b>: Anzo outputs values in SQL date format, YYYY-MM-DD.</li> <li>• <b>if error</b>: The date value to replace the string with if an error occurs with the conversion.</li> </ul>
TRYPARSEDECIMAL  value if error	<p>This function attempts to convert a string <b>value</b> to a decimal data type. If an instance cannot be converted, Anzo replaces the string with the value in <b>if error</b>.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The string value that contains decimal data.</li> <li>• <b>if error</b>: The decimal value to replace the string with if an error occurs with the conversion.</li> </ul>
TRYPARSEDOUBLE  value if error	<p>This function attempts to convert a string <b>value</b> to a double data type. If an instance cannot be converted, Anzo replaces the string with the value in <b>if error</b>.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The string value that contains double data.</li> <li>• <b>if error</b>: The double value to replace the string with if an error occurs with the conversion.</li> </ul>
TRYPARSEFLOAT	<p>This function attempts to convert a string <b>value</b> to a float data type. If an instance</p>

Function & Arguments	Description
value if error	<p>cannot be converted, Anzo replaces the string with the value in <b>if error</b>.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The string value that contains float values.</li> <li>• <b>if error</b>: The float value to replace the string with if an error occurs with the conversion.</li> </ul>
TRYPARSELONG  value if error	<p>This function attempts to convert a string <b>value</b> to a long data type. If an instance cannot be converted, Anzo replaces the string with the value in <b>if error</b>.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The string value that contains long data (-2,147,483,648 to 2,147,483,647).</li> <li>• <b>if error</b>: The long value to replace the string with if an error occurs with the conversion.</li> </ul>
TRYPARSESHORT  value if error	<p>This function attempts to convert a string <b>value</b> to a short data type. If an instance cannot be converted, Anzo replaces the string with the value in <b>if error</b>.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The string value that contains short data (-32,678 to 32,767).</li> <li>• <b>if error</b>: The short value to replace the string with if an error occurs with the conversion.</li> </ul>
URI()	<p>When specified in the Source column in mappings, this function transforms the property values to URI format by concatenating each of the components that you specify in the function. To ensure that values with spaces and other characters are encoded as valid URIs, the URI function is often used with the ENCODE function.</p> <p>For example, the following mapping for a "ticket_events" table transforms an "eventname" string to URI format by prepending "http://csi.com/" to the encoded event names:</p> <pre>String = URI("http://csi.com/", ENCODE(eventname)) Date Time = starttime Int =</pre>  <p>The screenshot shows a mapping configuration for a table named "ticket_events". It displays three fields: "String", "Date Time", and "Int". The "String" field is assigned the value "URI('http://csi.com/', ENCODE(eventname))". The "Date Time" field is assigned "starttime" and the "Int" field is assigned an empty value. A dropdown menu is open for the "String" field, showing the function "URI" with arguments "http://csi.com/" and "ENCODE(eventname)". The "ENCODE" function is highlighted in green, and its argument "eventname" is highlighted in red.</p>

Function & Arguments	Description
	<p>This example mapping results in triples such as:</p> <pre>&lt;ticketit_events&gt; &lt;eventname&gt; "http://csi.com/Rolling+Stones"</pre> <p>You can also enter URI() in the Target column to specify that the Expression in the Source column should be the URI of the entity that is being created. For example, the mapping below generates an entity URI by prepending "http://csi.com/Flight" to the flight number value:</p>  <p>The example URI specification results in triples such as:</p> <pre>&lt;http://csi.com/Flight1234&gt; &lt;FLIGHT_NUMBER&gt; 1234 &lt;http://csi.com/Flight1234&gt; &lt;ORIGIN_AIRPORT&gt; "BOS"</pre>

## Lookup and Mapping Functions

This section describes the lookup and map functions that Anzo supports.

Function & Arguments	Description
<p>LOOKUP</p> <p>from</p> <p>get</p> <p>fields</p> <p>values</p>	<p>This function enables you to look up values in a supplemental table. LOOKUP joins the <b>from</b> lookup table to a source table on the columns specified in the <b>fields</b> and <b>values</b> arguments. The function returns the value from <b>get</b> in the lookup table that corresponds to each row's value in the <b>values</b> argument.</p> <ul style="list-style-type: none"> <li>• <b>from</b>: The lookup table to perform the join against.</li> <li>• <b>get</b>: The field or property to retrieve from the lookup table in the <b>from</b> argument.</li> <li>• <b>fields</b>: The field or fields from the lookup table to compare with the values</li> </ul>

Function & Arguments	Description
	<p>from the primary table.</p> <ul style="list-style-type: none"> <li>• <b>values</b>: The values from the primary table to compare with the <b>fields</b> from the lookup table.</li> </ul>
MAKELIST expression	<p>This function maps multiple source columns to a single target property. The function does not create a list; it creates new rows, one for each column that is mapped to the target.</p> <ul style="list-style-type: none"> <li>• <b>expression</b>: The list of columns that you want to map to the target. After adding a source column press <b>Ctrl +</b> to select the next column.</li> </ul>
MAP value map	<p>This function retrieves values from a map that you define. The map is a collection of key/value pairs. The function uses the specified <b>value</b> as a key in the <b>map</b> and returns the value associated with the key.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The key or keys to use to look up the value from the map.</li> <li>• <b>map</b>: The map to use to look up the values. You can click the <b>map</b> argument name to open the Edit Map dialog box and define or change a map.</li> </ul>

## Numeric Functions

This section describes functions that operate on values with numeric data types.

Function & Arguments	Description
CEILING value	<p>This function rounds the <b>value</b> up to the next whole number if the value has a fractional part.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The source values that you want to round up to the next whole number.</li> </ul>
DIVIDE v1 v2	<p>This function divides the values of the numeric expressions that you specify (<b>v1/v2</b>).</p> <ul style="list-style-type: none"> <li>• <b>v1</b>: The divisor in the equation.</li> <li>• <b>v2</b>: The dividend in the equation.</li> </ul>
FLOOR	<p>This function rounds the <b>value</b> down to a whole number if the value has a fractional part.</p>

Function & Arguments	Description
value	<ul style="list-style-type: none"> <li>• <b>value</b>: The source values that you want to round down to a whole number.</li> </ul>
MULTIPLY v1 v2	<p>This function multiplies the values of the numeric expressions that you specify (<b>v1</b> x <b>v2</b>).</p> <ul style="list-style-type: none"> <li>• <b>v1</b>: The multiplicand in the equation.</li> <li>• <b>v2</b>: The multiplier in the equation.</li> </ul>
NUMERIC_ADD v1 v2	<p>This function adds the values of the numeric expressions that you specify (<b>v1</b> + <b>v2</b>).</p> <ul style="list-style-type: none"> <li>• <b>v1</b>: The first addend in the equation.</li> <li>• <b>v2</b>: The second addend in the equation.</li> </ul>
NUMERIC_SUBTRACT v1 v2	<p>This function subtracts the values of the numeric expressions that you specify (<b>v1</b> - <b>v2</b>).</p> <ul style="list-style-type: none"> <li>• <b>v1</b>: The minuend in the equation.</li> <li>• <b>v2</b>: The subtrahend in the equation.</li> </ul>
RANDOM value min range max range	<p>This function replaces <b>value</b> with a random integer from within the <b>min range</b> and <b>max range</b> that you specify.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The source values that you want to replace with a random integer.</li> <li>• <b>min range</b>: The integer that is the lowest number in the range that the function can choose from.</li> <li>• <b>max range</b>: The integer that is the highest number in the range that the function can choose from.</li> </ul>
ROUND value	<p>This function rounds the <b>value</b> up or down to the closest whole number.</p> <ul style="list-style-type: none"> <li>• <b>value</b>: The source values that you want to round up or down.</li> </ul>

## String Functions

This section describes functions that operate on values with string data types.

Function & Arguments	Description
<p>CONCATENATE</p> <p>text</p>	<p>This function concatenates multiple string values (<b>text</b>) and returns a single string.</p> <ul style="list-style-type: none"> <li>• <b>text</b>: The string values to concatenate, including any delimiters that you want to use. Press <b>Ctrl +</b> to enter multiple values.</li> </ul> <p>For example, the source mapping below concatenates PATIENTHOMESTATE and PATIENTHOMEZIP:</p> <p><b>CONCATENATE</b></p> <p>text PATIENTHOMESTATE PATIENTHOMEZIP</p>
<p>DATEPARSE</p> <p>date text</p> <p>date format</p>	<p>This function converts a string that contains a date value (<b>date text</b>) to the specified <b>date format</b>.</p> <ul style="list-style-type: none"> <li>• <b>date text</b>: The property that contains the date value in string format.</li> <li>• <b>date format</b>: The format that you want the date to follow. Specify days as "d," months as "M," and years as "y." For example, "yyyy-MM-dd."</li> </ul> <p>For example, the source mapping below converts the MovieReleaseDate values from strings to dates in the format "dd-MM-yyyy":</p> <p><b>DATEPARSE</b></p> <p>date text MovieReleaseDate date format "dd-MM-yyyy"</p> <p>The format that you specify for dates is flexible. For example, typing the format "dd-MMM-yy" displays values such as "01-JAN-19."</p>
<p>DATETIMEPARSE</p> <p>date text</p> <p>date format</p>	<p>This function converts a string that contains a datetime value (<b>date text</b>) to the specified <b>date format</b>.</p> <ul style="list-style-type: none"> <li>• <b>date text</b>: The property that contains the datetime value in string format.</li> <li>• <b>date format</b>: The format that you want the datetime to follow. For the date, specify days as "d," months as "M," and years as "y." For the time, specify "H" for hours, "m" for minutes, and "s" for seconds. For example, "yyyy-MM-dd HH:mm:ss."</li> </ul> <p>For example, the source mapping below converts the</p>



Function & Arguments	Description
	<p>PATIENTLASTPMODATE from a string value to a datetime value in the format "MM-dd-yyyy HH:mm:ss":</p> <p><b>DATETIMEPARSE</b>  date text <b>PATIENTLASTPMODATE</b>  date format "MM-dd-yyyy HH:mm:ss"</p>
LEFT  text  num chars	<p>This function starts on the left side of a <b>text</b> string, keeps the number of characters in <b>num chars</b>, and returns the truncated string.</p>
LOWER  value	<p>This function converts a string <b>value</b> to lower case letters.</p>
REGEX  input regex replace	<p>This function finds all patterns in the <b>input</b> string that match the specified regular expression (<b>regex</b>). It replaces the input patterns with the value in <b>replace</b> and returns the resulting string.</p> <ul style="list-style-type: none"> <li>• <b>input</b>: The source column or expression that evaluates to a string.</li> <li>• <b>regex</b>: The regular expression to use to find matches in the input string. For information about REGEX syntax, see the W3C <a href="#">Regular Expression Syntax</a> specification.</li> <li>• <b>replace</b>: The string that should replace the input patterns that match regex.</li> </ul> <p>For example, the source mapping below uses the REGEX function to search for the pattern "PS" in the COMPLAINTSTRING values and replaces each PS with a hyphen (-):</p> <p><b>REGEX</b>  input <b>COMPLAINTSTRING</b>  regex "PS"  replace "-"</p>
RIGHT	<p>This function starts on the right side of a <b>text</b> string, keeps the number of</p>

Function & Arguments	Description
text num chars	characters in <b>num chars</b> , and returns the truncated string.
SPLIT  string delimiter	<p>This function splits a <b>string</b> value into multiple values based on the specified <b>delimiter</b>.</p> <ul style="list-style-type: none"> <li>• <b>string</b>: The source column or function that evaluates to a string.</li> <li>• <b>delimiter</b>: The character to use to delimit the <b>string</b>.</li> </ul>
SPLITARRAY  string delimiter index	<p>This function splits a <b>string</b> value into an array based on the <b>delimiter</b>. From the array, the function retrieves only the portion of the value that you specify in the <b>index</b>.</p> <ul style="list-style-type: none"> <li>• <b>string</b>: The source column or function that evaluates to a string.</li> <li>• <b>delimiter</b>: The character to use to delimit the string.</li> <li>• <b>index</b>: An integer that specifies the portion of the array to retrieve. Indexes start at zero. The first portion of the array is 0, the second is 1, and so on. Choose an index that you know exists or the mapping becomes invalid.</li> </ul> <p>For example, the following source mapping retrieves only the last four digits of social security numbers:</p> <p><b>SPLITARRAY</b></p> <pre>string    SSN delimiter "-" index     2</pre>
STRLEN  term	This function returns the number of characters in the specified text string ( <b>term</b> ).
UPPER  value	This function converts a string <b>value</b> to upper case letters.

Related Topics

- [Creating a New Mapping](#)
- [Configuring Mappings to Ingest a Subset of the Source Data](#)
- [Transforming Data in Mappings](#)

Working with Pipelines

The topics in this section provide information about working with pipelines.

- [Creating a Dataset Pipeline](#)
- [Creating an ETL Pipeline](#)
- [Publishing a Pipeline or Subset of Jobs](#)
- [Canceling a Running Job](#)

Related Topics

- [Managing Dataset Editions](#)

Creating a Dataset Pipeline

This topic provides instructions for creating a new Dataset Pipeline to ingest data into Anzo. Dataset pipelines produce a new data set in the Dataset catalog and a file-based linked data set (FLDS) on the file store. This type of pipeline is created any time you ingest data by clicking the **Ingest** button for a data source.

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Then click the **Pipelines** tab. Anzo displays the Pipelines screen, which lists the existing pipelines. For example:

Data Sources		Schemas		Mappings		Pipelines			
<div><div></div><div>Search</div></div>		Sort By: Title		View: <div><div></div><div></div></div>		<div>Add Pipeline</div>			
<div></div>	Title	↑	Type	Version	Creator	Created Date	Updated Date	Tags	
	Load DB - northwind to Store		Dataset Pipel...	Derived from: Untitled Version	System Admi...	May 31, 2021	May 31, 2021	Auto-Gen	
	Load Movies to Store		Dataset Pipel...	Derived from: Untitled Version	System Admi...	Jun 7, 2021	Jun 7, 2021	Auto-Gen	
	Load Sample Movie Data to Store		Dataset Pipel...	Derived from: Untitled Version	System Admi...	Jun 2, 2021	Jun 2, 2021	Auto-Gen	
	Load Tickets to Store		Dataset Pipel...	Derived from: Untitled Version	System Admi...	May 31, 2021	May 31, 2021	Auto-Gen	

2. Click the **Add Pipeline** button at the top of the screen and select **Dataset Pipeline**. Anzo displays the Create Dataset Project screen.

Create Dataset Project

Title \*

The title of the project

Description

The description of the project

Engine Configuration \*

Select engine configuration

Graph datasource \*

Job Title \*

The title of the job

CANCEL

SAVE

3. Type a name for the pipeline in the **Title** field and enter an optional **Description**.
4. If necessary, click the **Engine Configuration** drop-down list and select the ETL engine for this pipeline.
5. If necessary, click the **Graph datasource** drop-down list and select the Anzo Data Store where you want Anzo to save the RDF files that are generated when jobs in this pipeline are published.
6. In the **Job Title** field, type a name for the first job in the pipeline.
7. Click **Save** to create the pipeline. Anzo displays the pipeline overview screen. For example:

Example Dataset Project

Not Versioned

Publish All

Overview

Jobs

History

Versions

Discussion

Description

None

Engine Configuration

Local Sparkler En...

Graph datasource

Store

General

Type

Dataset Pipeline

Creator

System Administrator

Updated

a few seconds ago

Released

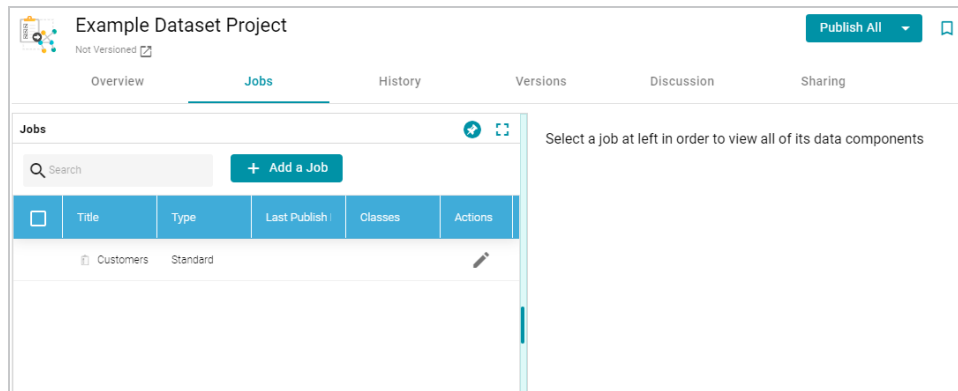
a few seconds ago

http://cambridgesemantics.com/Data...

Tags

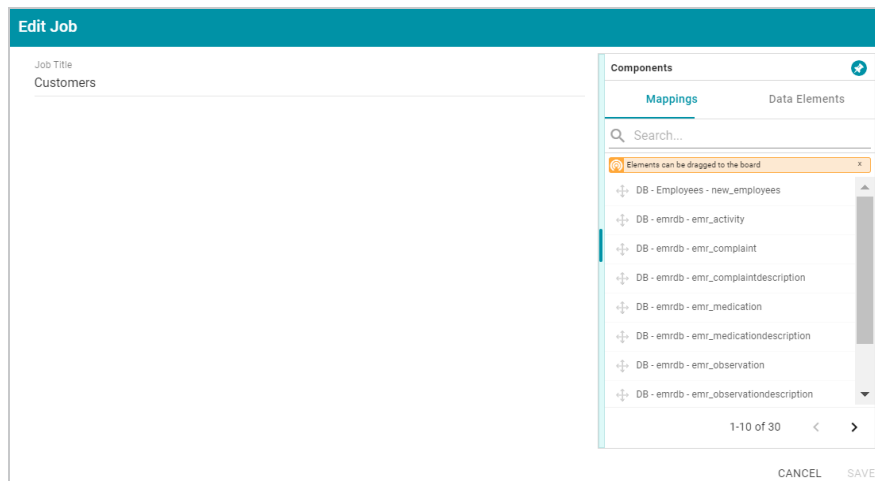
None

8. Click the **Jobs** tab to configure the jobs that this pipeline will run. Anzo displays the Jobs screen, which lists the job name that you specified when you created the project. For example:

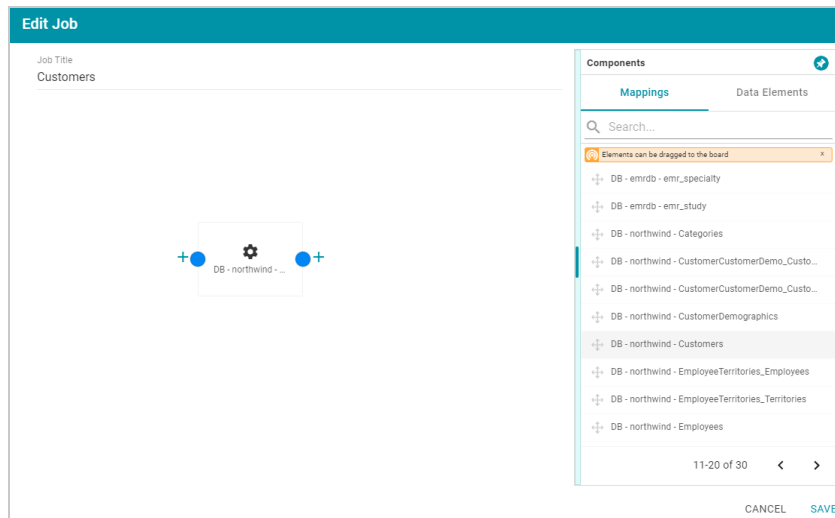


The Jobs pane on the left side of the screen lists the jobs in the project. Since this is a new project, the right side of the screen remains blank. Once this project is published, you can select a job to view the history of data components that the job has created.

9. To configure the first job to add the source, mapping, and target, click the edit icon (✎) in the Actions column for the initial job. Anzo opens the Edit Job dialog box. For example:



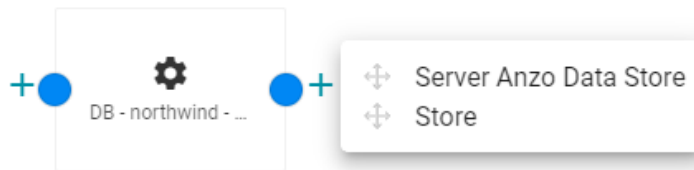
10. To configure the job, drag and drop a data source element from the Mappings or Data Elements tab in the Components list onto main part of the screen. For example, in the image below, the DB-northwind-Customers mapping is added to the job canvas:



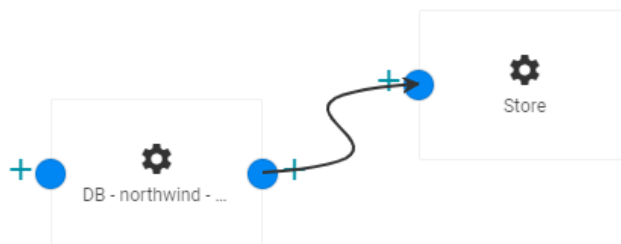
### Tip

If you drag a mapping onto the job canvas and there is only one source and one target for the mapping, Anzo automatically adds that source and target to the job.

11. To finish creating the job by adding any missing elements, click the plus icon on the right or left of an element on the canvas. Anzo suggests elements to add based on the existing element. For example, clicking the plus icon on the target side of the mapping element, presents two target choices:

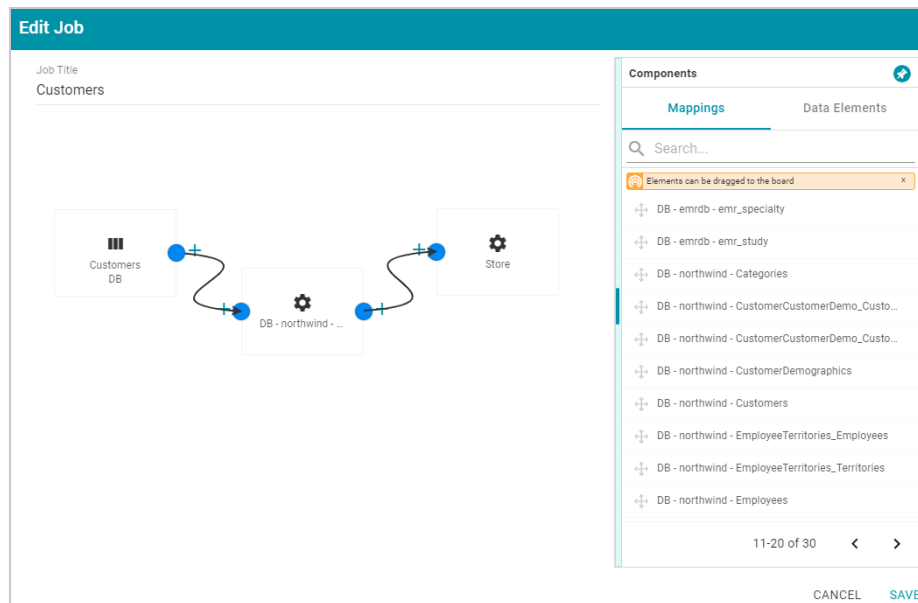


To add one of the options to the job, drag the option from the tooltip onto the canvas. For example:



To delete an element from the canvas, hover over the element and click the trashcan icon (🗑️).

12. Complete the job by adding any missing elements. For example:



13. Click **Save** to save the job and return to the Jobs screen.
14. If you want to create additional jobs for this project, click the **Add a Job** button and repeat the steps above to complete the job.
15. To run the pipeline and all of the jobs, click the **Publish All** button at the top of the screen. If you want to run a subset of jobs, click the checkbox next to the job that you want to run, and then click the **Publish** button at the top of the Jobs list and click **Run**.

The screenshot shows the 'Jobs' screen. At the top, there is a search bar and an 'Add a Job' button. Below the search bar, it says 'Selected: Customers' with a close button. There are 'Delete' and 'Publish' buttons. The main part of the screen is a table with the following columns: Title, Type, Last Publish Date, Classes, and Actions.

	Title	Type	Last Publish Date	Classes	Actions
<input checked="" type="checkbox"/>	Customers	Standard			
<input type="checkbox"/>	CustomerDemo	Standard			

When the pipeline completes, the data components created by the pipeline are displayed on the right side of the screen. The new (or updated) data set becomes available in the Dataset catalog. You can generate metrics on the graph data to start exploring the data. See [Generating a Dataset Data Profile](#) for more information. You can also add the new data set to a graphmart and load it to AnzoGraph so that you can access and analyze the data. See [Creating a Graphmart](#) for instructions.

## Creating an ETL Pipeline








This topic provides instructions for creating an ETL Pipeline that ingests data to a file or database. Create an ETL pipeline when you do not want to generate a new data set entry in the Anzo Dataset catalog or produce RDF files for AnzoGraph. Typically ETL pipelines are used to output data to a CSV file. ETL pipelines require a mapping that defines a file schema or database as the target.

For instructions on creating a Dataset Pipeline to ingest data into Anzo, see [Creating a Dataset Pipeline](#). For information about creating unstructured pipelines, see [Creating an Unstructured Pipeline](#).

## Publishing a Pipeline or Subset of Jobs

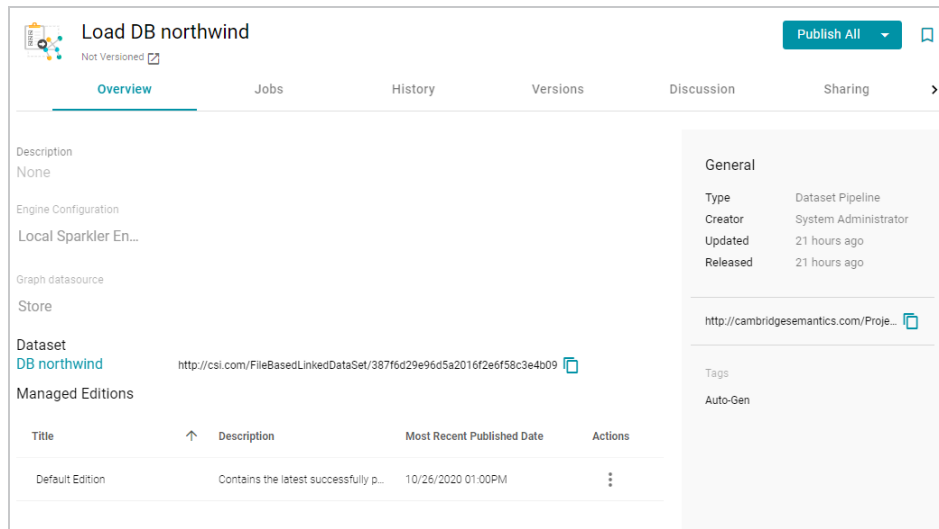
This topic provides guidance on publishing a pipeline or specific jobs in a pipeline.

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Then click the **Pipelines** tab. Anzo displays the Pipelines screen, which lists the existing pipelines. For example:

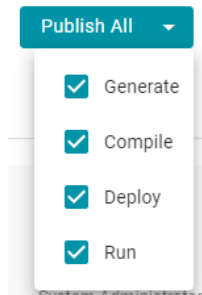
Data Sources		Schemas		Mappings		Pipelines		
 Search		Sort By: Title		View:  		<button>Add Pipeline</button>		
<input type="checkbox"/>	Title	Type	Version	Creator	Created Date	Updated Date	Tags	
	 Load DB - northwind to Store	Dataset Pipel...	Derived from: Untitled Version	System Admi...	May 31, 2021	May 31, 2021	Auto-Gen	
	 Load Movies to Store	Dataset Pipel...	Derived from: Untitled Version	System Admi...	Jun 7, 2021	Jun 7, 2021	Auto-Gen	
	 Load Sample Movie Data to Store	Dataset Pipel...	Derived from: Untitled Version	System Admi...	Jun 2, 2021	Jun 2, 2021	Auto-Gen	
	 Load Tickets to Store	Dataset Pipel...	Derived from: Untitled Version	System Admi...	May 31, 2021	May 31, 2021	Auto-Gen	

2. Click the name of the pipeline that you want to publish. Anzo displays the pipeline overview screen. For example:

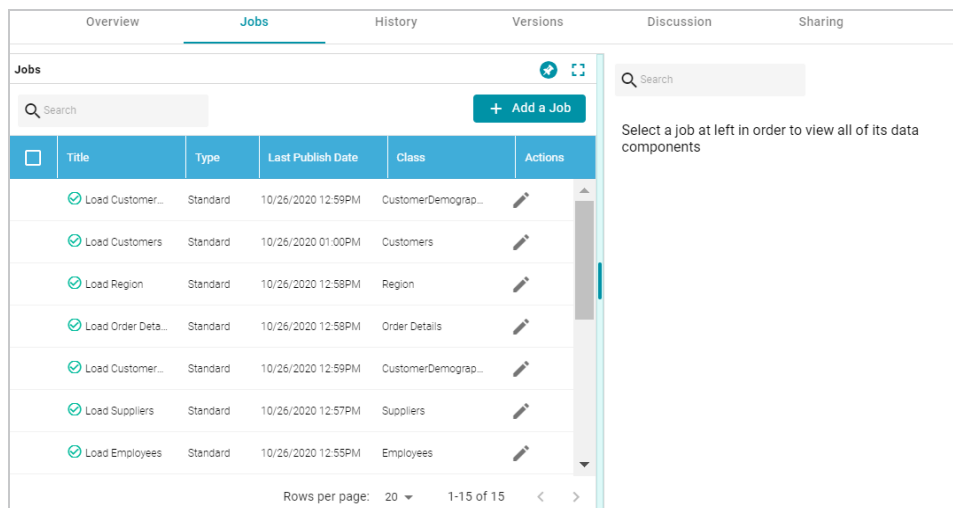




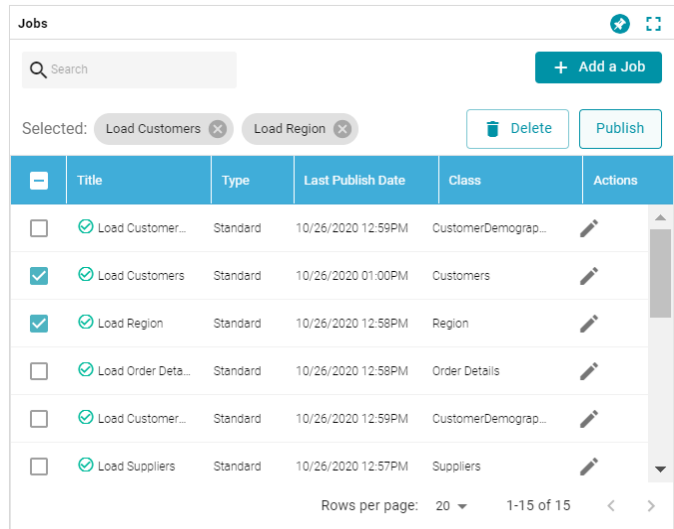
- To publish all of the jobs in the pipeline, click the **Publish All** button. To see the steps that will be executed when **Publish All** is clicked, click the arrow to the right of the name. For example, the image below shows that the ETL engine is configured to perform all steps when **Publish All** is clicked:



- To publish a subset of jobs instead of the entire pipeline, click the **Jobs** tab. The jobs are listed on the left side of the screen. For example:



In the list of jobs, select the checkbox next to each job that you want to publish, and then click the Publish button at the top of the table. For example:



The screenshot shows the 'Jobs' interface in Anzo. At the top, there is a search bar and an 'Add a Job' button. Below the search bar, there are filters for 'Selected: Load Customers' and 'Load Region', along with 'Delete' and 'Publish' buttons. The main part of the interface is a table with the following columns: Title, Type, Last Publish Date, Class, and Actions. The table lists several jobs, with checkboxes in the first column. The jobs listed are: 'Load Customer...', 'Load Customers', 'Load Region', 'Load Order Deta...', 'Load Customer...', and 'Load Suppliers'. The 'Load Customers' and 'Load Region' jobs are selected. At the bottom, there is a 'Rows per page' dropdown set to 20 and a pagination indicator showing '1-15 of 15'.

	Title	Type	Last Publish Date	Class	Actions
<input type="checkbox"/>	Load Customer...	Standard	10/26/2020 12:59PM	CustomerDemograp...	
<input checked="" type="checkbox"/>	Load Customers	Standard	10/26/2020 01:00PM	Customers	
<input checked="" type="checkbox"/>	Load Region	Standard	10/26/2020 12:58PM	Region	
<input type="checkbox"/>	Load Order Deta...	Standard	10/26/2020 12:58PM	Order Details	
<input type="checkbox"/>	Load Customer...	Standard	10/26/2020 12:59PM	CustomerDemograp...	
<input type="checkbox"/>	Load Suppliers	Standard	10/26/2020 12:57PM	Suppliers	

Rows per page: 20 1-15 of 15

When the pipeline or jobs finish, this run of the pipeline becomes the **Default Edition**. The Default Edition always contains the latest successfully published data for all of the jobs in the pipeline. If one or more of the jobs failed, those jobs are excluded from the Default Edition. If you publish the failed jobs at a later date or you create and publish additional jobs in the pipeline, the data from those jobs is also added to the Default Edition. For more information about editions, see [Managing Dataset Editions](#).

The new or updated data set also becomes available in the Dataset catalog. From the catalog, you can generate graph data profiles and create graphmarts.

## Canceling a Running Job

Follow the instructions below to cancel a running ETL job in the Anzo application.

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Then click the **Pipelines** tab. Anzo displays the Pipelines screen, which lists the existing pipelines. For example:

Data Sources

Schemas

Mappings

Pipelines

Search

Add Pipeline

<div><div></div></div>	Title	↑	Type	Creator	Created Date	Updated Date	Actions
<div><div></div><div></div></div>	<div><div></div><div>Load MySQL DB - emrdbsmall to</div></div>		Dataset Pipeline,...	System Adminis...	Oct 19, 2021	Oct 19, 2021	<div><div></div><div></div></div>
<div><div></div><div></div></div>	<div><div></div><div>Load MySQL DB - mysql to Store</div></div>		Dataset Pipeline,...	System Adminis...	Oct 19, 2021	Oct 19, 2021	<div><div></div><div></div></div>
<div><div></div><div></div></div>	<div><div></div><div>Load MySQL DB - northwind to S</div></div>		Dataset Pipeline,...	System Adminis...	Oct 19, 2021	Oct 19, 2021	<div><div></div><div></div></div>

- Click the Pipeline whose job you want to cancel, and then click the **History** tab. Note the jobs that are in progress. For example, the image below shows a job that has the Status of **running**:

SSB

# Load MySQL DB - northwind to Store

Not Versioned

Publishing

Overview

Jobs

History

Versions

Discussion

Search

<div></div>	Name	Start Time	End Time	Status	Application	Updated Date	Tags	
	LoadOrderDetails	10/19/2021 9:23		running				<div>View Logs</div>

Rows per page: 25

26-30 of 30

- To cancel the job that is running, select the checkbox for that row, and then click **Delete** at the bottom of the screen. For example:

Load MySQL DB - northwind to Store

Not Versioned

Publishing

Overview

Jobs

History

Versions

Discussion

Search

	Name	Start Time	End Time	Status	Application	Updated Date	Tags	
<input type="checkbox"/>	LoadOrderSubtot	10/19/2021 9:12	10/19/2021 9:13	success				<a href="#">View Logs</a>
<input type="checkbox"/>	LoadAlphabetica	10/19/2021 9:11	10/19/2021 9:12	success				<a href="#">View Logs</a>
<input checked="" type="checkbox"/>	LoadOrderDetails	10/19/2021 9:14		running				<a href="#">View Logs</a>

Rows per page: 25 1-25 of 30

1 Items Selected

Delete

Bookmark

### Note

Clicking **Delete** for an in-progress job does not remove the job from the Pipeline. It simply cancels the current run of that job.

Anzo cancels the job and removes the run history for that job.

## Related Topics

[Publishing a Pipeline or Subset of Jobs](#)

## Directly Loading Data Sources via Graphmarts

There are two ways to directly load a data source. You can build a graphmart from a selected source or you can add a data source to an existing, activated graphmart. Both procedures automatically generate data layers to extract, load, and transform the data to a knowledge graph.

### Tip

When you build a new graphmart from a data source, advanced graphmart options are made available that are not presented when you add a data source to an existing graphmart. See [Graphmart Options](#) for information about the additional options.

Click an option below to view instructions.

- [Creating a Graphmart from a Data Source](#)
- [Adding a Data Source to an Existing Graphmart](#)

## Related Topics

[Onboarding or Virtualizing Data with the Graph Data Interface](#)

### Creating a Graphmart from a Data Source

Follow the steps below if you want to create a new graphmart from a data source.

### Note

AnzoGraph uses the Graph Data Interface (GDI) Java plugin to connect directly to sources. For file-based sources, make sure the source files are available to AnzoGraph on the shared file store. For databases, if you have configured custom drivers to access those sources in Anzo, the same drivers need to be added to AnzoGraph. For instructions, see [Deploy Optional Drivers for Accessing Custom Database Sources](#) in the Deployment Guide.

1. If necessary, add the data source. See [Adding Data Sources](#) for instructions.
2. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:

Data Sources   Schemas   Mappings   Pipelines										
<div> <div> <input type="text" value="Search"/> </div> <div> </div> <div>Add Data Source</div> </div>										
<input type="checkbox"/>	Title		Description	Type	Schema	Creator	Created Date	Updated Date	Actions	
<input type="checkbox"/>	DB			Database Data ...	emrdb, new_h...	System Admini...	Dec 8, 2022	Dec 8, 2022		
<input type="checkbox"/>	Movies			CSV Data Sour...	Movies	System Admini...	Dec 8, 2022	Dec 8, 2022		
<input type="checkbox"/>	Tickets			CSV Data Sour...	Tickets	System Admini...	Dec 9, 2022	Dec 12, 2022		
<div>Rows per page: 25 1-3 of 3</div>										

3. Select the checkbox next to the data source that you want to ingest. Options are enabled at the bottom of the screen. For example:

Data Sources

Schemas

Mappings

Pipelines

Search

Add Data Source

	Title	Description	Type	Schema	Creator	Created Date	Updated Date	Actions
<input type="checkbox"/>	DB		Database Data ...	emrdb, new_h...	System Admini...	Dec 8, 2022	Dec 8, 2022	<a href="#">Bookmark</a> <a href="#">More</a>
<input checked="" type="checkbox"/>	Movies		CSV Data Sour...	Movies	System Admini...	Dec 8, 2022	Dec 8, 2022	<a href="#">Bookmark</a> <a href="#">More</a>
<input type="checkbox"/>	Tickets		CSV Data Sour...	Tickets	System Admini...	Dec 9, 2022	Dec 12, 2022	<a href="#">Bookmark</a> <a href="#">More</a>

Rows per page: 25

1-3 of 3

<

>

1 Items Selected

Delete

Bookmark

Create Graphmart

Add To Package

### Tip

Users with the **Batch Direct Data Loading** permission can select multiple data sources to ingest.

4. Click the **Create Graphmart** button. The Create Graphmart dialog box is displayed. For example:

### Create Graphmart

This will create a graphmart containing data layers for each of your selected data sources. Each layer will contain a graph representation of each data source.

Graphmart Name \*

Movies

Graphmart Description

☐ Export to Dataset

Advanced

CANCEL CREATE GRAPHMART

#### Note

If the selected data source includes more than one schema, the number of schema instances is shown at the top of the screen. For example:

This will create a graphmart containing data layers for each of your selected data sources. Each layer will contain a graph representation of each data source.

Selected: 1 Data Sources. 2 Schema Instances. [SHOW](#)

Graphmart Name \*

MySQL DB

Clicking **Show** displays the selected schemas. For example:

Selected: 1 Data Sources. 2 Schema Instances. [HIDE](#)

Choose schema instances (min. one is required):

☒ emrdbsmall-instance ☒ northwind-instance



If you would like to exclude one or more schemas, clear the checkbox for each schema that you want to exclude.

5. On the Create Graphmart dialog box, the **Graphmart Name** is populated with the name of the selected source. If multiple sources were selected, the Graphmart Name is blank. Edit the **Graphmart Name** if necessary and add an optional **Description** for the new graphmart.
6. If you would like to configure any of the advanced settings, click **Advanced** to view the options. For details about the each of the Advanced settings, see [Direct Load Advanced Settings Reference](#).
7. When you have finished configuring the workflow, click **Create Graphmart**. The new graphmart is created and activated and the data layers and steps are generated according to the chosen strategy. A *managed model* is also generated. See [Introduction to Managed Models](#) for information. If you chose to export a dataset, the new dataset is also added to the Datasets catalog.

Once the graphmart is online, the data can be analyzed. See [Access](#) for next steps. Or see [Working with Graphmarts](#) for information about managing graphmarts.

























## Adding a Data Source to an Existing Graphmart

Follow the steps below if you want to add a data source to an existing graphmart.


### Note

AnzoGraph uses the Graph Data Interface (GDI) Java plugin to connect directly to sources. For file-based sources, make sure the source files are available to AnzoGraph on the shared file store. For databases, if you have configured custom drivers to access those sources in Anzo, the same drivers need to be added to AnzoGraph. For instructions, see [Deploy Optional Drivers for Accessing Custom Database Sources](#) in the Deployment Guide.

1. If necessary, add the data source. See [Adding Data Sources](#) for instructions.
2. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing graphmarts. For example:


	<input type="text" value="Search"/>				<a href="#">Add Graphmart</a>	
<input type="checkbox"/>	Title	Description	Status	Statements	Creator	Actions
	Books		 Ready to use	156	System Administrator	 
	Flights		 Ready to use	265,339	System Administrator	 
	Movies		 Ready to use	104,166	System Administrator	 
	Northwind		 Ready to use	131,386	System Administrator	 
	Tickets		 Ready to use	3,696,548	System Administrator	 
<div> Rows per page: 25 1-5 of 5 </div>						

- Click the name of the graphmart that you want to add the data source to. The Overview is displayed. For example:


**Movies**

Not Versioned

INACTIVE
ACTIVE
Ready to use
AZG
Static

[Save a Copy](#)
[Create Dashboard](#)


Overview
Explore
Datasets
Data Layers
Dashboards
Data on Demand

Description  
None

Load Priority  
None

**Metrics**  
Profile data upon activation

**Disable Load Counts**  
Disable populating counts during load operations

**Data Loading Settings**  
Load layers that do not fail

**Ignore Source Errors**  
Ignore any step or view's source layer if they failed to load

**Leave Graphmart Online During Refresh**  
When refreshing layers, leave graphmart and layer online.

**Manual Refresh Graphmart**  
Once loaded, changes only pushed to AnzoGraph manually.(Only affects Journal Based Data)

**General**  
Type : Graphmart  
Creator : System Administrat...  
Last Accessed : 6 hours ago  
Last Updated : 6 hours ago  
Structure Modified : 7 hours ago  
Released : 7 hours ago

<http://cambridgesemantics.com/Graph>

[Dynamic Resource Log](#)


**AnzoGraph Server Details**  
AnzoGraph : AZG  
Status : Ready to use  
Last Accessed : 2 minutes ago  
Memory Used : 281.25 MB( 3% )  
Memory Total : 11.7 GB



Inactivity Deactivate Timeout  
None

4. If necessary, activate the graphmart. Graphmarts must be online to be able to add data sources to them.
5. Click the **Data Layers** tab. On the Data Layers tab, click **Add** on the right side of the screen and select **Data Source**. The Select Data Source dialog box is displayed. For example:

### Select Data Source

Select data sources and sample size to generate layers and load steps.



<input type="checkbox"/>	Title	Description	Type	Schema	Creator	Created Date	Updated Date
<input checked="" type="checkbox"/>	DB-QA		Database Data Sou...	northwind	System Administra...	Nov 16, 2022	Nov 16, 2022
<input type="checkbox"/>	Flights		CSV Data Source	Flights	System Administra...	Nov 16, 2022	Nov 16, 2022
<input type="checkbox"/>	Movies		CSV Data Source	Movies	System Administra...	Nov 16, 2022	Nov 16, 2022
<input type="checkbox"/>	Tickets		CSV Data Source	Tickets	System Administra...	Nov 16, 2022	Nov 16, 2022

Rows per page: 25 1-4 of 4

Advanced

CANCEL CREATE LAYER & STEP

6. Select the checkbox next to the data source that you want to add to the graphmart. (Users with the **Batch Direct Data Loading** permission can select multiple sources.) The selected schema or schemas are shown at the bottom of the screen. For example, in the image a below, a source with two schema instances is selected.

Search

	Title	Description	Type	Schema	Version	Creator	Created Date	Updated Date	
<input checked="" type="checkbox"/>	DB-QA		Database Dat...	emrdbsmall...		System Admi...	Nov 16, 2022	Nov 16, 2022	
<input type="checkbox"/>	Flights		CSV Data So...	Flights		System Admi...	Nov 16, 2022	Nov 16, 2022	
<input type="checkbox"/>	Movies		CSV Data So...	Movies		System Admi...	Nov 16, 2022	Nov 16, 2022	
<input type="checkbox"/>	Tickets		CSV Data So...	Tickets		System Admi...	Nov 16, 2022	Nov 16, 2022	

Rows per page: 25

1-4 of 4

1 Items Selected

2 of 2 Schema Instances selected.

SHOW

Advanced

CANCEL

CREATE LAYER & STEP

Clicking **Show** displays the selected schemas. For example:

2 of 2 Schema Instances selected.

HIDE

Choose schema instances (min. one is required and any change in selection of the datasources above will reset these):

☒ emrdbsmall-instance
 ☒ northwind-instance

If you would like to exclude one or more schemas, clear the checkbox for each schema that you want to exclude.

- If you want to configure any of the advanced settings, click **Advanced** to view the options. For details about the each of the Advanced settings, see [Direct Load Advanced Settings Reference](#).
- When you have finished configuring the workflow, click **Create Layer & Step**. The new layer is created and the steps are generated according to the chosen strategy. A *managed model* is also generated for the layer. See [Introduction to Managed Models](#) for information.

Once the layer is online, the data can be analyzed. See [Access](#) for next steps. Or see [Working with Graphmarts](#) for information about managing graphmarts.

## Direct Load Advanced Settings Reference

This topic describes the Advanced options that are available when you create or configure the Direct Data Load workflow to load a data source via auto-generated data layers.

Advanced ^

Graphmart Options

☐ Export to Dataset    ☐ Find Connections    ☐ Profile Data

Layer Generation Strategies

Each strategy results in the same graph data but uses a different method to perform the load. The queries created for each strategy are stored in their own layer. Only one of these layers should be enabled at a time.

☒ Single Step    ☐ Multiple Steps

Enabled Layer

Single Step ▼

Ontology URI optional

Used as the base for any class or property URIs in the generated model.

Load Options

☒ Enable Partitioning

Sampling Limit

Graphmart Options

- [Export to Dataset](#)
- [Find Connections](#)
- [Profile Data](#)

Layer Generation Strategies

- [Single Step](#)
- [Multiple Steps](#)
- [Enabled Layer](#)

Ontology URI

Enable Partitioning

Sampling Limit

Direct Load Advanced Settings Reference

209

## Graphmart Options

The Graphmart Options are available when you create a graphmart from a data source (as described in [Creating a Graphmart from a Data Source](#)). They are not available when you add a data source to an existing graphmart (as described in [Adding a Data Source to an Existing Graphmart](#)).

### Export to Dataset

This setting controls whether the automated workflow generates a dataset by exporting the graphmart. This option does not create an Export Step in the graphmart but it does generate a dataset in the selected data store and adds the dataset to the Datasets catalog. If you leave Export to Dataset disabled, a dataset is not automatically generated. For information about creating an Export Step to export a graphmart to a dataset at a later time, see [Exporting Data to an FLDS \(Export Step\)](#). If you enable Export to Dataset, the following settings are displayed:

☒ Export to Dataset

Advanced

**Dataset Anzo Data Store \***

Store

**Post Dataset Export Action \***

☒ Leave Graphmart Activated

☐ Deactivate Graphmart

☐ Delete Graphmart

**Dataset Format**

ttl.gz

Set the dataset's default data format, such as ttl or ttl.gz.

### Dataset Anzo Data Store

This required setting specifies the Anzo Data Store to export the dataset to. The data store must be a location on the shared file store that AnzoGraph has access to.

## Post Dataset Export Action

This required setting specifies how to treat the graphmart after the Export Step is processed. To leave the graphmart online, select **Leave Graphmart Activated** (the default value). To retain the graphmart but disable it and remove the data from AnzoGraph, select **Deactivate Graphmart**. And to designate the graphmart as temporary and remove it after the dataset is exported, select **Delete Graphmart**.

## Dataset Format

This setting specifies the file format for the RDF TTL files that are generated, i.e., whether they are compressed or not. The valid options are **ttl** for uncompressed and **ttl.gz** for compressed.

## Find Connections

This optional setting specifies whether to find relationships between tables in the schema (or between data sources if multiple sources are selected). Finding connections is useful if the schema does not define primary and foreign key relationships and you want the Graph Data Interface to create the connections.

### Note

When **Find Connections** is enabled, two models are created, one that contains the classes and properties and one that contains only the connections. To view the complete model, both models must be added to the Working Set in the Model viewer.

## Profile Data

This optional setting specifies whether to generate a Data Profile after the graphmart is activated. For information about the metrics that are run when a profile is generated, see [Data Profiling Metrics](#).

## Layer Generation Strategies

These settings control the strategy to use for auto-generating the data layer queries in the graphmart. Each strategy produces the same graph data but uses a different method for structuring the queries that produce the data. You can select both options if you want and review the resulting layers and steps. Each option results in a separate layer. However, only one of the resulting layers can be enabled by default.

### Single Step

This is the default strategy. When **Single Step** is selected, a layer with a single Direct Load Step is created. The single query loads all tables from the selected schema or schemas. The generated query is an RDF and Ontology Generator query. See [Generating a Graph and Ontology with a Direct Load Step](#) to learn more about the GDI RDF and Ontology Generator.

## Multiple Steps

When Multiple Steps is selected, the layer has a separate Direct Load Step for each table in the selected schema or schemas. The generated query in each step is also an RDF and Ontology Generator query. With this strategy, you can enable and disable certain steps to control which tables are included in the graphmart.

## Enabled Layer

This required setting specifies the data layer that should be enabled by default when the graphmart is activated.

## Ontology URI

This optional setting specifies the custom URI to use for the model that is automatically generated. The value must be a valid URI without a hash (#) or slash (\) character at the end. If you do not specify a custom URI, the Graph Data Interface generates a URI in the following format:

```
http://cambridgesemantics.com/Layer/<layer_ID>/Model
```

## Enable Partitioning

This option specifies whether to enable file partitions for file-based data sources. When file partitions are enabled, files will be partitioned and ingested in parallel for increased performance.

### Note

Multiline CSV files may fail to load when this option is enabled. If the following type of error message is returned when onboarding files, disable the Enable File Partitions setting:

```
File uses multiline records and cannot be segmented.  
Please disable segmenting for this file.
```

## Sampling Limit

This optional setting specifies the number of rows to scan before inferring the data types for each column.



# Onboarding or Virtualizing Data with the Graph Data Interface

The topics in this section introduce you to the Graph Data Interface (GDI) and provide instructions for manually exploring, analyzing, virtualizing, and ingesting data from data sources.

Tip

This section focuses on writing your own SPARQL queries to read, ingest, or virtualize data from various sources. The Graph Data Interface can also be used to automatically generate models, data layers, and steps with ingestion queries that you can edit. For more information about the automated workflow, see [Directly Loading Data Sources via Graphmarts](#).

Introduction to the GDI

213

General GDI Concepts and Usage

215

GDI Usage by Data Source Type

299

GDI Property Reference

403

## Introduction to the GDI

The Graph Data Interface (GDI) (sometimes called the Data Toolkit) is an extremely flexible and configurable AnzoGraph plugin that enables users to access a variety of data sources via federated SPARQL queries. Depending on the type of query you write, i.e., whether it is an INSERT query against the GDI service or a CONSTRUCT query against the view or virtualized service, you can ingest source data into Anzo or create a virtual graph that accesses the source only when it is needed without ingesting the data into Anzo.

The GDI has built-in, native support for various file format types, HTTP/REST endpoints, and common database types. Internally, the GDI API has a records-oriented view of data. This view enables the GDI to bridge graph operations to operations for data in other formats. Though the GDI views the source as rows in a table, ultimately it has the capability to convert the records to graph format, enabling the data to be incorporated into data layers to augment existing data.

## Supported Data Sources

This table below lists the data sources, file systems, and applications that the GDI supports.

Source	Description
HTTP/REST Endpoints	The GDI natively supports reading or ingesting data from HTTP/REST endpoints.

Source	Description
<b>Databases</b>	<p>Cambridge Semantics supplies JDBC drivers for the following databases:</p> <ul style="list-style-type: none"> <li>• Databricks</li> <li>• H2</li> <li>• IBM DB2</li> <li>• Microsoft SQL Server</li> <li>• MariaDB</li> <li>• Oracle</li> <li>• PostgreSQL</li> <li>• SAP Sybase (jTDS)</li> <li>• Snowflake</li> </ul> <p>To extend the service to access other databases, additional JDBC drivers can be added to AnzoGraph. For information about acquiring additional JDBC drivers, contact your Cambridge Semantics Customer Success manager. For instructions on deploying other drivers, see <a href="#">Deploy Optional Drivers for Accessing Custom Database Sources</a> in the Deployment Guide.</p>
<b>File Formats</b>	<p>The following file types are supported:</p> <ul style="list-style-type: none"> <li>• CSV and TSV</li> <li>• JSON and NDJSON</li> <li>• Parquet</li> <li>• SAS (SAS Transport XPT and SAS7BDAT formats)</li> <li>• XML</li> <li>• Raw text format</li> </ul>
<b>File Systems</b>	<p>The following types of file storage systems are supported:</p> <ul style="list-style-type: none"> <li>• Amazon S3</li> <li>• FTP &amp; FTPS</li> <li>• Google Cloud Storage</li> <li>• HDFS (Kerberized HDFS is not supported at this time.)</li> </ul>

Source	Description
	<ul style="list-style-type: none"> <li>• NFS</li> <li>• SFTP</li> <li>• WebDAV</li> </ul>
<b>Applications</b>	Queries against Elasticsearch and Kafka applications are supported.

## Related Topics

[General GDI Concepts and Usage](#)

### General GDI Concepts and Usage

The topics in this section help you get to know the Graph Data Interface (GDI) by introducing you to the main concepts and giving a general overview of the query syntax, available properties, and functionality that is applicable across query and data source types.

[Getting Started with GDI Queries](#)

[Generating a Graph and Ontology with a Direct Load Step](#)

[Data Type Formatting Options](#)

[Normalization Options](#)

[Data Linking Options](#)

[Pagination Options](#)

[Bindings and Hierarchies](#)

[Incremental Onboarding with the GDI](#)

[Reading Data Source Metadata](#)

### Getting Started with GDI Queries

This topic provides details about the structure to use when writing GDI queries. It focuses on the properties that are common to all types of data sources. It also includes example queries that demonstrate the data integration capabilities for different types of sources.

#### Tip

Rather than manually writing complex queries, you can use the GDI to automatically generate graphs and ontologies by including a few key statements in a relatively simple query. For information, see [Generating a Graph and Ontology with a Direct Load Step](#).

- [GDI Query Syntax](#)
- [GDI Query Examples](#)

## GDI Query Syntax

The following query syntax shows the structure of a GDI query. The clauses, patterns, and placeholders that are links are described below.

```
# PREFIX Clause
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>

# Result Clause
{
  [ GRAPH ${targetGraph} { ]
    triple_patterns
  [ ] ]
}
[ ${usingSources} ]

WHERE
{
  # SERVICE Clause: Include the following service call when reading or
  inserting data.
  SERVICE [ TOPDOWN ] <http://cambridgesemantics.com/services/DataToolkit>

  # View SERVICE Clause: Or use the service call below when constructing a
  view.
  SERVICE <http://cambridgesemantics.com/services/DataToolkitView>
    (${targetGraph})

  {
    ?data a s:source_type ;
  }
  # Based on the source_type, additional connection and input parameters are
```

```

# available. The options below are valid for all sources. For source-related
# options, see Source-Specific Properties in the GDI property reference.
  s:url "string" ;
  [ s:username "string" ; ]
  [ s:password "string" ; ]
  [ s:timeout int ; ]
  [ s:batching boolean | int ; ]
  [ s:paging [ pagination_options ; ]
  [ s:concurrency int | [ list_of_properties ] ; ]
  [ s:rate int | "string" ; ]
  [ s:locale "string" ; ]
  [ s:sampling int ; ]
  [ s:selector "string" | [ list ] ; ]
  [ s:model "string" ; ]
  [ s:key ("string") ; ]
  [ s:reference [ s:model "string" ; s:using ("string") ]
  [ s:formats [ datatype_formatting_options ] ; ]
  [ s:normalize boolean | [ normalization_rules ] ; ]
  [ s:count ?variable ; ]
  [ s:offset int ; ]
  [ s:limit int ; ]
  # Mapping variables
  ?mapping_variable ( [ "binding" ] [ datatype ] [ "datetime_format" ] ) ;
  ... ;
  .
  # Additional clauses such as BIND, VALUES, FILTER
}
}

```

### Note

For readability, the parameters below exclude the base URI

[<http://cambridgesemantics.com/ontologies/DataToolkit#>](http://cambridgesemantics.com/ontologies/DataToolkit#) as well as the `s:` prefix.

As shown in the examples, however, the `s:` prefix or full property URI does need to be included in queries.

Option	Data Type	Description
<b>PREFIX Clause</b>	N/A	<p>The PREFIX clause declares the standard and custom prefixes for GDI queries. Generally, queries include the following prefixes (or a subset of them) plus any data-specific declarations:</p> <pre> PREFIX s: &lt;http://cambridgesemantics.com/ontologies/DataToolkit#&gt; PREFIX rdf:    &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX rdfs:   &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX xsd:    &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX owl:  &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX anzo: &lt;http://openanzo.org/ontologies/2008/07/Anzo#&gt; PREFIX zowl: &lt;http://openanzo.org/ontologies/2009/05/AnzoOwl#&gt; PREFIX dc:     &lt;http://purl.org/dc/elements/1.1/&gt; </pre>
<b>Result Clause</b>	N/A	<p>The result clause defines the type of SPARQL query to run and the set of results to return, i.e., whether you want to read (SELECT or CONSTRUCT) from the source or ingest the data into Anzo (INSERT).</p>
<b>GRAPH <code>\${targetGraph}</code></b>	N/A	<p>Include the <code>GRAPH</code> keyword and target graph parameter <code>\${targetGraph}</code> when writing an INSERT query to ingest data into a graphmart. Anzo automatically populates the query with the appropriate target URIs when the query runs.</p>
<b><code>\${usingSources}</code></b>	N/A	<p>Include the source graph parameter <code>\${usingSources}</code> when writing a "topdown" query that passes values from the data that is in the graphmart to the data source. Anzo automatically populates the query with the appropriate FROM clauses when the query runs. When passing literal values to the remote source, you do not need to include the source graph parameter. The <a href="#">SERVICE Clause</a> description below includes more information about passing input to data sources.</p>

Option	Data Type	Description
<b>SERVICE Clause</b>	N/A	<p>Include the SERVICE call <code>SERVICE [ TOPDOWN ]</code> <code>&lt;http://cambridgesemantics.com/services/DataToolkit&gt;</code> to invoke the GDI service when you are running a SELECT, INSERT, or CONSTRUCT query that is not creating a view. When writing a CONSTRUCT query in a View Step, use the <code>DataToolkitView</code> service call, as described below in <a href="#">View SERVICE Clause</a>.</p> <p>Include the optional TOPDOWN keyword when you want to pass input values from the graphmart to the data source. When you include TOPDOWN in the service call, it indicates that the rest of the query produces values to send to the source. In this case, the GDI makes repeated calls to pass in each of the specified values and retrieve the data that is based on those values.</p>
<b>View SERVICE Clause</b>	N/A	<p>When writing a CONSTRUCT query that creates a view of the data (usually in a View Step), include the following SERVICE call: <code>SERVICE</code> <code>&lt;http://cambridgesemantics.com/services/DataToolkitView&gt;</code> <code>(\${targetGraph})</code>. Using the <code>DataToolkitView</code> call optimizes query execution because it tells the GDI to inspect the query and determine which filters to push to the data source. It also limits the result set and retrieves only the data that is needed, i.e., the source data is fully mapped but all of the mapped data is not necessarily returned.</p>
<b>source_type</b>	object	<p>The <code>?data a s:source_type</code> triple pattern specifies the type of data source that the query will run against. For example, <code>?data a s:DbSource</code>, specifies that the source type is a database. The list below describes the available types:</p> <ul style="list-style-type: none"> <li>• <b>DbSource</b> to connect to any type of database.</li> <li>• <b>FileSource</b> for flat files. The supported file types are CSV and TSV, JSON, NDJSON, XML, Parquet, and SAS (SAS Transport XPT and SAS7BDAT formats). The GDI automatically determines the file type from the file extensions. When querying file sources, make sure that the files are accessible to both Anzo and AnzoGraph.</li> </ul>

Option	Data Type	Description
		<ul style="list-style-type: none"> <li>• <b>HttpSource</b> to connect to HTTP endpoints.</li> <li>• <b>ElasticSource</b> to connect to Elasticsearch indexes on an Elasticsearch server.</li> <li>• <b>KafkaSource</b> to connect to Kafka streaming sources.</li> <li>• <b>MetadataSource</b> for metadata discovery.</li> </ul> <div> <b>Tip</b>            Certain connection and input parameters are available based on the specified source type. For details about the options for your source, see <a href="#">Source-Specific Properties</a>.         </div>
url	string	<p>This property specifies the URL for the data source, such as the database URL, Elasticsearch URL, or HTTP endpoint URL. For file-based sources, the <code>url</code> property specifies the file system location of the source file or directory of files. When specifying a directory (such as <code>s:url "/opt/shared-files/loads/"</code>), the GDI loads all of the file formats it recognizes. To specify a directory but limit the number or type of files that are read, you can include the <code>pattern</code> and/or <code>maxDepth</code> properties described in <a href="#">FileSource Properties</a>.</p> <div> <b>Important</b>            For security, it is a best practice to reference connection information (such as the url, username, and password) from a Query Context so that the sensitive details are abstracted from any requests. In addition, using a Query Context makes connection details reusable across queries. See <a href="#">Using Query Contexts in Queries</a> for more information. For example, the triple patterns below reference keys from a Query Context:         </div> <pre>?data a s:DbSource ;     s:url "     { {@db.eca4bfa83481f3638b93ab5fdf93ff9a.url} } " ;</pre>



Option	Data Type	Description
		<pre>s:username " {{@db.eca4bfa83481f3638b93ab5fdf93ff9a.user}}" ; s:password " {{@db.eca4bfa83481f3638b93ab5fdf93ff9a.passwor d}}" ;</pre>
<b>username</b>	string	If authentication is required to access the source, include this property to specify the user name.
<b>password</b>	string	This property lists the password for the given username.
<b>timeout</b>	int	This property can be used to specify the timeout (in milliseconds) to use for requests against the source. For example, <code>s:timeout 5000</code> configures a 5 second timeout.
<b>batching</b>	boolean or int	This property can be used to disable batching, or it can be used to change the default the batch size. By default, batching is set to 5000 ( <code>s:batching 5000</code> ). To disable batching, you can include <code>s:batching false</code> in the query. Typically users do not change the batching size. However, it can be useful to control the batch size when performing updates. To configure the size, include <code>s:batching int</code> in the query. For example, <code>s:batching 3000</code> .
<b>paging</b>	RDF list	This property can be used to configure paging so that the GDI can access large amounts of data across a number of smaller requests. For details about the <code>paging</code> property, see <a href="#">Pagination Options</a> .
<b>concurrency</b>	int or RDF list	This property can be included to configure the maximum level of concurrency for the query. The value can be an integer, such as <code>s:concurrency 8</code> . If the value is an integer, it configures a maximum limit on the number of slices that can execute the query. For finer-grained control over the number of nodes and slices to use, concurrency can also be included as an object with <code>limit</code> ,

Option	Data Type	Description
		<p>nodes, and/or <code>executorsPerNode</code> properties. For example, the following object configures a concurrency model that allows a maximum of 24 executors distributed across 4 nodes with 8 executors per node:</p> <pre>s:concurrency [   s:limit 24 ;   s:nodes 4 ;   s:executorsPerNode 8 ; ] ;</pre>
<b>rate</b>	int or string	<p>This property can be included to control the frequency with which a request is sent to the source. The limit applies to the number of requests a single slice can make. If you specify an integer for the rate, then the value is treated as the maximum number of requests to issue per minute. If you specify a string, you have more flexibility in configuring the rate. The sample values below show the types of values that are supported:</p> <pre>s:rate "90/minute" ; s:rate "90 per minute" ; s:rate "200000 every week" ; s:rate "10000 every 6 hours" ;</pre> <p>To enforce the rate limit, the GDI introduces a sleep between requests that is equal to the rate delay. The more executing slices, the longer the rate delay needs to be to enforce the limit in aggregate.</p> <p>Given the example of <code>s:rate "90/minute"</code>, the GDI would optimize the concurrency and only use 1 slice for execution with a rate delay of 666ms between requests. If <code>s:rate "240/minute"</code>, the GDI would use 3 executors with a rate delay of 750ms between requests.</p>
<b>locale</b>	string	This property can be used to specify the locale to use when parsing locale-dependent data such as numbers, dates, and times.
<b>sampling</b>	int	This property can be used to configure the number of records in the source to

Option	Data Type	Description
		examine for data type inferencing.
<b>selector</b>	string or RDF list	This property can be used as a binding component to identify the path to the source objects. For example, <code>s:selector "Sales.SalesOrderHeader"</code> targets the SalesOrderHeader table in the Sales schema. For more information about binding components and the selector property, see <a href="#">Using Binding Trees and Selector Paths</a> .
<b>model</b>	string	This property defines the class (or table) name for the type of data that is generated from the specified data source. For example, <code>s:model "employees"</code> . Model is optional when querying a single source. If your query targets multiple sources, however, and you want to define resource templates (primary keys) and object properties (foreign keys), you must specify the model value for each source.
<b>key</b>	string	This property can be used to define the primary key column for the source file or table. This column is leveraged in a resource template for the instances that are created from the source. For example, <code>s:key ("EMPLOYEE_ID")</code> . For more information about <code>key</code> , see <a href="#">Data Linking Options</a> .
<b>reference</b>	RDF list	This property can be used to specify a foreign key column. The reference property is an RDF list that includes the <code>model</code> property to list the target table and a <code>using</code> property that defines the foreign key column. For more information about <code>reference</code> , see <a href="#">Data Linking Options</a> .
<b>formats</b>	RDF list	To give users control over the data types that are used when coercing strings to other types, this property can be included in GDI queries to define the desired types. In addition, it can be used to describe the formats of date and time values in the source to ensure that they are recognized and parsed to the appropriate date, time, and/or dateTime values. For details about the <code>formats</code> property, see <a href="#">Data Type Formatting Options</a> .

Option	Data Type	Description
<b>normalize</b>	boolean and/or RDF list	To give users control over the labels and URIs that are generated, the GDI offers several options for normalizing the model and/or the fields that are created from the specified data source(s). For details about the <code>normalize</code> property, see <a href="#">Normalization Options</a> .
<b>count</b>	variable	If you want to turn the query into a COUNT query, you can include this property with a <code>?variable</code> to perform a count. For example, <code>s:count ?count</code> .
<b>offset</b>	int	This property can be used to offset the data that is returned by a number of rows.
<b>limit</b>	int	You can include this property to limit the number of results that are returned. <code>s:limit</code> maps to the SPARQL LIMIT clause.
<b>mapping_variable</b>	variable	<p>The mapping variables, in <code>?mapping_variable (["binding"] [datatype] ["datetime_format"])</code> format, define the triple patterns to output. When the specified <code>?variable</code> matches the source column name, the GDI uses the variable as the source data selector. If you specify an alternate variable name, a <a href="#">binding</a> needs to be specified to map the new variable to the source. You also have the option to transform the data using the <a href="#">datatype</a> and <a href="#">datetime_format</a> options.</p> <div> <p><b>Note</b></p> <p>The parentheses around the binding, data type, and format specifications are not required but are included in this document for readability.</p> </div>
<b>binding</b>	string	The <code>binding</code> is a literal value that binds a <a href="#">?mapping_variable</a> to a source column. If you specify a <code>?variable</code> that matches the source column name, then that variable name is the data selector and it is not necessary to specify a

Option	Data Type	Description
		<p>binding. If you specify an alternate variable name or there is a hierarchical path to the source column, then the binding is needed to map the new variable to that source column.</p> <p>For example for a flat source like CSV, the following pattern simply binds the source column AIRLINE to the lowercase variable <code>?airline: ?airline ("AIRLINE")</code>. For a database source, this example binds the <code>?subject</code> variable by navigating to the SUBJECT column in the FILM table in the dbo schema: <code>?subject ("dbo.FILM.SUBJECT")</code>. And for an HTTP source, this example binds the <code>?time</code> variable to the time object under the minutely data path: <code>?time ("minutely.data.time")</code>.</p> <div> <p><b>Note</b></p> <p>For <b>FileSource</b> and <b>HttpSource</b>, periods (.), forward slashes (/), and brackets ([ ]) are parsed as path notation. Therefore, if a source column name includes any of those characters they must be escaped in the binding. Use two backslashes (\\) as an escape character. For example, if a column name is <b>average/day</b>, the variable and binding pattern could be written as <code>?averagePerDay ("average\\/day")</code>.</p> <p>For <b>DbSource</b>, database, schema, and table names in bindings are parsed according to the specific rules for that database type. You do not need to escape characters in database names. However, database names with characters that do not match <code>(_ A-Z a-z)(_ A-Z a-z 0-9)*</code> should be quoted, such as <code>("'Adventure.Works'.Sales.'Daily.Totals'")</code>.</p> </div>

Option	Data Type	Description
<b>datatype</b>	URI	<p>The <code>datatype</code> is the data type to convert the column to. If you do not specify a data type, the GDI infers the type. The GDI supports the following types:</p> <ul style="list-style-type: none"> <li>• <code>xsd:int</code></li> <li>• <code>xsd:long</code></li> <li>• <code>xsd:float</code></li> <li>• <code>xsd:double</code></li> <li>• <code>xsd:boolean</code></li> <li>• <code>xsd:time</code></li> <li>• <code>xsd:dateTime</code></li> <li>• <code>xsd:date</code></li> <li>• <code>xsd:gMonthDay</code></li> <li>• <code>xsd:gYearMonth</code></li> <li>• <code>xsd:duration</code></li> <li>• <code>xsd:dayTimeDuration</code></li> <li>• <code>xsd:yearMonthDuration</code></li> <li>• <code>xsd:gMonth</code></li> <li>• <code>xsd:anyURI</code></li> </ul>
<b>datetime_format</b>	string	<p>This option is used to specify the format to use for date and time data types. The GDI supports Java date and time formats. Specify days as "d," months as "M," and years as "y." For the time, specify "H" for hours, "m" for minutes, and "s" for seconds. For example, "<code>yyyyMMdd HH:mm:ss</code>" or "<code>ddMMMyy</code>" to display date values such as "01JAN19."</p> <div> <p><b>Note</b></p> <p>The GDI's default base year is <b>2000</b>. If the source data has years with only two digits, such as <code>02-04-99</code>, the GDI prepends <b>20</b> to the digits. The value <code>02-04-99</code> is parsed to <code>02-04-2099</code>. To specify an alternate base year to use for two-digit values, you can include the notation <code>^nnnn</code> (e.g., <code>^1900</code>) in the format value. For example, to set the base year to 1900 instead of 2000, use a format value such as <code>xsd:date "dd-MMM-yy^1900"</code> or <code>xsd:date "dd-MMM-yy^1990"</code>. When one of those values is specified, <code>02-04-99</code> is parsed to <code>02-04-1999</code>.</p> </div>

## GDI Query Examples

The query below reads data from a sample HTTP source that compiles worldwide weather statistics. The source has several models available for retrieving data that is current, daily, historical, etc. To target current data, the query includes `s:selector "currently"` as an input parameter. In addition, the query

demonstrates the use of the "topdown" functionality, where the query sends values to the source to narrow the results. The query includes the TOPDOWN keyword in the GDI service call, and the VALUES clause specifies the latitude and longitude values for the cities to return data for. In addition, since this sample source requires parameters to be specified in the connection URL, the `s:url` value includes `?lat` and `?long` as parameters as part of the value.

```
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>
PREFIX ex:     <http://example.org/ontologies/City#>

SELECT
    ?city ?state ?summary ?temp ?rainChance
    ?humidity ?pressure ?windSpeed
WHERE
{
    SERVICE TOPDOWN <http://cambridgesemantics.com/services/DataToolkit>
    {
        ?data a s:HttpSource ;
        s:url "https://sampleEndpoint.com/forecast/{{?lat}},{{?long}}" ;
        s:selector "currently" ;
        ?lat ("latitude") ;
        ?long ("longitude") ;
        ?temp ("temperature") ;
        ?rainChance ( "precipProbability" ) ;
        ?humidity () ;
        ?pressure () ;
        ?windSpeed () .
    }
    VALUES( ?city ?state ?lat ?long )
    {
        ( "Lakeway" "TX" 30.374563 -97.975892 )
        ( "Boston" "MA" 42.358043 -71.060415 )
        ( "Seattle" "WA" 47.590720 -122.307053 )
    }
}
```

```

        ( "Chicago" "IL" 41.837741 -87.823296 )
        ( "Hilo" "HI" 19.702040 -155.090312 )
    }
}
ORDER BY ?city

```

The query returns the following results:

city	state	summary	temp	rainChance	humidity	pressure	windSpeed
Boston	MA	Overcast	79.81	0	0.6	1018.7	7.71
Chicago	IL	Clear	81.7	0	0.52	1021.1	5.13
Hilo	HI	Partly Cloudy	72.6	0.13	0.79	1018.6	4.86
Lakeway	TX	Partly Cloudy	92.43	0	0.48	1013.3	10.85
Seattle	WA	Mostly Cloudy	61.82	0	0.76	1018.2	4.57

5 rows

The example below ingests data into a data layer from a database source using an INSERT query in a Direct Load Step.

```

PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>
PREFIX :      <http://example.com/ontologies/kl_hosp#>

INSERT
{

```



```

GRAPH ${targetGraph}
{
  ?InputEvent_cv a :InputEvent_cv ;
    :row_id ?row_id ;
    :subject_id ?subject_id ;
    :hadm_id ?hadm_id ;
    :icustay_id ?icustay_id ;
    :charttime ?charttime ;
    :itemid ?itemid ;
    :amount ?amount ;
    :amountuom ?amountuom ;
    :rate ?rate ;
    :rateuom ?rateuom ;
    :storetime ?storetime ;
    :cgid ?cgid ;
    :orderid ?orderid ;
    :linkorderid ?linkorderid ;
    :stopped ?stopped ;
    :newbottle ?newbottle ;
    :originalamount ?originalamount ;
    :originalamountuom ?originalamountuom ;
    :originalroute ?originalroute ;
    :originalrate ?originalrate ;
    :originalrateuom ?originalrateuom ;
    :originalsite ?originalsite .
}
}
WHERE
{
  SERVICE <http://cambridgesemantics.com/services/DataToolkit>
  {
    ?data a s:DbSource ;
      s:url "{{@db.eca4bfa83481f3638b93ab5fdf93ff9a.url}}" ;
      s:username "{{@db.eca4bfa83481f3638b93ab5fdf93ff9a.user}}"
      s:password "{{@db.eca4bfa83481f3638b93ab5fdf93ff9a.password}}"
      s:selector "kl_hosp_schema.inputevents_cv" ;
      ?row_id (xsd:int) ;
      ?subject_id (xsd:int) ;
      ?hadm_id (xsd:int) ;

```

```

        ?icustay_id (xsd:int) ;
        ?charttime (xsd:dateTime) ;
        ?itemid (xsd:int) ;
        ?amount (xsd:float) ;
        ?amountuom (xsd:string) ;
        ?rate (xsd:float) ;
        ?rateuom (xsd:string) ;
        ?storetime (xsd:dateTime) ;
        ?cgid (xsd:int) ;
        ?orderid (xsd:int) ;
        ?linkorderid (xsd:int) ;
        ?stopped (xsd:string) ;
        ?newbottle (xsd:int) ;
        ?originalamount (xsd:float) ;
        ?originalamountuom (xsd:string) ;
        ?originalroute (xsd:string) ;
        ?originalrate (xsd:float) ;
        ?originalrateuom (xsd:string) ;
        ?originalsite (xsd:string) ;
    BIND(URI("http://example.com/inputevent_cv/{?row_id}") AS ?InputEvent_cv)
    BIND(URI("http://example.com/patients/{?subject_id}") AS ?patient)
    BIND(URI("http://example.com/admissions/{?hadm_id}") AS ?admission)
}
}

```

The following query ingests airport-related data from a CSV file.

```

PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>

INSERT
{
    GRAPH ${targetGraph}

```

```

{
  ?code a <http://anzograph.com/airport> ;
    <http://anzograph.com/airport/name> ?name ;
    <http://anzograph.com/airport/city> ?city ;
    <http://anzograph.com/airport/state> ?state ;
    <http://anzograph.com/airport/latitude> ?lat;
    <http://anzograph.com/airport/longitude> ?long.
}
}
WHERE
{
  SERVICE <http://cambridgesemantics.com/services/DataToolkit>
  {
    ?data a s:FileSource ;
      s:url "/opt/shared-files/airports.csv" ;
      ?iata_code ("IATA_CODE" xsd:string) ;
      ?name ("AIRPORT" xsd:string) ;
      ?city ("CITY" xsd:string) ;
      ?state ("STATE" xsd:string) ;
      ?lat ("LATITUDE" xsd:double) ;
      ?long ("LONGITUDE" xsd:double).
    BIND(IRI("http://anzograph.com/airport/{{?iata_code}}") as ?code)
  }
}

```

The query below creates a view of a database source.

```

PREFIX s:    <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX anzo: <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl: <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:   <http://purl.org/dc/elements/1.1/>
PREFIX ont:  <http://cambridgesemantics.com/ont/autogen/Rh/MIMIC-III-Data_
Source/mimic_iii_schema#>

CONSTRUCT

```

```

{
  ?caregiversURI a ont:caregivers ;
    ont:caregivers_cgid ?cgid ;
    ont:caregivers_description ?description ;
    ont:caregivers_label ?label .
}
WHERE
{
  GRAPH ?g
  {
    SERVICE <http://cambridgesemantics.com/services/DataToolkitView>
    (${targetGraph}) {
      {
        ?data a s:DbSource ;
          s:url "{{@db.eca4bfa83481f3638b93ab5fdf93dd9a.url}}" ;
          s:username "{{@db.eca4bfa83481f3638b93ab5fdf93dd9a.user}}"
          s:password "{{@db.eca4bfa83481f3638b93ab5fdf93dd9a.password}}"
          s:selector "mimic_iii_schema.caregivers" ;
          ?row_id (xsd:int) ;
          ?cgid (xsd:int) ;
          ?label (xsd:string) ;
          ?description (xsd:string) .
          BIND(IRI("http://cambridgesemantics.com/class/caregivers/{{?row_id}}") AS
?caregiversURI) }
      }
    }
  }
}

```

## Related Topics

[General GDI Concepts and Usage](#)

[Generating a Graph and Ontology with a Direct Load Step](#)

## Generating a Graph and Ontology with a Direct Load Step

With no mapping required, Anzo's Direct Load Step functionality automatically generates a graph and an ontology (model) for a data source. Using a relatively simple SPARQL query, the direct load option invokes the Graph Data Interface (GDI) RDF and Ontology Generators. The GDI Generators recognize the structure of a data source and automatically generate the necessary statements.

Invoking the Generators is preferable to producing a hand-written query, especially when the structure of the data is very complex, such as a JSON data source with many inner repeating structures or a database with many tables and keys. When the source contains complex structures, the GDI will generate only the required statements and avoid cross-products, optimizing query execution and memory usage. In addition, the GDI Generator parallelizes the load across the AnzoGraph cluster so that a data source (such as a database) can be ingested with a single query.

This topic provides details about invoking the GDI RDF and Ontology Generators. The Generators can be used with all of the supported data source types.

- [How to Use the GDI Generator in a Graphmart](#)
- [GDI Generator Query Syntax](#)
- [GDI Generator Example Queries](#)

## How to Use the GDI Generator in a Graphmart

To invoke the GDI Generator in a data layer, you add a **Direct Load Step** to the layer. In the Direct Load Step, you compose a SPARQL query that incorporates the GDI Generator parameters as detailed below in [GDI Generator Query Syntax](#).

### Tip

For instructions on adding steps to layers, see [Adding Steps to Data Layers](#).

## Why Use a Direct Load Step?

It is important to use a Direct Load Step with the RDF and Ontology Generators because it is the only type of step with the ability to manage the generated ontologies (models). An ontology that is generated in a Direct Load Step is automatically registered in Anzo. The registered model is linked to and managed by the data layer that contains the step. If an Ontology Generator query is changed, additional Direct Load Steps are added to the same layer, or the underlying source schema changes, the managed model is automatically updated when the graphmart is reloaded or refreshed. See [Managed Model Details](#) below for important details about layer-managed models.

## Managed Model Details

Though an ontology that is generated in a Direct Load Step is registered in Anzo and is available for viewing in the Model editor, **the model is owned and managed by the data layer that contains the Direct Load Step**. That means any manual changes made to the model outside of the step, such as from the Model

editor, will be overwritten any time the graphmart or layer is refreshed or reloaded. **Do not modify generated managed models except by editing (or adding) Direct Load Step queries.**

**There is only one managed model per layer.** If you include multiple Direct Load Steps in the same layer, they will all update the same ontology. This functionality can be useful if you want to align the data and generated model across multiple steps. If you have multiple sources that are not intended to align or update the same model, create separate layers.

**If you delete a layer that includes a managed model, the model is also deleted.** Use caution when referencing a managed model outside of a graphmart. For example, if you create a dataset and reference a managed model when you select the ontology, the reference will break if the data layer that manages the model is deleted.

## GDI Generator Query Syntax

The following query syntax shows the structure of a GDI Generator query. The clauses, patterns, and placeholders that are links are described below.

```
# PREFIX Clause
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>

#Result Clause
INSERT {
  GRAPH ${targetGraph} {
    ?s ?p ?o .
  }
}
WHERE
{
  SERVICE <http://cambridgesemantics.com/services/DataToolkit>
  {
    ?data a s:source_type ;
# Based on the source_type, additional connection and input parameters are
# available. The options below are valid for all sources. For source-related
# options, see Source-Specific Properties in the GDI property reference.
    s:url "string" ;
```

```

[ s:model "class_name_for_this_source" ; ]
[ s:username "string" ; ]
[ s:password "string" ; ]
[ s:timeout int ; ]
[ s:maxConnections int ; ]
[ s:batching boolean | int ; ]
[ s:paging [ pagination_options ; ]
[ s:concurrency int | [ list_of_properties ] ; ]
[ s:rate int | "string" ; ]
[ s:locale "string" ; ]
[ s:sampling int ; ]
[ s:selector "string" | [ list ] ; ]
[ s:key ("string") ; ]
[ s:reference [ s:model "string" ; s:using ("string") ]
[ s:formats [ datatype_formatting_options ] ; ]
[ s:normalize boolean | [ source_normalization_rules ] ; ]
[ s:count ?variable ; ]
[ s:offset int ; ]
[ s:limit int ; ] .

```

# Multiple data sources can be merged if they project a similar set  
# of output variables. Make sure each source has a unique subject  
variable.

```

[ ?unique_variable a s:source_type ;
...
. ]

```

```

?rdf a s:RdfGenerator, s:OntologyGenerator ;
s:as (?s ?p ?o);
s:ontology ontology_uri ;
s:base base_uri ;
[ s:normalize boolean | [ global_normalization_rules ] ; ]
.

```

# Additional clauses such as BIND, FILTER

```

}
}

```

### Note

For readability, the parameters below exclude the base URI

`<http://cambridgesemantics.com/ontologies/DataToolkit#>` as well as the `s :` prefix. As shown in the examples, however, the `s :` prefix or full property URI does need to be included in queries.

Option	Data Type	Description
<b>PREFIX Clause</b>	N/A	<p>The PREFIX clause declares the standard and custom prefixes for GDI Generator queries. Generally, queries include the following prefixes (or a subset of them) plus any data-specific declarations:</p> <pre>PREFIX s: &lt;http://cambridgesemantics.com/ontologies/DataToolkit#&gt; PREFIX owl: &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt;</pre>
<b>Result Clause</b>	N/A	<p>The result clause for Direct Load Steps is typically an INSERT query with the following graph pattern:</p> <pre>INSERT {   GRAPH \${targetGraph} {     ?s ?p ?o .   } }</pre> <p><b>Note</b></p> <p>It is important to include the <b>GRAPH</b> keyword and target graph parameter <b><code>\${targetGraph}</code></b> when you are writing an INSERT query. Anzo automatically replaces the <code>\$(targetGraph)</code> parameter with the appropriate target URI(s) when the query runs.</p>



Option	Data Type	Description
<b>source_type</b>	object	<p>The <code>?data a s:source_type</code> triple pattern specifies the type of data source that the query will run against. For example, <code>?data a s:DbSource</code>, specifies that the source type is a database. The list below describes the available types:</p> <ul style="list-style-type: none"> <li>• <b>DbSource</b> to connect to any type of database.</li> <li>• <b>FileSource</b> for flat files. The supported file types are CSV and TSV, JSON, NDJSON, XML, Parquet, and SAS (SAS Transport XPT and SAS7BDAT formats) . The GDI automatically determines the file type from the file extensions. When querying file sources, make sure that the files are accessible to both Anzo and AnzoGraph.</li> <li>• <b>HttpSource</b> to connect to HTTP endpoints.</li> <li>• <b>ElasticSource</b> to connect to Elasticsearch indexes on an Elasticsearch server.</li> <li>• <b>KafkaSource</b> to connect to Kafka streaming sources.</li> </ul> <div> <p><b>Tip</b></p> <p>Certain connection and input parameters are available based on the specified source type. For details about the options for your source, see <a href="#">Source-Specific Properties</a>.</p> </div>
<b>url</b>	string	<p>This property specifies the URL for the data source, such as the database URL, Elasticsearch URL, or HTTP endpoint URL. For file-based sources, the <code>url</code> property specifies the file system location of the source file or directory of files. When specifying a directory (such as <code>s:url "/opt/shared-files/loads/"</code>), the GDI loads all of the file formats it recognizes. To specify a directory but limit the number or type of files that are read, you can include the <code>pattern</code> and/or <code>maxDepth</code> properties described in <a href="#">FileSource Properties</a>.</p> <div> <p><b>Important</b></p> <p>For security, it is a best practice to reference connection</p> </div>

Option	Data Type	Description
		<p>information (such as the url, username, and password) from a Query Context so that the sensitive details are abstracted from any requests. In addition, using a Query Context makes connection details reusable across queries. See <a href="#">Using Query Contexts in Queries</a> for more information. For example, the triple patterns below reference keys from a Query Context:</p> <pre>?data a s:DbSource ;     s:url "     {{@db.eca4bfa83481f3638b93ab5fdf93ff9a.url}}" ;     s:username "     {{@db.eca4bfa83481f3638b93ab5fdf93ff9a.user}}"     ;     s:password "     {{@db.eca4bfa83481f3638b93ab5fdf93ff9a.passwor     d}}" ;</pre>
<b>model</b>	string	This property defines the class (or table) name for the type of data that is generated from the specified data source. For example, <code>s:model "employees"</code> . Model is optional when querying a single source. If your query targets multiple sources, however, and you want to define resource templates (primary keys) and object properties (foreign keys), you must specify the model value for each source.
<b>username</b>	string	If authentication is required to access the source, include this property to specify the user name.
<b>password</b>	string	This property lists the password for the given username.
<b>timeout</b>	int	This property can be used to specify the timeout (in milliseconds) to use for requests against the source. For example, <code>s:timeout 5000</code> configures a 5 second timeout.

Option	Data Type	Description
<b>maxConnections</b>	int	For database sources, this property can be used to set a limit on the maximum number of active connections to the source. For example, <code>s:maxConnections 16</code> sets the limit to 16 connections. The default value is 10.
<b>batching</b>	boolean or int	This property can be used to disable batching, or it can be used to change the default the batch size. By default, batching is set to 5000 ( <code>s:batching 5000</code> ). To disable batching, you can include <code>s:batching false</code> in the query. Typically users do not change the batching size. However, it can be useful to control the batch size when performing updates. To configure the size, include <code>s:batching int</code> in the query. For example, <code>s:batching 3000</code> .
<b>paging</b>	RDF list	This property can be used to configure paging so that the GDI can access large amounts of data across a number of smaller requests. For details about the <code>paging</code> property, see <a href="#">Pagination Options</a> .
<b>concurrency</b>	int or RDF list	<p>This property can be included to configure the maximum level of concurrency for the query. The value can be an integer, such as <code>s:concurrency 8</code>. If the value is an integer, it configures a maximum limit on the number of slices that can execute the query. For finer-grained control over the number of nodes and slices to use, concurrency can also be included as an object with <code>limit</code>, <code>nodes</code>, and/or <code>executorsPerNode</code> properties. For example, the following object configures a concurrency model that allows a maximum of 24 executors distributed across 4 nodes with 8 executors per node:</p> <pre>s:concurrency [   s:limit 24 ;   s:nodes 4 ;   s:executorsPerNode 8 ; ] ;</pre>

Option	Data Type	Description
<b>rate</b>	int or string	<p>This property can be included to control the frequency with which a request is sent to the source. The limit applies to the number of requests a single slice can make. If you specify an integer for the rate, then the value is treated as the maximum number of requests to issue per minute. If you specify a string, you have more flexibility in configuring the rate. The sample values below show the types of values that are supported:</p> <pre>s:rate "90/minute" ; s:rate "90 per minute" ; s:rate "200000 every week" ; s:rate "10000 every 6 hours" ;</pre> <p>To enforce the rate limit, the GDI introduces a sleep between requests that is equal to the rate delay. The more executing slices, the longer the rate delay needs to be to enforce the limit in aggregate.</p> <p>Given the example of <code>s:rate "90/minute"</code>, the GDI would optimize the concurrency and only use 1 slice for execution with a rate delay of 666ms between requests. If <code>s:rate "240/minute"</code>, the GDI would use 3 executors with a rate delay of 750ms between requests.</p>
<b>locale</b>	string	This property can be used to specify the locale to use when parsing locale-dependent data such as numbers, dates, and times.
<b>sampling</b>	int	This property can be used to configure the number of records in the source to examine for data type inferencing.
<b>selector</b>	string or RDF list	This property can be used as a binding component to identify the path to the source objects. For example, <code>s:selector "Sales.SalesOrderHeader"</code> targets the SalesOrderHeader table in the Sales schema. For more information about binding components and the selector property, see <a href="#">Using Binding Trees and Selector Paths</a> .
<b>key</b>	string	This property can be used to define the primary key column for the source

Option	Data Type	Description
		file or table. This column is leveraged in a resource template for the instances that are created from the source. For example, <code>s:key ("EMPLOYEE_ID")</code> . For more information about <code>key</code> , see <a href="#">Data Linking Options</a> .
<b>reference</b>	RDF list	This property can be used to specify a foreign key column. The reference property is an RDF list that includes the <code>model</code> property to list the target table and a <code>using</code> property that defines the foreign key column. For more information about <code>reference</code> , see <a href="#">Data Linking Options</a> .
<b>formats</b>	RDF list	To give users control over the data types that are used when coercing strings to other types, this property can be included in GDI queries to define the desired types. In addition, it can be used to describe the formats of date and time values in the source to ensure that they are recognized and parsed to the appropriate date, time, and/or dateTime values. For details about the <code>formats</code> property, see <a href="#">Data Type Formatting Options</a> .
<b>normalize</b>	RDF list	To give users control over the labels and URIs that are generated, the GDI offers several options for normalizing the model and/or the fields that are created from the specified data source(s). For details about the <code>normalize</code> property, see <a href="#">Normalization Options</a> .
<b>count</b>	variable	If you want to turn the query into a COUNT query, you can include this property with a <code>?variable</code> to perform a count. For example, <code>s:count ?count</code> .
<b>offset</b>	int	This property can be used to offset the data that is returned by a number of rows.
<b>limit</b>	int	You can include this property to limit the number of results that are returned. <code>s:limit</code> maps to the SPARQL LIMIT clause.
<b>RdfGenerator</b>	object	Include this property to invoke the RDF Generator. If you only want to

Option	Data Type	Description
		generate a model without RDF, you can exclude <code>RdfGenerator</code> .
<b>OntologyGenerator</b>	object	Include this property to invoke the Ontology Generator. If you only want to generate RDF without a model, you can exclude <code>OntologyGenerator</code> .
<b>as</b>	N/A	This property provides the variable bindings for the RDF Generator's projection to RDF. Typically the value is <code>s:as (?s ?p ?o)</code> to match the variables in the result clause.
<b>ontology</b>	URI	<p>This property specifies the URI to use as the base URI for any generated ontology artifacts. For example, <code>s:ontology &lt;http://abc.com/ontologies/MyOntology&gt;</code>.</p> <div> <p><b>Note</b></p> <p>In the graphmart, the data layer ID is appended to the ontology URI that is generated. The complete URI is based on the layer and cannot be customized.</p> </div>
<b>base</b>	URI	This property specifies the base URI for instance data. The base value should NOT end in <code>#</code> . The Generator will add a trailing slash ( <code>/</code> ) if one does not exist. For example, <code>s:base &lt;http://abc.com/&gt;</code> .

## GDJ Generator Example Queries

This section includes sample queries that may be useful as a starting point for writing your own RDF and Ontology Generator queries.

- [Basic Query that Generates RDF and Ontology for a JSON File](#)
- [Basic Query that Generates an Ontology for a Directory of CSV Files](#)
- [Query that Normalizes and Generates RDF and Ontology for a Database](#)
- [Query with Query Context that Normalizes and Generates RDF and Ontology for a Database](#)
- [Query for Multiple Sources that Generates RDF and Ontology with Resource Templates and Object Properties](#)

## Basic Query that Generates RDF and Ontology for a JSON File

```
PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>

INSERT {
  GRAPH ${targetGraph} {
    ?s ?p ?o .
  }
}

WHERE {
  SERVICE <http://cambridgesemantics.com/services/DataToolkit> {

    ?data a s:FileSource ;
      s:model "People" ;
      s:url "/opt/shared-files/json/people.json" .

    ?rdf a s:RdfGenerator , s:OntologyGenerator ;
      s:as (?s ?p ?o) ;
      s:ontology <http://cambridgesemantics.com/ontologies/People> ;
      s:base <http://cambridgesemantics.com/data/> .

  }
}
```

## Basic Query that Generates an Ontology for a Directory of CSV Files

```
PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>

INSERT {
  GRAPH ${targetGraph} {
    ?s ?p ?o .
  }
}

WHERE {
  SERVICE <http://cambridgesemantics.com/services/DataToolkit> {

    ?data a s:FileSource ;
      s:model "Sales" ;
```

```

s:url "/opt/shared-files/csv/sales" ;
s:format [
  s:delimiter "," ;
  s:headers true ;
  s:comment "#" ;
  s:quote "\"" ;
  s:maxColumns 22 ;
] .

?rdf a s:OntologyGenerator ;
s:as (?s ?p ?o) ;
s:ontology <http://cambridgesemantics.com/ontologies/Sales> ;
s:base <http://cambridgesemantics.com/data/> .
}
}

```

## Query that Normalizes and Generates RDF and Ontology for a Database

```

PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>

INSERT {
  GRAPH ${targetGraph} {
    ?s ?p ?o .
  }
}
WHERE {
  SERVICE <http://cambridgesemantics.com/services/DataToolkit> {

    ?data a s:DbSource ;
    s:url "jdbc:mysql://10.11.12.9/emrdbbig" ;
    s:username "root" ;
    s:password "sql1@#" ;
    s:normalize [
      s:model [
        s:removeStart "emr_" ;
        s:words "activity 'patient complaint' medication observation patient
specialty study" ;
      ] ;
    ] ;
  }
}

```



```

        s:field [
            s:removePartialPrefix true ;
            s:words "provider description start end drug complaint date
medication normal code
                        observation product active dose generic route admin
strength collection
                        activity home first last status first year birth death
directed complex
                        period age flag gender language" ;
        ] ;
    ] .

?rdf a s:RdfGenerator , s:OntologyGenerator ;
s:as (?s ?p ?o) ;
s:ontology <http://cambridgesemantics.com/ontologies/EMR> ;
s:base <http://cambridgesemantics.com/EMR> .
}
}

```

## Query with Query Context that Normalizes and Generates RDF and Ontology for a Database

The query below references a Query Context to supply the username and password for the database connection.

```

PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>

INSERT {
    GRAPH ${targetGraph} {
        ?s ?p ?o .
    }
}
WHERE {
    SERVICE <http://cambridgesemantics.com/services/DataToolkit> {

        ?data a s:DbSource ;
        s:url "jdbc:sqlserver://localhost;databaseName=AdventureWorks2012" ;
        s:username "{{@db.username}}" ;
    }
}

```

```

        s:password "{{@db.password}}" ;
        s:schema "Production", "HumanResources", "Person", "Sales",
"Purchasing" ;
        s:normalize [
            s:model [
                s:localNamePrefix "C_" ;
                s:localNameSeparator "_" ;
                s:match [ s:pattern "(.+)+Enlarged" ; s:replace "$1" ] ;
            ] ;
            s:field [
                s:localNamePrefix "P_" ;
                s:localNameSeparator "_" ;
                s:ignore "rowguid ModifiedDate" ;
                s:match (
                    [ s:pattern "(.+)+GUID$" ; s:replace "$1" ]
                    [ s:pattern "(.+)+ID$" ; s:replace "$1" ]
                ) ;
            ] ;
        ] .

?rdf a s:RdfGenerator, s:OntologyGenerator ;
s:as (?s ?p ?o) ;
s:ontology <http://cambridgesemantics.com/ontologies/AdventureWorks> ;
s:base <http://cambridgesemantics.com/AdventureWorks> .

}

}

```

## Query for Multiple Sources that Generates RDF and Ontology with Resource Templates and Object Properties

This query also includes global normalization rules for normalizing the data across all Data Sources.

```

PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>

INSERT {
    GRAPH ${targetGraph} {
        ?s ?p ?o .
    }
}

```

```

}
WHERE {
    SERVICE <http://cambridgesemantics.com/services/DataToolkit> {

        ?event a s:FileSource ;
            s:model "event" ;
            s:url   "/opt/shared-files/csv/events.csv" ;
            s:key   ("EVENT_ID") .

        ?listing a s:FileSource ;
            s:model "listing" ;
            s:url   " /opt/shared-files/csv/listings.csv" ;
            s:key   ("LIST_ID") ;
            s:reference [ s:model "event" ; s:using ("EVENT_ID") ; s:key ("EVENT_
ID") ] .

        ?date a s:FileSource ;
            s:model "date" ;
            s:url   "/opt/shared-files/csv/event_dates.csv" ;
            s:key   ("DATE_ID") ;
            s:reference [ s:model "event" ; s:using ("EVENT_ID") ; s:key ("EVENT_
ID") ] .

        ?venue a s:FileSource ;
            s:model "venue" ;
            s:url   " /opt/shared-files/csv/venues.csv" ;
            s:key   ("VENUE_ID") ;
            s:reference [ s:model "event" ; s:using ("EVENT_ID") ; s:key ("EVENT_
ID") ] .

        ?sale a s:FileSource ;
            s:model "sale" ;
            s:url   " /opt/shared-files/csv/sales.csv" ;
            s:key   ("SALE_ID") ;
            s:reference [ s:model "event" ; s:using ("EVENT_ID") ; s:key ("EVENT_
ID") ] ;
            s:reference [ s:model "listing" ; s:using ("LIST_ID") ; s:key ("LIST_
ID") ] .
    }
}

```

```

?rdf a s:RdfGenerator, s:OntologyGenerator ;
  s:as (?s ?p ?o) ;
  s:ontology <http://cambridgesemantics.com/tickets> ;
  s:base <http://cambridgesemantics.com/data> ;
  s:normalize [
    s:all [
      s:casing s:UPPER ;
      s:localNameSeparator "_" ;
    ] ;
  ] .
}

```

## Related Topics

[Directly Loading a Data Source \(Direct Load Step\)](#)

## Data Type Formatting Options

To give you control over the data types that are used when coercing strings to other types, the **formats** property can be included in GDI queries to define the desired types. In addition, formats can be used to describe the formats of date and time values in the source to ensure that they are recognized and parsed to the appropriate date, time, and/or dateTime values. You can also use the formats property to suppress the conversion so that the generated values are typed the same way as the source.

### Tip

The GDI takes locale into account when formatting the generated date and time values.

For sources that do not include data type specifications and natively treat values as strings, the GDI Generator automatically converts the values to the appropriate type. For example, if a CSV file includes the value "Feb-18-2022," the GDI parses the string to an `xsd:date` with the format "2022-02-18". A column with numbers is converted to an `xsd:int` type and a column with a decimal value is converted to `xsd:float`. The formats property usage is described below.

- [Formats Syntax](#)
- [Formats Examples](#)

Formats Syntax

```
s:formats [  
  s:strict boolean ; [  
    xsd:data_type "format"  
  | xsd:data_type boolean ;  
    [ ... ; ]  
  ]  
] ;
```

Option	Data Type	Description
strict	boolean	<p>This property enables or disables the automatic type conversion feature. By default, strict is set to false (<code>s:strict false</code>), meaning the GDI's automatic type conversion feature is enabled. When strict is false, any formats specified in <code>s:formats [ ]</code> augment the GDI's built-in date and time formats. You can selectively disable certain type conversions, however, by including <code>xsd:data_type false</code>. For example, <code>xsd:dateTime false</code> disables the parsing of strings to <code>dateTime</code>.</p> <p>When strict is true (<code>s:strict true</code>), the auto conversion logic is essentially disabled and the generated data will be represented the same way it is in the source. When strict is true, you can selectively enable certain conversions by including <code>xsd:data_type true</code> or by defining <code>xsd:data_type "format"</code>. In this case, values that do not match any of the formats provided will be typed as <code>xsd:string</code>.</p>
xsd:data_type "format"	N/A	<p>Include <code>xsd:data_type "format"</code> when you want to describe the formats of date and time values in the source. The GDI supports Java date and time format notation. For example, if dates in the source are formatted like "yyyy-MM-dd," include the statement <code>xsd:date "yyyy-MM-dd"</code>. If the source uses multiple formats for dates, e.g., 18-MAR-1978 and 03/18/1978, you can list multiple formats for <code>xsd:date</code>, such as <code>xsd:date "dd-MMM-yyyy", "MM/dd/yyyy"</code>.</p> <div><p><b>Note</b></p><p>The GDI's default base year is <b>2000</b>. If the source data has years</p></div>

Option	Data Type	Description
		<p>with only two digits, such as 02-04-99, the GDI prepends <b>20</b> to the digits. The value 02-04-99 is parsed to 02-04-2099. To specify an alternate base year to use for two-digit values, you can include the notation <code>^nnnn</code> (e.g., <code>^1900</code>) in the format value. For example, to set the base year to 1900 instead of 2000, use a format value such as <code>xsd:date "dd-MMM-yy^1900"</code> or <code>xsd:date "dd-MMM-yy^1990"</code>. When one of those values is specified, 02-04-99 is parsed to 02-04-1999.</p>
<b>xsd:data_</b> <b>type boolean</b>	N/A	<p>When <code>strict</code> is <code>false</code> or not set, you can disable specific type conversions by listing data types and setting their values to <code>false</code>. For example, if you want the GDI to convert strings to integers or floats when possible but you want the dates in the source to be preserved as strings, you can include <code>xsd:date false</code> to disable the conversion of strings to dates.</p> <p>When <code>strict</code> is <code>true</code>, you can enable specific type conversions by listing data types and setting their values to <code>true</code>. For example, if you want the GDI to preserve the strings in the source except for when the string is a number, you can include <code>xsd:int true</code> to enable the conversion of strings to integers.</p>

## Formats Examples

The example below sets `strict` to `true` and forces the GDI to parse values only to the data types that are enabled with `true`. It also defines the format to look for when converting strings to `dateTime`:

```
s:formats [
  s:strict true ;
  xsd:int true ;
  xsd:dateTime true ;
  xsd:dateTime "yyyy-MM-dd-HH-mm-ss" ;
] ;
```

The example below does not set `strict`, so the default value of `false` is used. The data type definitions specify the formats of the values to parse as date, time, and dateTime values. The example also disables the conversion from string to long:

```
s:formats [  
  xsd:date "MM/dd/yyyy", "MMM dd", "MMM dd yyyy" ;  
  xsd:time "HH[:mm][:ss][ ]a" ;  
  xsd:dateTime "M/d/yyyy HH:mm:ss a", "yyyy-MM-dd-HH-mm-ss" ;  
  xsd:long false ;  
] ;
```

## Related Topics

[Generating a Graph and Ontology with a Direct Load Step](#)

## Normalization Options

To give users control over the labels and URIs that are generated, the GDI offers several options for normalizing the model and/or the fields that are created from the specified data source(s). Normalization rules can be specified at the source level to normalize the data from each source independently, or they can be used at the RDF Generator level to apply global rules across all specified data sources.

### Tip

You can also specify normalization rules at both the source and global level in the same query. If you include multi-valued rules (such as `ignore`, `words`, or `preserve`) at both levels, the Generator combines all values from both instances of the rule. If you specify single value rules at both levels and the values are conflicting, the Generator applies the value at the source level.

Including the **normalize** parameter is optional. If you include it, you can specify any combination of rules. See [Default Normalization Behavior](#) below for details about the Generator's default behavior when normalization rules are not specified in your query.

- [Default Normalization Behavior](#)
- [Normalize Syntax](#)
- [Normalize Examples](#)

## Default Normalization Behavior

The GDI Generator normalizes data according to the following rules by default. If you do not include the `s:normalize` parameter in your query, these are the rules that are applied:

```

s:normalize [
  s:all [
    s:removePrefix true ;
    s:removePartialPrefix false ;
    s:allowWhiteSpace false ;
    s:allowPunctuation false ;
    s:allowSymbols false ;
    s:separator " " ;
    s:singularize false ;
    s:casing s:UpperCamel ;
    s:localNameSeparator "." ;
  ]
]

```

## Normalize Syntax

```

s:normalize boolean | [
  s:model | s:field | s:all
  [
    s:removeStart "string" ;
    s:removeEnd "string" ;
    s:removePrefix boolean ;
    s:removePartialPrefix boolean ;
    s:match [ s:pattern "regex" ; s:replace "regex" ] ;
    s:disambiguationLevel int ;
    s:ignore "string" ;
    s:words "string" ;
    s:preserve "string" ;
    s:split "string" ;
    s:allowWhiteSpace boolean ;
    s:allowPunctuation boolean ;
    s:allowSymbols boolean ;
    s:singularize boolean ;
    s:casing property ;
    s:separator "string" ;
    s:localNamePrefix "string" ;
    s:localNameSeparator "string" ;
  ] ;
] ;

```



Property	Data Type	Description
<b>boolean</b>	N/A	Normalize is enabled by default for all GDI Generator queries. If you want to disable normalization, you can include <code>s:normalize false</code> . If normalization is disabled, the names in the source will be used verbatim both for labeling and in generating the local names for property and class URIs. However, when normalization is disabled, the labels in the data source are used verbatim. In addition, the Generator creates hard-to-read, URL-encoded local names for property and class URIs.
<b>s:model   s:field   s:all</b>	N/A	This property defines whether the specified normalization rules should be applied to the model only ( <code>s:model</code> ), field identifiers only ( <code>s:fields</code> ), or both the model and field identifiers ( <code>s:all</code> ). If not specified, the default behavior is to apply the rules to the model and fields.
<b>removeStart</b>	string	If you want to remove text from the beginning of identifiers, include the <b>removeStart</b> rule to specify the string to remove. For example, <code>s:removeStart "temp_"</code> .
<b>removeEnd</b>	string	If you want to remove text from the end of identifiers, include the <b>removeEnd</b> rule to specify the string to remove. For example, <code>s:removeEnd "NEW"</code> .
<b>removePrefix</b>	boolean	If there are property identifiers that share a prefix with the class, the RDF Generator automatically removes the shared prefix from the property name; the <b>removePrefix</b> rule is set to <code>true</code> by default. For example, if there is an <code>EMPLOYEE</code> class with an <code>EMPLOYEE_ID</code> column, the shared prefix "EMPLOYEE" is removed from the generated property so that it becomes "ID." If you do not want the Generator to remove prefixes, you can include <code>s:removePrefix false</code> .
<b>removePartialPrefix</b>	boolean	If there are property identifiers that share a partial prefix with the class, you can enable <b>removePartialPrefix</b> to remove the partial prefix from the property name. The <b>removePartialPrefix</b> rule is set to <code>false</code> by default. If you want the Generator to remove partial prefixes, you can include <code>s:removePrefix true</code> .

Property	Data Type	Description
<b>match</b>	RDF list	<p>This rule provides a way to use regular expressions (REGEX) to match a <b>pattern</b> against source identifiers and <b>replace</b> the matched text in the normalized name:</p> <pre>s:match [   s:pattern "java_regex_to_match" ; s:replace "java_ regex_to_replace" ; ] ;</pre> <p>The <code>s:pattern</code> property defines the Java REGEX pattern to match against, and <code>s:replace</code> defines the Java REGEX replacement pattern. As shown in the example below, the match rule can also be configured with an <code>rdf:List</code> of objects to perform match evaluation in a certain order:</p> <pre>s:match (   [ s:pattern "(.+)GUID\$" ; s:replace "\$1" ; ]   [ s:pattern "(.+)ID\$" ; s:replace "\$1" ; ] )</pre>
<b>disambiguationLevel</b>	int	<p>This rule specifies the number of levels to use to resolve ambiguities between similarly named elements in a hierarchical source. For example, an element named "Data" appears in two contexts: "Currently" and "Hourly." By default, the Generator retains all levels, meaning two classes are generated: "Currently Data" and "Hourly Data." If <code>s:disambiguationLevel</code> is set to 0, a single class named "Data" is generated and both the Currently and Hourly classes have a "Data" property. The <code>disambiguationLevel</code> value is also used to determine the number of hierarchy levels to use when encoding the local name of the generated URI.</p>
<b>ignore</b>	string	<p>This rule can be used to list identifiers that should be ignored. Properties and classes will not be generated for identifiers that match the specified string(s). The ignore rule is a multi-valued property. For simplicity, you can enter a list by separating words with a space, rather than quoting each term and separating them with a comma. For multi-word identifiers, use single quotes. For example, <code>s:ignore "sample example 'test column' old"</code>.</p>

Property	Data Type	Description
<b>words</b>	string	<p>Since many sources do not encode word boundaries very well, the <b>words</b> rule can be used to list the set of words that should be separate identifiers. This rule tells the Generator which words may be encountered. The words rule is a multi-valued property. For simplicity, you can enter a list by separating words with a space, rather than quoting each term and separating them with a comma. For multi-word identifiers, use single quotes. For example:</p> <pre>s:words "activity 'patient complaint' medication observation patient signal specialty study" ;</pre>
<b>preserve</b>	string	<p>This rule can be used to identify any words whose casing should be preserved in the input identifiers. For example, if <b>casing</b> is set to <code>lower</code> but you want preserve the original upper casing of certain words, you can specify the words to preserve. The preserve rule is a multi-valued property. For simplicity, you can enter a list by separating words with a space, rather than quoting each term and separating them with a comma. For multi-word identifiers, use single quotes. For example: <code>s:preserve "ABC 'Laundry List' TriG"</code>. The preserve rule is case-insensitive. You do not have to match the casing of the words to preserve.</p>
<b>split</b>	string	<p>This rule specifies the string that should be used to split source identifiers into individual terms. If neither <code>split</code> nor <code>words</code> is specified, input identifiers are split on casing changes and character class changes.</p>
<b>allowWhiteSpace</b>	boolean	<p>This rule specifies whether or not white space should be preserved in identifiers after they have been split into individual terms. This rule is set to <code>false</code> by default, meaning white space is not preserved. You can specify <code>s:allowWhiteSpace true</code> to preserve spaces.</p>
<b>allowPunctuation</b>	boolean	<p>This rule specifies whether or not punctuation should be preserved in identifiers after they have been split into individual terms. This rule is set to <code>false</code> by default, meaning punctuation is not preserved. You can specify <code>s:allowPunctuation true</code> to preserve punctuation.</p>

Property	Data Type	Description
<b>allowSymbols</b>	boolean	This rule specifies whether or not symbols should be preserved in identifiers after they have been split into individual terms. This rule is set to <code>false</code> by default, meaning symbols are not preserved. You can specify <code>s:allowSymbols true</code> to preserve symbols.
<b>singularize</b>	boolean	This rule specifies whether or not to change any plural identifiers to singular. This rule is set to <code>false</code> by default, meaning plural identifiers are preserved. You can specify <code>s:singularize true</code> to change plural terms to the singular version of the term.
<b>casing</b>	object	<p>This rule specifies how the generated labels should be cased. By default, the Generator outputs labels in upper camel case (<code>s:casing s:UpperCamel</code>). To use a different casing, specify any of the following properties:</p> <ul style="list-style-type: none"> <li>• <b>default:</b> This object preserves the casing from the source. Labels will not be converted.</li> <li>• <b>UPPER:</b> This object converts all characters to uppercase. For example, "uppercase" becomes "UPPERCASE."</li> <li>• <b>lower:</b> This object converts all characters to lowercase. For example, "Lower Case" becomes "lower case".</li> <li>• <b>UpperCamel:</b> This is the default casing value and converts labels to upper camel case, where terms are concatenated and the first letter of each word is capitalized. For example, "upper camel case" becomes "UpperCamelCase."</li> <li>• <b>lowerCamel:</b> This object converts labels to lower camel case, where terms are concatenated and the first letter of the first word is lower case. The first letter of subsequent terms is capitalized. For example, "lower camel case" becomes "lowerCamelCase."</li> </ul>
<b>separator</b>	string	This rule specifies the character or characters to use to separate terms in the generated label. The default separator is a space ( <code>s:separator " "</code> ).

Property	Data Type	Description
<b>localNamePrefix</b>	string	This rule specifies a string to use as the prefix for local names when generating a URI.
<b>localNameSeparator</b>	string	This rule specifies the string to use for separating local names when encoding hierarchies according to the specified <a href="#">disambiguationLevel</a> . By default, <code>localNameSeparator</code> is a period ( <code>s:localNameSeparator "."</code> ). If <code>localNameSeparator</code> is empty, hierarchical context will not be encoded into the local name of any properties or child classes. The result would be an ontology where only the class or property name is used to determine the local name. For example, a property URI would look like <code>ont:employeeID</code> rather than <code>ont:Employee.employeeID</code> . The result could lead to "conflicts" in the generated ontology, but those "conflicts" may be desired as properties with same name are reused across the generated ontology.

## Normalize Examples

The example below uses the `normalize` property to normalize data at both the model and field level.

```
s:normalize [
  s:model [
    s:localNamePrefix "C_" ;
    s:localNameSeparator "_" ;
    s:match [ s:pattern "(.+)"Enlarged" ; s:replace "$1" ] ;
  ] ;
  s:field [
    s:localNamePrefix "P_" ;
    s:localNameSeparator "_";
    s:ignore "rowguid ModifiedDate" ;
    s:match (
      [ s:pattern "(.+)"GUID$" ; s:replace "$1" ]
      [ s:pattern "(.+)"ID$" ; s:replace "$1" ]
    ) ;
  ] ;
] ;
```

## Related Topics

Data Linking Options

When a data source does not define keys (such as a CSV or JSON source), the GDI provides properties that enable you to create a connected knowledge graph by defining relationships, resource templates (primary keys) and object properties (foreign keys), when you are loading data from multiple sources. The properties that are available are described below.

- [Data Linking Syntax](#)
- [Data Linking Examples](#)

Data Linking Syntax

```
s:key ("column_name") ;
s:reference [
  s:model "table_to_reference" ;
  s:using ("foreign_key_column")
]
```

Option	Data Type	Description
key	string	Include this property when you want to define the primary key column for the source file or table. This column is leveraged in a resource template for the instances that are created from the source. For example, <code>s:key ("EMPLOYEE_ID")</code> . If none of the columns contain unique values, you can specify a combination of columns that would create a unique value. For example, <code>s:key ("FlightNumber", "TailNumber")</code> .
reference	RDF list	<div>Include this property when you want to specify a foreign key column. The reference property is an RDF list that includes the <code>model</code> property to list the target table and a <code>using</code> property that defines the foreign key column in the source table.</div> <div><pre>s:reference [ s:model "table_to_reference" ; s:using ("foreign_key_column") ]</pre></div> <div><div>Tip</div><div>You can also include an optional <code>key</code> property within the <code>s:reference</code> list</div></div>

Option	Data Type	Description
		<p>that defines the key column in the target table and can be used as a way to expose additional metadata that helps inform the GDI how to name the object property. For example:</p> <pre>s:reference [ s:model "Employees" ; s:using ("EMPLOYEE_ID") ; s:key ("EMPLOYEE_ID") ]</pre>

## Data Linking Examples

For example, the query snippet below defines two data sources. The `s:model` property defines the table/class for each source, and the `s:key` defines the primary key for each table/class. The `s:reference` property for the "venue" table defines a foreign key relationship from `venue.EVENT_ID` to `event.EVENT_ID`.

```
?event a s:FileSource ;
  s:model "event" ;
  s:url "/opt/shared-files/csv/events.csv" ;
  s:key ("EVENT_ID") .

?venue a s:FileSource ;
  s:model "venue" ;
  s:url " /opt/shared-files/csv/venues.csv" ;
  s:key ("VENUE_ID") ;
  s:reference [ s:model "event" ; s:using ("EVENT_ID") ] .
```

The following query for multiple file sources generates RDF and an ontology with resource templates and object properties. The query also includes global normalization rules for normalizing the data across all sources (see [Normalization Options](#) for information about normalization).

```
PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>

INSERT {
  GRAPH ${targetGraph} {
    ?s ?p ?o .
  }
}
```

```

}
WHERE {
    SERVICE <http://cambridgesemantics.com/services/DataToolkit> {

        ?event a s:FileSource ;
            s:model "event" ;
            s:url   "/opt/shared-files/csv/events.csv" ;
            s:key   ("EVENT_ID") .

        ?listing a s:FileSource ;
            s:model "listing" ;
            s:url   " /opt/shared-files/csv/listings.csv" ;
            s:key   ("LIST_ID") ;
            s:reference [ s:model "event" ; s:using ("EVENT_ID") ; s:key ("EVENT_
ID") ] .

        ?date a s:FileSource ;
            s:model "date" ;
            s:url   "/opt/shared-files/csv/event_dates.csv" ;
            s:key   ("DATE_ID") ;
            s:reference [ s:model "event" ; s:using ("EVENT_ID") ; s:key ("EVENT_
ID") ] .

        ?venue a s:FileSource ;
            s:model "venue" ;
            s:url   " /opt/shared-files/csv/venues.csv" ;
            s:key   ("VENUE_ID") ;
            s:reference [ s:model "event" ; s:using ("EVENT_ID") ; s:key ("EVENT_
ID") ] .

        ?sale a s:FileSource ;
            s:model "sale" ;
            s:url   " /opt/shared-files/csv/sales.csv" ;
            s:key   ("SALE_ID") ;
            s:reference [ s:model "event" ; s:using ("EVENT_ID") ; s:key ("EVENT_
ID") ] ;
            s:reference [ s:model "listing" ; s:using ("LIST_ID") ; s:key ("LIST_
ID") ] .
    }
}

```



```

?rdf a s:RdfGenerator, s:OntologyGenerator ;
  s:as (?s ?p ?o) ;
  s:ontology <http://cambridgesemantics.com/tickets> ;
  s:base <http://cambridgesemantics.com/data> ;
  s:normalize [
    s:all [
      s:casing s:UPPER ;
      s:localNameSeparator "_" ;
    ] ;
  ] .
}

```

## Pagination Options

The GDI exposes paging models that enable you to access large amounts of data across a number of smaller requests. Paging is configured by including the **paging** property in a query and configuring a combination of the pagination options described below. The GDI supports keyset-based, page-based, cursor-based, and offset-based pagination. Paging is supported for all data source types.

- [Paging Syntax](#)
- [Paging Examples](#)

## Paging Syntax

```

s:paging [
  s:key (?variable) ;
  s:page ?variable ;
  s:cursor ?variable ;
  s:offset ?variable ;
  s:size int ;
  s:limit ?variable ;
] ;

```

Option	Data Type	Description
<b>key</b>	variable	Include this property if you want to configure keyset-based pagination where a key is specified to act as a delimiter of the page. The <code>s:key</code> value is a variable that is bound to an expression that defines how to delimit the data. It is usually

Option	Data Type	Description
		calculated by an aggregate expression and/or filter that can be pushed to the source. The aggregate expression is typically MAX, but MIN can also be used to page through data in reverse order, such as when working with temporal data. See <a href="#">Key-Based Examples</a> below for examples that configure paging using the <code>s:key</code> property.
<b>page</b>	variable	Include this property if you want to configure page-based pagination where the set is divided into pages. The <code>s:page</code> property value is a variable that the GDI can use to track the current page across requests. See <a href="#">Page-Based Example</a> below for an example that configures paging using the <code>s:page</code> property.
<b>cursor</b>	variable	Include this property if you want to configure cursor-based pagination. The <code>s:cursor</code> property value is a variable that is bound against the source to capture the "cursor" value. The GDI uses this value as input to the source to deliver the next page of data. See <a href="#">Cursor-Based Example</a> below for an example that configures paging with the <code>s:cursor</code> property.
<b>offset</b>	variable	Include this property along with the <a href="#">limit</a> property if you want to configure offset-based pagination. The <code>s:offset</code> property value is a variable that the GDI can use to track the current offset across requests. See <a href="#">Offset-Based Example</a> below for an example that configures paging using the <code>s:offset</code> property.
<b>size</b>	int	This property can be included with any of the paging models to configure the maximum size of each page. For example, <code>s:size 5000</code> limits the page size to 5,000 rows.
<b>limit</b>	variable	This property can be included to define the variable that the GDI should use to push the page size back to the source.

## Paging Examples

- [Key-Based Examples](#)
- [Page-Based Example](#)

- [Cursor-Based Example](#)
- [Offset-Based Example](#)

## Key-Based Examples

The example SERVICE clause below pages data based on the `?LastID` key, which is calculated by finding the maximum value of `SalesOrderID` and binding it to `?LastID`. A FILTER is used to filter for data where the `SalesOrderID` is greater than `?LastID`.

```
SERVICE <http://cambridgesemantics.com/services/DataToolkit>
{
  BIND(MAX(?SalesOrderID) AS ?LastID)
  FILTER(?SalesOrderID > ?LastID)

  ?SalesOrderHeaderEnlarged a s:DbSource ;
    s:url "jdbc:sqlserver://..." ;
    s:table "Sales.SalesOrderHeaderEnlarged" ;
    s:paging [
      s:key (?LastID) ;
      s:size 5000 ;
    ] ;
    ?SalesOrderID (xsd:int) ;
    ?RevisionNumber (xsd:int) ;
    ?OrderDate ("OrderDate" xsd:dateTime) ;
    ?DueDate (xsd:dateTime) .
}
```

The SERVICE clause below shows an example where key-based paging is configured to page through temporal data in reverse order. The `s:limit` property is configured on the `s:HttpSource` to limit the overall number of results returned across all pages. This query retrieves at most 1000 records (`s:limit 1000`), 100 rows (`s:size 100`) at a time.

```
SERVICE <http://cambridgesemantics.com/services/DataToolkit>
{
  BIND(MIN(?Timestamp) AS ?LastTimestamp)

  ?api a s:HttpSource ;
    s:url "http://slack.com/api/messages/latest" ;
```

```

s:parameter [ s:name "before" ; s:value ?LastTimestamp ] ;
s:parameter [ s:name "limit" ; s:value ?limit ] ;
s:limit 1000 ;
s:paging [
  s:key (?LastTimestamp) ;
  s:limit ?limit ;
  s:size 100 ;
] ;
?Message (xsd:string) ;
?Author (xsd:string) ;
?Timestamp (xsd:dateTime) .
}

```

## Page-Based Example

The SERVICE clause below shows an example that uses the `s:page` property to configure page-based paging where the page size is 100 rows. This query retrieves at most 1000 records (`s:limit 1000`), 100 rows (`s:size 100`) at a time.

```

SERVICE <http://cambridgesemantics.com/services/DataToolkit>
{
  ?api a s:HttpSource ;
  s:url "http://slack.com/api/messages" ;
  s:parameter [ s:name "page" ; s:value ?page ] ;
  s:parameter [ s:name "size" ; s:value ?limit ] ;
  s:limit 1000 ;
  s:paging [
    s:page ?page ;
    s:limit ?limit ;
    s:size 100 ;
  ] ;
  ?Message (xsd:string) ;
  ?Author (xsd:string) ;
  ?Timestamp (xsd:dateTime) .
}

```

## Cursor-Based Example

The SERVICE clause below shows an example that uses the `s:cursor` property to configure cursor-based paging.

```
SERVICE <http://cambridgesemantics.com/services/DataToolkit>
{
  ?api a s:HttpSource ;
    s:url "http://slack.com/api/messages" ;
    s:parameter [ s:name "cursor" ; s:value ?cursor ] ;
    s:parameter [ s:name "limit" ; s:value ?limit ] ;
    s:limit 1000 ;
    s:paging [
      s:cursor ?cursor ;
      s:limit ?limit ;
      s:size 100 ;
    ] ;
    ?Message (xsd:string) ;
    ?Author (xsd:string) ;
    ?Timestamp (xsd:dateTime) ;
    ?cursor ("next_cursor" xsd:string) .
}
```

## Offset-Based Example

The SERVICE clause below shows an example that uses the `s:offset` property to configure offset-based paging.

```
SERVICE <http://cambridgesemantics.com/services/DataToolkit>
{
  ?api a s:HttpSource ;
    s:url "http://slack.com/api/messages" ;
    s:parameter [ s:name "offset" ; s:value ?offset ] ;
    s:parameter [ s:name "limit" ; s:value ?limit ] ;
    s:limit 1000 ;
    s:paging [
      s:offset ?offset ;
      s:limit ?limit ;
    ] ;
}
```

```

        s:size 100 ;
    ] ;
    ?Message (xsd:string) ;
    ?Author (xsd:string) ;
    ?Timestamp (xsd:dateTime) .
}

```

## Bindings and Hierarchies

As part of the Graph Data Interface's (GDI) flexibility, there are multiple ways to express binding hierarchies in queries. This topic describes the options for expressing hierarchies.

- [Using Binding Trees and Selector Paths](#)
- [Unpacking JSON with Bindings and Arrays](#)
- [Returning Hierarchies as JSON Strings](#)

### Using Binding Trees and Selector Paths

One way to express hierarchies in queries is to use brackets ( [ ] ) to group objects into binding trees. For example, the WHERE clause snippet below organizes mapping variable objects into an `hourly/data` hierarchy by nesting the `?data` patterns inside the `?hourly [ ]` tree:

```

WHERE
{
    SERVICE <http://cambridgesemantics.com/services/DataToolkit>
    {
        ?data a s:HttpSource;
        s:url "https://sampleEndpoint.com/forecast/" ;
        ?latitude (xsd:double) ;
        ?longitude (xsd:double) ;
        ?timezone (xsd:string) ;
        ?hourly
        [
            ?data
            [
                ?time (xsd:long) ;
                ?summary (xsd:string) ;
                ?rainIntensity ("precipIntensity" xsd:double) ;
                ?rainProbability ("precipProbability" xsd:double) ;
                ?temperature (xsd:double) ;
            ]
        ]
    }
}

```

```

        ?feelsLike ("apparentTemperature" xsd:double) ;
        ?humidity (xsd:double) ;
        ?pressure (xsd:double) ;
        ?windSpeed (xsd:double) ;
    ] ;
] .
}
}

```

When constructing object binding trees, if you choose to introduce the hierarchy with a variable name that is not an exact match to the source label, include a **selector** property to list the value from the source. For example, in the WHERE clause snippet below, `s:selector` is included to select `eventHeader` in the source as `?event` in the query and `statLocation` as `?location`.

```

WHERE
{
    SERVICE <http://cambridgesemantics.com/services/DataToolkit>
    {
        ?data a s:FileSource ;
        s:url "/mnt/data/json/part_1.json" ;
        ?event
    [
        s:selector "eventHeader" ;
        ?eventId (xsd:string) ;
        ?eventName (xsd:string) ;
        ?eventVersion (xsd:string) ;
        ?eventTime (xsd:dateTime) ;
    ] ;
    ?location
    [
        s:selector "statLocation" ;
        ?locationId (xsd:string) ;
        ?lineNo (xsd:int) ;
        ?statNo (xsd:int) ;
        ?statId (xsd:int) ;
    ] .
    }
}

```

As an alternative to grouping objects in binding trees, the **selector** property also supports using dot notation to specify paths. For example, the WHERE clause snippet below rewrites the first example query to express the same `hourly/data` hierarchy as a path in the `s:selector` value:

```
WHERE
{
  SERVICE <http://cambridgesemantics.com/services/DataToolkit>
  {
    ?data a s:HttpSource;
    s:url "https://sampleEndpoint.com/forecast/" ;
    ?latitude (xsd:double) ;
    ?longitude (xsd:double) ;
    ?timezone (xsd:string) ;
    s:selector: "hourly.data" ;
    ?time (xsd:long) ;
    ?summary (xsd:string) ;
    ?rainIntensity ("precipIntensity" xsd:double) ;
    ?rainProbability ("precipProbability" xsd:double) ;
    ?temperature (xsd:double) ;
    ?feelsLike ("apparentTemperature" xsd:double) ;
    ?humidity (xsd:double) ;
    ?pressure (xsd:double) ;
    ?windSpeed (xsd:double) .
  }
}
```

You can also include the `$` character to anchor the selector at the root of the file. For example, `s:selector "data"` captures all `data` elements anywhere in the file. But `s:selector "$.data"` captures only the `data` elements that are at the root of the hierarchy.

## Unpacking JSON with Bindings and Arrays

In addition to object binding trees and selectors, the GDI offers additional syntax for reading or ingesting JSON sources with nested objects and arrays. For example, following the JSON sample file below is a query that captures each value in the arrays:

```
{
  "payload" :
  {
    "IBP_IndEvent_MSR" :
```



```

{
  "unit" : "ms",
  "value" : [ 0, 1 ]
},
"IBP_IndEvent_RMF" :
{
  "unit" : "-",
  "value" : [ 0.012, 1.398, 3.1415 ]
}
}
}

```

To read the JSON file above, the following query uses an object binding (`?values [ ]`) to drill down to the value arrays in the source. An `@` selector is specified in the `?value` variable binding (`?value ("@" xsd:double)`) to retrieve each of the array values. For an array of primitive values, the `@` selector captures each value in the array. If the source value was an array of objects, the `@` selector would retrieve a JSON representation for each object in the array. In addition to creating a new binding context for the primitive array values, the `?values` object binding also includes `?index ("!array::index")` to capture the index array with the primitive value.

```

PREFIX s:    <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
SELECT *
WHERE {
  SERVICE <http://cambridgesemantics.com/services/DataToolkit> {
    ?data a s:FileSource ;
    s:url "/mnt/data/json/array-index.json" ;
    s:selector "payload.*" ;
    ?unit (xsd:string) ;
    ?values [
      s:selector "value" ;
      ?value ("@" xsd:double) ;
      ?index ("!array::index") ;
    ] .
  }
}

```

The results of the query are shown below:

unit	value	index
-----	-----	-----
ms	0	0
ms	1	1
-	0.012	0
-	1.398	1
-	3.1415	2

If you do not want to retrieve all of the values in an array, you can include the specific index number to retrieve instead of using the @ symbol. In the variable binding, the index number is appended in brackets ([ ]) to the binding column name. For example, the following variable binding retrieves the second index value (the third value in the array) from a "projects" array: `?project ("projects[2]")`. The next example uses the following JSON file:

```
{
  "field1" : "value1" ,
  "arrayfield" : [
    "arrayvalue1",
    "arrayvalue2"
  ]
}
```

To retrieve only the second value in the array, the following query appends the index value 1 to the array column name, `arrayfield`:

```
PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>
SELECT *
WHERE {
  SERVICE <http://cambridgesemantics.com/services/DataToolkit> {
    ?json a s:FileSource ;
    s:url "/mnt/data/json/array-index-2.json" ;
    ?field1 (xsd:string) ;
    ?arrayval ("arrayfield[1]" xsd:string) .
  }
}
```

The results of the query are shown below:

```

field1    | arrayval
-----+-----
value1    | arrayvalue2

```

## Returning Hierarchies as JSON Strings

When working with schema-less sources, you can also capture a tree of data as a JSON string. For example, the query snippet below targets an HTTP endpoint. In this case, the properties under the **hourly** class of data are unknown. So the query binds all of the data below hourly to the **?hourly** variable by including empty parentheses. As a result, the GDI returns a JSON string representation of all of the properties and instance data under hourly:

```

WHERE
{
  SERVICE <http://cambridgesemantics.com/services/DataToolkit>
  {
    ?data a s:HttpSource;
    s:url "https://sampleEndpoint.com/forecast/" ;
    ?latitude (xsd:double) ;
    ?longitude (xsd:double) ;
    ?timezone (xsd:string) ;
    ?hourly () .
  }
}

```

For example, the results look like this:

```

latitude    | longitude    | timezone          | hourly
-----+-----+-----+-----
30.374563   | -97.975892   | America/Chicago   | {"summary":"\
cloudy
throughout the day.\",\"icon\":\"partly-cloudy-day\", \"data\":
[{"time\":\"1595559600\",
summary\":\"Clear\", \"icon\":\"clear-night\", \"precipIntensity\":\"0\",
\"precipProbability\":\"0\", \"temperature\":\"88.39\", \"apparentTemperature\":\"91.72\",
\"dewPoint\":\"67.42\", \"humidity\":\"0.5\", \"pressure\":\"1011.7\", \"windSpeed\":\"7.48\",
\"windGust\":\"16.71\", \"windBearing\":\"109\", \"cloudCover\":\"0.06\", \"uvIndex\":\"0\",
\"visibility\":\"10\", \"ozone\":\"285.2\"}, {\"time\":\"1595563200\", \"summary\":\"Clear\",

```

```
"icon":"\"clear-night\"", "precipIntensity":"2.0E-4", "precipProbability":"0.01",
"precipType":"\"rain\"", "temperature":"86.69", "apparentTemperature":"90.1",
"dewPoint":"67.84", "humidity":"0.54", "pressure":"1012", "windSpeed":"7.05",
"windGust":"17.56", "windBearing":"110", "cloudCover":"0.12", "uvIndex":"0",
"visibility":"10", "ozone":"284.9"}, ...
```

Similar to the example above, you can write a query that specifically captures some of the properties in a hierarchy and then returns the rest of the properties and their values as a JSON string representation. To do so, use "@" as the binding path. For example:

```
WHERE
{
  SERVICE <http://cambridgesemantics.com/services/DataToolkit>
  {
    ?data a s:HttpSource;
    s:url
    "https://api.darksky.net/forecast/bdbe3f638eb908c9b94919537dad5945/30.374563,-
    97.975892" ;
    ?latitude (xsd:double) ;
    ?longitude (xsd:double) ;
    ?timezone (xsd:string) ;
    ?hourly [
      s:selector "hourly.data" ;
      ?time (xsd:long) ;
      ?summary (xsd:string) ;
      ?hourly_data ("@") ;
    ] .
  }
}
```

Sample results are shown below:

latitude	longitude	timezone	time	summary	
30.374563	-97.975892	America/Chicago	1595559600	Clear	
{ "time": "1595559600", "summary": "\"Clear\"",					
"icon": "\"clear-					

```

night\","", "precipIntensity": "0", "precipProbability": "0", "temperature": "88.39",

"apparentTemperature": "91.72", "dewPoint": "67.42", "humidity": "0.5", "pressure": "1011.7", "windSpeed": "7.48",

"windGust": "16.71", "windBearing": "109", "cloudCover": "0.06", "uvIndex": "0", "visibility": "10", "ozone": "285.2"}

30.374563 | -97.975892 | America/Chicago | 1595563200 | Clear |
{"time": "1595563200", "summary": "\"Clear\"",
"icon": "\"clear-night\"", "precipIntensity": "2.0E-4", "precipProbability": "0.01", "precipType": "\"rain\"", "temperature": "86.69",

"apparentTemperature": "90.1", "dewPoint": "67.84", "humidity": "0.54", "pressure": "1012", "windSpeed": "7.05", "windGust": "17.56",

"windBearing": "110", "cloudCover": "0.12", "uvIndex": "0", "visibility": "10", "ozone": "284.9"}

30.374563 | -97.975892 | America/Chicago | 1595566800 | Partly Cloudy |
{"time": "1595566800", "summary": "\"Partly Cloudy\"",
"icon": "\"partly-cloudy-night\"", "precipIntensity": "3.0E-4", "precipProbability": "0.01",
"precipType": "\"rain\"", "temperature": "85.63", "apparentTemperature": "89.21",

"dewPoint": "68.33", "humidity": "0.56", "pressure": "1012.6", "windSpeed": "6.48", "windGust": "17.92", "windBearing": "110",
"cloudCover": "0.34", "uvIndex": "0", "visibility": "10", "ozone": "284.5"}
...

```

## Related Topics

[General GDI Concepts and Usage](#)

[Getting Started with GDI Queries](#)

## Incremental Onboarding with the GDI

When loading data from a database or file-based data source with a Graph Data Interface (GDI) query, you can add a few statements to the query to load a portion of the data incrementally rather than all of the data at once. As data is added or changed in the source, new data can be ingested without having to reload all of the

previously ingested data. Because incremental ingestion is configured as a filter in a SPARQL query, it is extremely flexible, allowing for various conditions to be defined for diverse data sources. When the data is ingested, the GDI evaluates the current state of the data and then loads only the data that meets the conditions defined in the query.

This topic provides example incremental queries to get you started and includes instructions for configuring a Data Layer as an incremental ingestion workflow.

- [Incremental DbSource Example](#)
- [Incremental FileSource Example](#)
- [Setting Up a Data Layer to Ingest Data Incrementally](#)

## Incremental DbSource Example

The following query from a Direct Load Step ingests data from a database. All of the values for the requested columns in the ORDER\_DETAILS table will be loaded.

```
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>

INSERT {
  GRAPH ${targetGraph} {
    ?s ?p ?o .
  }
}
${usingSources}
WHERE {
  SERVICE <http://cambridgesemantics.com/services/DataToolkit> {

    ?data a s:DbSource ;
      s:url "jdbc:oracle:thin:@10.10.10.10:1111/XE" ;
      s:username "northwind" ;
      s:password "NORTHWIND123" ;
      s:schema "NORTHWIND" ;
      s:table "ORDER_DETAILS" ;
      ?database ("!") ;
      ?schema ("!") ;
      ?table ("!") ;
```

```

        ?OrderID (xsd:int) ;
        ?ProductID (xsd:int) ;
        ?UnitPrice (xsd:double) ;
        ?Quantity (xsd:short) ;
        ?Discount xsd:double .

    BIND( IRI("http://cambridgesemantics.com/orders/{{?OrderID}}") AS
?resource)

    ?rdf a s:RdfGenerator, s:OntologyGenerator ;
    s:as (?s ?p ?o) ;
    s:ontology <http://cambridgesemantics.com/ontologies/northwind> ;
    s:base <http://cambridgesemantics.com/data> .
}
}

```

The query below adds statements that configure the same Direct Load Step to ingest data incrementally. It captures the maximum order ID as the incremental value. When the source is updated with records that increase the order ID, only the records with larger order IDs than the previous maximum value will be ingested when the Graphmart is refreshed or reloaded. In the query:

- A ?MaxID variable is bound to the result of MAX(?OrderID): `BIND (MAX(?OrderID) AS ?MaxID).`
- The ?MaxID variable is defined as the incremental value: `?MaxID a s:IncrementalValue.`
- A filter clause is added to create a condition that ingests only the records where the order ID is greater than the previously ingested maximum ID: `FILTER (?OrderID > ?MaxID).`

```

PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>

INSERT {
    GRAPH ${targetGraph} {
        ?s ?p ?o .
    }
}
${usingSources}

```

```

WHERE {
  SERVICE <http://cambridgesemantics.com/services/DataToolkit> {

    ?MaxID a s:IncrementalValue .
    FILTER (?OrderID > ?MaxID)
    BIND (MAX(?OrderID) AS ?MaxID)

    ?data a s:DbSource ;
      s:url "jdbc:oracle:thin:@10.10.10.10:1111/XE" ;
      s:username "northwind" ;
      s:password "NORTHWIND123" ;
      s:schema "NORTHWIND" ;
      s:table "ORDER_DETAILS" ;
      ?database ("!") ;
      ?schema ("!") ;
      ?table ("!") ;
      ?OrderID (xsd:int) ;
      ?ProductID (xsd:int) ;
      ?UnitPrice (xsd:double) ;
      ?Quantity (xsd:short) ;
      ?Discount xsd:double .

    BIND(IRI("http://cambridgesemantics.com/orders/{{?OrderID}}") AS
?resource)

    ?rdf a s:RdfGenerator, s:OntologyGenerator ;
    s:as (?s ?p ?o) ;
    s:ontology <http://cambridgesemantics.com/ontologies/northwind> ;
    s:base <http://cambridgesemantics.com/data> .
  }
}

```

## Incremental FileSource Example

The following query from a Direct Load Step ingests data from all of the CSV files in the `/nfs/data/fmcsa` directory:



```

PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>

DELETE {
  GRAPH ${targetGraph} {
  }
}
INSERT {
  GRAPH ${targetGraph} {
    ?s ?p ?o .
  }
}
WHERE {
  SERVICE <http://cambridgesemantics.com/services/DataToolkit> {

    ?data a s:FileSource ;
      s:model "fmcsa" ;
      s:url "/nfs/data/fmcsa" ;
      s:pattern "*.csv" .

    ?rdf a s:RdfGenerator , s:OntologyGenerator ;
      s:as (?s ?p ?o) ;
      s:ontology <http://cambridgesemantics.com/ontologies/fmcsa> ;
      s:base <http://cambridgesemantics.com/data/> .
  }
}

```

The query below adds statements that configure the same Direct Load Step to ingest data incrementally. It uses a "last modified" strategy to determine what files are new or modified and should be ingested the next time the Graphmart is refreshed or reloaded. In the query:

- The modified timestamp metadata on the files is captured with `?Modified ("!")`.
- The `?LastRun` variable is bound to the result of the `NOW()` function: `BIND (NOW() AS ?LastRun)`.
- A filter clause is added to check whether the modified timestamp is later than the timestamp from the last time the query was run: `FILTER (?Modified > ?LastRun)`.
- `?LastRun` is defined as the incremental value: `?LastRun a s:IncrementalValue`.

```

PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>

DELETE {
    GRAPH ${targetGraph} {
    }
}
INSERT {
    GRAPH ${targetGraph} {
        ?s ?p ?o .
    }
}
WHERE {
    SERVICE <http://cambridgesemantics.com/services/DataToolkit> {

        ?LastRun a s:IncrementalValue .
        FILTER (?Modified > ?LastRun)
        BIND (NOW() AS ?LastRun)

        ?data a s:FileSource ;
            s:model "fmcsa" ;
            s:url "/nfs/data/fmcsa" ;
            s:pattern "*.csv" ;
            ?Modified ("!") .

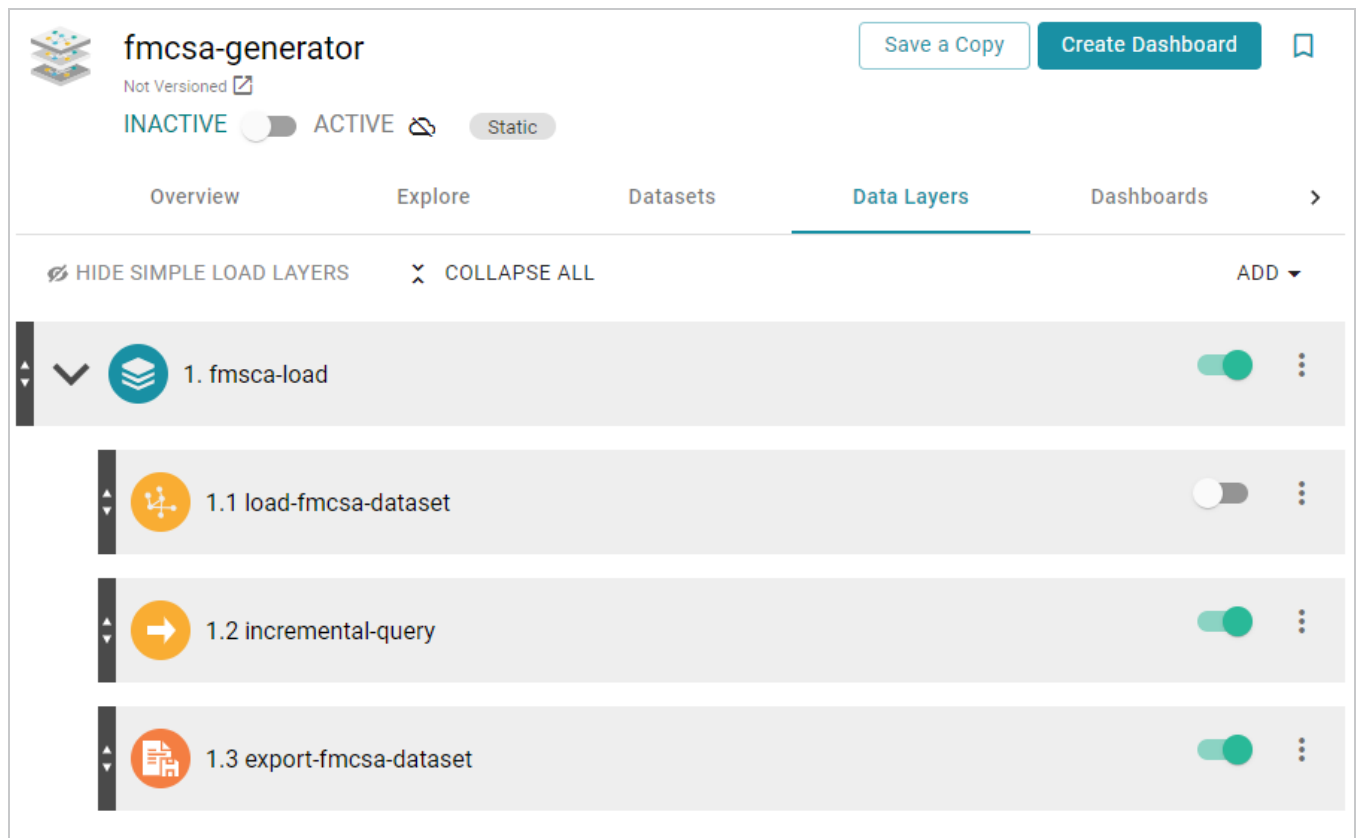
        ?rdf a s:RdfGenerator , s:OntologyGenerator ;
            s:as (?s ?p ?o) ;
            s:ontology <http://cambridgesemantics.com/ontologies/fmcsa> ;
            s:base <http://cambridgesemantics.com/data/> .
    }
}

```

## Setting Up a Data Layer to Ingest Data Incrementally

1. Create a new empty dataset in the Anzo Data Store. For instructions, see [Adding an Empty Dataset for an Export Step](#).

2. In the Graphmart where you want to add a GDI query that ingests data incrementally, add a new Data Layer.
3. In the new layer, add a Load Dataset Step as the first step. The Linked Dataset for this step should be the empty dataset that you created in the first step.
4. Now, add a Direct Load Step as the next step in the layer. Edit the query template in the step to compose the incremental query.
5. As the last step in the layer, add an Export Step. The Target FLDS for the step should also be the empty dataset that you created in the first step. For example, the image below shows a Graphmart with a layer that is set up to ingest data incrementally.



6. Activate the Graphmart to ingest the data and export it to an FLDS. Once the Graphmart is activated, enable the Load Dataset Step.

## Related Topics

[Generating a Graph and Ontology with a Direct Load Step](#)

## Reading Data Source Metadata

If you want to retrieve instance data from a source but are unsure about the data model, schema, or the exact names of columns and their data types, you can use the Graph Data Interface (GDI) to explore the source's metadata. The GDI can be used to return a list of the catalogs (schemas), models, columns, data types, and other data source information.

This topic describes the metadata query syntax and provides several example queries.

- [Metadata Query Syntax](#)
- [Metadata Query Examples](#)

### Metadata Query Syntax

The following query syntax shows the structure of a metadata query. The clauses, patterns, and placeholders in blue are described below.

```
# PREFIX Clause
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>

# Result Clause
SELECT *
WHERE
{
    # SERVICE Clause: Include the following service call
    SERVICE <http://cambridgesemantics.com/services/DataToolkit>
    {
        [] s:select ?metadata .

        ?data a s:source_type ;
            s:url "string" ;
            [ s:username "string" ; ]
            [ s:password "string" ; ]
    }
}
```

```

?metadata a s:MetadataSource ;
    s:from ?data ;

# The metadata selector below specifies the type of metadata to return.
?catalogs | ?fields | ?models [
    ?metadata_type datatype ;
    ... ;
] .
}
}

```

Option	Data Type	Description
<b>PREFIX Clause</b>	N/A	<p>The PREFIX clause declares the standard and custom prefixes for GDI queries. Generally, queries include the following prefixes (or a subset of them) plus any data-specific declarations:</p> <pre> PREFIX s: &lt;http://cambridgesemantics.com/ontologies/DataToolkit#&gt; PREFIX rdf:    &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX rdfs:    &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX xsd:     &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX owl:   &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX anzo: &lt;http://openanzo.org/ontologies/2008/07/Anzo#&gt; PREFIX zowl: &lt;http://openanzo.org/ontologies/2009/05/AnzoOwl#&gt; PREFIX dc:      &lt;http://purl.org/dc/elements/1.1/&gt; </pre>
<b>Result Clause</b>	N/A	The result clause defines the results to return. For metadata queries, the result clause is typically <code>SELECT *</code> .
<b>SERVICE Clause</b>		Include the required GDI SERVICE call in the WHERE clause. The rest of the WHERE clause defines the patterns to look for in the source.

Option	Data Type	Description
<code>[] s:select ?metadata</code>	N/A	Include this required triple pattern in metadata queries. The select property specifies the source that should be used to return data.
<code>source_type</code>	object	<p>The <code>?data a s:source_type</code> triple pattern specifies the type of data source that the query will run against. For example, <code>?data a s:DbSource</code>, specifies that the source type is a database. The list below describes the available types:</p> <ul style="list-style-type: none"> <li>• <b>DbSource</b> to connect to any type of database.</li> <li>• <b>FileSource</b> for flat files. The supported file types are CSV and TSV, JSON, NDJSON, XML, Parquet, and SAS (SAS Transport XPT and SAS7BDAT formats) . The GDI automatically determines the file type from the file extensions. When querying file sources, make sure that the files are accessible to both Anzo and AnzoGraph.</li> <li>• <b>HttpSource</b> to connect to HTTP endpoints.</li> <li>• <b>ElasticSource</b> to connect to Elasticsearch indexes on an Elasticsearch server.</li> <li>• <b>KafkaSource</b> to connect to Kafka streaming sources.</li> </ul> <div> <p><b>Tip</b></p> <p>Certain connection and input parameters are available based on the specified source type. For details about the options for your source, see <a href="#">Source-Specific Properties</a>.</p> </div>
<code>url</code>	string	<p>This property specifies the URL for the data source, such as the database URL, Elasticsearch URL, or HTTP endpoint URL. For file-based sources, the <code>url</code> property specifies the file system location of the source file or directory of files.</p> <div> <p><b>Important</b></p> <p>For security, it is a best practice to reference connection information (such as the url, username, and password) from a Query Context so that the sensitive details are abstracted from any requests. In addition,</p> </div>

Option	Data Type	Description
		<p>using a Query Context makes connection details reusable across queries. See <a href="#">Using Query Contexts in Queries</a> for more information. For example, the triple patterns below reference keys from a Query Context:</p> <pre> ?data a s:DbSource ;   s:url "   { {@db.eca4bfa83481f3638b93ab5fdf93ff9a.url} } " ;   s:username "   { {@db.eca4bfa83481f3638b93ab5fdf93ff9a.user} } " ;   s:password "   { {@db.eca4bfa83481f3638b93ab5fdf93ff9a.password} } " ; </pre>
<b>username</b>	string	If authentication is required to access the source, include this property to specify the user name.
<b>password</b>	string	This property lists the password for the given username.
<b>catalogs</b>	variable	This selector narrows the results to schema-related metadata such as the schema names. Even when additional metadata types ( <a href="#">metadata_type datatype</a> ) are specified as objects, only catalog (schema) information is returned.
<b>fields</b>	variable	This selector is the broadest and most flexible option. Using the <code>fields</code> selector enables users to return any and all of the source metadata information, depending on the specified metadata types ( <a href="#">metadata_type datatype</a> ).
<b>models</b>	variable	This selector narrows the results to model-related metadata such as the model names. Even when additional metadata types ( <a href="#">metadata_type datatype</a> ) are specified as objects, only model information is returned.
<b>metadata_a_type</b>	N/A	The triple patterns in the array for the metadata selector specify the type of metadata to return as well as the data type for the return value. The following list

Option	Data Type	Description
<b>datatype</b>		<p>shows all of the valid options. You can include any combination of properties. The results that are returned depend on the type of data source and whether the information exists in the source. The parentheses around the data type are not required but are included in this document for readability.</p> <ul style="list-style-type: none"> <li>• <b>?model (xsd:string)</b>: Returns model names in string format. For file sources, this property returns file names.</li> <li>• <b>?field (xsd:string)</b>: Returns column names.</li> <li>• <b>?catalog (xsd:string)</b>: Returns schema names.</li> <li>• <b>?datatype (owl:Thing)</b>: Returns the data types of the columns.</li> <li>• <b>?keys (xsd:string)</b>: Returns primary and foreign key columns. For compound keys, the GDI returns a comma-separated list of columns comprising the key.</li> <li>• <b>?format (xsd:string)</b>: Returns the format of the source.</li> <li>• <b>?cardinality (xsd:string)</b>: Returns the cardinality of relationships between tables: optional, many, or required.</li> <li>• <b>?count (xsd:int)</b>: Returns the number of times the field appears in the source.</li> <li>• <b>?order (xsd:int)</b>: Returns the order in which the field was encountered.</li> </ul>

## Metadata Query Examples

This section includes sample metadata queries that run against different types of data sources.

- [List Database Schemas](#)
- [Explore a Database Schema](#)
- [Explore a Directory of SAS Files](#)
- [Explore an HTTP Endpoint](#)
- [Explore a Directory of CSV Files](#)



## List Database Schemas

The query below sends a metadata query to a MySQL database to return a list of the schemas that are available:

```
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>

SELECT *
WHERE
{
    SERVICE <http://cambridgesemantics.com/services/DataToolkit>
    {
        [] s:select ?metadata .

        ?data a s:DbSource ;
            s:url "jdbc:mysql://10.100.2.9:5555/?user=root&password=Mysql1l@#" .

        ?metadata a s:MetadataSource ;
            s:from ?data ;
        ?catalogs [
            ?catalog (xsd:string) ;
            ?order (xsd:int) ;
        ] .
    }
}
ORDER BY ?catalog
```

The query returns the following results:

catalog	order
-----	-----
BANKTEST_DB	1

EMR		4
GOLFCLUB_DB		8
NORTHWIND		10
SPORTDB		13
SQLPOCKET_DB		14
WORDPRESS_DB		16
classicmodels		2
crm_national_patients		3
emrdbbig		5
emrdbsmall		6
emrnational_schema		7
mysql		9
optum		11
performance_schema		12
sys		15
16 rows		

## Explore a Database Schema

Using the list of schemas that were returned in the example above ([List Database Schemas](#)), the query below returns metadata about the columns in one of the schemas. To narrow the results to a schema, the schema name (NORTHWIND) is added to the connection URL.

```

PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>

SELECT *
WHERE
{
    SERVICE <http://cambridgesemantics.com/services/DataToolkit>
    {
        [] s:select ?metadata .
    }
}

```

```

    ?data a s:DbSource ;
        s:url "jdbc:mysql://10.100.2.9:5555/NORTHWIND?user=root&password=Mysql123" ;

    ?metadata a s:MetadataSource ;
        s:from ?data ;

    ?fields [
        ?model (xsd:string) ;
        ?field (xsd:string) ;
        ?datatype (owl:Thing) ;
    ] .
}
}
ORDER BY ?model

```

The query returns the following results:

model	field	datatype
-----	-----	-----
Alphabetical list of products	CategoryID	int
http://www.w3.org/2001/XMLSchema#int		
Alphabetical list of products	Discontinued	boolean
http://www.w3.org/2001/XMLSchema#boolean		
Alphabetical list of products	SupplierID	int
http://www.w3.org/2001/XMLSchema#int		
Alphabetical list of products	UnitPrice	decimal
http://www.w3.org/2001/XMLSchema#decimal		
Alphabetical list of products	ProductName	string
http://www.w3.org/2001/XMLSchema#string		
Alphabetical list of products	QuantityPerUnit	string
http://www.w3.org/2001/XMLSchema#string		
Alphabetical list of products	UnitsOnOrder	short
http://www.w3.org/2001/XMLSchema#short		
Alphabetical list of products	CategoryName	string
http://www.w3.org/2001/XMLSchema#string		
Alphabetical list of products	ProductID	int
http://www.w3.org/2001/XMLSchema#int		
Alphabetical list of products	ReorderLevel	int

```

http://www.w3.org/2001/XMLSchema#short
Alphabetical list of products | UnitsInStock |
http://www.w3.org/2001/XMLSchema#short
Categories | CategoryID |
http://www.w3.org/2001/XMLSchema#int
Categories | Description |
http://www.w3.org/2001/XMLSchema#string
Categories | Picture |
http://www.w3.org/2001/XMLSchema#base64Binary
Categories | CategoryName |
http://www.w3.org/2001/XMLSchema#string
Categories | categoryid |
Category Sales for 1997 | CategoryName |
http://www.w3.org/2001/XMLSchema#string
Category Sales for 1997 | CategorySales |
http://www.w3.org/2001/XMLSchema#double
Current Product List | ProductName |
http://www.w3.org/2001/XMLSchema#string
Current Product List | ProductID |
http://www.w3.org/2001/XMLSchema#int
...
201 rows

```

## Explore a Directory of SAS Files

The query below explores a directory of SAS files to return the model, catalog (schema), field, data type, and cardinality information. The query also orders the results by model name, which is the file name for file sources of a data model does not exist.

```

PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>

```

```

SELECT *
WHERE
{
    SERVICE <http://cambridgesemantics.com/services/DataToolkit>
    {
        [] s:select ?metadata .

        ?data a s:FileSource ;
            s:url "/opt/shared-files/sas" .

        ?metadata a s:MetadataSource ;
            s:from ?data ;

        ?fields [
            ?model (xsd:string) ;
            ?field (xsd:string) ;
            ?catalog (xsd:string) ;
            ?datatype (owl:Thing) ;
            ?cardinality (xsd:string) ;
        ] .
    }
}
ORDER BY ?model

```

The query returns the following results:

model	field	catalog	datatype	cardinality
demand	P1	les/sas	http://www.w3.org/2001/XMLSchema#double	REQUIRED
demand	P2	les/sas	http://www.w3.org/2001/XMLSchema#double	REQUIRED
demand	P3	les/sas	http://www.w3.org/2001/XMLSchema#double	REQUIRED
demand	Y	les/sas	http://www.w3.org/2001/XMLSchema#double	REQUIRED
demand	Q1	les/sas	http://www.w3.org/2001/XMLSchema#double	

```

REQUIRED
demand | Q2          | les/sas | http://www.w3.org/2001/XMLSchema#double |
REQUIRED
demand | Q3          | les/sas | http://www.w3.org/2001/XMLSchema#double |
REQUIRED
demo   | YEAR        | les/sas | http://www.w3.org/2001/XMLSchema#long    |
REQUIRED
demo   | QTR         | les/sas | http://www.w3.org/2001/XMLSchema#long    |
REQUIRED
demo   | GDP         | les/sas | http://www.w3.org/2001/XMLSchema#double  |
REQUIRED
demo   | PR          | les/sas | http://www.w3.org/2001/XMLSchema#double  |
REQUIRED
demo   | M1          | les/sas | http://www.w3.org/2001/XMLSchema#double  |
REQUIRED
demo   | RS          | les/sas | http://www.w3.org/2001/XMLSchema#double  |
REQUIRED
airline | YEAR        | les/sas | http://www.w3.org/2001/XMLSchema#long    |
REQUIRED
airline | Y           | les/sas | http://www.w3.org/2001/XMLSchema#double  |
REQUIRED
airline | W           | les/sas | http://www.w3.org/2001/XMLSchema#double  |
REQUIRED
airline | R           | les/sas | http://www.w3.org/2001/XMLSchema#double  |
REQUIRED
airline | L           | les/sas | http://www.w3.org/2001/XMLSchema#double  |
REQUIRED
airline | K           | les/sas | http://www.w3.org/2001/XMLSchema#double  |
REQUIRED
cars    | MPG         | les/sas | http://www.w3.org/2001/XMLSchema#long    |
REQUIRED
cars    | CYL         | les/sas | http://www.w3.org/2001/XMLSchema#long    |
REQUIRED
...
50 rows

```

## Explore an HTTP Endpoint

The query below explores the metadata for a sample HTTP source that compiles worldwide weather statistics.

```
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>

SELECT *
WHERE
{
    SERVICE <http://cambridgesemantics.com/services/DataToolkit>
    {
        [] s:select ?metadata .

        ?data a s:HttpSource ;
              s:url "https://sampleEndpoint.com/forecast/30.374563,-97.975892" .

        ?metadata a s:MetadataSource ;
                  s:from ?data ;

        ?fields [
            ?model (xsd:string) ;
            ?field (xsd:string) ;
            ?datatype (owl:Thing) ;
            ?cardinality (xsd:string) ;
            ?order (xsd:int) ;
        ] .
    }
}
ORDER BY ?model ?order
```

The query returns the following results:

model	field	datatype
cardinality	order	
-----+-----+-----		
--+-----+-----		
currently	time	http://www.w3.org/2001/XMLSchema#int
REQUIRED	6	
currently	summary	
http://www.w3.org/2001/XMLSchema#string	REQUIRED	7
currently	icon	
http://www.w3.org/2001/XMLSchema#string	REQUIRED	8
currently	nearestStormDistance	http://www.w3.org/2001/XMLSchema#int
REQUIRED	9	
currently	nearestStormBearing	http://www.w3.org/2001/XMLSchema#int
REQUIRED	10	
currently	precipIntensity	http://www.w3.org/2001/XMLSchema#int
REQUIRED	11	
currently	precipProbability	http://www.w3.org/2001/XMLSchema#int
REQUIRED	12	
currently	temperature	http://www.w3.org/2001/XMLSchema#float
REQUIRED	13	
currently	apparentTemperature	http://www.w3.org/2001/XMLSchema#float
REQUIRED	14	
currently	dewPoint	http://www.w3.org/2001/XMLSchema#float
REQUIRED	15	
currently	humidity	http://www.w3.org/2001/XMLSchema#float
REQUIRED	16	
currently	pressure	http://www.w3.org/2001/XMLSchema#float
REQUIRED	17	
currently	windSpeed	http://www.w3.org/2001/XMLSchema#float
REQUIRED	18	
currently	windGust	http://www.w3.org/2001/XMLSchema#float
REQUIRED	19	
currently	windBearing	http://www.w3.org/2001/XMLSchema#int
REQUIRED	20	
currently	cloudCover	http://www.w3.org/2001/XMLSchema#float
REQUIRED	21	
currently	uvIndex	http://www.w3.org/2001/XMLSchema#int
REQUIRED	22	
currently	visibility	http://www.w3.org/2001/XMLSchema#int



REQUIRED		23	
currently	ozone		http://www.w3.org/2001/XMLSchema#float
REQUIRED		24	
daily	summary		
http://www.w3.org/2001/XMLSchema#string	REQUIRED		75
daily	icon		
http://www.w3.org/2001/XMLSchema#string	REQUIRED		76
daily	data		
MANY		77	
data	time		http://www.w3.org/2001/XMLSchema#int
REQUIRED		29	
data	precipIntensity		http://www.w3.org/2001/XMLSchema#float
REQUIRED		30	
data	precipProbability		http://www.w3.org/2001/XMLSchema#float
REQUIRED		31	
data	summary		
http://www.w3.org/2001/XMLSchema#string	OPTIONAL		32
...			
81 rows			

The following query retrieves the model, field, and data type metadata for the United States from the publicly available [Data API Covid Tracking Project](#).

```

PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>

SELECT *
WHERE
{
    SERVICE <http://cambridgesemantics.com/services/DataToolkit>
    {
        [] s:select ?metadata .
    }
}

```

```

?data a s:HttpSource ;
      s:url "https://covidtracking.com/api/v1/us/current.csv" .

?metadata a s:MetadataSource ;
      s:from ?data ;

?fields [
      ?model (xsd:string) ;
      ?field (xsd:string) ;
      ?datatype (owl:Thing) ;
] .
}
}

```

The query returns the following results:

model	field	datatype
us	date	http://www.w3.org/2001/XMLSchema#string
us	states	http://www.w3.org/2001/XMLSchema#string
us	positive	http://www.w3.org/2001/XMLSchema#string
us	negative	http://www.w3.org/2001/XMLSchema#string
us	pending	http://www.w3.org/2001/XMLSchema#string
us	hospitalizedCurrently	http://www.w3.org/2001/XMLSchema#string
us	hospitalizedCumulative	http://www.w3.org/2001/XMLSchema#string
us	inIcuCurrently	http://www.w3.org/2001/XMLSchema#string
us	inIcuCumulative	http://www.w3.org/2001/XMLSchema#string
us	onVentilatorCurrently	http://www.w3.org/2001/XMLSchema#string
us	onVentilatorCumulative	http://www.w3.org/2001/XMLSchema#string
us	recovered	http://www.w3.org/2001/XMLSchema#string
us	dateChecked	http://www.w3.org/2001/XMLSchema#string
us	death	http://www.w3.org/2001/XMLSchema#string
us	hospitalized	http://www.w3.org/2001/XMLSchema#string
us	lastModified	http://www.w3.org/2001/XMLSchema#string
us	total	http://www.w3.org/2001/XMLSchema#string
us	totalTestResults	http://www.w3.org/2001/XMLSchema#string
us	posNeg	http://www.w3.org/2001/XMLSchema#string
us	deathIncrease	http://www.w3.org/2001/XMLSchema#string
us	hospitalizedIncrease	http://www.w3.org/2001/XMLSchema#string

```

us      | negativeIncrease      | http://www.w3.org/2001/XMLSchema#string
us      | positiveIncrease       | http://www.w3.org/2001/XMLSchema#string
us      | totalTestResultsIncrease | http://www.w3.org/2001/XMLSchema#string
us      | hash                   | http://www.w3.org/2001/XMLSchema#string
25 rows

```

## Explore a Directory of CSV Files

The query below explores a directory of CSV files to return the model, field, and data type. The query also orders the results by model name, which is the file name for file sources of a data model does not exist. In addition, the query includes `s:sampling true`, which means the GDI will scan the entire file or files before returning results.

```

PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>

SELECT *
WHERE
{
    SERVICE <http://cambridgesemantics.com/services/DataToolkit>
    {
        [] s:select ?metadata .

        ?data a s:FileSource ;
              s:url "/opt/shared-files/movie-csv" .

        ?metadata a s:MetadataSource ;
              s:from ?data ;

        # Sample the whole file
        s:sampling true ;
    }
}

```

```

# Sample the first N records #
# s:sampling 1000 ;

?fields [
    ?model (xsd:string) ;
    ?field (xsd:string) ;
    ?datatype (owl:Thing) ;
] .
}
}
ORDER BY ?model

```

The query returns the following results:

model	field	datatype
-----	-----	-----
MovieActors1	MovieID	
http://www.w3.org/2001/XMLSchema#int		
MovieActors1	MovieTitle	
http://www.w3.org/2001/XMLSchema#string		
MovieActors1	ActorID	
http://www.w3.org/2001/XMLSchema#int		
MovieActors1	ActorName	
http://www.w3.org/2001/XMLSchema#string		
MovieActors2	MovieID	
http://www.w3.org/2001/XMLSchema#int		
MovieActors2	MovieTitle	
http://www.w3.org/2001/XMLSchema#string		
MovieActors2	ActorID	
http://www.w3.org/2001/XMLSchema#int		
MovieActors2	ActorName	
http://www.w3.org/2001/XMLSchema#string		
MovieActors2	ActorCategory	
http://www.w3.org/2001/XMLSchema#string		
MovieCategory	MovieID	
http://www.w3.org/2001/XMLSchema#int		
MovieCategory	MovieTitle	
http://www.w3.org/2001/XMLSchema#string		

MovieCategory	MoveCategoryID	
http://www.w3.org/2001/XMLSchema#int		
MovieCategory	MovieCategory	
http://www.w3.org/2001/XMLSchema#string		
MovieCinematographers	MovieID	
http://www.w3.org/2001/XMLSchema#int		
MovieCinematographers	MovieTitle	
http://www.w3.org/2001/XMLSchema#string		
MovieCinematographers	MovieCinematographerID	
http://www.w3.org/2001/XMLSchema#int		
MovieCinematographers	MovieCinematographerName	
http://www.w3.org/2001/XMLSchema#string		
MovieComposers	MovieID	
http://www.w3.org/2001/XMLSchema#int		
MovieComposers	MovieTitle	
http://www.w3.org/2001/XMLSchema#string		
MovieComposers	MovieComposerID	
http://www.w3.org/2001/XMLSchema#int		
MovieComposers	MovieComposerName	
http://www.w3.org/2001/XMLSchema#string		
MovieDirectors	MovieID	
http://www.w3.org/2001/XMLSchema#int		
MovieDirectors	MovieTitle	
http://www.w3.org/2001/XMLSchema#string		
...		
79 rows		

The following example shows a query that returns metadata for an Elasticsearch source.

```

PREFIX s:    <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX anzo:  <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:  <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:    <http://purl.org/dc/elements/1.1/>
PREFIX ex:    <http://example.org/ontologies/City#>
PREFIX es:    <http://elastic.co/search/>

```

```

PREFIX :      <http://example.org/cities/>

SELECT *
WHERE {
    SERVICE <http://cambridgesemantics.com/services/DataToolkit>
    {
        [] s:select ?_fields .

        ?data a es:ElasticSource ;
            es:url "http://localhost:9200/" ;
            es:index "account" ;
            ?account_number xsd:long ;
            ?age (xsd:long) ;
            ?balance (xsd:long) ;
            ?address (xsd:string) ;
            ?city (xsd:string) ;
            ?state (xsd:string) ;
            ?email (xsd:string) ;
            ?employer (xsd:string) ;
            ?firstname (xsd:string) ;
            ?lastname (xsd:string) ;
            ?gender (xsd:string) .

        ?_fields a s:MetadataSource ;
            s:from ?data ;
            ?fields [
                ?catalog () ;
                ?model () ;
                ?field () ;
                ?cardinality () ;
                ?datatype () ;
                ?type () ;
                ?object () ;
            ] .
    }
}
ORDER BY ?catalog ?model ?field

```

For instructions on querying the instance data based on the data source metadata, see [Getting Started with GDI Queries](#).

## GDI Usage by Data Source Type

The topics in this section provide more advanced GDI usage information by including descriptions for all of the query options for each type of supported data source.

### Tip

Rather than manually writing complex queries, you can use the GDI to automatically generate graphs and ontologies by including a few key statements in a relatively simple query. For information, see [Generating a Graph and Ontology with a Direct Load Step](#).

- [Querying a Database Source](#)
- [Querying an HTTP Source](#)
- [Querying an Elasticsearch Source](#)
- [Querying File Sources](#)

## Querying a Database Source

This topic provides details about the structure to use when writing GDI queries to read or ingest data from database data sources. It also includes example queries that may be useful as a starting point for writing your own GDI queries.

- [Supported Databases](#)
- [Query Syntax](#)
- [Query Examples](#)

## Supported Databases

The GDI supports querying any database through a JDBC connection. AnzoGraph installations include JDBC drivers for the following databases:

- Databricks
- H2
- IBM DB2
- Microsoft SQL Server
- MariaDB
- Oracle

- PostgreSQL
- SAP Sybase (jTDS)
- Snowflake

To extend the service to access other databases, additional JDBC drivers can be added to AnzoGraph. For information about acquiring additional JDBC drivers, contact your Cambridge Semantics Customer Success manager. For instructions on deploying other drivers, see [Deploy Optional Drivers for Accessing Custom Database Sources](#) in the Deployment Guide.

## Query Syntax

The following query syntax shows the structure of a GDI query for database sources. The clauses, patterns, and placeholders that are links are described below.

```
# PREFIX Clause
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>

# Result Clause
{
  [ GRAPH ${targetGraph} { ]
    triple_patterns
  [ } ]
}
[ ${usingSources} ]

WHERE
{
  # SERVICE Clause: Include the following service call when reading or
  inserting data.
    SERVICE [ TOPDOWN ] <http://cambridgesemantics.com/services/DataToolkit>

  # View SERVICE Clause: Or use the service call below when constructing a
  view.
```



```

SERVICE <http://cambridgesemantics.com/services/DataToolkitView>
({targetGraph})

{
  ?data a s:DbSource ;
    s:url "string" ;
    s:username "string" ;
    s:password "string" ;
    [ s:token "string" ; ]
    [ s:driver "string" ; ]
    [ s:property [ s:name "string" ; s:value "string" ] ; ]
    [ s:timeout int ; ]
    [ s:maxConnections int ; ]
    [ s:batching boolean | int ; ]
    [ s:concurrency int | [ list_of_properties ] ; ]
    [ s:rate int | "string" ; ]
    [ s:partitionBy "string" | ?variable | s:auto ; ]
    [ s:paging [ pagination_options ; ]
    [ s:locale "string" ; ]
    [ s:sampling int ; ]
    [ s:selector "string" | [ list ] ; ]
    [ s:model "string" ; ]
    [ s:key ("string") ; ]
    [ s:reference [ s:model "string" ; s:using ("string") ]
    [ s:formats [ datatype_formatting_options ] ; ]
    [ s:normalize boolean | [ normalization_rules ] ; ]
    [ s:query "string" ; ]
    [ s:database "string" ; ]
    [ s:schema "string" ; ]
    [ s:table "string" ; ]
    [ s:count ?variable ; ]
    [ s:offset int ; ]
    [ s:orderBy "string" | ?variable ; ]
    [ s:limit int ; ]
    # Mapping variables
    ?mapping_variable ( [ "binding" ] [ datatype ] [ "datetime_format" ] ) ;
    ... ;
    .
  # Additional clauses such as BIND, VALUES, FILTER

```

```
}
}
```

### Note

For readability, the parameters below exclude the base URI

<http://cambridgesemantics.com/ontologies/DataToolkit#> as well as the `s:` prefix.

As shown in the examples, however, the `s:` prefix or full property URI does need to be included in queries.

Option	Data Type	Description
<b>PREFIX Clause</b>	N/A	<p>The PREFIX clause declares the standard and custom prefixes for GDI service queries. Generally, queries include the following prefixes (or a subset of them) plus any data-specific declarations:</p> <pre> PREFIX s: &lt;http://cambridgesemantics.com/ontologies/DataToolkit#&gt; PREFIX rdf:    &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX rdfs:   &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX xsd:    &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX owl:  &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX anzo: &lt;http://openanzo.org/ontologies/2008/07/Anzo#&gt; PREFIX zowl: &lt;http://openanzo.org/ontologies/2009/05/AnzoOwl#&gt; PREFIX dc:     &lt;http://purl.org/dc/elements/1.1/&gt; </pre>
<b>Result Clause</b>	N/A	<p>The result clause defines the type of SPARQL query to run and the set of results to return, i.e., whether you want to read (SELECT or CONSTRUCT) from the source or ingest the data into Anzo (INSERT).</p>
<b>GRAPH \${targetGraph}</b>	N/A	<p>Include the GRAPH keyword and target graph parameter <code>\${targetGraph}</code> when writing an INSERT query to ingest data into a graphmart. Anzo automatically populates the query with the appropriate target URIs when the query runs.</p>

Option	Data Type	Description
<b><code>\${usingSources}</code></b>	N/A	Include the source graph parameter <code>\${usingSources}</code> when writing a "topdown" query that passes values from the data that is in the graphmart to the data source. Anzo automatically populates the query with the appropriate FROM clauses when the query runs. When passing literal values to the remote source, you do not need to include the source graph parameter. The <a href="#">SERVICE Clause</a> description below includes more information about passing input to data sources.
<b>SERVICE Clause</b>	N/A	<p>Include the SERVICE call <code>SERVICE [ TOPDOWN ]</code></p> <p><code>&lt;http://cambridgesemantics.com/services/DataToolkit&gt;</code> to invoke the GDI service when you are running a SELECT, INSERT, or CONSTRUCT query that is not creating a view. When writing a CONSTRUCT query in a View Step, use the <code>DataToolkitView</code> service call, as described below in <a href="#">View SERVICE Clause</a>.</p> <p>Include the optional TOPDOWN keyword when you want to pass input values from the graphmart to the data source. When you include TOPDOWN in the service call, it indicates that the rest of the query produces values to send to the source. In this case, the GDI makes repeated calls to pass in each of the specified values and retrieve the data that is based on those values.</p>
<b>View SERVICE Clause</b>	N/A	<p>When writing a CONSTRUCT query that creates a view of the data (usually in a View Step), include the following SERVICE call: <code>SERVICE</code></p> <p><code>&lt;http://cambridgesemantics.com/services/DataToolkitView&gt;</code> (<code>\${targetGraph}</code>). Using the <code>DataToolkitView</code> call optimizes query execution because it tells the GDI to inspect the query and determine which filters to push to the data source. It also limits the result set and retrieves only the data that is needed, i.e., the source data is fully mapped but all of the mapped data is not necessarily returned.</p>
<b>url</b>	string	This property specifies the URL to use to access the database.

Option	Data Type	Description
		<p><b>Important</b></p> <p>For security, it is a best practice to reference connection information (such as the url, username, and password) from a Query Context so that the sensitive details are abstracted from any requests. In addition, using a Query Context makes connection details reusable across queries. See <a href="#">Using Query Contexts in Queries</a> for more information. For example, the triple patterns below reference a Query Context and add a JDBC driver level connection property:</p> <pre>?data a s:DbSource ;     s:url "     {{@db.eca4bfa83481f3638b93ab5fdf93ff9a.url}}" ;     s:username "     {{@db.eca4bfa83481f3638b93ab5fdf93ff9a.user}}" ;     s:password "     {{@db.eca4bfa83481f3638b93ab5fdf93ff9a.passwor     d}}" ;     s:property [ s:name "access" ; s:value "all" ]     ;</pre>
<b>username</b>	string	<p>This property lists the user name to use for the connection to the database.</p> <p><b>Tip</b></p> <p>If you want to group the username and password properties, you can wrap them with <code>s:credentials [ ]</code>. For example:</p> <pre>s:credentials [     s:username "username" ;     s:password "password" ; ] ;</pre>
<b>password</b>	string	This property lists the password for the given username.
<b>token</b>	string	For connections that require a bearer token, this property can be included to

Option	Data Type	Description
		specify the token.
<b>driver</b>	string	This property can be included to specify the JDBC driver to use.
<b>property</b>	RDF list	<p>This property can be included to list any JDBC driver-specific connection properties. To incorporate <code>property</code>, use the following syntax:</p> <pre>s:property [ s:name "custom_driver_property_name" ; s:value "custom_value" ]</pre>
<b>timeout</b>	int	This property can be used to specify the timeout (in milliseconds) to use for requests against the source. For example, <code>s:timeout 5000</code> configures a 5 second timeout.
<b>maxConnections</b>	int	This property can be used to set a limit on the maximum number of active connections to the source. For example, <code>s:maxConnections 16</code> sets the limit to 16 connections. The default value is 10.
<b>batching</b>	boolean or int	This property can be used to disable batching, or it can be used to change the default the batch size. By default, batching is set to 5000 ( <code>s:batching 5000</code> ). To disable batching, you can include <code>s:batching false</code> in the query. Typically users do not change the batching size. However, it can be useful to control the batch size when performing updates. To configure the size, include <code>s:batching int</code> in the query. For example, <code>s:batching 3000</code> .
<b>concurrency</b>	int or RDF list	This property can be included to configure the maximum level of concurrency for the query. The value can be an integer, such as <code>s:concurrency 8</code> . If the value is an integer, it configures a maximum limit on the number of slices that can execute the query. For finer-grained control over the number of nodes and slices to use, concurrency can also be included as an object with <code>limit</code> , <code>nodes</code> , and/or <code>executorsPerNode</code> properties. For example, the following object configures a concurrency model that allows a maximum of 24 executors

Option	Data Type	Description
		<p>distributed across 4 nodes with 8 executors per node:</p> <pre>s:concurrency [   s:limit 24 ;   s:nodes 4 ;   s:executorsPerNode 8 ; ] ;</pre>
<b>rate</b>	int or string	<p>This property can be included to control the frequency with which a request is sent to the source. The limit applies to the number of requests a single slice can make. If you specify an integer for the rate, then the value is treated as the maximum number of requests to issue per minute. If you specify a string, you have more flexibility in configuring the rate. The sample values below show the types of values that are supported:</p> <pre>s:rate "90/minute" ; s:rate "90 per minute" ; s:rate "200000 every week" ; s:rate "10000 every 6 hours" ;</pre> <p>To enforce the rate limit, the GDI introduces a sleep between requests that is equal to the rate delay. The more executing slices, the longer the rate delay needs to be to enforce the limit in aggregate.</p> <p>Given the example of <code>s:rate "90/minute"</code>, the GDI would optimize the concurrency and only use 1 slice for execution with a rate delay of 666ms between requests. If <code>s:rate "240/minute"</code>, the GDI would use 3 executors with a rate delay of 750ms between requests.</p>
<b>partitionBy</b>	string, variable, object	<p>The GDI attempts to partition queries automatically across the available cores (slices) in AnzoGraph. To determine how to partition the query, the GDI uses metadata from the source database. It looks for any column in an index, preferring the primary key column if it is interpolable. However, it only considers the first column in any index on the table. After determining the</p>

Option	Data Type	Description
		partition column, the GDI does a MIN/MAX on the column as well as a basic sizing query. To specify which column or columns the GDI should partition on, you can include the <code>partitionBy</code> property in the query. The property supports a list of source field names, bound variables, or the object <code>s:auto</code> , which forces the GDI to partition the data when the source does not define partitioning metadata.
<b>paging</b>	RDF list	This property can be used to configure paging so that the GDI can access large amounts of data across a number of smaller requests. For details about the <code>paging</code> property, see <a href="#">Pagination Options</a> .
<b>locale</b>	string	This property can be used to specify the locale to use when parsing locale-dependent data such as numbers, dates, and times.
<b>sampling</b>	int	This property can be used to configure the number of records in the source to examine for data type inferencing.
<b>selector</b>	string or RDF list	This property can be used as a binding component to identify the path to the source objects. For example, <code>s:selector "Sales.SalesOrderHeader"</code> targets the SalesOrderHeader table in the Sales schema. As an alternative to including the selector property for identifying the target data, you could use the <a href="#">database</a> , <a href="#">schema</a> , and/or <a href="#">table</a> properties.
<b>model</b>	string	This property defines the class (or table) name for the type of data that is generated from the specified data source. For example, <code>s:model "employees"</code> . Model is optional when querying a single source. If your query targets multiple sources, however, and you want to define resource templates (primary keys) and object properties (foreign keys), you must specify the model value for each source.
<b>key</b>	string	This property can be used to define the primary key column for the source file or table. This column is leveraged in a resource template for the instances that are created from the source. For example, <code>s:key ("EMPLOYEE_ID")</code> . For

Option	Data Type	Description
		more information about <code>key</code> , see <a href="#">Data Linking Options</a> .
<b>reference</b>	RDF list	This property can be used to specify a foreign key column. The reference property is an RDF list that includes the <code>model</code> property to list the target table and a <code>using</code> property that defines the foreign key column. For more information about <code>reference</code> , see <a href="#">Data Linking Options</a> .
<b>formats</b>	RDF list	To give users control over the data types that are used when coercing strings to other types, this property can be included in GDI queries to define the desired types. In addition, it can be used to describe the formats of date and time values in the source to ensure that they are recognized and parsed to the appropriate date, time, and/or <code>dateTime</code> values. For details about the <code>formats</code> property, see <a href="#">Data Type Formatting Options</a> .
<b>normalize</b>	RDF list	To give users control over the labels and URIs that are generated, the GDI offers several options for normalizing the model and/or the fields that are created from the specified data source(s). For details about the <code>normalize</code> property, see <a href="#">Normalization Options</a> .
<b>query</b>	string	<p>If you want to access the source data by running an SQL query, you can include this property to specify the query string to run. The language does not have to be SQL if the source supports another language. However, some GDI features where the query is dynamically altered may not work with a non-SQL language. Including <code>{{?variable}}</code> substitutions is supported within <code>s:query</code> strings.</p> <div> <p><b>Note</b></p> <p>If you include <code>s:query</code> without also specifying <code>table</code> or <code>partitionBy</code>, the GDI may not partition the query and query execution may be less performant than if the partition column was specified. When using <code>s:query</code>, specifying the table in <code>s:table</code> and the column to partition the table on in <code>s:partitionBy</code> is a good practice, especially when querying large tables.</p> </div>



Option	Data Type	Description
<b>database</b>	string	This property can be used to specify the database to target in the source if the database is not listed in the <code>s:url</code> or <code>s:selector</code> strings.
<b>schema</b>	string	This property can be included to specify the target schema to query. If you include <code>s:schema "schema_name"</code> without specifying <code>s:table</code> (described below) or <code>s:query</code> , all tables in the schema are queried.
<b>table</b>	string	This property can be included to specify the target table or tables for the query.
<b>count</b>	variable	If you want to turn the query into a COUNT query, you can include this property with a <code>?variable</code> to perform a count. For example, <code>s:count ?count</code> .
<b>offset</b>	int	This property can be used to offset the data that is returned by a number of rows.
<b>orderBy</b>	string, variable, list	You can include this property to order the result set by a field name, a bound variable, or a list of names or bound variables.
<b>limit</b>	int	You can include this property to limit the number of results that are returned. <code>s:limit</code> maps to the SPARQL LIMIT clause.
<b>mapping_variable</b>	variable	<p>The mapping variables, in <code>?variable (["binding"] [datatype] ["datetime_format"])</code> format, define the triple patterns to output. When the specified <code>?variable</code> matches the source column name, the GDI uses the variable as the source data selector. If you specify an alternate variable name, a <a href="#">binding</a> needs to be specified to map the new variable to the source. You also have the option to transform the data using the <a href="#">datatype</a> and <a href="#">datetime_format</a> options.</p> <div> <p><b>Note</b></p> <p>The parentheses around the binding, data type, and format specifications are not required but are included in this document for</p> </div>

Option	Data Type	Description
		<div>readability.</div>
<b>binding</b>	string	<p>The <b>binding</b> is a literal value that binds a ?variable to a source column. If you specify a ?variable that matches the source column name, then that variable name is the data selector and it is not necessary to specify a binding. If you specify an alternate variable name or there is a hierarchical path to the source column that is not already identified by the <a href="#">selector</a>, <a href="#">database</a>, <a href="#">schema</a>, <a href="#">table</a>, or <a href="#">query</a> properties, then the binding is needed to map the new variable to that source column. For example, <code>?subject ("dbo.FILM.SUBJECT")</code> binds the ?subject variable by navigating to the SUBJECT column in the FILM table in the dbo schema.</p> <div> <p><b>Note</b></p> <p>Database, schema, and table names in bindings are parsed according to the specific rules for that database type. You do not need to escape characters in database names. However, database names with characters that do not match <code>(_ A-Z a-z)(_ A-Z a-z 0-9)*</code> should be quoted, such as <code>("'Adventure.Works'.Sales.'Daily.Totals'")</code>.</p> </div>
<b>datatype</b>	URI	<p>The <code>datatype</code> is the data type to convert the column to. If you do not specify a data type, the GDI infers the type. The GDI supports the following types:</p> <ul style="list-style-type: none"> <li>• xsd:int</li> <li>• xsd:long</li> <li>• xsd:float</li> <li>• xsd:double</li> <li>• xsd:boolean</li> <li>• xsd:time</li> <li>• xsd:dateTime</li> <li>• xsd:date</li> <li>• xsd:gMonthDay</li> <li>• xsd:gYearMonth</li> <li>• xsd:duration</li> <li>• xsd:dayTimeDuration</li> <li>• xsd:yearMonthDuration</li> <li>• xsd:gMonth</li> <li>• xsd:anyURI</li> </ul>

Option	Data Type	Description
<b>datetime_format</b>	string	<p>This option is used to specify the format to use for date and time data types. The GDI supports Java date and time formats. Specify days as "d," months as "M," and years as "y." For the time, specify "H" for hours, "m" for minutes, and "s" for seconds. For example, "yyyyMMdd HH:mm:ss" or "ddMMMy" to display date values such as "01JAN19."</p> <div> <p><b>Note</b></p> <p>The GDI's default base year is <b>2000</b>. If the source data has years with only two digits, such as 02-04-99, the GDI prepends <b>20</b> to the digits. The value 02-04-99 is parsed to 02-04-2099. To specify an alternate base year to use for two-digit values, you can include the notation ^nnnn (e.g., ^1900) in the format value. For example, to set the base year to 1900 instead of 2000, use a format value such as xsd:date "dd-MMM-yy^1900" or xsd:date "dd-MMM-yy^1990". When one of those values is specified, 02-04-99 is parsed to 02-04-1999.</p> </div>

## Query Examples

The example below selects data from the AdventureWorks2012 database. The `s:selector` property is used to specify the table (`salesOrderHeader` in the `Sales` schema) to target.

```
PREFIX s:    <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT (COUNT(*) as ?count)
WHERE
{
    SERVICE <http://cambridgesemantics.com/services/DataToolkit>
    {
        ?SalesOrderHeader a s:DbSource ;
        s:url "{@db.eca4bfa83481f3638b93ab5fdf93ff9a.url}" ;
        s:username "{@db.eca4bfa83481f3638b93ab5fdf93ff9a.user}" ;
        s:password "{@db.eca4bfa83481f3638b93ab5fdf93ff9a.password}" ;
        s:selector "Sales.SalesOrderHeader" ;
    }
}
```

```

        ?SalesOrderID (xsd:int) ;
        ?RevisionNumber (xsd:int) ;
        ?OrderDate (xsd:dateTime) ;
        ?DueDate (xsd:dateTime) ;
        ?TerritoryID (xsd:int) ;
        ?TotalDue (xsd:decimal) .
    FILTER(?TerritoryID IN (1, 2, 3))
    FILTER(?TotalDue < 11.0 || ?TotalDue > 250)
}
}

```

The example below ingests data from a database. To define the data to target, the query includes the `s:query` property to run an SQL query. The `s:table` and `partitionBy` properties are also included to aid the GDI in partitioning the query.

```

PREFIX s:    <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
PREFIX ont:
<http://cambridgesemantics.com/Layer/2f1e926b130a402db6fc10fa54199d49/Model#>

INSERT {
  GRAPH ${targetGraph} {
    ?resource a ont:EmrPatient ;
    ont:EmrPatient.patientid ?PATIENTID ;
    ont:EmrPatient.gender ?GENDER ;
    ont:EmrPatient.language ?LANGUAGE ;
    ont:EmrPatient.patientfirstdocactivitydate ?PATIENTFIRSTDOCACTIVITYDATE .
  }
}
WHERE {
  SERVICE <http://cambridgesemantics.com/services/DataToolkit> {
    ?data a s:DbSource ;
    s:url "{{@db.eca4bfa83481f3638b93ab5fdf93ff9a.url}}" ;
    s:username "{{@db.eca4bfa83481f3638b93ab5fdf93ff9a.user}}" ;
    s:password "{{@db.eca4bfa83481f3638b93ab5fdf93ff9a.password}}" ;
    s:query "select * from emrdbsmall.emr_patient where emr_patient.PATIENTID
< 500" ;
    s:partitionBy "PATIENTID" ;
    s:table "emrdbsmall.emr_patient" ;

```

```

    ?PATIENTID (xsd:int) ;
    ?GENDER (xsd:string) ;
    ?LANGUAGE (xsd:string) ;
    ?PATIENTFIRSTDOCACTIVITYDATE (xsd:dateTime "M/d/yyyy HH:mm:ss") .

    BIND(IRI
("http://cambridgesemantics.com/Layer/2f1e926b130a402db6fc10fa54199d49/
{{?PATIENTID}}") AS ?resource)
  }
}

```

## Related Topics

[Generating a Graph and Ontology with a Direct Load Step](#)

## Querying an HTTP Source

This topic provides details about the structure to use when writing GDI queries to read or ingest data from HTTP data sources. It also includes example queries that may be useful as a starting point for writing your own GDI queries.

- [Query Syntax](#)
- [Mapping the Content Property to JSON](#)
- [Query Examples](#)

## Query Syntax

The following query syntax shows the structure of a GDI query for HTTP sources. The clauses, patterns, and placeholders that are links are described below.

```

# PREFIX Clause
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>

```

```

# Result Clause
{
  [ GRAPH ${targetGraph} { ]
  triple_patterns
  [ ] ]
}
[ ${usingSources} ]

WHERE
{
  # SERVICE Clause: Include the following service call when reading or
  inserting data.
  SERVICE [ TOPDOWN ] <http://cambridgesemantics.com/services/DataToolkit>

  # View SERVICE Clause: Or use the service call below when constructing a
  view.
  SERVICE <http://cambridgesemantics.com/services/DataToolkitView>
  (${targetGraph})

  {
    ?data a s:HttpSource ;
    s:url "string" ;
    [ s:authorization [
      a s:BearerToken ; s:token "string" ;
      | a s:AWSSignature ; s:accessKey "string" ; s:region "string" ;
        s:secretKey "string" ; s:serviceName "string" ;
        s:sessionToken "string" ;
      | a s:BasicAuth ; s:username "string" ; s:password "string" ;
    ] ; ]
    [ s:trust "string" ; ]
    [ s:proxy "string" | [ s:host "string" ; s:port int ] ]
    [ s:header [ s:name: "string" ; s:value "string" ] ; ]
    [ s:mimetype "string" ; ]
    [ s:contentType "string" ; ]
    [ s:content ""string"" ; ]
    [ s:parameter [ s:name "string" ; s:value "string" ] ; ]
    [ s:method "string" ; ]
    [ s:encoding "string" ; ]
    [ s:form [ s:name: "string" ; s:value "string" ] ; ]
  }
}

```

```

[ s:format [ source_format_options ; ] ; ]
[ s:timeout int ; ]
[ s:batching boolean | int ; ]
[ s:paging [ pagination_options ; ]
[ s:concurrency int | [ list_of_properties ] ; ]
[ s:rate int | "string" ; ]
[ s:partitionBy "string" | ?variable ; ]
[ s:locale "string" ; ]
[ s:sampling int ; ]
[ s:selector "string" | [ list ] ; ]
[ s:model "string" ; ]
[ s:key ("string") ; ]
[ s:reference [ s:model "string" ; s:using ("string") ]
[ s:formats [ datatype_formatting_options ] ; ]
[ s:normalize boolean | [ normalization_rules ] ; ]
[ s:count ?variable ; ]
[ s:offset int ; ]
[ s:orderBy "string" | ?variable ; ]
[ s:limit int ; ]
# Mapping variables
?mapping_variable ( [ "binding" ] [ datatype ] [ "datetime_format" ] ) ;
... ;
.
# Additional clauses such as BIND, VALUES, FILTER
}
}

```

### Note

For readability, the parameters below exclude the base URI

<<http://cambridgesemantics.com/ontologies/DataToolkit#>> as well as the `s:` prefix.

As shown in the examples, however, the `s:` prefix or full property URI does need to be included in queries.

Option	Data Type	Description
<b>PREFIX Clause</b>	N/A	The PREFIX clause declares the standard and custom prefixes for GDI service queries. Generally, queries include the following prefixes (or a subset of them) plus

Option	Data Type	Description
		<p>any data-specific declarations:</p> <pre> PREFIX s: &lt;http://cambridgesemantics.com/ontologies/DataToolkit#&gt; PREFIX rdf:    &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX rdfs:   &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX xsd:    &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX owl:  &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX anzo: &lt;http://openanzo.org/ontologies/2008/07/Anzo#&gt; PREFIX zowl: &lt;http://openanzo.org/ontologies/2009/05/AnzoOwl#&gt; PREFIX dc:     &lt;http://purl.org/dc/elements/1.1/&gt; </pre>
<b>Result Clause</b>	N/A	The result clause defines the type of SPARQL query to run and the set of results to return, i.e., whether you want to read (SELECT or CONSTRUCT) from the source or ingest the data into Anzo (INSERT).
<b>GRAPH \${targetGraph}</b>	N/A	Include the GRAPH keyword and target graph parameter <code>\${targetGraph}</code> when writing an INSERT query to ingest data into a graphmart. Anzo automatically populates the query with the appropriate target URIs when the query runs.
<b>\${usingSources}</b>	N/A	Include the source graph parameter <code>\${usingSources}</code> when writing a "topdown" query that passes values from the data that is in the graphmart to the data source. Anzo automatically populates the query with the appropriate FROM clauses when the query runs. When passing literal values to the remote source, you do not need to include the source graph parameter. The <a href="#">SERVICE Clause</a> description below includes more information about passing input to data sources.
<b>SERVICE Clause</b>	N/A	Include the SERVICE call <code>SERVICE [ TOPDOWN ]</code> <code>&lt;http://cambridgesemantics.com/services/DataToolkit&gt;</code> to invoke the GDI service when you are running a SELECT, INSERT, or CONSTRUCT query that is not creating a view. When writing a CONSTRUCT query in a View Step, use



Option	Data Type	Description
		<p>the <code>DataToolkitView</code> service call, as described below in <a href="#">View SERVICE Clause</a>.</p> <p>Include the optional TOPDOWN keyword when you want to pass input values from the graphmart to the data source. When you include TOPDOWN in the service call, it indicates that the rest of the query produces values to send to the source. In this case, the GDI makes repeated calls to pass in each of the specified values and retrieve the data that is based on those values.</p>
<b>View SERVICE Clause</b>	N/A	<p>When writing a CONSTRUCT query that creates a view of the data (usually in a View Step), include the following SERVICE call: <code>SERVICE</code></p> <pre>&lt;http://cambridgesemantics.com/services/DataToolkitView&gt; (\${targetGraph}).</pre> <p>Using the <code>DataToolkitView</code> call optimizes query execution because it tells the GDI to inspect the query and determine which filters to push to the data source. It also limits the result set and retrieves only the data that is needed, i.e., the source data is fully mapped but all of the mapped data is not necessarily returned.</p>
<b>url</b>	string	<p>This property specifies the URL to use to access the source. Query binding variables can be inserted into the url string by surrounding the variable name with double curly braces. For example, "<code>{{?name}}</code>".</p> <div> <p><b>Important</b></p> <p>For security, it is a best practice to reference connection information (such as the url, username, and password) from a Query Context so that the sensitive details are abstracted from any requests. In addition, using a Query Context makes connection details reusable across queries. See <a href="#">Using Query Contexts in Queries</a> for more information. For example:</p> <pre>?data a s:HttpSource ;   s:url "   {{@db.eca4bfa83481f3638b93ab5fdf93ff9a.url}}" ;   s:authorization [ a s:BasicAuth ;     s:username "     {{@db.eca4bfa83481f3638b93ab5fdf93ff9a.user}}" ;</pre> </div>

Option	Data Type	Description
		<pre>s:password " {{@db.eca4bfa83481f3638b93ab5fdf93ff9a.password}}" ; ] ;</pre>
<b>authorization</b>	RDF list	<p>This property specifies the type of authorization to use and the values for authentication. The options are <a href="#">BearerToken</a>, <a href="#">AWSSignature</a>, or <a href="#">BasicAuth</a>.</p> <pre>s:authorization [ a s:BearerToken   s:AWSSignature   s:BasicAuth ] ;</pre>
<b>BearerToken</b>	string	<p>Specify this property when a bearer token is used for authentication, and include the token property.</p> <pre>s:authorization [ a s:BearerToken ; s:token "string" ] ;</pre>
<b>AWSSignature</b>	RDF list	<p>For authorization to AWS service endpoints, specify this property and include the appropriate authentication properties from the list below:</p> <ul style="list-style-type: none"> <li>• <b>accessKey</b>: Include this property to specify the AWS access key.</li> <li>• <b>region</b>: Include this property to specify the AWS region.</li> <li>• <b>secretKey</b>: Include this property to specify the AWS secret key.</li> <li>• <b>serviceName</b>: Include this property to specify the AWS service name.</li> <li>• <b>sessionToken</b>: Include this property to specify the AWS session token.</li> </ul> <pre>s:authorization [ a s:AWSSignature ; s:accessKey "string" ; s:region "string" ; s:secretKey "string" ; s:serviceName "string" ; s:sessionToken "string" ; ] ;</pre>
<b>BasicAuth</b>	RDF	Specify this property when basic authentication is used, and include the username

Option	Data Type	Description
	list	<p>and password properties.</p> <pre>s:authorization [ a s:BasicAuth ;   s:username "string" ;   s:password "string" ; ] ;</pre>
<b>trust</b>	string	Include this property to set the level of trust for the source's SSL certificate. The value can be either "system" or "all".
<b>proxy</b>	string or RDF list	Include this property to specify proxy information if a proxy is used. The value can be a string, such as <code>s:proxy "host_url:port_number"</code> , or an RDF list that includes <code>host</code> and <code>port</code> properties, such as <code>s:proxy [ s:host "host_url" ; s:port port_number ]</code> .
<b>header</b>	RDF list	<p>You can use this property to specify name-value pairs to include as headers in the request. For example:</p> <pre>s:header [ s:name "Accept" ; s:value "application/json" ]</pre> <p>If you are creating a view, you can include variables in the <code>s:header</code> list. When another query is run against a view with variables, that query can map the variables through the view by including predicates in the CONSTRUCT clause.</p>
<b>mimetype</b>	string	You can include this property to specify the MIME type of the source. For example, <code>s:mimetype "text/html"</code> .
<b>contentType</b>	string	Include this property to specify the content type of the body of the request. For example, <code>s:contentType "application/sparql-query"</code> or <code>s:contentType "application/json"</code> .
<b>content</b>	string or RDF	This property can be included to send content to the source in the body of the request. For example, <code>content</code> can be a SPARQL query, JSON arrays, or a list of

Option	Data Type	Description
	list	key-value pairs. Content can also be configured with an inline object (blank node) that gets translated to JSON. For more information, see <a href="#">Mapping the Content Property to JSON</a> below.
<b>parameter</b>	RDF list	<p>You can include this property to list any URL parameters as name-value pairs. For example, the <code>s:parameter</code> property below adds <code>format</code> to return results in CSV format and the <code>named-graph-uri</code> parameter to target a specific layer in a graphmart.</p> <pre>s:parameter [ s:name "format" ; s:value "csv" ] ,               [ s:name "named-graph-uri" ; s:value                 "http://cambridgesemantics.com/Layer/d541..." ] ;</pre> <p>If you are creating a view, you can include variables in the <code>s:parameter</code> list. When another query is run against a view with variables, that query can map the variables through the view by including predicates in the CONSTRUCT clause.</p>
<b>method</b>	string	You can include this property to specify the HTTP method. For example, <code>s:method "GET"</code> or <code>s:method "POST"</code> .
<b>encoding</b>	string	When targeting a file, you can include this property to specify the character encoding used by the file. The default value is <code>s:encoding "utf8"</code> .
<b>form</b>	RDF list	<p>To send data to the HTTP endpoint, you can use this property to post the data. Form is a list of name-value pairs. When including <code>s:form</code>, you must also include <code>s:contentType "multipart/form-data"</code>. The GDI sends the form object as an <code>application/x-www-form-urlencoded</code> string that contains the specified parameters. The query snippet below shows an example that incorporates <code>s:form</code>.</p> <pre>SERVICE &lt;http://cambridgesemantics.com/services/DataToolkit&gt; {   VALUES (?reviews_text) {     ("Horrible, terrible, will never use again.")</pre>

Option	Data Type	Description
		<pre> ("Wonderful, magnificent, will recommend to everyone!") ("Simply OK. I might purchase this product again.") }  ?data a s:HttpSource ;   s:url "https://api.meaningcloud.com/sentiment-2.1" ;   s:contentType "multipart/form-data" ;   s:form [ s:name "key" ; s:value "9eab751142..." ],         [ s:name "lang" ; s:value "auto" ],         [ s:name "txt" ; s:value ?reviews_text ] ;   ?confidence () ;   ?score_tag () ;   ?subjectivity () ;   ?irony () ;   ?agreement () . } </pre>
<b>format</b>	RDF list	If the data is file-based, you can include the <code>format</code> property to add parameters that describe the source. See <a href="#">File Source Format Options</a> for details about the supported parameters.
<b>timeout</b>	int	This property can be used to specify the timeout (in milliseconds) to use for requests against the source. For example, <code>s:timeout 5000</code> configures a 5 second timeout.
<b>batching</b>	boolean or int	This property can be used to disable batching, or it can be used to change the default the batch size. By default, batching is set to 5000 ( <code>s:batching 5000</code> ). To disable batching, you can include <code>s:batching false</code> in the query. Typically users do not change the batching size. However, it can be useful to control the batch size when performing updates. To configure the size, include <code>s:batching int</code> in the query. For example, <code>s:batching 3000</code> .
<b>paging</b>	RDF list	This property can be used to configure paging so that the GDI can access large amounts of data across a number of smaller requests. For details about the <code>paging</code> property, see <a href="#">Pagination Options</a> .

Option	Data Type	Description
<b>concurrency</b>	int or RDF list	<p>This property can be included to configure the maximum level of concurrency for the query. The value can be an integer, such as <code>s:concurrency 8</code>. If the value is an integer, it configures a maximum limit on the number of slices that can execute the query. For finer-grained control over the number of nodes and slices to use, concurrency can also be included as an object with <code>limit</code>, <code>nodes</code>, and/or <code>executorsPerNode</code> properties. For example, the following object configures a concurrency model that allows a maximum of 24 executors distributed across 4 nodes with 8 executors per node:</p> <pre>s:concurrency [   s:limit 24 ;   s:nodes 4 ;   s:executorsPerNode 8 ; ] ;</pre>
<b>rate</b>	int or string	<p>This property can be included to control the frequency with which a request is sent to the source. The limit applies to the number of requests a single slice can make. If you specify an integer for the rate, then the value is treated as the maximum number of requests to issue per minute. If you specify a string, you have more flexibility in configuring the rate. The sample values below show the types of values that are supported:</p> <pre>s:rate "90/minute" ; s:rate "90 per minute" ; s:rate "200000 every week" ; s:rate "10000 every 6 hours" ;</pre> <p>To enforce the rate limit, the GDI introduces a sleep between requests that is equal to the rate delay. The more executing slices, the longer the rate delay needs to be to enforce the limit in aggregate.</p> <p>Given the example of <code>s:rate "90/minute"</code>, the GDI would optimize the concurrency and only use 1 slice for execution with a rate delay of 666ms between requests. If <code>s:rate "240/minute"</code>, the GDI would use 3 executors with a rate delay of 750ms between requests.</p>

Option	Data Type	Description
<b>partitionBy</b>	string, variable, list	The GDI attempts to partition queries automatically across the available cores (slices) in AnzoGraph. To determine how to partition the query, the GDI uses metadata from the source. It looks for any column in an index, preferring the primary key column if it is interpolable. However, it only considers the first column in any index on the table. After determining the partition column, the GDI does a MIN/MAX on the column as well as a basic sizing query. To specify which column or columns the GDI should partition on, you can include the <code>partitionBy</code> property in the query. The property supports a list of source field names, bound variables, or the object <code>s:auto</code> , which forces the GDI to partition the data when the source does not define partitioning metadata.
<b>locale</b>	string	This property can be used to specify the locale to use when parsing locale-dependent data such as numbers, dates, and times.
<b>sampling</b>	int	This property can be used to configure the number of records in the source to examine for data type inferencing.
<b>selector</b>	string or RDF list	This property can be used as a binding component to identify the path to the source objects. For example, <code>s:selector "Sales.SalesOrderHeader"</code> targets the SalesOrderHeader table in the Sales schema. For more information about binding components and the selector property, see <a href="#">Using Binding Trees and Selector Paths</a> .
<b>model</b>	string	This property defines the class (or table) name for the type of data that is generated from the specified data source. For example, <code>s:model "employees"</code> . Model is optional when querying a single source. If your query targets multiple sources, however, and you want to define resource templates (primary keys) and object properties (foreign keys), you must specify the model value for each source.
<b>key</b>	string	This property can be used to define the primary key column for the source file or table. This column is leveraged in a resource template for the instances that are created from the source. For example, <code>s:key ("EMPLOYEE_ID")</code> . For more information about <code>key</code> , see <a href="#">Data Linking Options</a> .

Option	Data Type	Description
<b>reference</b>	RDF list	This property can be used to specify a foreign key column. The reference property is an RDF list that includes the <code>model</code> property to list the target table and a <code>using</code> property that defines the foreign key column. For more information about reference, see <a href="#">Data Linking Options</a> .
<b>formats</b>	RDF list	To give users control over the data types that are used when coercing strings to other types, this property can be included in GDI queries to define the desired types. In addition, it can be used to describe the formats of date and time values in the source to ensure that they are recognized and parsed to the appropriate date, time, and/or dateTime values. For details about the <code>formats</code> property, see <a href="#">Data Type Formatting Options</a> .
<b>normalize</b>	RDF list	To give users control over the labels and URIs that are generated, the GDI offers several options for normalizing the model and/or the fields that are created from the specified data source(s). For details about the <code>normalize</code> property, see <a href="#">Normalization Options</a> .
<b>count</b>	variable	If you want to turn the query into a COUNT query, you can include this property with a <code>?variable</code> to perform a count. For example, <code>s:count ?count</code> .
<b>offset</b>	int	This property can be used to offset the data that is returned by a number of rows.
<b>orderBy</b>	string, variable, list	You can include this property to order the result set by a field name, a bound variable, or a list of names or bound variables.
<b>limit</b>	int	You can include this property to limit the number of results that are returned. <code>s:limit</code> maps to the SPARQL LIMIT clause.
<b>mapping_variable</b>	variable	The mapping variables, in <code>?mapping_variable (["binding"] [datatype]</code>



Option	Data Type	Description
		<p><code>["datetime_format"])</code> format, define the triple patterns to output. When the specified <code>?variable</code> matches the source column name, the GDI uses the variable as the source data selector. If you specify an alternate variable name, a <a href="#">binding</a> needs to be specified to map the new variable to the source. You also have the option to transform the data using the <a href="#">datatype</a> and <a href="#">datetime_format</a> options.</p> <div> <p><b>Note</b></p> <p>The parentheses around the binding, data type, and format specifications are not required but are included in this document for readability.</p> </div>
<b>binding</b>	string	<p>The <code>binding</code> is a literal value that binds a <a href="#">?mapping_variable</a> to a source column. If you specify a <code>?variable</code> that matches the source column name, then that variable name is the data selector and it is not necessary to specify a binding. If you specify an alternate variable name or there is a hierarchical path to the source column, then the binding is needed to map the new variable to that source column.</p> <p>For example for CSV, the following pattern simply binds the source column AIRLINE to the lowercase variable <code>?airline</code>: <code>?airline ("AIRLINE")</code>.</p> <div> <p><b>Note</b></p> <p>For FileSource, periods (<code>.</code>), forward slashes (<code>/</code>), and brackets (<code>[ ]</code>) are parsed as path notation. Therefore, if a source column name includes any of those characters they must be escaped in the binding. Use two backslashes (<code>\\</code>) as an escape character. For example, if a column name is <b>average/day</b>, the variable and binding pattern could be written as <code>?averagePerDay ("average\\/day")</code>.</p> </div>

Option	Data Type	Description
<b>datatype</b>	URI	<p>The <code>datatype</code> is the data type to convert the column to. If you do not specify a data type, the GDI infers the type. The GDI supports the following types:</p> <ul style="list-style-type: none"> <li>• <code>xsd:int</code></li> <li>• <code>xsd:long</code></li> <li>• <code>xsd:float</code></li> <li>• <code>xsd:double</code></li> <li>• <code>xsd:boolean</code></li> <li>• <code>xsd:time</code></li> <li>• <code>xsd:dateTime</code></li> <li>• <code>xsd:date</code></li> <li>• <code>xsd:gMonthDay</code></li> <li>• <code>xsd:gYearMonth</code></li> <li>• <code>xsd:duration</code></li> <li>• <code>xsd:dayTimeDuration</code></li> <li>• <code>xsd:yearMonthDuration</code></li> <li>• <code>xsd:gMonth</code></li> <li>• <code>xsd:anyURI</code></li> </ul>
<b>datetime_format</b>	string	<p>This option is used to specify the format to use for date and time data types. The GDI supports Java date and time formats. Specify days as "d," months as "M," and years as "y." For the time, specify "H" for hours, "m" for minutes, and "s" for seconds. For example, "<code>yyyyMMdd HH:mm:ss</code>" or "<code>ddMMMyy</code>" to display date values such as "01JAN19."</p> <div> <p><b>Note</b></p> <p>The GDI's default base year is <b>2000</b>. If the source data has years with only two digits, such as <code>02-04-99</code>, the GDI prepends <b>20</b> to the digits. The value <code>02-04-99</code> is parsed to <code>02-04-2099</code>. To specify an alternate base year to use for two-digit values, you can include the notation <code>^nnnn</code> (e.g., <code>^1900</code>) in the format value. For example, to set the base year to 1900 instead of 2000, use a format value such as <code>xsd:date "dd-MMM-yy^1900"</code> or <code>xsd:date "dd-MMM-yy^1900"</code>. When one of those values is specified, <code>02-04-99</code> is parsed to <code>02-04-1999</code>.</p> </div>

## Mapping the Content Property to JSON

The `s:content` property can be configured with an inline object (blank node) that gets translated to JSON in the request body. This mapping allows for creation of embedded objects and arrays as well as a mechanism for iterating over all available input so that HTTP endpoints that support batching can be used more effectively.

## Using Blank Nodes

Blank nodes are used to create an object in the output JSON. The local name of any predicate used within `content` becomes a key in the generated JSON object. Blank nodes can be embedded within each other, allowing the hierarchical nature of JSON to be represented. For example:

```
s:content [ ex:firstName "Mary" ; ex:lastName "Barry" ] ;
```

Or

```
s:content [ ex:person [ ex:firstName "Mary" ] ] ;
```

## Using Variables

Variables can be also used in the object position to construct a request from input at runtime. For example:

```
s:content [ ex:firstName ?firstName ; ex:lastName ?lastName ] ;
```

The values for the variables can come from a TOPDOWN variable, a VALUES clause in the SERVICE block, or another data source. Any unbound variables in the input will not be added to the generated JSON object.

## Using RDF Lists

An RDF list can also be used to create an array in the output JSON. For example:

```
s:content [ ex:allKnownNames ( ?firstName ?lastName ?nickName ) ]
```

An RDF list can also be embedded inside another list to create an array in the output JSON and populate it with items evaluated against a repeating pattern across all available input rows for a slice. That pattern can be a variable, which generates an array of primitive values, or a blank node, which generates an array of mapped JSON objects. For example:

```
s:content [ ex:documents ((?id)) ] ;
```

Or

```
s:content [ ex:documents ([ ex:id ?id ; ex:title ?title ]) ] ;
```

## Example

The following example query demonstrates the use of `s:content` to generate JSON. The query also includes the `s:concurrency` property to restrict execution to a single slice. Without limiting execution when there are a small number of inputs (as in the `VALUES` clause), each input row gets executed on its own. As the inputs increase, each slice operates over a larger number of inputs until the default `s:batching 5000` is applied.

```
PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX api: <http://contoso.com/api/>

SELECT *
WHERE {
    SERVICE TOPDOWN <http://cambridgesemantics.com/services/DataToolkit>
    {
        VALUES (?firstName ?lastName ?dob ?email)
        {
            ("Gray" "Hay" "1978-03-18"^^xsd:date "gray@abc.com")
            ("Ana" "Bana" "1974-10-20"^^xsd:date "ana@abc.com")
            ("George" "Forge" "1975-08-13"^^xsd:date "george@abc.com")
            ("Miles" "Giles" "1977-04-12"^^xsd:date "miles@abc.com")
        }

        ?data a s:HttpSource ;
        s:url "https://postman-echo.com/post" ;
        s:header [ s:name "Accept" ; s:value "application/json" ] ;
        s:concurrency 1 ;
        s:content
        ([[
            api:dateOfBirth ?dob ;
            api:email ?email ;
            api:year 2020 ;
            api:person [ api:firstName ?firstName ; api:lastName ?lastName ] ;
        ]]) ;
        s:selector "data" ;
        ?firstName ("person.firstName" xsd:string) ;
        ?lastName ("person.lastName" xsd:string) ;
    }
}
```

```

        ?dob ("dateOfBirth" xsd:date) ;
        ?email ("email" xsd:string) ;
        ?year ("year" xsd:int) .
    }
}

```

The content portion of the request that the query generates is shown below:

```

[[{
  "firstName": "Gray" ,
  "lastName": "Hay" ,
  "dateOfBirth": "1978-03-18" ,
  "email": "gray@abc.com" ,
  "year": 2020
},
{
  "firstName": "Ana" ,
  "lastName": "Bana" ,
  "dateOfBirth": "1974-10-20" ,
  "email": "ana@abc.com" ,
  "year": 2020
},
{
  ...
}]

```

## Query Examples

- [Topdown Query with URL Parameters](#)
- [Generator Query against an Anzo SPARQL Endpoint](#)
- [API Queries](#)

## Topdown Query with URL Parameters

The query below reads data from a sample HTTP source that compiles worldwide weather statistics. The source has several models available for retrieving data that is current, daily, historical, etc. To target current data, the query includes `s:selector "currently"`. In addition, the query demonstrates the use of the "topdown" functionality, where the query sends values to the source to narrow the results. The VALUES

clause specifies the latitude and longitude values for the cities to return data for. In addition, since this sample source requires parameters to be specified in the connection URL, the `s:url` value includes `?lat` and `?long` as parameters as part of the value.

```
PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ex: <http://example.org/ontologies/City#>

SELECT
    ?city ?state ?temp ?rainChance
    ?humidity ?pressure ?windSpeed
WHERE
{
    SERVICE TOPDOWN <http://cambridgesemantics.com/services/DataToolkit>
    {
        ?data a s:HttpSource ;
        s:url "https://sampleEndpoint.com/forecast/{{?lat}},{{?long}}" ;
        s:selector "currently" ;
        ?lat ("latitude") ;
        ?long ("longitude") ;
        ?temp ("temperature") ;
        ?rainChance ( "precipProbability" ) ;
        ?humidity ( ) ;
        ?pressure ( ) ;
        ?windSpeed ( ) .
    }
    VALUES( ?city ?state ?lat ?long )
    {
        ( "Lakeway" "TX" 30.374563 -97.975892 )
        ( "Boston" "MA" 42.358043 -71.060415 )
        ( "Seattle" "WA" 47.590720 -122.307053 )
        ( "Chicago" "IL" 41.837741 -87.823296 )
        ( "Hilo" "HI" 19.702040 -155.090312 )
    }
}
ORDER BY ?city
```

## Generator Query against an Anzo SPARQL Endpoint

The example below is a GDI Generator query that retrieves graphmart data from a remote Anzo SPARQL endpoint.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

INSERT {
  GRAPH ${targetGraph} {
    ?s ?p ?o }
}
${usingSources}
WHERE {
  SERVICE <http://cambridgesemantics.com/services/DataToolkit>
  {
    ?data a s:HttpSource ;
      s:url
    "https://10.10.0.10/sparql/graphmart/http%3A%2F%2Fcambridgesemantics.com%2FGrap
    hmart%2F1686168b-3eaf-4fdc-9730-1903717b9e62";
      s:trust "all" ;
      s:username "user";
      s:password "pass";
      s:contentType "application/sparql-query" ;
      s:header [ s:name "Accept" ; s:value "text/csv" ] ;
    s:content ""
      PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
      SELECT ?s ?p ?o
      WHERE {
        ?s ?p ?o .
        FILTER(ISLITERAL(?o))
      }
    "" .
    ?rdf a s:RdfGenerator, s:OntologyGenerator ;
      s:as (?s ?p ?o) ;
```

```

s:ontology <http://cambridgesemantics.com/ontologies/TopMovies> ;
s:base <http://cambridgesemantics.com/data> .

}

}

```

## API Queries

The following example queries the Google Recognize API to request transcriptions for voice recordings that are stored in a Google bucket.

```

PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX anzo: <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl: <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

INSERT {
  GRAPH ${targetGraph}{
    ?record <http://google.com/transcript> ?transcript .
    ?record <http://google.com/confidence> ?confidence .
    ?record <http://google.com/file> ?file .
  }
}
${usingSources}
WHERE {
  BIND(<gs://csi-se/demo/emergency-test.mp3> as ?file)
  BIND(UUID() as ?record)
  {
    SERVICE <http://cambridgesemantics.com/services/DataToolkit>
    {
      ?data a s:HttpSource ;
      s:selector "results.alternatives" ;
      s:url "https://speech.googleapis.com/v1plbeta1/speech:recognize" ;
      s:authorization [ a s:BearerToken ; s:token ""ya29..." ] ;
      s:content ""
    }
  }
}

```



```

{
  "config": {
    "encoding": "MP3",
    "sampleRateHertz": 16000,
    "languageCode": "en-US",
    "enableWordTimeOffsets": false
  },
  "audio": {
    "uri": "gs://csi-se/demo/emergency-test.mp3"
  }
}
""" ;

    ?confidence ("confidence") ;
    ?transcript ("transcript") .

  }
}

```

The example below includes the `header` and `content` properties to send a request that contains small text snippets for sentiment analysis.

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ont: <http://cambridgesemantics.com/ontologies/Sentiment_Analysis#>

INSERT {
  GRAPH ${targetGraph} {
    ?requirement a ont:Sentiment ;
    ont:p_Sentiment_Type ?sentiment ;
    ont:p_Sentiment_Score ?polarity .
  }
}
${usingSources}
WHERE {
  ?requirement a
<http://cambridgesemantics.com/Layer/3b4163e7f53149d5a815627be5d409bd/Model#Req
uirements> ;

```

```
<http://cambridgesemantics.com/Layer/3b4163e7f53149d5a815627be5d409bd/Model#Requirements.reqText> ?requirement_text .
```

```
SERVICE TOPDOWN <http://cambridgesemantics.com/services/DataToolkit> {  
  ?data a s:HttpSource ;  
    s:url "https://text-analysis12.p.rapidapi.com/sentiment-  
analysis/api/v1.1" ;  
    s:method "POST" ;  
    s:header [ s:name "Accept" ; s:value "application/json" ] ,  
              [ s:name "X-RapidAPI-Key" ; s:value "key" ] ,  
              [ s:name "X-RapidAPI-Host" ; s:value "text-  
analysis12.p.rapidapi.com" ] ;  
    s:contentType "application/json" ;  
    s:content ""{ "text": "{?requirement_text}" , "language": "english"  
}"" ;  
    ?polarity ("aggregate_sentiment/compound" xsd:double);  
    ?sentiment () .  
}  
}
```

## Querying an Elasticsearch Source

This topic provides details about the structure to use when writing GDI queries to read or ingest data from Elasticsearch data sources. It also includes example queries that may be useful as a starting point for writing your own GDI queries.

- [Query Syntax](#)
- [Query DSL and Filter Mapping](#)
- [Query Examples](#)

### Query Syntax

The following query syntax shows the structure of a GDI query for Elasticsearch sources. The clauses, patterns, and placeholders that are links are described below.

```
# PREFIX Clause  
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>  
PREFIX es:     <http://elastic.co/search/>  
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
```

```

PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>

# Result Clause
{
  [ GRAPH ${targetGraph} { ]
    triple_patterns
  [ } ]
}
[ ${usingSources} ]

WHERE
{
  # SERVICE Clause: Include the following service call when reading or
  inserting data.
  SERVICE [ TOPDOWN ] <http://cambridgesemantics.com/services/DataToolkit>

  # View SERVICE Clause: Or use the service call below when constructing a
  view.
  SERVICE <http://cambridgesemantics.com/services/DataToolkitView>
  (${targetGraph})

  {
    ?data a s:ElasticSource ;
    s:url "string" ;
    [ s:username "string" ; ]
    [ s:password "string" ; ]
    [ s:property [ s:name "string" ; s:value "string" ; ]
    [ es:aggregations [ rdf_list ] ; ]
    [ es:config "string" ; ]
    [ es:document "string" ; ]
    [ es:field "string" | ?variable ; ]
    [ es:highlight [ rdf_list ] ; ]
    [ es:html boolean ; ]
    [ es:index "string" ; ]
    [ es:minScore float ; ]
  }
}

```

```

[ es:query "string" | [ rdf_list ] ; ]
[ es:routing "string" ; ]
[ es:searchAfter [ rdf_list ] ; ]
[ es:size int ; ]
[ es:source boolean | [ rdf_list ] ; ]
[ s:timeout int ; ]
[ s:batching boolean | int ; ]
[ s:paging [ pagination_options ] ; ]
[ s:concurrency int | [ list_of_properties ] ; ]
[ s:rate int | "string" ; ]
[ s:partitionBy "string" | ?variable ; ]
[ s:locale "string" ; ]
[ s:sampling int ; ]
[ s:selector "string" | [ list ] ; ]
[ s:model "string" ; ]
[ s:key ("string") ; ]
[ s:reference [ s:model "string" ; s:using ("string") ]
[ s:formats [ datatype_formatting_options ] ; ]
[ s:normalize boolean | [ normalization_rules ] ; ]
[ s:count ?variable ; ]
[ s:offset int ; ]
[ s:orderBy "string" | ?variable ; ]
[ s:limit int ; ]
# Mapping variables
?mapping_variable ( [ "binding" ] [ datatype ] [ "datetime_format" ] ) ;
... ;
.
# Additional clauses such as BIND, VALUES, FILTER
}
}

```

### Note

For readability, the parameters below exclude the base URIs

<<http://cambridgesemantics.com/ontologies/DataToolkit#>> and

<<http://elastic.co/search/>> as well as the `s:` and `es:` prefixes. As shown in the examples, however, the prefixes or full property URIs do need to be included in queries.

Option	Data Type	Description
<b>PREFIX Clause</b>	N/A	<p>The PREFIX clause declares the standard and custom prefixes for GDI service queries against Elasticsearch. Generally, queries include the following prefixes (or a subset of them) plus any data-specific declarations:</p> <pre> PREFIX s: &lt;http://cambridgesemantics.com/ontologies/DataToolkit#&gt; PREFIX es:    &lt;http://elastic.co/search/&gt; PREFIX rdf:   &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX rdfs:  &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX xsd:   &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX owl: &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX anzo: &lt;http://openanzo.org/ontologies/2008/07/Anzo#&gt; PREFIX zowl: &lt;http://openanzo.org/ontologies/2009/05/AnzoOwl#&gt; PREFIX dc:    &lt;http://purl.org/dc/elements/1.1/&gt; </pre>
<b>Result Clause</b>	N/A	<p>The result clause defines the type of SPARQL query to run and the set of results to return, i.e., whether you want to read (SELECT or CONSTRUCT) from the source or ingest the data into Anzo (INSERT).</p>
<b>GRAPH <code>\${targetGraph}</code></b>	N/A	<p>Include the GRAPH keyword and target graph parameter <code>\${targetGraph}</code> when writing an INSERT query to ingest data into a graphmart. Anzo automatically populates the query with the appropriate target URIs when the query runs.</p>
<b><code>\${usingSources}</code></b>	N/A	<p>Include the source graph parameter <code>\${usingSources}</code> when writing a "topdown" query that passes values from the data that is in the graphmart to the data source. Anzo automatically populates the query with the appropriate FROM clauses when the query runs. When passing literal values to the remote source, you do not need to include the source graph parameter. The <a href="#">SERVICE Clause</a> description below includes more information about passing</p>

Option	Data Type	Description
		input to data sources.
<b>SERVICE Clause</b>	N/A	<p>Include the SERVICE call <code>SERVICE [ TOPDOWN ]</code> <code>&lt;http://cambridgesemantics.com/services/DataToolkit&gt;</code> to invoke the GDI service when you are running a SELECT, INSERT, or CONSTRUCT query that is not creating a view. When writing a CONSTRUCT query in a View Step, use the <code>DataToolkitView</code> service call, as described below in <a href="#">View SERVICE Clause</a>.</p> <p>Include the optional TOPDOWN keyword when you want to pass input values from the graphmart to the data source. When you include TOPDOWN in the service call, it indicates that the rest of the query produces values to send to the source. In this case, the GDI makes repeated calls to pass in each of the specified values and retrieve the data that is based on those values.</p>
<b>View SERVICE Clause</b>	N/A	<p>When writing a CONSTRUCT query that creates a view of the data (usually in a View Step), include the following SERVICE call: <code>SERVICE &lt;http://cambridgesemantics.com/services/DataToolkitView&gt; (\$ {targetGraph})</code>. Using the <code>DataToolkitView</code> call optimizes query execution because it tells the GDI to inspect the query and determine which filters to push to the data source. It also limits the result set and retrieves only the data that is needed, i.e., the source data is fully mapped but all of the mapped data is not necessarily returned.</p>
<b>url</b>	string	<p>This property specifies the URL to use to access the source.</p> <div> <p><b>Important</b></p> <p>For security, it is a best practice to reference connection information (such as the url, username, and password) from a Query Context so that the sensitive details are abstracted from any requests. In addition, using a Query Context makes connection details reusable across queries. See <a href="#">Using Query</a></p> </div>

Option	Data Type	Description
		<p><a href="#">Contexts in Queries</a> for more information. For example:</p> <pre>?data a s:ElasticSource ;     s:url "{{@es.hostname}}:{{@es.port}}" ;     s:username "{{@es.username}}" ;     s:password "{{@es.password}}" ;</pre>
<b>username</b>	string	<p>This property lists the user name to use for the connection to the Elasticsearch.</p> <p><b>Tip</b> If you want to group the username and password properties, you can wrap them with <code>s:credentials [ ]</code>. For example:</p> <pre>s:credentials [     s:username "username" ;     s:password "password" ; ] ;</pre>
<b>password</b>	string	This property lists the password for the given username.
<b>property</b>	RDF list	<p>This property can be included to list any source-specific configuration values.</p> <pre>s:property [ s:name "custom_property_name" ; s:value "custom_value" ]</pre>
<b>aggregations</b>	RDF list	You can include this property to calculate aggregations over the specified bindings. For information about aggregations, see <a href="#">Aggregations</a> in the Elasticsearch documentation.
<b>config</b>	string	To enable you to use explicit mappings, you can include this property to specify the URL to the index configuration file to employ. For example, <code>es:config "/opt/shared/elastic/mapping.json"</code> .

Option	Data Type	Description
<b>document</b>	string	This property lists the document(s) to search.
<b>field</b>	string or variable	This property defines the field to operate on. The value can be a string or bound variable.
<b>highlight</b>	RDF list	You can include this property to define how results are highlighted. For information about the available properties, see <a href="#">Highlighting Elasticsearch Results</a> .
<b>html</b>	boolean	This property controls whether to output HTML for highlighted results. Defaults to <code>true</code> .
<b>index</b>	string	This property can be included to specify the index to search.
<b>minScore</b>	float	This property defines the minimum score for matching documents. Documents with a lower score are not included in the search results.
<b>query</b>	string or RDF list	This property defines the query to execute. The value can be a string or a query object that maps to the Elasticsearch <a href="#">Query DSL</a> . To generate the final query, the GDI combines <code>es:query</code> with any filters it can push to the Elasticsearch DSL. For more information about the <code>query</code> property and mapping Elasticsearch filters to SPARQL FILTER clauses, see <a href="#">Query DSL and Filter Mapping</a> below.
<b>routing</b>	string	This property can be included to route a document to a specific shard or to limit the search to a particular shard.
<b>searchAfter</b>	RDF list	You can include this property to define the key values to start searching from.
<b>size</b>	int	This property maps to the <code>size</code> parameter in the Elasticsearch Search API and configures the batch size or maximum number of hits to return in a single call. Defaults to <code>10</code> and typically does not need to be changed.



Option	Data Type	Description
<b>source</b>	boolean or RDF list	This property can be included to specify the source data to include in results. The value can be a boolean, list of fields, or a list of variable bindings. When <code>true</code> , all source data is returned. When <code>false</code> , no source data is returned.
<b>timeout</b>	int	This property can be used to specify the timeout (in milliseconds) to use for requests against the source. For example, <code>s:timeout 5000</code> configures a 5 second timeout.
<b>batching</b>	boolean or int	This property can be used to disable batching, or it can be used to change the default the batch size. By default, batching is set to 5000 ( <code>s:batching 5000</code> ). To disable batching, you can include <code>s:batching false</code> in the query. Typically users do not change the batching size. However, it can be useful to control the batch size when performing updates. To configure the size, include <code>s:batching int</code> in the query. For example, <code>s:batching 3000</code> .
<b>paging</b>	RDF list	This property can be used to configure paging so that the GDI can access large amounts of data across a number of smaller requests. For details about the <code>paging</code> property, see <a href="#">Pagination Options</a> .
<b>concurrency</b>	int or RDF list	<p>This property can be included to configure the maximum level of concurrency for the query. The value can be an integer, such as <code>s:concurrency 8</code>. If the value is an integer, it configures a maximum limit on the number of slices that can execute the query. For finer-grained control over the number of nodes and slices to use, concurrency can also be included as an object with <code>limit</code>, <code>nodes</code>, and/or <code>executorsPerNode</code> properties. For example, the following object configures a concurrency model that allows a maximum of 24 executors distributed across 4 nodes with 8 executors per node:</p> <pre>s:concurrency [   s:limit 24 ;   s:nodes 4 ;   s:executorsPerNode 8 ;</pre>

Option	Data Type	Description
		<pre>] ;</pre>
<b>rate</b>	int or string	<p>This property can be included to control the frequency with which a request is sent to the source. The limit applies to the number of requests a single slice can make. If you specify an integer for the rate, then the value is treated as the maximum number of requests to issue per minute. If you specify a string, you have more flexibility in configuring the rate. The sample values below show the types of values that are supported:</p> <pre>s:rate "90/minute" ; s:rate "90 per minute" ; s:rate "200000 every week" ; s:rate "10000 every 6 hours" ;</pre> <p>To enforce the rate limit, the GDI introduces a sleep between requests that is equal to the rate delay. The more executing slices, the longer the rate delay needs to be to enforce the limit in aggregate.</p> <p>Given the example of <code>s:rate "90/minute"</code>, the GDI would optimize the concurrency and only use 1 slice for execution with a rate delay of 666ms between requests. If <code>s:rate "240/minute"</code>, the GDI would use 3 executors with a rate delay of 750ms between requests.</p>
<b>partitionBy</b>	string, variable, list	<p>The GDI attempts to partition queries automatically across the available cores (slices) in AnzoGraph. To determine how to partition the query, the GDI uses metadata from the source database. It looks for any column in an index, preferring the primary key column if it is interpolable. However, it only considers the first column in any index on the table. After determining the partition column, the GDI does a MIN/MAX on the column as well as a basic sizing query. To specify which column or columns the GDI should partition on, you can include the <code>partitionBy</code> property in the query. The property supports a list of source field names, bound variables, or the object <code>s:auto</code>, which forces the GDI to partition the data when the source does not define</p>

Option	Data Type	Description
		partitioning metadata.
<b>locale</b>	string	This property can be used to specify the locale to use when parsing locale-dependent data such as numbers, dates, and times.
<b>sampling</b>	int	This property can be used to configure the number of records in the source to examine for data type inferencing.
<b>selector</b>	string or RDF list	This property can be used as a binding component to identify the path to the source objects. For example, <code>s:selector "Sales.SalesOrderHeader"</code> targets the SalesOrderHeader table in the Sales schema. For more information about binding components and the selector property, see <a href="#">Using Binding Trees and Selector Paths</a> .
<b>model</b>	string	This property defines the class (or table) name for the type of data that is generated from the specified data source. For example, <code>s:model "employees"</code> . Model is optional when querying a single source. If your query targets multiple sources, however, and you want to define resource templates (primary keys) and object properties (foreign keys), you must specify the model value for each source.
<b>key</b>	string	This property can be used to define the primary key column for the source file or table. This column is leveraged in a resource template for the instances that are created from the source. For example, <code>s:key ("EMPLOYEE_ID")</code> . For more information about <code>key</code> , see <a href="#">Data Linking Options</a> .
<b>reference</b>	RDF list	This property can be used to specify a foreign key column. The reference property is an RDF list that includes the <code>model</code> property to list the target table and a <code>using</code> property that defines the foreign key column. For more information about <code>reference</code> , see <a href="#">Data Linking Options</a> .
<b>formats</b>	RDF list	To give users control over the data types that are used when coercing strings

Option	Data Type	Description
		to other types, this property can be included in GDI queries to define the desired types. In addition, it can be used to describe the formats of date and time values in the source to ensure that they are recognized and parsed to the appropriate date, time, and/or dateTime values. For details about the <code>formats</code> property, see <a href="#">Data Type Formatting Options</a> .
<b>normalize</b>	RDF list	To give users control over the labels and URIs that are generated, the GDI offers several options for normalizing the model and/or the fields that are created from the specified data source(s). For details about the <code>normalize</code> property, see <a href="#">Normalization Options</a> .
<b>count</b>	variable	If you want to turn the query into a COUNT query, you can include this property with a <code>?variable</code> to perform a count. For example, <code>s:count ?count</code> . The GDI runs an Elasticsearch <a href="#">value count aggregation</a> .
<b>offset</b>	int	This property can be used to offset the data that is returned by a number of rows.
<b>orderBy</b>	string, variable, list	You can include this property to order the result set by a field name, a bound variable, or a list of names or bound variables.
<b>limit</b>	int	You can include this property to limit the number of results that are returned. <code>s:limit</code> maps to the SPARQL LIMIT clause.
<b>mapping_variable</b>	variable	The mapping variables, in <code>?mapping_variable (["binding"] [datatype] ["datetime_format"])</code> format, define the triple patterns to output. When the specified <code>?variable</code> matches the source column name, the GDI uses the variable as the source data selector. If you specify an alternate variable name, a <a href="#">binding</a> needs to be specified to map the new variable to the source. You also have the option to transform the data using the <a href="#">datatype</a> and <a href="#">datetime_format</a> options.

Option	Data Type	Description
		<p><b>Note</b></p> <p>The parentheses around the binding, data type, and format specifications are not required but are included in this document for readability.</p>
<b>binding</b>	string	<p>The <code>binding</code> is a literal value that binds a <code>?mapping_variable</code> to a source column. If you specify a <code>?variable</code> that matches the source column name, then that variable name is the data selector and it is not necessary to specify a binding. If you specify an alternate variable name or there is a hierarchical path to the source column, then the binding is needed to map the new variable to that source column.</p>
<b>datatype</b>	URI	<p>The <code>datatype</code> is the data type to convert the column to. If you do not specify a data type, the GDI infers the type. The GDI supports the following types:</p> <ul style="list-style-type: none"> <li>• <code>xsd:int</code></li> <li>• <code>xsd:long</code></li> <li>• <code>xsd:float</code></li> <li>• <code>xsd:double</code></li> <li>• <code>xsd:boolean</code></li> <li>• <code>xsd:time</code></li> <li>• <code>xsd:dateTime</code></li> <li>• <code>xsd:date</code></li> <li>• <code>xsd:gMonthDay</code></li> <li>• <code>xsd:gYearMonth</code></li> <li>• <code>xsd:duration</code></li> <li>• <code>xsd:dayTimeDuration</code></li> <li>• <code>xsd:yearMonthDuration</code></li> <li>• <code>xsd:gMonth</code></li> <li>• <code>xsd:anyURI</code></li> </ul>
<b>datetime_format</b>	string	<p>This option is used to specify the format to use for date and time data types. The GDI supports Java date and time formats. Specify days as "d," months as "M," and years as "y." For the time, specify "H" for hours, "m" for minutes, and "s" for seconds. For example, "<code>yyyyMMdd HH:mm:ss</code>" or "<code>ddMMMyy</code>" to display date values such as "01JAN19."</p> <p><b>Note</b></p> <p>The GDI's default base year is <b>2000</b>. If the source data has years</p>

Option	Data Type	Description
		<p>with only two digits, such as 02-04-99, the GDI prepends <b>20</b> to the digits. The value 02-04-99 is parsed to 02-04-2099. To specify an alternate base year to use for two-digit values, you can include the notation ^nnnn (e.g., ^1900) in the format value. For example, to set the base year to 1900 instead of 2000, use a format value such as xsd:date "dd-MMM-yy^1900" or xsd:date "dd-MMM-yy^1990". When one of those values is specified, 02-04-99 is parsed to 02-04-1999.</p>

Query DSL and Filter Mapping

The vocabulary used in GDI queries against an ElasticSource closely mimics the Elasticsearch [Query DSL](#). The table below shows a side-by-side view of a DSL query that is mapped to SPARQL using the `es:query` property:

DSL	SPARQL
<pre>{   "query": {     "bool" : {       "must" : {         "term" : { "user.id" : "kimchy" }       },       "filter": {         "term" : { "tags" : "production" }       },       "must_not" : {         "range" : {           "age" : { "gte" : 10, "lte" : 20 }         }       },     }   } }</pre>	<pre>es:query [   a es:BoolQuery ;   es:must [     a es:TermQuery ;     es:field "user.id" ;     es:value "kimchy" ;   ] ;   es:filter [     a es:TermQuery ;     es:field "tags" ;     es:value "production" ;   ] ;   es:mustNot [     a es:RangeQuery ;     es:field "age" ;     es:gte 10 ;     es:lte 20 ;   ] ; ]</pre>

DSL	SPARQL
<pre>       "should" : [         { "term" : { "tags" : "env1" } },         { "term" : { "tags" : "deployed" } }       ],       "minimum_should_match" : 1,       "boost" : 1.0     }   } } </pre>	<pre>     ] ;     es:should (       [ a es:TermQuery ; es:field "tags" ; es:value "env1" ]       [ a es:TermQuery ; es:field "tags" ; es:value "deployed" ]     ) ;     es:minimumShouldMatch 1 ;     es:boost 1.0 ;   ] ; </pre>

The following example SERVICE clause with comments provides details about how the GDI `es:query` property can be mapped to DSL:

```

SERVICE <http://cambridgesemantics.com/services/DataToolkit> {
  ?data a es:ElasticSource ;
  s:url "http://localhost:9200/" ;
# When the value of es:query is a simple literal,
# it is mapped to an Elastic query string query.
  es:query "literal"

# When the value of es:query is an RDF list,
# you can specify other query types,
# such as a match query:
  es:query [
    a es:MatchQuery ;
    es:field "title" | ?title ; # field can be a literal or bound variable
    es:query "moby dick" ;
  ] ;
# or a boolean query:
  es:query [
    a es:BoolQuery ;
    es:should ([
      a es:RangeQuery ;

```

```

        es:field ?amount ;
        es:gt 500 ;
        es:lt 1000 ;
    ] [
        a es:TermQuery ;
        es:field ?status ;
        es:value 'late' ;
    ] ) ;
] ;
}

```

## Filter Mapping

Filtering can be performed inside the `es:query` list or you can add a `FILTER` clause to the query. For example, the table below shows the SPARQL snippet above expressed as a `FILTER` clause.

SPARQL Query	FILTER Clause
<pre> es:query [   a es:BoolQuery ;   es:must [     a es:TermQuery ;     es:field "user.id" ;     es:value "kimchy" ;   ] ;   es:filter [     a es:TermQuery ;     es:field "tags" ;     es:value "production" ;   ] ;   es:mustNot [     a es:RangeQuery ;     es:field "age" ;     es:gte 10 ;     es:lte 20 ;   ] ;   es:should (     [ a es:TermQuery ; es:field "tags" ; </pre>	<pre> FILTER(?user_id = "kimchy" &amp;&amp;       ?tags = "production" &amp;&amp;       !(?age &gt;= 10 &amp;&amp; ?age &lt;= 20) &amp;&amp;       (?tags == "env1"    ?tags == "deployed")) </pre>



SPARQL Query	FILTER Clause
<pre> es:value "env1" ]   [ a es:TermQuery ; es:field "tags" ; es:value "deployed" ]   ) ; es:minimumShouldMatch 1 ; es:boost 1.0 ; ] ; </pre>	

The table below shows each of the supported ElasticSource FILTER translations. Only expressions matching the list below will be translated by the GDI. If the expression is of the form `value <= ?field`, the inequality is flipped to `?field > value` before translating.

es:query Expression	FILTER Clause Expression
<code>es:query [ a es:BoolQuery ; es:mustNot expr ]</code>	<code>!expr</code>
<code>es:query [ a es:BoolQuery ; es:must (left right) ]</code>	<code>left &amp;&amp; right</code>
<code>es:query [ a es:BoolQuery ; es:should (left right) ]</code>	<code>left    right</code>
<code>es:query [ a es:RangeQuery ; es:field ?field ; es:lt value ]</code>	<code>?field &lt; value</code>
<code>es:query [ a es:RangeQuery ; es:field ?field ; es:lte value ]</code>	<code>?field &lt;= value</code>
<code>es:query [ a es:TermQuery ; es:field ?field ; es:value value ]</code>	<code>?field = value</code>
<code>es:query [ a es:BoolQuery ; es:mustNot [ a es:TermQuery ; es:field ?field ; es:value value ] ]</code>	<code>?field != value</code>
<code>es:query [ a es:RangeQuery ; es:field ?field ; es:gte value ]</code>	<code>?field &gt;= value</code>

es:query Expression	FILTER Clause Expression
es:query [ a es:RangeQuery ; es:field ?field ; es:gt value ]	?field > value
es:query [ a es:QueryStringQuery ; es:field ?field ; es:query pattern ; es:defaultOperator "AND" ]	REGEX(?field, pattern, "q")
es:query [ a es:TermsQuery ; es:field ?field ; es:value value, ... ]	IN(?field, value, ...)
es:query [ a es:BoolQuery ; es:mustNot [ a es:TermsQuery ; es:field ?field ; es:value value, ... ] ]	NOT IN(?field, value, ...)
es:query [ a es:MatchQuery ; es:field ?field ; es:query value ; es:lenient true ]	CONTAINS (?field, value)
es:query [ a es:PrefixQuery ; es:field ?field ; es:value value ]	STRSTARTS (?field, value)
es:query [ a es:ExistsQuery ; es:field ?field ]	BOUND(?field)

## Query Examples

- [General Query](#)
- [Aggregations](#)
- [Highlighting](#)

## General Query

The following example queries any Elasticsearch indexes that are loaded in the graphmart for which you run the query. No configuration is needed because Anzo manages the indexes that it loads and uses predictable naming conventions and aliases.

```
PREFIX docm:
<http://cambridgesemantics.com/ontologies/2011/07/DocumentMetadata#>
PREFIX es: <http://elastic.co/search/>
```

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX services: <http://cambridgesemantics.com/services/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?title
WHERE {
  SERVICE <http://cambridgesemantics.com/services/DataToolkit> {
    ?data a es:ElasticSource;
    s:selector "unstructuredfile" ;
    es:query "string" ; # input the text search string
    es:fields "fullText" ;
    ?title (xsd:string) .
  }
}

```

## Aggregations

The following example query performs terms aggregations.

```

PREFIX es: <http://elastic.co/search/>
PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT *
WHERE {
  SERVICE <http://cambridgesemantics.com/services/DataToolkit> {
    ?data a es:ElasticSource;
    s:url "https://{{@es.hostname}}:{{@es.port}}/" ;
    s:username "{{@es.username}}" ;
    s:password "{{@es.password}}" ;
    es:index "templated_consumption_es" ;
    es:query "*ELM*" ;
    ?instance () ;
    es:aggregations [
      ?artifactTypes [
        a es:TermsAggregation ;
        es:field ?artifactType ;

```

```

        es:meta [
            ?label "artifactType" ;
        ] ;
        ?value () ;
        ?count () ;
    ] ;
    ?fileTypes [
        a es:TermsAggregation ;
        es:field ?fileType ;
        es:meta [
            ?label "fileType" ;
        ] ;
        ?value () ;
        ?count () ;
    ] ;
    ?managedBys [
        a es:TermsAggregation ;
        es:field ?managedBy ;
        es:meta [
            ?label "managedBy" ;
        ] ;
        ?value () ;
        ?count () ;
    ] ;
] .
}
}

```

## Highlighting

The following example configures highlighting for fragments from the actor field.

```

PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX es: <http://elastic.co/search/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT *
WHERE {
    SERVICE TOPDOWN <http://cambridgesemantics.com/services/DataToolkit>

```

```

{
  ?data a es:ElasticSource ;
    es:url "http://localhost:9200/" ;
    es:index "films" ;
    es:html false ;
    es:query "Clint" ;
    es:field ?actor, ?director ;
    es:highlight [
      es:field ?actor ;
      es:type "plain" ;
      es:fragmentSize 200 ;
      es:numberOfFragments 10 ;
      es:preTags "<mark hit='true'>" ;
      es:postTags "</mark>" ;
    ] ;
    s:selector "film" ;
    ?actor (xsd:string) ;
    ?awards (xsd:string) ;
    ?director (xsd:string) ;
    ?image (xsd:string) ;
    ?length (xsd:long) ;
    ?popularity (xsd:long) ;
    ?subject (xsd:string) ;
    ?title (xsd:string) ;
    ?year (xsd:long) ;
    ?score () ;
    ?id () ;
    ?highlights [
      ?field () ;
      ?fragment () ;
    ] .
  FILTER(?year = 1990 || ?length > 103)
  FILTER(REGEX(?title, "Manhattan", "q") || REGEX(?subject, "Comedy", "q") ||
  REGEX(?subject, "Drama", "q"))
}
}

```

## Highlighting Elasticsearch Results

By including the **highlight** property in ElasticSource GDI queries, you can configure the response to include highlights for search results. For general information about highlighting Elasticsearch responses, see [Highlighting](#) in the Elasticsearch documentation. Highlight property usage is described below.

- [Highlight Syntax](#)
- [Highlight Examples](#)

## Highlight Syntax

```
es:highlight [  
  es:boundaryChars "string" ;  
  es:boundaryMaxScan int ;  
  es:boundaryScannerLocale "string" ;  
  es:boundaryScannerType "string" ;  
  es:field "string" ;  
  es:forceSource boolean ;  
  es:fragmentSize int ;  
  es:fragmenter "string" ;  
  es:highlightFilter boolean ;  
  es:highlightQuery "string" | [ rdf_list ] ;  
  es:highlighterType "string" ;  
  es:noMatchSize int ;  
  es:numberOfFragments int ;  
  es:order "string" ;  
  es:phraseLimit int ;  
  es:postTags "string" ;  
  es:preTags "string" ;  
  es:requireFieldMatch boolean ;  
]
```

Option	Data Type	Description
<b>boundaryChars</b>	string	This property can be used to define the boundary characters to look for. Defaults to . , ! ? \t\n.
<b>boundaryMaxScan</b>	int	This property can be used to place a limit on the number of characters to scan when looking for boundary characters.

Option	Data Type	Description
		Defaults to 20.
<b>boundaryScannerLocale</b>	string	This property defines the language tag (such as "en-US" or "fr-FR") to apply when searching for sentence and word boundaries.
<b>boundaryScannerType</b>	string	<p>If <a href="#">highlighterType</a> is <code>unified</code> or <code>fvh</code>, this property can be used to specify how to break the highlighted fragments. This property is ignored when the highlighter type is <code>plain</code>. The list below describes the valid values:</p> <ul style="list-style-type: none"> <li>• <b>chars</b>: Valid when the highlighter type is fast vector highlighter (fvh) (<code>es:highlighterType "fvh"</code>). Specifies that the highlighting boundaries are the characters specified by <a href="#">boundaryChars</a>. The <a href="#">boundaryMaxScan</a> value controls how far to scan for boundary characters. This is the default value for fvh.</li> <li>• <b>sentence</b>: This is the default value for the unified highlighter. It configures highlighted fragments to break at the next sentence boundary. You can specify the locale to use with <a href="#">boundaryScannerLocale</a>. When used with the unified highlighter, the sentence scanner splits sentences bigger than <a href="#">fragmentSize</a> at the first word boundary next to <code>fragmentSize</code>. You can set <code>fragmentSize</code> to 0 to avoid splitting sentences.</li> <li>• <b>word</b>: Configures highlighted fragments to break at the next word boundary. You can specify the locale to use with <a href="#">boundaryScannerLocale</a>.</li> </ul>
<b>field</b>	string or variable	This property specifies the field to retrieve highlights for. It can include a <code>?variable</code> (which the GDI maps to the full path of the field in the Elasticsearch document), a field name, or

Option	Data Type	Description
		<p>a field name pattern. For example:</p> <pre> es:highlight [   es:field ?actor ;   es:field "film.actor" ;   es:field "film.*" ;   es:field "*" ; ] </pre>
<b>forceSource</b>	boolean	This property controls whether to highlight based on the source even if the field is stored separately. Defaults to <code>false</code> .
<b>fragmentSize</b>	int	This property specifies the number of characters to include in highlighted fragments. Defaults to <code>100</code> .
<b>fragmenter</b>	string	<p>If <b>highlighterType</b> is <code>plain</code>, this property can be used to specify how to break up text in highlight snippets. The list below describes the valid values:</p> <ul style="list-style-type: none"> <li>• <b>simple</b>: Breaks text into fragments that are the same size (as specified by <b>fragmentSize</b>).</li> <li>• <b>span</b>: The default value. Breaks text into fragments that are the same size but tries to avoid breaking text between highlighted terms.</li> </ul>
<b>highlightFilter</b>	boolean	This property controls whether to highlight filter results.
<b>highlightQuery</b>	string or object	This property specifies the highlight query. The value can be a string or a query object that maps to the Elasticsearch query DSL.
<b>highlighterType</b>	string	This property defines the type of highlighter to use, <code>"plain"</code> , <code>"unified"</code> , or <code>"fvh"</code> .



Option	Data Type	Description
<b>noMatchSize</b>	int	This property specifies the number of characters to return from the beginning of the field if there are no matching fragments to highlight. Defaults to 0 (nothing is returned).
<b>numberOfFragments</b>	int	This property can be used to set the maximum number of fragments to generate. If this property is set to 0, no fragments are returned. Instead, the entire field contents are highlighted and returned, which can be useful if you want to highlight short text (such as a title or address) for which fragmentation is not required. Defaults to 5. If the number of fragments is 0, <a href="#">fragmentSize</a> is ignored.
<b>order</b>	string	This property can be included to sort highlighted fragments by score. When <code>es:order "score"</code> , the most relevant fragments are output first. Defaults to "none"; fragments are output in the order they appear in the field.
<b>phraseLimit</b>	int	If <a href="#">highlighterType</a> is <code>fvh</code> , this property can be used to limit the number of matching phrases to consider. Limiting the number of phrases prevents the <code>fvh</code> highlighter from analyzing too many phrases and consuming too much memory. Defaults to 256.
<b>postTags</b>	string	This property is used in conjunction with <a href="#">preTags</a> to define the HTML tags to use for the highlighted elements. This property defines the closing tag to use after the highlighted text. Defaults to <code>&lt;/em&gt;</code> .
<b>preTags</b>	string	This property is used in conjunction with <a href="#">postTags</a> to define the HTML tags to use for the highlighted elements. This property defines the opening tag to use before the highlighted text. Defaults to <code>&lt;em&gt;</code> .

Option	Data Type	Description
<b>requireFieldMatch</b>	boolean	This property controls whether to highlight only the fields that contain a query match. Defaults to <code>true</code> . If <code>false</code> , all fields are highlighted.

## Highlight Examples

The following example configures highlighting for fragments from the actor field.

```

PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX es: <http://elastic.co/search/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT *
WHERE {
  SERVICE TOPDOWN <http://cambridgesemantics.com/services/DataToolkit>
  {
    ?data a es:ElasticSource ;
    es:url "http://localhost:9200/" ;
    es:index "films" ;
    es:html false ;
    es:query "Clint" ;
    es:field ?actor, ?director ;
    es:highlight [
      es:field ?actor ;
      es:type "plain" ;
      es:fragmentSize 200 ;
      es:numberOfFragments 10 ;
      es:preTags "<mark hit='true'>" ;
      es:postTags "</mark>" ;
    ] ;
    s:selector "film" ;
    ?actor (xsd:string) ;
    ?awards (xsd:string) ;
    ?director (xsd:string) ;
    ?image (xsd:string) ;
    ?length (xsd:long) ;
    ?popularity (xsd:long) ;
  }
}
```

```

    ?subject (xsd:string) ;
    ?title (xsd:string) ;
    ?year (xsd:long) ;
    ?score () ;
    ?id () ;
    ?highlights [
        ?field () ;
        ?fragment () ;
    ] .
FILTER(?year = 1990 || ?length > 103)
FILTER(REGEX(?title, "Manhattan", "q") || REGEX(?subject, "Comedy", "q") ||
REGEX(?subject, "Drama", "q"))
}
}

```

## Querying File Sources

The Graph Data Interface (GDI) uses the [Apache Commons VFS](#) library to work with file systems. Many of the properties specified in queries against file sources reflect the requirements of the VFS library for that source. The topics in this section provide guidance on writing GDI queries for each of the supported file types.

[Querying CSV and TSV Files](#)

[Querying JSON and NDJSON Files](#)

[File Source Format Options](#)

[File Storage Connection Options](#)

## Querying CSV and TSV Files

This topic provides details about the structure to use when writing GDI queries to read or ingest data from CSV or TSV files. It also includes example queries that may be useful as a starting point for writing your own GDI queries.

- [Query Structure](#)
- [Query Examples](#)

## Query Structure

The following query syntax shows the structure of a GDI query for CSV and TSV sources. The clauses, patterns, and placeholders that are links are described below.

```

# PREFIX Clause
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>

# Result Clause
{
  [ GRAPH ${targetGraph} { ]
    triple_patterns
  [ } ]
}
[ ${usingSources} ]

WHERE
{
  # SERVICE Clause: Include the following service call when reading or
inserting data.
    SERVICE [ TOPDOWN ] <http://cambridgesemantics.com/services/DataToolkit>

  # View SERVICE Clause: Or use the service call below when constructing a
view.
    SERVICE <http://cambridgesemantics.com/services/DataToolkitView>
    (${targetGraph})

  {
    ?data a s:FileSource ;
      s:url "string" ;
      [ s:options [ file_storage_connection_options ] ; ]
      [ s:pattern "string" ; ]
      [ s:maxDepth int ; ]
      [ s:format [ source_format_options ; ] ; ]
      [ s:mimetype "string" ; ]
      [ s:username "string" ; ]
      [ s:password "string" ; ]
  }
}

```

```

[ s:timeout int ; ]
[ s:batching boolean | int ; ]
[ s:paging [ pagination_options ; ]
[ s:concurrency int | [ list_of_properties ] ; ]
[ s:rate int | "string" ; ]
[ s:locale "string" ; ]
[ s:sampling int ; ]
[ s:selector "string" | [ list ] ; ]
[ s:model "string" ; ]
[ s:key ("string") ; ]
[ s:reference [ s:model "string" ; s:using ("string") ]
[ s:formats [ datatype_formatting_options ] ; ]
[ s:normalize boolean | [ normalization_rules ] ; ]
[ s:count ?variable ; ]
[ s:offset int ; ]
[ s:limit int ; ]
# Mapping variables
?mapping_variable ( [ "binding" ] [ datatype ] [ "datetime_format" ] ) ;
... ;
.
# Additional clauses such as BIND, VALUES, FILTER
}
}

```

### Note

For readability, the parameters below exclude the base URI

<<http://cambridgesemantics.com/ontologies/DataToolkit#>> as well as the `s:` prefix.

As shown in the examples, however, the `s:` prefix or full property URI does need to be included in queries.

Option	Data Type	Description
<b>PREFIX Clause</b>	N/A	The PREFIX clause declares the standard and custom prefixes for GDI queries. Generally, queries include the following prefixes (or a subset of them) plus any data-specific declarations:

Option	Data Type	Description
		<pre> PREFIX s: &lt;http://cambridgesemantics.com/ontologies/DataToolkit#&gt; PREFIX rdf:      &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX rdfs:     &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX xsd:      &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX owl:    &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX anzo: &lt;http://openanzo.org/ontologies/2008/07/Anzo#&gt; PREFIX zowl: &lt;http://openanzo.org/ontologies/2009/05/AnzoOwl#&gt; PREFIX dc:       &lt;http://purl.org/dc/elements/1.1/&gt; </pre>
<b>Result Clause</b>	N/A	The result clause defines the type of SPARQL query to run and the set of results to return, i.e., whether you want to read (SELECT or CONSTRUCT) from the source or ingest the data into Anzo (INSERT).
<b>GRAPH \${targetGraph}</b>	N/A	Include the GRAPH keyword and target graph parameter <code>\${targetGraph}</code> when writing an INSERT query to ingest data into a graphmart. Anzo automatically populates the query with the appropriate target URIs when the query runs.
<b>\${usingSources}</b>	N/A	Include the source graph parameter <code>\${usingSources}</code> when writing a "topdown" query that passes values from the data that is in the graphmart to the data source. Anzo automatically populates the query with the appropriate FROM clauses when the query runs. When passing literal values to the remote source, you do not need to include the source graph parameter. The <a href="#">SERVICE Clause</a> description below includes more information about passing input to data sources.
<b>SERVICE Clause</b>	N/A	Include the SERVICE call <code>SERVICE [ TOPDOWN ]</code> <pre>&lt;http://cambridgesemantics.com/services/DataToolkit&gt; to</pre>

Option	Data Type	Description
		<p>invoke the GDI service when you are running a SELECT, INSERT, or CONSTRUCT query that is not creating a view. When writing a CONSTRUCT query in a View Step, use the <code>DataToolkitView</code> service call, as described below in <a href="#">View SERVICE Clause</a>.</p> <p>Include the optional TOPDOWN keyword when you want to pass input values from the graphmart to the data source. When you include TOPDOWN in the service call, it indicates that the rest of the query produces values to send to the source. In this case, the GDI makes repeated calls to pass in each of the specified values and retrieve the data that is based on those values.</p>
<b>View SERVICE Clause</b>	N/A	<p>When writing a CONSTRUCT query that creates a view of the data (usually in a View Step), include the following SERVICE call: <code>SERVICE &lt;http://cambridgesemantics.com/services/DataToolkitView&gt; (\${targetGraph})</code>. Using the <code>DataToolkitView</code> call optimizes query execution because it tells the GDI to inspect the query and determine which filters to push to the data source. It also limits the result set and retrieves only the data that is needed, i.e., the source data is fully mapped but all of the mapped data is not necessarily returned.</p>
<b>url</b>	string	<p>This property specifies the file system location of the source file or directory of files. When specifying a directory (such as <code>s:url "/opt/shared-files/loads/"</code>), the GDI loads all of the file formats it recognizes. To specify a directory but limit the number or type of files that are read, you can include the <a href="#">pattern</a> and/or <a href="#">maxDepth</a> properties.</p>
<b>options</b>	RDF list	<p>If additional connection information needs to be provided to access the file storage system, include the <code>options</code> property to list any storage-specific connection parameters. See <a href="#">File Storage Connection Options</a> for information about the supported properties for each storage type.</p>
<b>pattern</b>	string	<p>This property can be used to specify a wildcard pattern for matching file</p>

Option	Data Type	Description
		names. For example, <code>s:pattern "common_prefix*.csv"</code> . You can include one <code>s:pattern</code> property per <code>FileSource</code> . The GDI supports Unix file globbing syntax outside of parentheses. Within parentheses, full Java regular expression language is supported. For example, including <code>s:pattern "data/**/customer_*.csv"</code> tells the GDI to load all files that match the pattern "customer_*.csv" from any number of subdirectories under the <code>data</code> directory. Similarly <code>s:pattern "(\\d+)/transaction_*.csv"</code> tells the GDI to load all files that match the pattern "transaction_*.csv" in all subdirectories.
<b>maxDepth</b>	int	This property can be used to limit the directory traversal depth. By default, when <code>s:url</code> specifies a directory (and a <code>s:pattern</code> that limits that traversal depth is not specified), all subdirectories are processed. To process only the files in the top level directory, set <code>maxDepth</code> to 0 ( <code>s:maxDepth 0</code> ). To process the files in the top level directory plus the first-level subdirectories, set <code>maxDepth</code> to 1 ( <code>s:maxDepth 1</code> ), and so on.
<b>format</b>	RDF list	You can include the <code>format</code> property to add parameters that describe the source files. See <a href="#">File Source Format Options</a> for details about the supported parameters.
<b>mimetype</b>	string	This property can be included to specify the MIME type of the data. If you are querying TSV files that do not have a <code>.tsv</code> file extension, include the <code>mimetype</code> property with a value of <code>text/tsv</code> ( <code>s:mimetype "text/tsv"</code> ).
<b>username</b>	string	If authentication is required to access the source, include this property to specify the user name.
<b>password</b>	string	This property lists the password for the given username.
<b>timeout</b>	int	This property can be used to specify the timeout (in milliseconds) to use for requests against the source. For example, <code>s:timeout 5000</code> configures a 5



Option	Data Type	Description
		second timeout.
<b>batching</b>	boolean or int	This property can be used to disable batching, or it can be used to change the default the batch size. By default, batching is set to 5000 ( <code>s:batching 5000</code> ). To disable batching, you can include <code>s:batching false</code> in the query. Typically users do not change the batching size. However, it can be useful to control the batch size when performing updates. To configure the size, include <code>s:batching int</code> in the query. For example, <code>s:batching 3000</code> .
<b>paging</b>	RDF list	This property can be used to configure paging so that the GDI can access large amounts of data across a number of smaller requests. For details about the <code>paging</code> property, see <a href="#">Pagination Options</a> .
<b>concurrency</b>	int or RDF list	<p>This property can be included to configure the maximum level of concurrency for the query. The value can be an integer, such as <code>s:concurrency 8</code>. If the value is an integer, it configures a maximum limit on the number of slices that can execute the query. For finer-grained control over the number of nodes and slices to use, concurrency can also be included as an object with <code>limit</code>, <code>nodes</code>, and/or <code>executorsPerNode</code> properties. For example, the following object configures a concurrency model that allows a maximum of 24 executors distributed across 4 nodes with 8 executors per node:</p> <pre>s:concurrency [   s:limit 24 ;   s:nodes 4 ;   s:executorsPerNode 8 ; ] ;</pre>
<b>rate</b>	int or string	This property can be included to control the frequency with which a request is sent to the source. The limit applies to the number of requests a single slice can make. If you specify an integer for the rate, then the value is treated as the maximum number of requests to issue per minute. If you specify a string,

Option	Data Type	Description
		<p>you have more flexibility in configuring the rate. The sample values below show the types of values that are supported:</p> <pre>s:rate "90/minute" ; s:rate "90 per minute" ; s:rate "200000 every week" ; s:rate "10000 every 6 hours" ;</pre> <p>To enforce the rate limit, the GDI introduces a sleep between requests that is equal to the rate delay. The more executing slices, the longer the rate delay needs to be to enforce the limit in aggregate.</p> <p>Given the example of <code>s:rate "90/minute"</code>, the GDI would optimize the concurrency and only use 1 slice for execution with a rate delay of 666ms between requests. If <code>s:rate "240/minute"</code>, the GDI would use 3 executors with a rate delay of 750ms between requests.</p>
<b>locale</b>	string	This property can be used to specify the locale to use when parsing locale-dependent data such as numbers, dates, and times.
<b>sampling</b>	int	This property can be used to configure the number of records in the source to examine for data type inferencing.
<b>selector</b>	string or RDF list	This property can be used as a binding component to identify the path to the source objects. For example, <code>s:selector "Sales.SalesOrderHeader"</code> targets the SalesOrderHeader table in the Sales schema. For more information about binding components and the selector property, see <a href="#">Using Binding Trees and Selector Paths</a> .
<b>model</b>	string	This property defines the class (or table) name for the type of data that is generated from the specified data source. For example, <code>s:model "employees"</code> . Model is optional when querying a single source. If your query targets multiple sources, however, and you want to define resource templates (primary keys) and object properties (foreign keys), you must

Option	Data Type	Description
		specify the model value for each source.
<b>key</b>	string	This property can be used to define the primary key column for the source file or table. This column is leveraged in a resource template for the instances that are created from the source. For example, <code>s:key ("EMPLOYEE_ID")</code> . For more information about <code>key</code> , see <a href="#">Data Linking Options</a> .
<b>reference</b>	RDF list	This property can be used to specify a foreign key column. The reference property is an RDF list that includes the <code>model</code> property to list the target table and a <code>using</code> property that defines the foreign key column. For more information about <code>reference</code> , see <a href="#">Data Linking Options</a> .
<b>formats</b>	RDF list	To give users control over the data types that are used when coercing strings to other types, this property can be included in GDI queries to define the desired types. In addition, it can be used to describe the formats of date and time values in the source to ensure that they are recognized and parsed to the appropriate date, time, and/or dateTime values. For details about the <code>formats</code> property, see <a href="#">Data Type Formatting Options</a> .
<b>normalize</b>	RDF list	To give users control over the labels and URIs that are generated, the GDI offers several options for normalizing the model and/or the fields that are created from the specified data source(s). For details about the <code>normalize</code> property, see <a href="#">Normalization Options</a> .
<b>count</b>	variable	If you want to turn the query into a COUNT query, you can include this property with a <code>?variable</code> to perform a count. For example, <code>s:count ?count</code> .
<b>offset</b>	int	This property can be used to offset the data that is returned by a number of rows.
<b>limit</b>	int	You can include this property to limit the number of results that are returned. <code>s:limit</code> maps to the SPARQL LIMIT clause.

Option	Data Type	Description
<b>mapping_variable</b>	variable	<p>The mapping variables, in <code>?mapping_variable ("binding" [datatype] ["datetime_format"])</code> format, define the triple patterns to output. When the specified <code>?variable</code> matches the source column name, the GDI uses the variable as the source data selector. If you specify an alternate variable name, a <a href="#">binding</a> needs to be specified to map the new variable to the source. You also have the option to transform the data using the <a href="#">datatype</a> and <a href="#">datetime_format</a> options.</p> <div> <p><b>Note</b></p> <p>The parentheses around the binding, data type, and format specifications are not required but are included in this document for readability.</p> </div>
<b>binding</b>	string	<p>The <code>binding</code> is a literal value that binds a <a href="#">?mapping_variable</a> to a source column. If you specify a <code>?variable</code> that matches the source column name, then that variable name is the data selector and it is not necessary to specify a binding. If you specify an alternate variable name or there is a hierarchical path to the source column, then the binding is needed to map the new variable to that source column.</p> <p>For example for CSV, the following pattern simply binds the source column AIRLINE to the lowercase variable <code>?airline</code>: <code>?airline ("AIRLINE")</code>.</p> <div> <p><b>Note</b></p> <p>For FileSource, periods (.), forward slashes (/), and brackets ([ ]) are parsed as path notation. Therefore, if a source column name includes any of those characters they must be escaped in the binding. Use two backslashes (\\) as an escape character. For example, if a column name is <b>average/day</b>, the variable and binding pattern could be written as <code>?averagePerDay ("average\\/day")</code>.</p> </div>

Option	Data Type	Description
<b>datatype</b>	URI	<p>The <code>datatype</code> is the data type to convert the column to. If you do not specify a data type, the GDI infers the type. The GDI supports the following types:</p> <ul style="list-style-type: none"> <li>• <code>xsd:int</code></li> <li>• <code>xsd:long</code></li> <li>• <code>xsd:float</code></li> <li>• <code>xsd:double</code></li> <li>• <code>xsd:boolean</code></li> <li>• <code>xsd:time</code></li> <li>• <code>xsd:dateTime</code></li> <li>• <code>xsd:date</code></li> <li>• <code>xsd:gMonthDay</code></li> <li>• <code>xsd:gYearMonth</code></li> <li>• <code>xsd:duration</code></li> <li>• <code>xsd:dayTimeDuration</code></li> <li>• <code>xsd:yearMonthDuration</code></li> <li>• <code>xsd:gMonth</code></li> <li>• <code>xsd:anyURI</code></li> </ul>
<b>datetime_format</b>	string	<p>This option is used to specify the format to use for date and time data types. The GDI supports Java date and time formats. Specify days as "d," months as "M," and years as "y." For the time, specify "H" for hours, "m" for minutes, and "s" for seconds. For example, "<code>yyyyMMdd HH:mm:ss</code>" or "<code>ddMMMyy</code>" to display date values such as "01JAN19."</p> <div> <p><b>Note</b></p> <p>The GDI's default base year is <b>2000</b>. If the source data has years with only two digits, such as <code>02-04-99</code>, the GDI prepends <b>20</b> to the digits. The value <code>02-04-99</code> is parsed to <code>02-04-2099</code>. To specify an alternate base year to use for two-digit values, you can include the notation <code>^nnnn</code> (e.g., <code>^1900</code>) in the format value. For example, to set the base year to 1900 instead of 2000, use a format value such as <code>xsd:date "dd-MMM-yy^1900"</code> or <code>xsd:date "dd-MMM-yy^1990"</code>. When one of those values is specified, <code>02-04-99</code> is parsed to <code>02-04-1999</code>.</p> </div>

## Query Examples

The example below ingests a directory of CSV files into a Graphmart. The pattern property (`s:pattern "post_[0-9]*_[0-9]*.csv"`) is used to narrow down the set of files to load. Since the files use a pipe (|) as the delimiter rather than a comma (,), the delimiter property is also included to specify the | character.

```
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX snvoc:  <http://www.ldbc.eu/ldbc_socialnet/1.0/vocabulary/>
PREFIX sntag:  <http://www.ldbc.eu/ldbc_socialnet/1.0/tag/>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
```

```
INSERT {
  GRAPH ${targetGraph}
  {
    ?postIRI a snvoc:Post, snvoc:Message ;
    snvoc:creationDate ?creationDate ;
    snvoc:id ?id ;
    snvoc:imageFile ?imageFile ;
    snvoc:locationIP ?locationIP ;
    snvoc:browserUsed ?browserUsed ;
    snvoc:language ?language ;
    snvoc:content ?content ;
    snvoc:length ?length ;
  }
}
WHERE {
  SERVICE <http://cambridgesemantics.com/services/DataToolkit>
  {
    ?data a s:FileSource ;
    s:url "/opt/shared-files/data/csv/post_6_0/" ;
    s:pattern "post_[0-9]*_[0-9]*.csv" ;
    s:format [ s:delimiter "|" ] ;
    ?creationDate (xsd:dateTime) ;
    ?id (xsd:string) ;
    ?imageFile (xsd:string) ;
    ?locationIP (xsd:string) ;
    ?browserUsed (xsd:string) ;
    ?language (xsd:string) ;
  }
}
```

```

        ?content (xsd:string) ;
        ?length(xsd:string) .
    BIND( IRI("http://www.ldbc.eu/ldbc_socialnet/1.0/data/Post/{ {?id}}") as
    ?postIRI)
}
}

```

The example below is similar to the query above but it specifies the formats to use for the `xsd:date` values.

```

PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX anzo:   <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:   <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:     <http://purl.org/dc/elements/1.1/>
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX kd:     <http://cambridgesemantics.com/ont/autogen/Rh/Kaggle_Diabetes#>

INSERT {
    GRAPH ${targetGraph}
    {
        ?URI a kd:Diagnosis ;
        kd:Diagnosis_DiagnosisGuid ?diagnosis_guid ;
        kd:Diagnosis_PatientGuid ?patient_guid ;
        kd:Diagnosis_ICD9Code ?icd9Code ;
        kd:Diagnosis_DiagnosisDescription ?diagnosisDescription ;
        kd:Diagnosis_StartDate ?cus_start_date ;
        kd:Diagnosis_EndDate ?Date_End ;
        kd:Diagnosis_Acute ?acute ;
        kd:Diagnosis_UserGuid ?UserGuid .
    }
}
WHERE {
    SERVICE <http://cambridgesemantics.com/services/DataToolkit>
    {
        ?csv a s:FileSource ;
        s:url "/opt/shared-files/source_data/kaggle_diabetes/" ;
        s:pattern "Diagnosis.csv" ;
    }
}

```

```

s:format [ s:delimiter "," ] ;
?diagnosis_guid ("DiagnosisGuid" xsd:string) ;
?patient_guid ("PatientGuid" xsd:string) ;
?icd9Code ("ICD9Code" xsd:string) ;
?diagnosisDescription ("DiagnosisDescription" xsd:string) ;
?acute ("Acute" xsd:int ) ;
?UserGuid ("UserGuid" xsd:string) ;
?cus_start_date ("CUSTOMER_START_DATE" xsd:date "yyyy-MM-dd") ;
?Date_End ("Date End" xsd:date "MM/dd/yy") .
}
BIND(IRI(CONCAT("urn://cambridgesemantics.com/kaggle_
diabetes/patient/",ENCODE_FOR_URI(?diagnosis_guid))) as ?URI)
}

```

## Related Topics

[Generating a Graph and Ontology with a Direct Load Step](#)

## Querying JSON and NDJSON Files

This topic provides details about the structure to use when writing GDI queries to read or ingest data from JSON or NDJSON files. It also includes example queries that may be useful as a starting point for writing your own GDI queries.

- [Query Syntax](#)
- [Hierarchical Bindings and Arrays](#)
- [Capturing Property Keys](#)
- [Query Examples](#)

## Query Syntax

The following query syntax shows the structure of a GDI query for JSON sources. The clauses, patterns, and placeholders that are links are described below.

```

# PREFIX Clause
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>

```



```

PREFIX owl:    <http://www.w3.org/2002/07/owl#>
PREFIX anzo:    <http://openanzo.org/ontologies/2008/07/Anzo#>
PREFIX zowl:    <http://openanzo.org/ontologies/2009/05/AnzoOwl#>
PREFIX dc:      <http://purl.org/dc/elements/1.1/>

# Result Clause
{
  [ GRAPH ${targetGraph} { ]
    triple_patterns
  [ } ]
}
[ ${usingSources} ]

WHERE
{
  # SERVICE Clause: Include the following service call when reading or
inserting data.
    SERVICE [ TOPDOWN ] <http://cambridgesemantics.com/services/DataToolkit>

  # View SERVICE Clause: Or use the service call below when constructing a
view.
    SERVICE <http://cambridgesemantics.com/services/DataToolkitView>
    (${targetGraph})

  {
    ?data a s:FileSource ;
      s:url "string" ;
      [ s:options [ file_storage_connection_options ] ; ]
      [ s:pattern "string" ; ]
      [ s:maxDepth int ; ]
      [ s:format [ source_format_options ; ] ; ]
      [ s:mimetype "string" ; ]
      [ s:username "string" ; ]
      [ s:password "string" ; ]
      [ s:timeout int ; ]
      [ s:batching boolean | int ; ]
      [ s:paging [ pagination_options ; ]
      [ s:concurrency int | [ list_of_properties ] ; ]
      [ s:rate int | "string" ; ]
  }
}

```

```

[ s:locale "string" ; ]
[ s:sampling int ; ]
[ s:selector "string" | [ list ] ; ]
[ s:model "string" ; ]
[ s:key ("string") ; ]
[ s:reference [ s:model "string" ; s:using ("string") ]
[ s:formats [ datatype_formatting_options ] ; ]
[ s:normalize boolean | [ normalization_rules ] ; ]
[ s:count ?variable ; ]
[ s:offset int ; ]
[ s:limit int ; ]
# Mapping variables and hierarchical bindings
?mapping_variable ( [ "binding" ] [ datatype ] [ "datetime_format" ] ) ;
... ;
.
# Additional clauses such as BIND, VALUES, FILTER
}
}

```

### Note

For readability, the parameters below exclude the base URI

<<http://cambridgesemantics.com/ontologies/DataToolkit#>> as well as the `s:` prefix.

As shown in the examples, however, the `s:` prefix or full property URI does need to be included in queries.

Option	Data Type	Description
<b>PREFIX Clause</b>	N/A	<p>The PREFIX clause declares the standard and custom prefixes for GDI queries. Generally, queries include the following prefixes (or a subset of them) plus any data-specific declarations:</p> <pre> PREFIX s: &lt;<a href="http://cambridgesemantics.com/ontologies/DataToolkit#">http://cambridgesemantics.com/ontologies/DataToolkit#</a>&gt; PREFIX rdf:      &lt;<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a>&gt; PREFIX rdfs:     &lt;<a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a>&gt; </pre>

Option	Data Type	Description
		<pre> PREFIX xsd:      &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX owl:    &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX anzo: &lt;http://openanzo.org/ontologies/2008/07/Anzo#&gt; PREFIX zowl: &lt;http://openanzo.org/ontologies/2009/05/AnzoOwl#&gt; PREFIX dc:       &lt;http://purl.org/dc/elements/1.1/&gt; </pre>
<b>Result Clause</b>	N/A	The result clause defines the type of SPARQL query to run and the set of results to return, i.e., whether you want to read (SELECT or CONSTRUCT) from the source or ingest the data into Anzo (INSERT).
<b>GRAPH \${targetGraph}</b>	N/A	Include the GRAPH keyword and target graph parameter <code>\${targetGraph}</code> when writing an INSERT query to ingest data into a graphmart. Anzo automatically populates the query with the appropriate target URIs when the query runs.
<b>\${usingSources}</b>	N/A	Include the source graph parameter <code>\${usingSources}</code> when writing a "topdown" query that passes values from the data that is in the graphmart to the data source. Anzo automatically populates the query with the appropriate FROM clauses when the query runs. When passing literal values to the remote source, you do not need to include the source graph parameter. The <a href="#">SERVICE Clause</a> description below includes more information about passing input to data sources.
<b>SERVICE Clause</b>	N/A	<p>Include the SERVICE call <code>SERVICE [ TOPDOWN ]</code></p> <pre>&lt;http://cambridgesemantics.com/services/DataToolkit&gt;</pre> <p>to invoke the GDI service when you are running a SELECT, INSERT, or CONSTRUCT query that is not creating a view. When writing a CONSTRUCT query in a View Step, use the <code>DataToolkitView</code> service call, as described below in <a href="#">View SERVICE Clause</a>.</p> <p>Include the optional TOPDOWN keyword when you want to pass input</p>

Option	Data Type	Description
		values from the graphmart to the data source. When you include TOPDOWN in the service call, it indicates that the rest of the query produces values to send to the source. In this case, the GDI makes repeated calls to pass in each of the specified values and retrieve the data that is based on those values.
<b>View SERVICE Clause</b>	N/A	When writing a CONSTRUCT query that creates a view of the data (usually in a View Step), include the following SERVICE call: <code>SERVICE &lt;http://cambridgesemantics.com/services/DataToolkitView&gt; (\$ {targetGraph})</code> . Using the <code>DataToolkitView</code> call optimizes query execution because it tells the GDI to inspect the query and determine which filters to push to the data source. It also limits the result set and retrieves only the data that is needed, i.e., the source data is fully mapped but all of the mapped data is not necessarily returned.
<b>url</b>	string	This property specifies the file system location of the source file or directory of files. When specifying a directory (such as <code>s:url "/opt/shared-files/loads/"</code> ), the GDI loads all of the file formats it recognizes. To specify a directory but limit the number or type of files that are read, you can include the <a href="#">pattern</a> and/or <a href="#">maxDepth</a> properties.
<b>options</b>	RDF list	If additional connection information needs to be provided to access the file storage system, include the <code>options</code> property to list any storage-specific connection parameters. See <a href="#">File Storage Connection Options</a> for information about the supported properties for each storage type.
<b>pattern</b>	string	This property is used to specify a wildcard pattern for matching file names. For example, <code>s:pattern "common_prefix*.json"</code> . You can include one <code>s:pattern</code> property per FileSource. The GDI supports Unix file globbing syntax outside of parentheses. Within parentheses, full Java regular expression language is supported. For example, including <code>s:pattern "data/**/customer_*.json"</code> tells the GDI to load all files that match the

Option	Data Type	Description
		pattern "customer_*.json" from any number of subdirectories under the <code>data</code> directory. Similarly <code>s:pattern "(\d+)/transaction_*.json"</code> tells the GDI to load all files that match the pattern "transaction_*.json" in all subdirectories.
<b>maxDepth</b>	int	This property can be used to limit the directory traversal depth. By default, when <code>s:url</code> specifies a directory (and a <code>s:pattern</code> that limits that traversal depth is not specified), all subdirectories are processed. To process only the files in the top level directory, set <code>maxDepth</code> to 0 ( <code>s:maxDepth 0</code> ). To process the files in the top level directory plus the first-level subdirectories, set <code>maxDepth</code> to 1 ( <code>s:maxDepth 1</code> ), and so on.
<b>format</b>	RDF list	You can include the <code>format</code> property to add parameters that describe the source files. See <a href="#">File Source Format Options</a> for details about the supported parameters.
<b>mimetype</b>	string	This property can be included to specify the MIME type of the data.
<b>username</b>	string	If authentication is required to access the source, include this property to specify the user name.
<b>password</b>	string	This property lists the password for the given username.
<b>timeout</b>	int	This property can be used to specify the timeout (in milliseconds) to use for requests against the source. For example, <code>s:timeout 5000</code> configures a 5 second timeout.
<b>batching</b>	boolean or int	This property can be used to disable batching, or it can be used to change the default the batch size. By default, batching is set to 5000 ( <code>s:batching 5000</code> ). To disable batching, you can include <code>s:batching false</code> in the query. Typically users do not change the batching size. However, it can be useful to control the batch size when performing updates. To configure the size, include <code>s:batching int</code> in the query. For example, <code>s:batching</code>

Option	Data Type	Description
		3000.
<b>paging</b>	RDF list	This property can be used to configure paging so that the GDI can access large amounts of data across a number of smaller requests. For details about the <code>paging</code> property, see <a href="#">Pagination Options</a> .
<b>concurrency</b>	int or RDF list	<p>This property can be included to configure the maximum level of concurrency for the query. The value can be an integer, such as <code>s:concurrency 8</code>. If the value is an integer, it configures a maximum limit on the number of slices that can execute the query. For finer-grained control over the number of nodes and slices to use, concurrency can also be included as an object with <code>limit</code>, <code>nodes</code>, and/or <code>executorsPerNode</code> properties. For example, the following object configures a concurrency model that allows a maximum of 24 executors distributed across 4 nodes with 8 executors per node:</p> <pre>s:concurrency [   s:limit 24 ;   s:nodes 4 ;   s:executorsPerNode 8 ; ] ;</pre>
<b>rate</b>	int or string	<p>This property can be included to control the frequency with which a request is sent to the source. The limit applies to the number of requests a single slice can make. If you specify an integer for the rate, then the value is treated as the maximum number of requests to issue per minute. If you specify a string, you have more flexibility in configuring the rate. The sample values below show the types of values that are supported:</p> <pre>s:rate "90/minute" ; s:rate "90 per minute" ; s:rate "200000 every week" ; s:rate "10000 every 6 hours" ;</pre> <p>To enforce the rate limit, the GDI introduces a sleep between requests that</p>

Option	Data Type	Description
		<p>is equal to the rate delay. The more executing slices, the longer the rate delay needs to be to enforce the limit in aggregate.</p> <p>Given the example of <code>s:rate "90/minute"</code>, the GDI would optimize the concurrency and only use 1 slice for execution with a rate delay of 666ms between requests. If <code>s:rate "240/minute"</code>, the GDI would use 3 executors with a rate delay of 750ms between requests.</p>
<b>locale</b>	string	This property can be used to specify the locale to use when parsing locale-dependent data such as numbers, dates, and times.
<b>sampling</b>	int	This property can be used to configure the number of records in the source to examine for data type inferencing.
<b>selector</b>	string or RDF list	<p>This property can be used for JSON path extraction to traverse nested structures and target specific data. For example, <code>s:selector "projects"</code> targets the <code>projects</code> class of data. To express a hierarchy, use dot notation. For example, <code>s:selector "region.state.city"</code> navigates a hierarchy to target <code>city</code> data. For more information about binding components and the selector property, see <a href="#">Using Binding Trees and Selector Paths</a>.</p>
<b>model</b>	string	<p>This property defines the class (or table) name for the type of data that is generated from the specified data source. For example, <code>s:model "employees"</code>. Model is optional when querying a single source. If your query targets multiple sources, however, and you want to define resource templates (primary keys) and object properties (foreign keys), you must specify the model value for each source.</p>
<b>key</b>	string	<p>This property can be used to define the primary key column for the source file or table. This column is leveraged in a resource template for the instances that are created from the source. For example, <code>s:key ("EMPLOYEE_ID")</code>. For more information about <code>key</code>, see <a href="#">Data Linking Options</a>.</p>

Option	Data Type	Description
<b>reference</b>	RDF list	This property can be used to specify a foreign key column. The reference property is an RDF list that includes the <code>model</code> property to list the target table and a <code>using</code> property that defines the foreign key column. For more information about <code>reference</code> , see <a href="#">Data Linking Options</a> .
<b>formats</b>	RDF list	To give users control over the data types that are used when coercing strings to other types, this property can be included in GDI queries to define the desired types. In addition, it can be used to describe the formats of date and time values in the source to ensure that they are recognized and parsed to the appropriate date, time, and/or dateTime values. For details about the <code>formats</code> property, see <a href="#">Data Type Formatting Options</a> .
<b>normalize</b>	RDF list	To give users control over the labels and URIs that are generated, the GDI offers several options for normalizing the model and/or the fields that are created from the specified data source(s). For details about the <code>normalize</code> property, see <a href="#">Normalization Options</a> .
<b>count</b>	variable	If you want to turn the query into a COUNT query, you can include this property with a <code>?variable</code> to perform a count. For example, <code>s:count ?count</code> .
<b>offset</b>	int	This property can be used to offset the data that is returned by a number of rows.
<b>limit</b>	int	You can include this property to limit the number of results that are returned. <code>s:limit</code> maps to the SPARQL LIMIT clause.
<b>mapping_variable</b>	variable	The mapping variables, in <code>?mapping_variable (["binding"] [datatype] ["datetime_format"])</code> format, define the triple patterns to output. When the specified <code>?variable</code> matches the source column name, the GDI uses the variable as the source data selector. If you specify an alternate variable name, a <a href="#">binding</a> needs to be specified to map the new variable to the source. You also have the option to transform the data using



Option	Data Type	Description
		<p>the <a href="#">datatype</a> and <a href="#">datetime_format</a> options.</p> <div> <p><b>Tip</b></p> <p>See <a href="#">Hierarchical Bindings and Arrays</a> below for more information about configuring mapping variables and unpacking JSON files with nested objects and arrays.</p> </div>
<b>binding</b>	string	<p>The <code>binding</code> is a literal value that binds a <code>?mapping_variable</code> to a source column. If you specify a <code>?variable</code> that matches the source column name, then that variable name is the data selector and it is not necessary to specify a binding. If you specify an alternate variable name or there is a hierarchical path to the source column, then the binding is needed to map the new variable to that source column.</p> <p>For example for CSV, the following pattern simply binds the source column AIRLINE to the lowercase variable <code>?airline</code>: <code>?airline ("AIRLINE")</code>.</p> <div> <p><b>Note</b></p> <p>For FileSource, periods (<code>.</code>), forward slashes (<code>/</code>), and brackets (<code>[ ]</code>) are parsed as path notation. Therefore, if a source column name includes any of those characters they must be escaped in the binding. Use two backslashes (<code>\\</code>) as an escape character. For example, if a column name is <b>average/day</b>, the variable and binding pattern could be written as <code>?averagePerDay ("average\\/day")</code>.</p> </div>

Option	Data Type	Description
<b>datatype</b>	URI	<p>The <code>datatype</code> is the data type to convert the column to. If you do not specify a data type, the GDI infers the type. The GDI supports the following types:</p> <ul style="list-style-type: none"> <li>• <code>xsd:int</code></li> <li>• <code>xsd:long</code></li> <li>• <code>xsd:float</code></li> <li>• <code>xsd:double</code></li> <li>• <code>xsd:boolean</code></li> <li>• <code>xsd:time</code></li> <li>• <code>xsd:dateTime</code></li> <li>• <code>xsd:date</code></li> <li>• <code>xsd:gMonthDay</code></li> <li>• <code>xsd:gYearMonth</code></li> <li>• <code>xsd:duration</code></li> <li>• <code>xsd:dayTimeDuration</code></li> <li>• <code>xsd:yearMonthDuration</code></li> <li>• <code>xsd:gMonth</code></li> <li>• <code>xsd:anyURI</code></li> </ul>
<b>datetime_format</b>	string	<p>This option is used to specify the format to use for date and time data types. The GDI supports Java date and time formats. Specify days as "d," months as "M," and years as "y." For the time, specify "H" for hours, "m" for minutes, and "s" for seconds. For example, "<code>yyyyMMdd HH:mm:ss</code>" or "<code>ddMMMyy</code>" to display date values such as "01JAN19."</p> <div> <p><b>Note</b></p> <p>The GDI's default base year is <b>2000</b>. If the source data has years with only two digits, such as <code>02-04-99</code>, the GDI prepends <b>20</b> to the digits. The value <code>02-04-99</code> is parsed to <code>02-04-2099</code>. To specify an alternate base year to use for two-digit values, you can include the notation <code>^nnnn</code> (e.g., <code>^1900</code>) in the format value. For example, to set the base year to 1900 instead of 2000, use a format value such as <code>xsd:date "dd-MMM-yy^1900"</code> or <code>xsd:date "dd-MMM-yy^1990"</code>. When one of those values is specified, <code>02-04-99</code> is parsed to <code>02-04-1999</code>.</p> </div>

## Hierarchical Bindings and Arrays

When configuring the mapping variables in a query, the GDI provides syntax for unpacking JSON files with nested objects and arrays. One way to express hierarchies in queries is to use brackets ( [ ] ) to group objects into binding trees. For example, the WHERE clause snippet below organizes mapping variable objects into an `hourly/data` hierarchy by nesting the `?data` patterns inside the `?hourly [ ]` tree:

```
WHERE
{
  SERVICE <http://cambridgesemantics.com/services/DataToolkit>
  {
    ?data a s:FileSource;
    s:url "/mnt/data/json/weather.json" ;
    ?latitude (xsd:double) ;
    ?longitude (xsd:double) ;
    ?timezone (xsd:string) ;
    ?hourly
  [
    ?data
  [
    ?time (xsd:long) ;
    ?rainProbability ("precipProbability" xsd:double) ;
    ?temperature (xsd:double) ;
    ?feelsLike ("apparentTemperature" xsd:double) ;
    ?windSpeed (xsd:double) ;
  ] ;
  ] .
  }
}
```

When constructing object binding trees, if you choose to introduce the hierarchy with a variable name that is not an exact match to the source label, include a **selector** property to list the value from the source. For example, in the WHERE clause snippet below, `s:selector` is included to select `eventHeader` in the source as `?event` in the query and `statLocation` as `?location`.

```
WHERE
{
  SERVICE <http://cambridgesemantics.com/services/DataToolkit>
```

```

{
  ?data a s:FileSource ;
  s:url "/mnt/data/json/part_1.json" ;
  ?event
[
    s:selector "eventHeader" ;
    ?eventId (xsd:string) ;
    ?eventName (xsd:string) ;
    ?eventVersion (xsd:string) ;
    ?eventTime (xsd:dateTime) ;
  ] ;
  ?location
[
    s:selector "statLocation" ;
    ?locationId (xsd:string) ;
    ?lineNo (xsd:int) ;
    ?statNo (xsd:int) ;
    ?statId (xsd:int) ;
  ] .
}
}

```

As an alternative to grouping objects in binding trees, the **selector** property also supports using dot notation to specify paths. For example, the WHERE clause snippet below rewrites the first example query to express the same `hourly/data` hierarchy as a path in the `s:selector` value:

```

WHERE
{
  SERVICE <http://cambridgesemantics.com/services/DataToolkit>
  {
    ?data a s:FileSource;
    s:url "/mnt/data/json/weather.json" ;
    ?latitude (xsd:double) ;
    ?longitude (xsd:double) ;
    ?timezone (xsd:string) ;
    s:selector: "hourly.data" ;
    ?time (xsd:long) ;
    ?rainProbability ("precipProbability" xsd:double) ;
    ?temperature (xsd:double) ;
  }
}

```

```

        ?feelsLike ("apparentTemperature" xsd:double) ;
        ?windSpeed (xsd:double) .
    }
}

```

In addition to object binding trees and selectors, the GDI offers additional syntax for reading or ingesting JSON sources with nested objects and arrays. For example, following the JSON sample file below is a query that captures each value in the arrays:

```

{
  "payload" :
  {
    "IBP_IndEvent_MSR" :
    {
      "unit" : "ms",
      "value" : [ 0, 1 ]
    },
    "IBP_IndEvent_RMF" :
    {
      "unit" : "-",
      "value" : [ 0.012, 1.398, 3.1415 ]
    }
  }
}

```

To read the JSON file above, the following query uses an object binding (`?values [ ]`) to drill down to the value arrays in the source. An `@` selector is specified in the `?value` variable binding (`?value ("@" xsd:double)`) to retrieve each of the array values. For an array of primitive values, the `@` selector captures each value in the array. If the source value was an array of objects, the `@` selector would retrieve a JSON representation for each object in the array. In addition to creating a new binding context for the primitive array values, the `?values` object binding also includes `?index ("!array::index")` to capture the index array with the primitive value.

```

PREFIX s:    <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
SELECT *
WHERE {
  SERVICE <http://cambridgesemantics.com/services/DataToolkit> {

```

```

?data a s:FileSource ;
s:url "/mnt/data/json/array-index.json" ;
s:selector "payload.*" ;
?unit (xsd:string) ;
?values [
    s:selector "value" ;
    ?value ("@" xsd:double) ;
    ?index ("!array::index") ;
] .
}
}

```

The results of the query are shown below:

unit	value	index
-----	-----	-----
ms	0	0
ms	1	1
-	0.012	0
-	1.398	1
-	3.1415	2

If you do not want to retrieve all of the values in an array, you can include the specific index number to retrieve instead of using the @ symbol. In the variable binding, the index number is appended in brackets ([ ]) to the binding column name. For example, the following variable binding retrieves the second index value (the third value in the array) from a "projects" array: `?project ("projects[2]")`. The next example uses the following JSON file:

```

{
  "field1" : "value1" ,
  "arrayfield" : [
    "arrayvalue1",
    "arrayvalue2"
  ]
}

```

To retrieve only the second value in the array, the following query appends the index value 1 to the array column name, `arrayfield`:

```

PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>
SELECT *
WHERE {
    SERVICE <http://cambridgesemantics.com/services/DataToolkit> {
        ?json a s:FileSource ;
        s:url "/mnt/data/json/array-index-2.json" ;
        ?field1 (xsd:string) ;
        ?arrayval ("arrayfield[1]" xsd:string) .
    }
}

```

The results of the query are shown below:

```

field1      | arrayval
-----+-----
value1      | arrayvalue2

```

## Capturing Property Keys

In GDI Generator queries, the names of property keys can be captured from files by including a variable as the `s:selector` and using the same variable as the `s:key`. For example, the GDI query below ingests the following simple JSON file.

```

# company.json
{
  "AAPL": {
    "name": "Apple Corp"
  },
  "MSFT": {
    "name": "Microsoft"
  },
  "IBM": {
    "name": "IBM"
  }
}

```

In the query, the keys "AAPL," "MSFT," and "IBM" are selected as the `?TickerSymbol` variable and the key is set to the same value.

```

PREFIX s: <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

INSERT ${targetGraph} {
    ?s ?p ?o .
}
}
WHERE {
    SERVICE <http://cambridgesemantics.com/services/DataToolkit> {
        ?data a s:FileSource ;
        s:url "/opt/shared/data/company.json" ;
        s:selector "?TickerSymbol" ;
        s:key (?TickerSymbol) ;
        s:model "Company" ;
        ?TickerSymbol (xsd:string) ;
        ?name (xsd:string) .

        ?rdf a s:RdfGenerator, s:OntologyGenerator;
        s:as (?s ?p ?o) ;
        s:ontology <http://cambridgesemantics.com/ontologies/company> ;
        s:base ${targetGraph} .
    }
}

```

Selecting the predicates and objects from the graph shows the tickerSymbol predicate and value.

```

SELECT ?p ?o
${usingSources}
WHERE { ?s ?p ?o . }
ORDER BY desc(?o)

```

p	o
http://anzograph.com/ontologies/company#Company.name	Microsoft
http://anzograph.com/ontologies/company#Company.tickerSymbol	MSFT
http://anzograph.com/ontologies/company#Company.name	IBM
http://anzograph.com/ontologies/company#Company.tickerSymbol	IBM
http://anzograph.com/ontologies/company#Company.name	Apple Corp



```
http://anzograph.com/ontologies/company#Company.tickerSymbol | AAPL
```

```
...
```

## Query Examples

The example query below reads a JSON file that contains data about weather. Since the file is hierarchical, the `s:selector` property is included to specify the path to `data` in the `hourly/data` hierarchy:

```
PREFIX s:    <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
SELECT *
WHERE
{
  SERVICE <http://cambridgesemantics.com/services/DataToolkit>
  {
    ?data a s:FileSource;
    s:url "/mnt/data/json/weather.json" ;
    ?latitude (xsd:double) ;
    ?longitude (xsd:double) ;
    ?timezone (xsd:string) ;
    s:selector: "hourly.data" ;
    ?time (xsd:long) ;
    ?rainProbability ("precipProbability" xsd:double) ;
    ?temperature (xsd:double) ;
    ?feelsLike ("apparentTemperature" xsd:double) ;
    ?windSpeed (xsd:double) .
  }
}
```

The following example query ingests data from a JSON file that contains data about the New York Times best selling books.

```
PREFIX s:    <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX books: <http://cambridgesemantics.com/ontologies/NYT_Bestsellers_
Ontology#>
INSERT {
  GRAPH ${targetGraph}{
    ?book a books:Book ;
```

```

        books:p_Title ?title ;
        books:p_Description ?description ;
        books:p_Author ?author ;
        books:p_Publisher ?publisher ;
        books:p_Date ?rawdate .
    }
}
WHERE {
    SERVICE <http://cambridgesemantics.com/services/DataToolkit>
    {
        ?data a s:FileSource ;
        s:url "/mnt/data/json/nyt_best_sellers.json" ;
        ?title () ;
        ?author () ;
        ?description () ;
        ?publisher () ;
        ?price() ;
        ?rawdate ("bestsellers_date.$date.$numberLong") .
    }
    BIND(IRI(CONCAT("http://cambridgesemantics.com/ontologies/NYT_Bestsellers_
Ontology/", ENCODE_FOR_URI(?title))) AS ?book) .
}

```

A snippet of the file's contents is shown below:

```

{
  "_id": {
    "$oid": "5b4aa4ead3089013507db18b"
  },
  "bestsellers_date": {
    "$date": {
      "$numberLong": "1211587200000"
    }
  },
  "published_date": {
    "$date": {
      "$numberLong": "1212883200000"
    }
  },
}

```

```

    "amazon_product_url": "http://www.amazon.com/Odd-Hours-Dean-
Koontz/dp/0553807056?tag=NYTBS-20",
    "author": "Dean R Koontz",
    "description": "Odd Thomas, who can communicate with the dead, confronts evil
forces in a California coastal town.",
    "price": {
        "$numberInt": "27"
    },
    "publisher": "Bantam",
    "title": "ODD HOURS",
    "rank": {
        "$numberInt": "1"
    },
    "rank_last_week": {
        "$numberInt": "0"
    },
    "weeks_on_list": {
        "$numberInt": "1"
    }
}

```

## Related Topics

[Bindings and Hierarchies](#)

[Generating a Graph and Ontology with a Direct Load Step](#)

## File Source Format Options

For file sources, you can include the **format** property to list additional parameters that describe the source. The supported format parameters are described below.

```

s:format [
    s:delimiter "string" ;
    s:headers boolean ;
    s:columns "string" ;
    s:start int ;
    s:skip int ;
    s:comment "string" ;
    s:quote "string" ;
    s:escape "string" ;

```

```
s:maxColumns int ;
s:segment boolean ;
] ;
```

Option	Data Type	Description
<b>delimiter</b>	string	This property specifies the string that is used to delimit columns in the file(s). For example, <code>s:delimiter " " </code> .
<b>headers</b>	boolean	This property indicates whether or not the file(s) include headers. By default the headers value is <code>true (s:headers true)</code> . For files that do not have headers, specify <code>s:headers false</code> .
<b>columns</b>	string	If you want the GDI to target only certain columns in the source file(s), you can include the <b>columns</b> property to list the names of columns to include. The value is a single string that is a comma-separated list. For example, <code>s:columns "employee_id, name, address, start date, title"</code> .
<b>start</b>	int	If the file includes headers that take up more than one row, include the <code>start</code> property to specify the row number where the data starts to exclude headers. For example, <code>s:start 8</code> .
<b>skip</b>	int	This property can be used to specify the number of rows/records to skip before reading or ingesting the file(s). By default, skip is set to 0 ( <code>s:skip 0</code> ).
<b>comment</b>	string	This property specifies the string that is used as the comment character in the file(s). The comment value is set to <code>#</code> by default ( <code>s:comment "#"</code> ).
<b>quote</b>	string	This property is used to specify the string that is used as the quote character.
<b>escape</b>	string	This property is used to specify the escape string that is used in the file

Option	Data Type	Description
		(s). For example, <code>s:escape "\\</code> .
<b>maxColumns</b>	int	This property can be used to set a limit on the maximum number of columns to read or ingest. The maxColumns property is set to -1 (unlimited) by default ( <code>s:maxColumns -1</code> ).
<b>segment</b>	boolean	This property indicates whether or not the file(s) can be segmented. For example, some CSV files that contain embedded newlines cannot be segmented. By default, segment is not set.

## File Storage Connection Options

If you are querying a **FileSource** and additional connection information needs to be provided to access the file storage system, include the **options** property in the query and define the necessary storage-specific connection parameters. The parameters that the GDI supports for each type of storage system are pulled directly from the Java API for that system. The supported properties for each storage type are listed below.

- [Amazon S3](#)
- [FTP & FTPS](#)
- [Google Cloud Storage](#)
- [HDFS](#)
- [SFTP](#)
- [WebDAV](#)

# Amazon S3

```
s:options [  
  s:accessKey "string" ;  
  s:region "string" ;  
  s:secretKey "string" ;  
  s:serviceName "string" ;  
  s:sessionToken "string" ;  
  s:createBucket boolean ;  
  s:disableChunkedEncoding boolean ;  
  s:serverSideEncryption boolean ;  
  s:useHttps boolean ;  
] ;
```

Option	Data Type	Description
accessKey	string	The accessKey property can be included to specify the access key.
region	string	The region property can be included to specify the region.
secretKey	string	The secretKey property can be included to specify the secret key.
serviceName	string	For connections to AWS service endpoints, the serviceName property can be included to specify the service name.
sessionToken	string	The sessionToken property can be included to specify the session token.
createBucket	boolean	Refer to the S3 API documentation.
disableChunkedEncoding	boolean	For increased performance, Amazon S3 requests use

Option	Data Type	Description
		chunked encoding by default. To disable chunked encoding, you can include <code>s:disableChunkedEncoding true</code> in the query.
<b>serverSideEncryption</b>	boolean	Refer to the S3 API documentation.
<b>useHttps</b>	boolean	Refer to the S3 API documentation.

---

## FTP & FTPS

```
s:options [  
  s:autodetectUtf8 boolean ;  
  s:connectTimeout int ;  
  s:controlEncoding "string" ;  
  s:dataTimeout int ;  
  s:defaultDateFormat "string" ;  
  s:entryParser "string" ;  
  s:fileType "string" ;  
  s:passiveMode boolean ;  
  s:proxy "string" ;  
  s:recentDateFormat "string" ;  
  s:remoteVerification boolean ;  
  s:serverLanguageCode "string" ;  
  s:serverTimeZoneId "string" ;  
  s:shortMonthNames "string" ;  
  s:socketTimeout int ;  
  s:userDirIsRoot boolean ;  
  s:dataChannelProtectionLevel "string" ;  
  s:ftpsMode "string" ;  
  s:keyManager "string" ;  
  s:trustManager "string" ;  
] ;
```

Option	Data Type	Description
<b>autodetectUtf8</b>	boolean	For FTP connections, the <code>autodetectUtf8</code> property can be included to indicate whether the FTP server is set to UTF-8 mode or Auto-detect encoding.
<b>connectTimeout</b>	int	For FTP connections, you can include the <code>connectTimeout</code> property to specify the maximum number of seconds to hold a connection before timing out.



Option	Data Type	Description
<b>controlEncoding</b>	string	Refer to the FTP API documentation.
<b>dataTimeout</b>	int	For FTP connections, you can include the <code>dataTimeout</code> property to specify the maximum number of seconds to transfer data before timing out.
<b>defaultDateFormat</b>	string	Refer to the FTP API documentation.
<b>entryParser</b>	string	Refer to the FTP API documentation.
<b>fileType</b>	string	Refer to the FTP API documentation.
<b>passiveMode</b>	boolean	For FTP connections, the <code>passiveMode</code> property can be included to indicate whether the data transfer mode is passive or active. If you use passive mode, set <code>passiveMode</code> to true ( <code>s:passiveMode true</code> ).
<b>proxy</b>	string	If you are using an FTP proxy, include the <code>proxy</code> property to specify the proxy connection details.
<b>recentDateFormat</b>	string	Refer to the FTP API documentation.
<b>remoteVerification</b>	boolean	For FTP connections, the <code>remoteVerification</code> property can be included to indicate whether remote authentication is enabled. If you use remote authentication, set <code>remoteVerification</code> to true ( <code>s:remoteVerification true</code> ).
<b>serverLanguageCode</b>	string	If the FTP server language is not set to English, include the <code>serverLanguageCode</code> property to specify the language code for the server. For example, <code>s:serverLanguageCode "ES"</code> .

Option	Data Type	Description
<b>serverTimeZoneId</b>	string	For FTP connections, the <code>serverTimeZoneId</code> property can be included to specify the timezone ID for the server.
<b>shortMonthNames</b>	string	Refer to the FTP API documentation.
<b>socketTimeout</b>	int	For FTP connections, you can include the <code>socketTimeout</code> property to specify the maximum number of seconds to transfer data before timing out.
<b>userDirIsRoot</b>	boolean	Refer to the FTP API documentation.
<b>dataChannelProtectionLevel</b>	string	For FTPS connections, the <code>dataChannelProtectionLevel</code> property specifies the Data Channel Protection Level for the server.
<b>ftpsMode</b>	string	For FTPS connections, the <code>ftpsMode</code> property specifies whether the FTPS is in implicit or explicit mode.
<b>keyManager</b>	string	For FTPS connections, the <code>keyManager</code> property specifies the KeyManager value for making an SSL connection to the server.
<b>trustManager</b>	string	For FTPS connections, the <code>trustManager</code> property specifies the TrustManager value for the SSL connection to the server.

---

# Google Cloud Storage

```
s:options [  
  s:serviceAccountKey "string" ;  
] ;
```

Option	Data Type	Description
serviceAccountKey	string	For connections to GCS, the serviceAccountKey property can be included to specify the key for the service account.

---

## HDFS

```
s:options [  
  s:configName "string" ;  
  s:configPath "string" ;  
  s:configURL "string" ;  
] ;
```

Option	Data Type	Description
configName	string	For connections to HDFS, the <code>configName</code> property can be included to specify the name of the configuration file to read.
configPath	string	For connections to HDFS, the <code>configPath</code> property can be included to list the path to the specified configuration file.
configURL	string	Refer to the HDFS API documentation.

---

## SFTP

```
s:options [  
  s:compression "string" ;  
  s:configRepository "string" ;  
  s:fileNameEncoding "string" ;  
  s:identityProvider "string" ;  
  s:identityRepositoryFactory "string" ;  
  s:keyExchangeAlgorithm "string" ;  
  s:knownHosts "string" ;  
  s:loadOpenSSHConfig boolean ;  
  s:preferredAuthentications "string" ;  
  s:sessionTimeout int ;  
  s:strictHostKeyChecking "string" ;  
  s:userInfo "string" ;  
] ;
```

Option	Data Type	Description
<b>compression</b>	string	Refer to the SFTP API documentation.
<b>configRepository</b>	string	Refer to the SFTP API documentation.
<b>fileNameEncoding</b>	string	Refer to the SFTP API documentation.
<b>identityProvider</b>	string	Refer to the SFTP API documentation.
<b>identityRepositoryFactory</b>	string	Refer to the SFTP API documentation.
<b>keyExchangeAlgorithm</b>	string	For SFTP connections, you can include the <code>keyExchangeAlgorithm</code> property to specify the key exchange algorithm to use.
<b>knownHosts</b>	string	Refer to the SFTP API documentation.

Option	Data Type	Description
<b>loadOpenSSHConfig</b>	boolean	For SFTP connections, the <code>loadOpenSSHConfig</code> property indicates whether to read the <code>~/.ssh/config</code> file.
<b>preferredAuthentications</b>	string	For SFTP connections, the <code>preferredAuthentications</code> property can be included to specify the authentication order to use.
<b>sessionTimeout</b>	int	For SFTP connections, you can include the <code>sessionTimeout</code> property to specify the maximum number of seconds to leave the session open before timing out.
<b>strictHostKeyChecking</b>	string	For SFTP connections, you can include the <code>strictHostKeyChecking</code> property to specify how host keys are checked.
<b>userInfo</b>	string	Refer to the SFTP API documentation.

---

# WebDAV

```
s:options [  
  s:creatorName "string" ;  
  s:versioning boolean ;  
] ;
```

Option	Data Type	Description
creatorName	string	For WebDAV connections, the <code>creatorName</code> property can be included to add a description of the creator of the resource.
versioning	boolean	Refer to the WebDAV API documentation.

## Related Topics

[Querying File Sources](#)

[Generating a Graph and Ontology with a Direct Load Step](#)

[Getting Started with GDI Queries](#)

## GDI Property Reference

This topic describes the Graph Data Interface (GDI) properties that are available to use in queries. The first section describes the options that are available regardless of data source type, and the second section describes the source-specific options.

- [Universal Properties](#)
- [Source-Specific Properties](#)

## Universal Properties

The table below lists the properties that are valid in queries against all data source types.

Option	Data Type	Description
batching	boolean or int	This property can be used to disable batching, or it can be used to change the default the batch size. By default, batching is set to 5000 ( <code>s:batching 5000</code> ).

Option	Data Type	Description
		<p>To disable batching, you can include <code>s:batching false</code> in the query. Typically users do not change the batching size. However, it can be useful to control the batch size when performing updates. To configure the size, include <code>s:batching int</code> in the query. For example, <code>s:batching 3000</code>.</p>
<b>concurrency</b>	int or RDF list	<p>This property can be included to configure the maximum level of concurrency for the query. The value can be an integer, such as <code>s:concurrency 8</code>. If the value is an integer, it configures a maximum limit on the number of slices that can execute the query. For finer-grained control over the number of nodes and slices to use, concurrency can also be included as an object with <code>limit</code>, <code>nodes</code>, and/or <code>executorsPerNode</code> properties. For example, the following object configures a concurrency model that allows a maximum of 24 executors distributed across 4 nodes with 8 executors per node:</p> <pre>s:concurrency [   s:limit 24 ;   s:nodes 4 ;   s:executorsPerNode 8 ; ] ;</pre>
<b>count</b>	variable	<p>If you want to turn the query into a COUNT query, you can include this property with a <code>?variable</code> to perform a count. For example, <code>s:count ?count</code>.</p>
<b>errors</b>	boolean	<p>Controls whether the GDI ignores errors (such as query or file errors) or stops processing the query when an error is encountered. This property is set to <code>true</code> by default (<code>s:errors true</code>). Processing stops when an error is encountered. To ignore errors, you can include <code>s:errors false</code>.</p>
<b>formats</b>	RDF list	<p>To give users control over the data types that are used when coercing strings to other types, this property can be included in GDI queries to define the desired types. In addition, it can be used to describe the formats of date and time values in the source to ensure that they are recognized and parsed to the appropriate date, time, and/or dateTime values. For details about the <code>formats</code> property, see</p>



Option	Data Type	Description
<a href="#">Data Type Formatting Options.</a>		
<b>key</b>	string	This property can be used to define the primary key column for the source file or table. This column is leveraged in a resource template for the instances that are created from the source. For example, <code>s:key ("EMPLOYEE_ID")</code> . For more information about <code>key</code> , see <a href="#">Data Linking Options</a> .
<b>limit</b>	int	You can include this property to limit the number of results that are returned. <code>s:limit</code> maps to the SPARQL LIMIT clause.
<b>locale</b>	string	This property can be used to specify the locale to use when parsing locale-dependent data such as numbers, dates, and times.
<b>model</b>	string	This property defines the class (or table) name for the type of data that is generated from the specified data source. For example, <code>s:model "employees"</code> . Model is optional when querying a single source. If your query targets multiple sources, however, and you want to define resource templates (primary keys) and object properties (foreign keys), you must specify the model value for each source.
<b>normalize</b>	boolean and/or RDF list	To give users control over the labels and URIs that are generated, the GDI offers several options for normalizing the model and/or the fields that are created from the specified data source(s). For details about the <code>normalize</code> property, see <a href="#">Normalization Options</a> .
<b>offset</b>	int	This property can be used to offset the data that is returned by a number of rows.
<b>paging</b>	RDF list	This property can be used to configure paging so that the GDI can access large amounts of data across a number of smaller requests. For details about the <code>paging</code> property, see <a href="#">Pagination Options</a> .
<b>password</b>	string	This property lists the password for the given username.

Option	Data Type	Description
<b>rate</b>	int or string	<p>This property can be included to control the frequency with which a request is sent to the source. The limit applies to the number of requests a single slice can make. If you specify an integer for the rate, then the value is treated as the maximum number of requests to issue per minute. If you specify a string, you have more flexibility in configuring the rate. The sample values below show the types of values that are supported:</p> <pre>s:rate "90/minute" ; s:rate "90 per minute" ; s:rate "200000 every week" ; s:rate "10000 every 6 hours" ;</pre> <p>To enforce the rate limit, the GDI introduces a sleep between requests that is equal to the rate delay. The more executing slices, the longer the rate delay needs to be to enforce the limit in aggregate.</p> <p>Given the example of <code>s:rate "90/minute"</code>, the GDI would optimize the concurrency and only use 1 slice for execution with a rate delay of 666ms between requests. If <code>s:rate "240/minute"</code>, the GDI would use 3 executors with a rate delay of 750ms between requests.</p>
<b>reference</b>	RDF list	<p>This property can be used to specify a foreign key column. The reference property is an RDF list that includes the <code>model</code> property to list the target table and a <code>using</code> property that defines the foreign key column. For more information about <code>reference</code>, see <a href="#">Data Linking Options</a>.</p>
<b>sampling</b>	int	<p>This property can be used to configure the number of records in the source to examine for data type inferencing.</p>
<b>selector</b>	string or RDF list	<p>This property can be used as a binding component to identify the path to the source objects. For example, <code>s:selector "Sales.SalesOrderHeader"</code> targets the SalesOrderHeader table in the Sales schema. For more information about binding components and the selector property, see <a href="#">Using Binding Trees and Selector Paths</a>.</p>

Option	Data Type	Description
<b>timeout</b>	int	This property can be used to specify the timeout (in milliseconds) to use for requests against the source. For example, <code>s:timeout 5000</code> configures a 5 second timeout.
<b>url</b>	string	<p>This property specifies the URL for the data source, such as the database URL, Elasticsearch URL, or HTTP endpoint URL. For file-based sources, the <code>url</code> property specifies the file system location of the source file or directory of files. When specifying a directory (such as <code>s:url "/opt/shared-files/loads/"</code>), the GDI loads all of the file formats it recognizes. To specify a directory but limit the number or type of files that are read, you can include the <a href="#">pattern</a> and/or <a href="#">maxDepth</a> properties.</p> <div> <p><b>Important</b></p> <p>For security, it is a best practice to reference connection information (such as the url, username, and password) from a Query Context so that the sensitive details are abstracted from any requests. In addition, using a Query Context makes connection details reusable across queries. See <a href="#">Using Query Contexts in Queries</a> for more information. For example, the triple patterns below reference keys from a Query Context:</p> <pre>?data a s:DbSource ;   s:url "   {{@db.eca4bfa83481f3638b93ab5fdf93ff9a.url}}" ;   s:username "   {{@db.eca4bfa83481f3638b93ab5fdf93ff9a.user}}" ;   s:password "   {{@db.eca4bfa83481f3638b93ab5fdf93ff9a.password}}" ;</pre> </div>
<b>username</b>	string	If authentication is required to access the source, include this property to specify the user name.

## Source-Specific Properties

The options described in this section are grouped by data source type. For more information about writing queries against different sources, see [GDI Usage by Data Source Type](#).

- [DbSource Properties](#)
- [FileSource Properties](#)
- [HttpSource Properties](#)
- [ElasticSource Properties](#)

### DbSource Properties

The table below lists the properties that are available for queries against database data sources. For more information about database sources, see [Querying a Database Source](#).

Option	Data Type	Description
<b>database</b>	string	This property can be used to specify the database to target in the source if the database is not listed in the <code>s:url</code> or <code>s:selector</code> strings.
<b>driver</b>	string	This property can be included to specify the JDBC driver to use.
<b>orderBy</b>	string, variable, list	You can include this property to order the result set by a field name, a bound variable, or a list of names or bound variables.
<b>maxConnections</b>	int	This property can be used to set a limit on the maximum number of active connections to the source. For example, <code>s:maxConnections 16</code> sets the limit to 16 connections. The default value is 10.
<b>partitionBy</b>	string, variable, list	The GDI attempts to partition queries automatically across the available cores (slices) in AnzoGraph. To determine how to partition the query, the GDI uses metadata from the source database. It looks for any column in an index, preferring the primary key column if it is interpolable. However, it only considers the first column in any index on the table. After determining the partition column, the GDI does a MIN/MAX on the column as well as a basic

Option	Data Type	Description
		sizing query. To specify which column or columns the GDI should partition on, you can include the <code>partitionBy</code> property in the query. The property supports a list of source field names, bound variables, or the object <code>s:auto</code> , which forces the GDI to partition the data when the source does not define partitioning metadata.
<b>property</b>	RDF list	<p>This property can be included to list any JDBC driver-specific connection properties. To incorporate <code>property</code>, use the following syntax:</p> <pre>s:property [ s:name "custom_driver_property_name" ; s:value "custom_value" ]</pre>
<b>query</b>	string	<p>If you want to access the source data by running an SQL query, you can include this property to specify the query string to run. The language does not have to be SQL if the source supports another language. However, some GDI features where the query is dynamically altered may not work with a non-SQL language. Including <code>{{?variable}}</code> substitutions is supported within <code>s:query</code> strings.</p> <div> <p><b>Note</b></p> <p>If you include <code>s:query</code> without also specifying <code>table</code> or <code>partitionBy</code>, the GDI may not partition the query and query execution may be less performant than if the partition column was specified. When using <code>s:query</code>, specifying the table in <code>s:table</code> and the column to partition the table on in <code>s:partitionBy</code> is a good practice, especially when querying large tables.</p> </div>
<b>schema</b>	string	This property can be included to specify the target schema to query. If you include <code>s:schema "schema_name"</code> without specifying <code>s:table</code> (described below) or <code>s:query</code> , all tables in the schema are queried.
<b>table</b>	string	This property can be included to specify the target table or tables for the query.

## FileSource Properties

The table below lists the properties that are available for queries against file-based data sources. For more information about file sources, see [Querying File Sources](#).

Option	Data Type	Description
<b>format</b>	RDF list	You can include the <code>format</code> property to add parameters that describe the source files. See <a href="#">File Source Format Options</a> for details about the supported parameters.
<b>maxDepth</b>	int	This property can be used to limit the directory traversal depth. By default, when <code>s:url</code> specifies a directory (and a <code>s:pattern</code> that limits that traversal depth is not specified), all subdirectories are processed. To process only the files in the top level directory, set <code>maxDepth</code> to 0 ( <code>s:maxDepth 0</code> ). To process the files in the top level directory plus the first-level subdirectories, set <code>maxDepth</code> to 1 ( <code>s:maxDepth 1</code> ), and so on.
<b>mimetype</b>	string	This property can be included to specify the MIME type of the data. If you are querying TSV files that do not have a <code>.tsv</code> file extension, include the <code>mimetype</code> property with a value of <code>text/tsv</code> ( <code>s:mimetype "text/tsv"</code> ).
<b>options</b>	RDF list	If additional connection information needs to be provided to access the file storage system, include the <code>options</code> property to list any storage-specific connection parameters. See <a href="#">File Storage Connection Options</a> for information about the supported properties for each storage type.
<b>pattern</b>	string	This property can be used to specify a wildcard pattern for matching file names. For example, <code>s:pattern "common_prefix*.csv"</code> . You can include one <code>s:pattern</code> property per <code>FileSource</code> . The GDI supports Unix file globbing syntax outside of parentheses. Within parentheses, full Java regular expression language is supported. For example, including <code>s:pattern "data/**/customer_*.csv"</code> tells the GDI to load all files that match the pattern <code>"customer_*.csv"</code> from any number of subdirectories under the <code>data</code> directory. Similarly <code>s:pattern "(\\d+)/transaction_*.csv"</code> tells the

Option	Data Type	Description
		GDI to load all files that match the pattern "transaction_*.csv" in all subdirectories.

## HttpSource Properties

The table below lists the properties that are available for queries against HTTP data sources. For more information about HTTP sources, see [Querying an HTTP Source](#).

Option	Data Type	Description
<b>authorization</b>	RDF list	<p>This property specifies the type of authorization to use and the values for authentication. The options are <a href="#">BearerToken</a>, <a href="#">AWSSignature</a>, or <a href="#">BasicAuth</a>.</p> <pre>s:authorization [ a s:BearerToken   s:AWSSignature   s:BasicAuth ] ;</pre>
<b>AWSSignature</b>	RDF list	<p>For authorization to AWS service endpoints, specify this property and include the appropriate authentication properties from the list below:</p> <ul style="list-style-type: none"> <li>• <b>accessKey</b>: Include this property to specify the AWS access key.</li> <li>• <b>region</b>: Include this property to specify the AWS region.</li> <li>• <b>secretKey</b>: Include this property to specify the AWS secret key.</li> <li>• <b>serviceName</b>: Include this property to specify the AWS service name.</li> <li>• <b>sessionToken</b>: Include this property to specify the AWS session token.</li> </ul> <pre>s:authorization [   a s:AWSSignature ; s:accessKey "string" ;   s:region "string" ; s:secretKey "string" ;   s:serviceName "string" ; s:sessionToken "string" ; ] ;</pre>
<b>BasicAuth</b>	RDF list	Specify this property when basic authentication is used, and include the username and password properties.

Option	Data Type	Description
		<pre>s:authorization [ a s:BasicAuth ;   s:username "string" ;   s:password "string" ; ] ;</pre>
<b>BearerToken</b>	string	<p>Specify this property when a bearer token is used for authentication, and include the token property.</p> <pre>s:authorization [ a s:BearerToken ;   s:token "string" ] ;</pre>
<b>content</b>	string or RDF list	<p>This property can be included to send content to the source in the body of the request. For example, <code>content</code> can be a SPARQL query, JSON arrays, or a list of key-value pairs. Content can also be configured with an inline object (blank node) that gets translated to JSON. For more information, see <a href="#">Mapping the Content Property to JSON</a>.</p>
<b>contentType</b>	string	<p>Include this property to specify the content type of the body of the request. For example, <code>s:contentType "application/sparql-query"</code> or <code>s:contentType "application/json"</code>.</p>
<b>encoding</b>	string	<p>When targeting a file, you can include this property to specify the character encoding used by the file. The default value is <code>s:encoding "utf8"</code>.</p>
<b>form</b>	RDF list	<p>To send data to the HTTP endpoint, you can use this property to post the data. Form is a list of name-value pairs. When including <code>s:form</code>, you must also include <code>s:contentType "multipart/form-data"</code>. The GDI sends the form object as an <code>application/x-www-form-urlencoded</code> string that contains the specified parameters. The query snippet below shows an example that incorporates <code>s:form</code>.</p> <pre>SERVICE</pre>



Option	Data Type	Description
		<pre> &lt;http://cambridgesemantics.com/services/DataToolkit&gt; {   VALUES (?reviews_text) {     ("Horrible, terrible, will never use again.")     ("Wonderful, magnificent, will recommend to everyone!")     ("Simply OK. I might purchase this product again.") }    ?data a s:HttpSource ;     s:url "https://api.meaningcloud.com/sentiment-2.1" ;     s:contentType "multipart/form-data" ;     s:form [ s:name "key" ; s:value "9eab751142..." ],             [ s:name "lang" ; s:value "auto" ],             [ s:name "txt" ; s:value ?reviews_text ] ;     ?confidence ();     ?score_tag ();     ?subjectivity ();     ?irony ();     ?agreement () . } </pre>
<b>format</b>	RDF list	If the data is file-based, you can include the <code>format</code> property to add parameters that describe the source. See <a href="#">File Source Format Options</a> for details about the supported parameters.
<b>header</b>	RDF list	<p>You can use this property to specify name-value pairs to include as headers in the request. For example:</p> <pre>s:header [ s:name "Accept" ; s:value "application/json" ]</pre> <p>If you are creating a view, you can include variables in the <code>s:header</code> list. When another query is run against a view with variables, that query can map the variables through the view by including predicates in the CONSTRUCT clause.</p>
<b>method</b>	string	You can include this property to specify the HTTP method. For example, <code>s:method "GET"</code> or <code>s:method "POST"</code> .

Option	Data Type	Description
<b>mimetype</b>	string	You can include this property to specify the MIME type of the source. For example, <code>s:mimetype "text/html"</code> .
<b>orderBy</b>	string, variable, list	You can include this property to order the result set by a field name, a bound variable, or a list of names or bound variables.
<b>parameter</b>	RDF list	<p>You can include this property to list any URL parameters as name-value pairs. For example, the <code>s:parameter</code> property below adds <code>format</code> to return results in CSV format and the <code>named-graph-uri</code> parameter to target a specific layer in a graphmart.</p> <pre>s:parameter [ s:name "format" ; s:value "csv" ] ,              [ s:name "named-graph-uri" ; s:value                "http://cambridgesemantics.com/Layer/d541..." ] ;</pre> <p>If you are creating a view, you can include variables in the <code>s:parameter</code> list. When another query is run against a view with variables, that query can map the variables through the view by including predicates in the CONSTRUCT clause.</p>
<b>partition By</b>	string, variable, list	The GDI attempts to partition queries automatically across the available cores (slices) in AnzoGraph. To determine how to partition the query, the GDI uses metadata from the source. It looks for any column in an index, preferring the primary key column if it is interpolable. However, it only considers the first column in any index on the table. After determining the partition column, the GDI does a MIN/MAX on the column as well as a basic sizing query. To specify which column or columns the GDI should partition on, you can include the <code>partitionBy</code> property in the query. The property supports a list of source field names, bound variables, or the object <code>s:auto</code> , which forces the GDI to partition the data when the source does not define partitioning metadata.

Option	Data Type	Description
<b>proxy</b>	string or RDF list	Include this property to specify proxy information if a proxy is used. The value can be a string, such as <code>s:proxy "host_url:port_number"</code> , or an RDF list that includes <code>host</code> and <code>port</code> properties, such as <code>s:proxy [ s:host "host_url" ; s:port port_number ]</code> .
<b>trust</b>	string	Include this property to set the level of trust for the source's SSL certificate. The value can be either <code>"system"</code> or <code>"all"</code> .

## ElasticSource Properties

The table below lists the properties that are available for queries against Elasticsearch data sources. For more information about Elasticsearch sources, see [Querying an Elasticsearch Source](#).

Option	Data Type	Description
<b>aggregations</b>	object	You can include this property to calculate aggregations over the specified bindings. For information about aggregations, see <a href="#">Aggregations</a> in the Elasticsearch documentation.
<b>config</b>	string	To enable you to use explicit mappings, you can include this property to specify the URL to the index configuration file to employ. For example, <code>es:config "/opt/shared/elastic/mapping.json"</code> .
<b>document</b>	string	This property lists the document(s) to search.
<b>field</b>	string or variable	This property defines the field to operate on. The value can be a string or bound variable.
<b>highlight</b>	RDF list	You can include this property to define how results are highlighted. For information about the available properties, see <a href="#">Highlighting Elasticsearch Results</a> .
<b>html</b>	boolean	This property controls whether to output HTML for highlighted results. Defaults to <code>true</code> .

Option	Data Type	Description
<b>index</b>	string	This property can be included to specify the index to search.
<b>minScore</b>	float	This property defines the minimum score for matching documents. Documents with a lower score are not included in the search results.
<b>query</b>	string or RDF list	This property defines the query to execute. The value can be a string or a query object that maps to the Elasticsearch <a href="#">Query DSL</a> . To generate the final query, the GDI combines <code>es:query</code> with any filters it can push to the Elasticsearch DSL. For more information about the <code>query</code> property and mapping Elasticsearch filters to SPARQL FILTER clauses, see <a href="#">Query DSL and Filter Mapping</a> .
<b>routing</b>	string	This property can be included to route a document to a specific shard or to limit the search to a particular shard.
<b>searchAfter</b>		You can include this property to define the key values to start searching from.
<b>size</b>	int	This property maps to the <code>size</code> parameter in the Elasticsearch Search API and configures the batch size or maximum number of hits to return in a single call. Defaults to 10 and typically does not need to be changed.
<b>source</b>	boolean or RDF list	This property can be included to specify the source data to include in results. The value can be a boolean, list of fields, or a list of variable bindings. When <code>true</code> , all source data is returned. When <code>false</code> , no source data is returned.
<b>url</b>	string	The Elasticsearch endpoint URL.

# Onboarding Unstructured Data

Anzo processes unstructured data through configurable text analytics and natural language processing (NLP) pipelines that find and extract data and convert it to the graph data model. Anzo can process all common file types such as Office documents, PDFs, web pages, and email messages, and can analyze text within Excel, databases, and knowledgebases, or XML columns, properties, and fields. Anzo finds, analyzes, extracts, and ingests concepts, entities, sentiment, topics, classifications, events, facts, and thousands of types of relationships.

**Note**

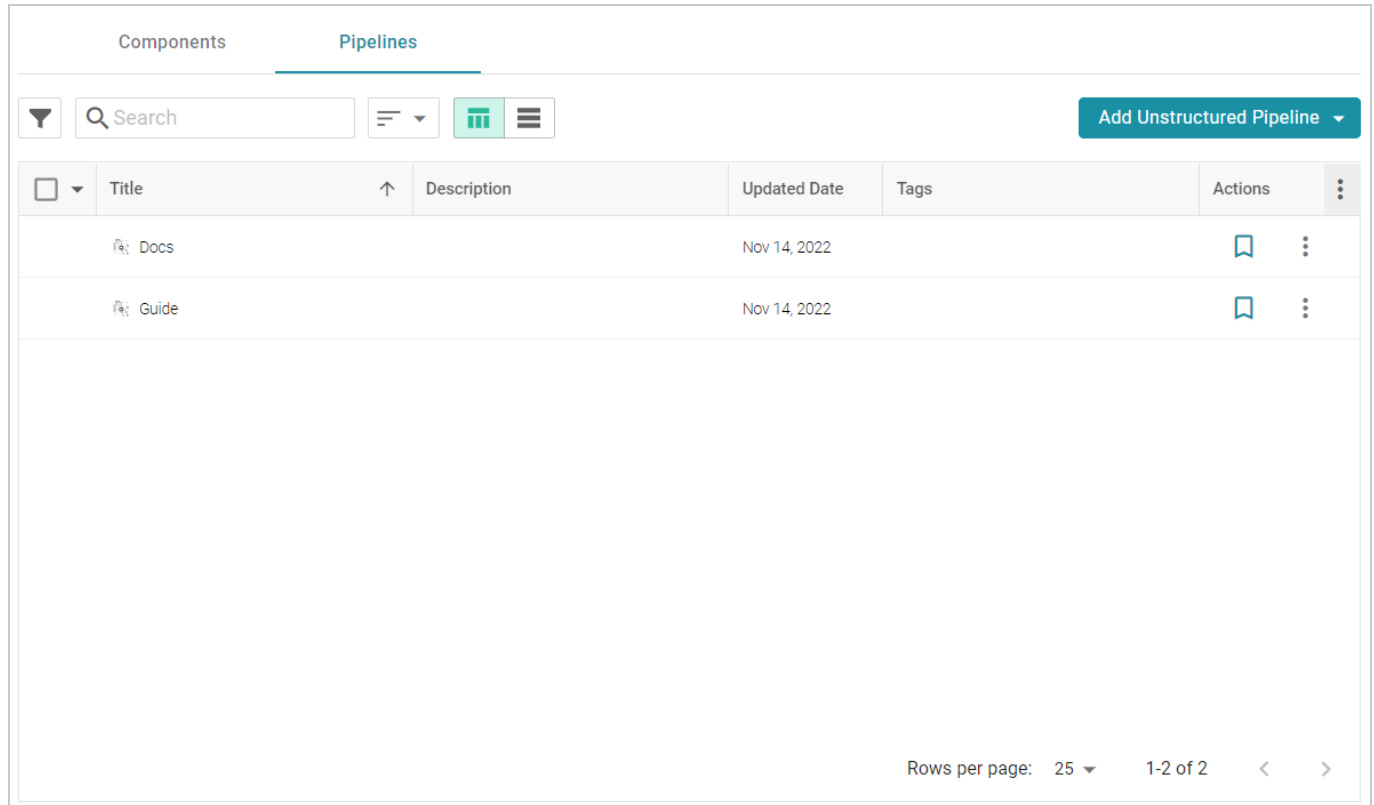
The topics in this section provide instructions for creating pipelines and onboarding unstructured data. For information about setting up the Anzo Unstructured environment, see the Deployment Guide.

- [Creating an Unstructured Pipeline ..... 418](#)
- [Running an Unstructured Pipeline ..... 441](#)
- [Pipeline Settings Reference ..... 444](#)
- [Annotator Settings Reference ..... 450](#)

## Creating an Unstructured Pipeline

Follow the instructions below to create a new unstructured pipeline.

1. In the Anzo application, expand the **Onboard** menu and click **Unstructured Data**. Anzo displays the Pipelines screen, which lists any existing unstructured pipelines. For example:



Components		Pipelines					
	<input type="text" value="Search"/>				<button>Add Unstructured Pipeline</button>		
<input type="checkbox"/>	Title	↑	Description	Updated Date	Tags	Actions	
	Docs			Nov 14, 2022			
	Guide			Nov 14, 2022			
Rows per page: 25 1-2 of 2 < >							

2. Click the **Add Unstructured Pipeline** button and select **Distributed Unstructured Pipeline**. Anzo opens the Create Distributed Unstructured Pipeline dialog box. For example:

### Create Distributed Unstructured Pipeline

Title \*

Description

Target Anzo Data Store \*

The datasource to use for autocreating linked datasets from this pipeline

☐ Deploy Unstructured Infrastructure Dynamically

Static Elastic Search Config

A static elastic search config to use in post processing. If none is provided, then user will be prompted to pick a cloud location for dynamic ES spinup while triggering pipeline.

CANCEL

SAVE

- In the **Title** field, type a name for the pipeline.

#### Note

This Title serves as a key to identify this pipeline and its corpus in multiple contexts. Specify a Title that is unique and stable. The pipeline's corpus dataset name is derived from this Title.

- Type an optional description for the pipeline in the **Description** field.
- If necessary, click the **Target Anzo Data Store** field and select the Anzo Data Store for this pipeline.
- If the environment is configured for dynamic Kubernetes-based deployments of the Anzo Unstructured infrastructure, select the **Deploy Unstructured Infrastructure Dynamically** checkbox and leave the **Static Elasticsearch Config** field blank.
- If necessary, click the **Static Elasticsearch Config** field and select the Elasticsearch connection to use for this pipeline. If you use dynamic deployments to deploy Elasticsearch instances on-demand, leave this field blank. Anzo will prompt the user to choose a Cloud Location when the pipeline is run.

8. Click **Save** to create the Pipeline. Anzo displays the pipeline Overview screen. For example:

At least one Crawler is required for a valid Pipeline

Anzo Docs

Not Versioned

Run Pipeline

Overview

Crawlers

Annotators

History

Versions

Discussion

Description

PDF versions of the documentation

Target Anzo Data Store

Store

☐ Deploy Unstructured Infrastructure Dynamically

Static Elastic Search Config

ES

+ Advanced

General

Type

Unstructured Pipeline

Creator

System Administrator

Updated

a few seconds ago

Released

an hour ago

http://cambridgesemantics.com/DistributedU

Dynamic Resource Log

Tags

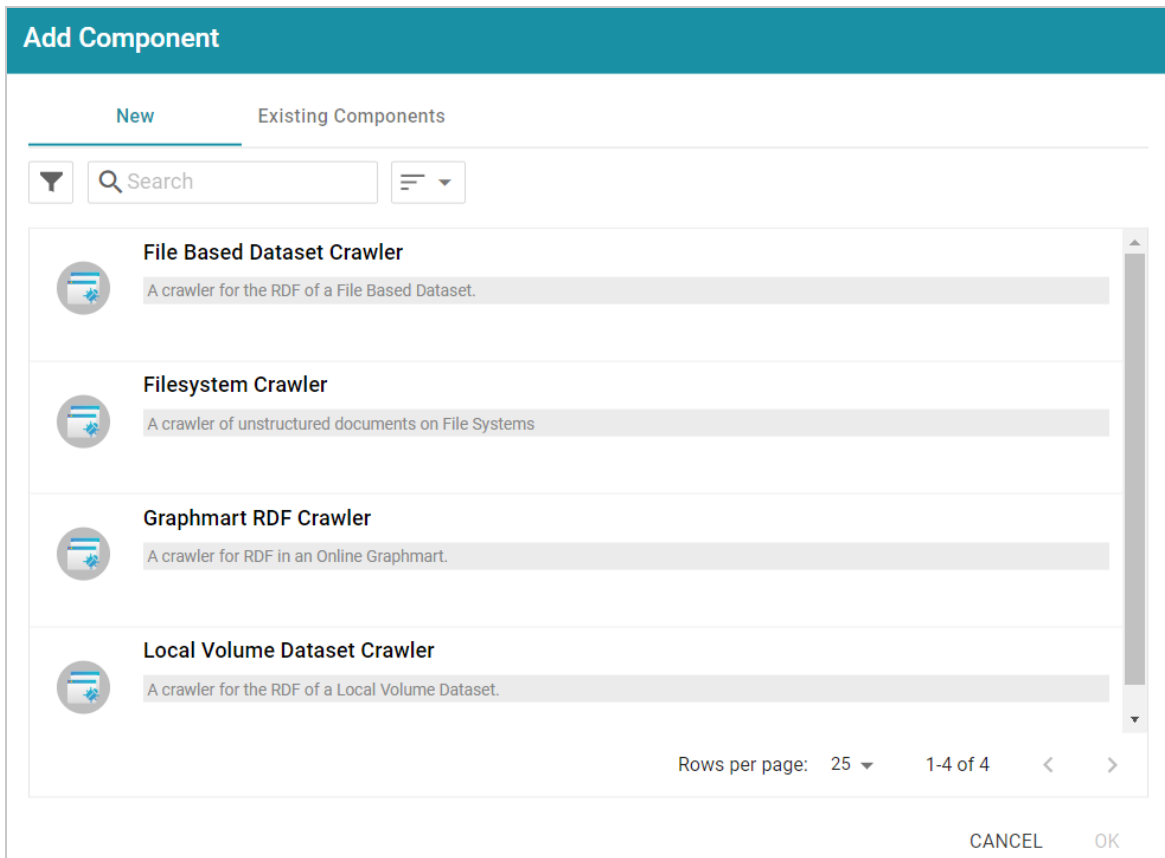
None

### Note

A pipeline saves automatically and constantly undergoes validation to make sure that it is valid based on the current configuration. Anzo displays validation issues in red on the top of the screen. The warnings will disappear as you add components to the pipeline.

9. If necessary, click **Advanced** to configure the advanced pipeline settings. For details about the advanced settings, see [Pipeline Settings Reference](#).
10. Click the **Crawlers** tab and follow the substeps below to add a crawler to the pipeline:
- Click **Add Input**. Anzo displays the Add Component dialog box. The **New** tab is selected and lists all available crawlers. The **Existing Components** tab lists crawlers that have been previously configured for other pipelines.





b. To add a new crawler, select the crawler. To add an existing crawler, click the **Existing Components** tab and select a crawler. The list below describes each of the crawlers:

- **File Based Dataset Crawler:** Include this crawler to process data from a file-based linked data set (FLDS) on a file store.
- **Filesystem Crawler:** Include this crawler to process documents, such as email messages, PDF, XML, PowerPoint, Excel, OneNote, or Word files, and images, that are available on a file store.
- **Graphmart RDF Crawler:** Include this crawler to process RDF in an online graphmart or specific data layer.
- **Local Volume Dataset Crawler:** Include this crawler to process RDF data that is stored as a linked data set (LDS) in an Anzo journal.

c. After selecting a crawler, click **OK**. Anzo opens the Create dialog box for that crawler so that you can configure it. The list below provides details about the settings for each crawler.

### File Based Dataset Crawler

This section describes the settings that are available on the Create File Based Dataset Crawler screen:

The screenshot shows a web form titled "Create File Based Dataset Crawler". It contains several input fields: "Title \*" (required), "Description", "Backing Dataset \*" (required), "Backing Ontology \*" (required), "RDF Resource Type", and "Link Property". Each field has a small help text below it: "The backing dataset" for Backing Dataset, "The backing ontology" for Backing Ontology, "RDF Class" for RDF Resource Type, and "Property(s) to match on for filepath reference" for Link Property. The form also includes "CANCEL" and "SAVE" buttons at the bottom right.

**Create File Based Dataset Crawler**

Title \*

Description

Backing Dataset \*  
The backing dataset

Backing Ontology \*  
The backing ontology

RDF Resource Type  
RDF Class

Link Property  
Property(s) to match on for filepath reference

CANCEL SAVE

- **Title:** Required field that specifies the unique name for this crawler.
- **Description:** Optional field that provides a description of this crawler.
- **Backing Dataset:** Required field that specifies the Anzo dataset to crawl.
- **Backing Ontology:** Required field that specifies the model for the dataset.
- **RDF Resource Type:** Required field that specifies the resource type or class of data to target with this crawler.
- **Link Property:** Optional field that specifies any link properties to crawl. A link property is a property whose value identifies the location of a linked document. When linked properties are specified, the crawler will crawl the linked documents. For example, in the triples below, **fileLocation** is a link property:

```
<urn://someUnstructuredDocument> <urn://someProperty>
"file metadata" ;
<urn://fileLocation> "/path/to/file.pdf" .
```

#### Note

In typical use cases, this crawler is configured to define either a Link Property or a Content Property but not both.

- **Content Property:** Optional field that identifies any content properties to crawl. A content property is a property whose value is a string literal and you want the crawler to crawl and annotate those strings. For example, in the triples below, **longDescription** is a content property:

```
<urn://someUnstructuredDocument> <urn://someProperty>
"file metadata" ;
<urn://longDescription> "this is some interesting, likely
long, unstructured text
with a lot of information, and I want it to be annotated"
.
```

- **Base Path Connection:** Required field whose value depends on whether you specified a Link Property or a Content Property:
  - If a **Link Property** was specified, the Base Path Connection is the base path to use for resolving relative file paths in the Link Property values. For example, using the example triples:

```
<urn://someUnstructuredDocument> <urn://someProperty>
"file metadata" ;
<urn://fileLocation> "/path/to/file.pdf" .
```

The `<urn://fileLocation>` value of `/path/to/file.pdf` could be a relative path to a location like

`s3://location/bucket/path/to/file.pdf` or

`/opt/anzoshare/data/path/to/file.pdf`. Therefore, the Base Path needs to be specified to resolve any relative paths and locate the linked documents.

- If a **Content Property** was specified, the Base Path Connection is a directory on the file store where the crawler can save a copy of the Content Property strings for the Anzo Unstructured worker instances. Saving the content to a shared file location avoids the overhead of sending the strings to the workers over the network.

---

## Filesystem Crawler

This section describes the settings that are available on the Create Filesystem Crawler screen:

**Create Filesystem Crawler**

Title \*

Description

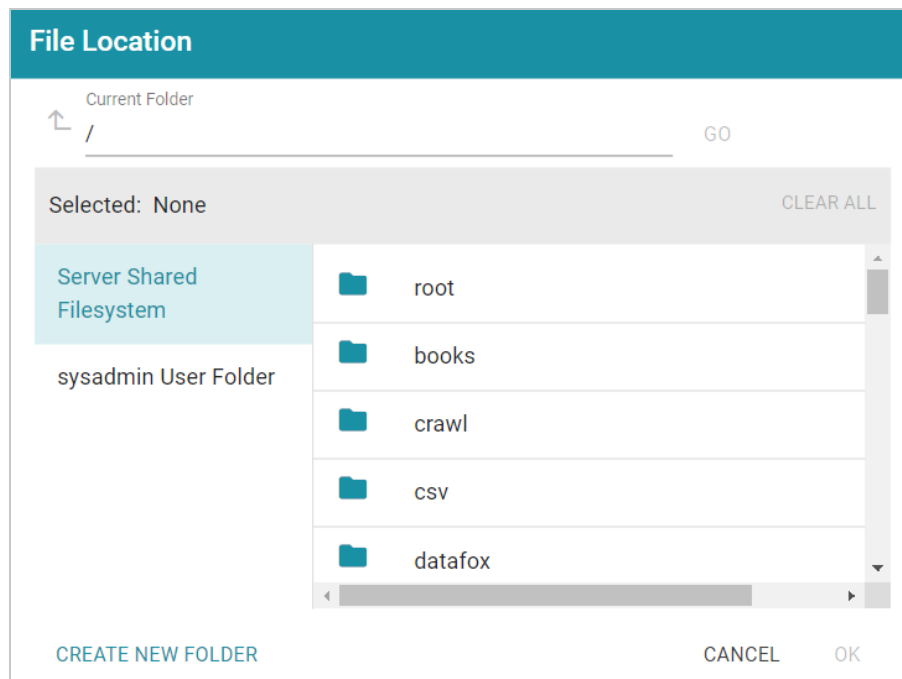
File Crawl Location \* [BROWSE](#)

The file location to crawl

☒ Crawl subfolders

[CANCEL](#) [SAVE](#)

- **Title:** Required field that specifies the unique name for this crawler.
- **Description:** Optional field that provides a description of this crawler.
- **File Crawl Location:** Required field that specifies the file system crawl location. Click the field to open the File Location dialog box:



On the left side of the screen, select the storage location for the files to crawl. On the right side of the screen, navigate to the directory that contains the files. Select a directory, and then click **OK**.

- **Crawl subfolders**: Optional field that specifies whether to crawl the subdirectories under the VFS Crawl Location. To crawl the subdirectories, select the **Crawl subfolders** checkbox. To ignore subdirectories, clear the **Crawl subfolders** checkbox.

---

## Graphmart RDF Crawler

This section describes the settings that are available on the Create Graphmart RDF Crawler screen:

---

### Create Graphmart RDF Crawler

Title \*

Description

Backing Graphmart

A backing graphmart

Backing Layer

A backing layer

Backing Ontology \*

The backing ontology

RDF Resource Type

RDF Class

CANCEL

SAVE

- **Title:** Required field that specifies the unique name for this crawler.
- **Description:** Optional field that provides a description of this crawler.
- **Backing Graphmart:** Optional field that specifies the graphmart to crawl. To configure the crawler to crawl at the graphmart level, select one or more graphmarts in the **Backing Graphmart** field and leave the **Backing Layer** field blank.
- **Backing Layer:** Optional field that specifies the data layer or layers that you want the pipeline to crawl. To crawl specific layers and not an entire graphmart, make sure that you leave the **Backing Graphmart** field blank and select the layers to crawl in the **Backing Layer** field. If you specify both a Backing Graphmart and a Backing Layer, the Backing Graphmart value supersedes the Backing Layer value, resulting in the entire graphmart being crawled.

- **Backing Ontology:** Required field that specifies the model for the Backing Graphmart or Data Layer.
- **RDF Resource Type:** Required field that specifies the resource type or class of data to target with this crawler.
- **Link Property:** Optional field that specifies any link properties to crawl. A link property is a property whose value identifies the location of a linked document. When linked properties are specified, the crawler will crawl the linked documents. For example, in the triples below, **fileLocation** is a link property:

```
<urn://someUnstructuredDocument> <urn://someProperty>
"file metadata" ;
<urn://fileLocation> "/path/to/file.pdf" .
```

#### Note

In typical use cases, this crawler is configured to define either a Link Property or a Content Property but not both.

- **Content Property:** Optional field that identifies any content properties to crawl. A content property is a property whose value is a string literal and you want the crawler to crawl and annotate those strings. For example, in the triples below, **longDescription** is a content property:

```
<urn://someUnstructuredDocument> <urn://someProperty>
"file metadata" ;
<urn://longDescription> "this is some interesting, likely
long, unstructured text
with a lot of information, and I want it to be annotated"
.
```

- **Base Path Connection:** Required field whose value depends on whether you specified a Link Property or a Content Property:
  - If a **Link Property** was specified, the Base Path Connection is the base path to use for resolving relative file paths in the Link Property values. For example, using the example triples:

```
<urn://someUnstructuredDocument> <urn://someProperty>
"file metadata" ;
<urn://fileLocation> "/path/to/file.pdf" .
```

The `<urn://fileLocation>` value of `/path/to/file.pdf` could be a relative path to a location like

`s3://location/bucket/path/to/file.pdf` or  
`/opt/anzoshare/data/path/to/file.pdf`. Therefore, the Base Path needs to be specified to resolve any relative paths and locate the linked documents.

- If a **Content Property** was specified, the Base Path Connection is a directory on the file store where the crawler can save a copy of the Content Property strings for the Anzo Unstructured worker instances. Saving the content to a shared file location avoids the overhead of sending the strings to the workers over the network.

---

## Local Volume Dataset Crawler

This section describes the settings that are available on the Create Local Volume Dataset Crawler screen:

---



Create Local Volume Dataset Crawler

Title \*

Description

Backing Dataset \*

The backing dataset

Backing Ontology \*

The backing ontology

RDF Resource Type

RDF Class

Link Property

Property(s) to match on for filepath reference

CANCEL

SAVE

- **Title:** Required field that specifies the unique name for this crawler.
- **Description:** Optional field that provides a description of this crawler.
- **Backing Dataset:** Required field that specifies the Anzo dataset to crawl.
- **Backing Ontology:** Required field that specifies the model for the dataset.
- **RDF Resource Type:** Required field that specifies the resource type or class of data to target with this crawler.
- **Link Property:** Optional field that specifies any link properties to crawl. A link property is a property whose value identifies the location of a linked document. When linked properties are specified, the crawler will crawl the linked documents. For example, in the triples below, **fileLocation** is a link property:

```

<urn://someUnstructuredDocument> <urn://someProperty>
"file metadata" ;
<urn://fileLocation> "/path/to/file.pdf" .

```

### Note

In typical use cases, this crawler is configured to define either a Link Property or a Content Property but not both.

- **Content Property:** Optional field that identifies any content properties to crawl. A content property is a property whose value is a string literal and you want the crawler to crawl and annotate those strings. For example, in the triples below, **longDescription** is a content property:

```
<urn://someUnstructuredDocument> <urn://someProperty>
"file metadata" ;
<urn://longDescription> "this is some interesting, likely
long, unstructured text
with a lot of information, and I want it to be annotated"
.
```

- **Base Path Connection:** Required field whose value depends on whether you specified a Link Property or a Content Property:
  - If a **Link Property** was specified, the Base Path Connection is the base path to use for resolving relative file paths in the Link Property values. For example, using the example triples:

```
<urn://someUnstructuredDocument> <urn://someProperty>
"file metadata" ;
<urn://fileLocation> "/path/to/file.pdf" .
```

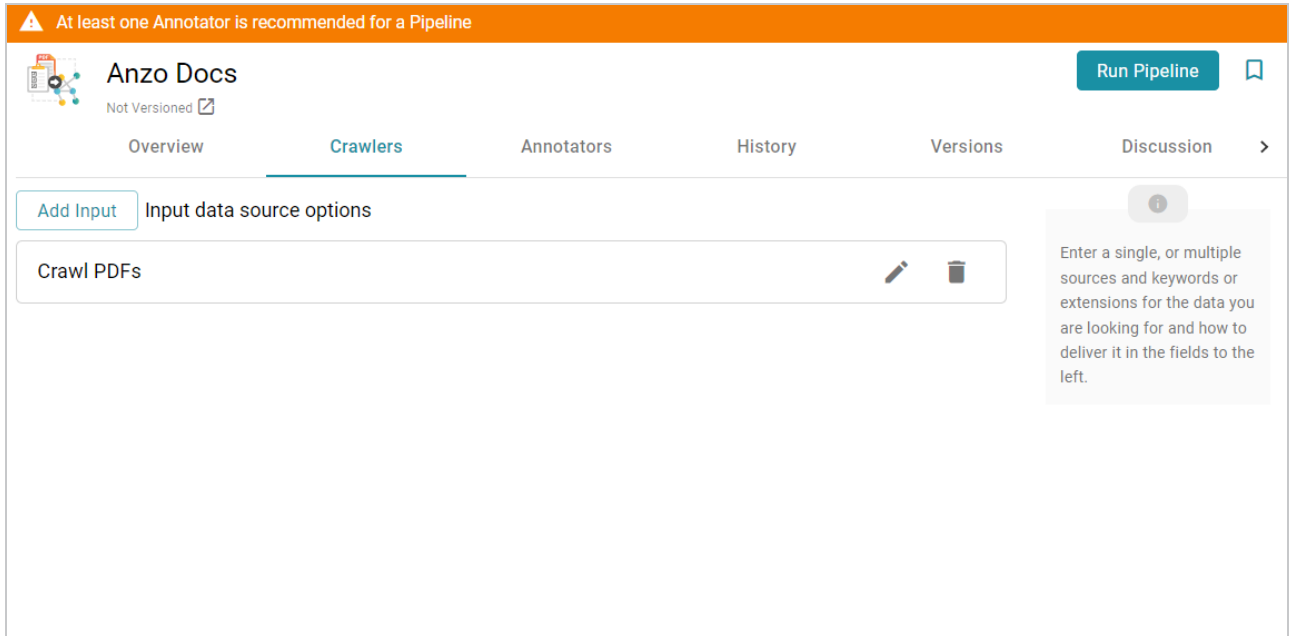
The `<urn://fileLocation>` value of `/path/to/file.pdf` could be a relative path to a location like

`s3://location/bucket/path/to/file.pdf` or  
`/opt/anzoshare/data/path/to/file.pdf`. Therefore, the Base Path needs to be specified to resolve any relative paths and locate the linked documents.

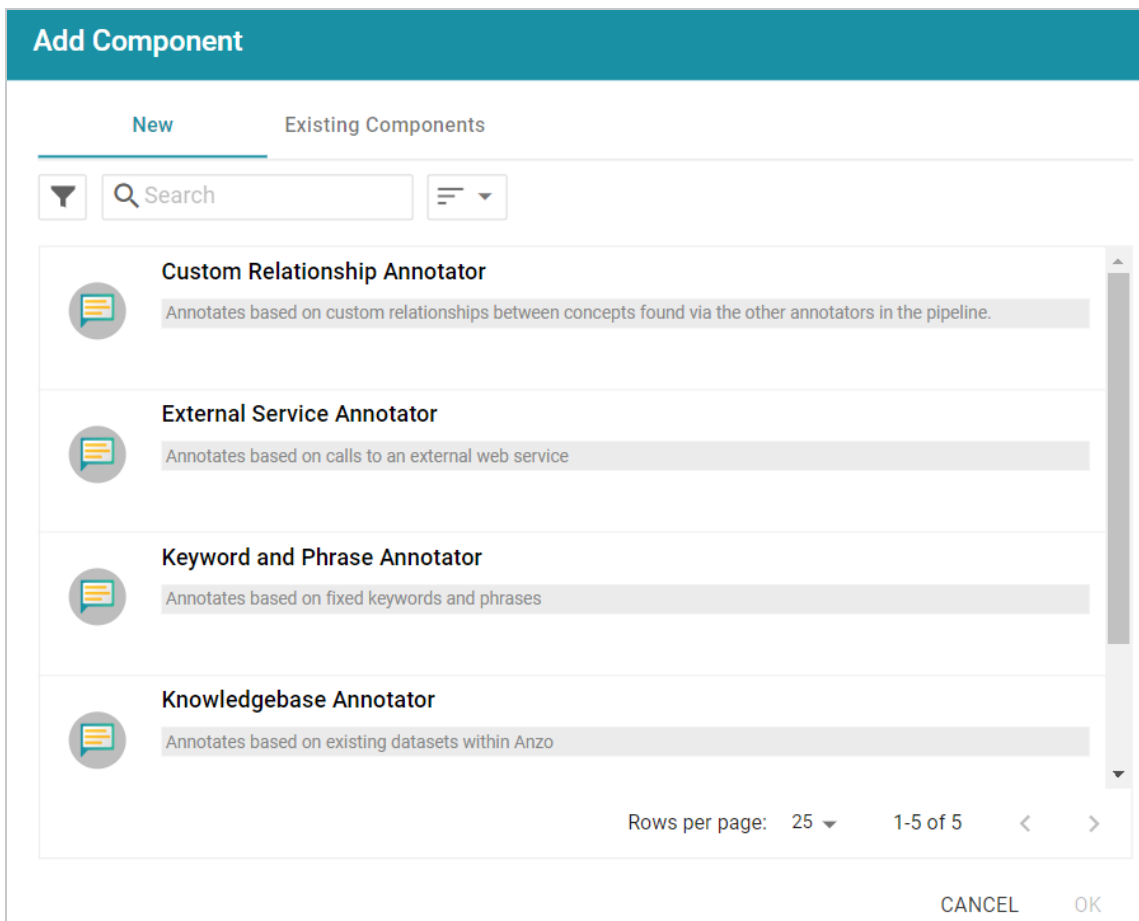
- If a **Content Property** was specified, the Base Path Connection is a directory on the file store where the crawler can save a copy of the Content Property strings for the Anzo Unstructured worker instances. Saving the

content to a shared file location avoids the overhead of sending the strings to the workers over the network.

- d. When you have finished configuring the crawler, click **Save**. Anzo adds the crawler to the pipeline and returns to the Crawlers screen. For example:



- e. If you want to change the crawler configuration, click the Edit icon (✎) for the crawler and modify the settings as needed. If you want to add another crawler to the pipeline, repeat substeps a – d.
11. Click the **Annotators** tab and follow the substeps below to add an annotator to the pipeline:
    - a. Click **Add Output** to select an annotator. Anzo opens the Add Component dialog box. The **New** tab is selected and lists the available annotators and the **Existing Components** tab lists annotators that have been previously configured for other pipelines.



b. To add a new annotator to the pipeline, click the annotator name to select it. To add an existing annotator to the pipeline, click the **Existing Components** tab, and then select an annotator. The list below describes each of the default annotators:

- **Custom Relationship Annotator:** Include this annotator to map relationships between annotations based on the number of characters between the annotations.
- **External Service Annotator:** Include this annotator to hit an HTTP endpoint that provides annotations.
- **Keyword and Phrase Annotator:** Include this annotator to create annotations based on the phrases that you specify.
- **Knowledgebase Annotator:** Include this annotator to link structured and unstructured data by finding instances in data layers, graphmarts, or Anzo linked datasets. Based on the names and aliases of entities present or patterns that are indicative of the entities, this annotator marks up the documents with the structured entities linked.

- **Regex Annotator:** Include this annotator to use regular expression rules to identify entities such as email addresses, URLs, phone numbers, or any other entity that can be matched using a regular expression.
- c. After selecting an annotator, click **OK**. Anzo opens the Create dialog box for the component. Complete the fields to configure the annotator. The list below provides details about the settings for the annotators that are typically used in pipelines.

## External Service Annotator

This section describes the settings that are available on the Create External Service Annotator screen.

### Tip

For information about the options that are presented when you edit an External Service Annotator, see [Annotator Settings Reference](#).

### Create External Service Annotator

Title \*

Description

HTTP Request Config \*

Config for connecting and sending a request to the external NLP server

Document ID Response Path \*

the Document ID path for entities returned in the service response

Entity Name Path \*

The entity Name path for entities returned in the service response

Entity Class Path \*

The entity Class path for entities returned in the service response

CANCEL

SAVE

- **Title:** Required field that specifies the unique name for this annotator.
- **Description:** Optional field that provides a description of this annotator.
- **HTTP Request Config:** Required field that specifies the HTTP source object that contains the URL and method to use when sending data for annotations.
- **Document ID Response Path:** Required field that specifies where to find the document ID in the response.
- **Entity Name Path:** Required field that specifies the annotation object name path.
- **Entity Class Path:** Required field that specifies the class URI for an annotation.

---

## Keyword and Phrase Annotator

---

This section describes the settings that are available on the Create Keyword and Phrase Annotator screen.

#### Tip

For information about the options that are presented when you edit a Keyword and Phrase Annotator, see [Annotator Settings Reference](#).

### Create Keyword and Phrase Annotator

Title \*

Description

Phrase \*

Phrase to look for

ADD

CANCEL SAVE

- **Title:** Required field that specifies the unique name for this annotator.
- **Description:** Optional field that provides a description of this annotator.
- **Phrase:** Required field that specifies the terms or phrases to annotate. Type a word or phrase in the field and then click **Add** to add the phrase. You can add any number of phrases.

## Knowledgebase Annotator

This section describes the settings that are available on the Create Knowledgebase Annotator screen.

#### Tip

For information about the options that are presented when you edit a Knowledgebase Annotator, see [Annotator Settings Reference](#).

Create Knowledgebase Annotator

Title \*

Description

Backing Graphmart

A backing graphmart

Backing Layer

A backing layer

Backing Ontology \*

The backing ontology

Term Class

The owl:Class of the knowledge base terms

CANCEL

SAVE

- **Title:** Required field that specifies the unique name for this annotator.
- **Description:** Optional field that provides a description of this annotator.
- **Backing Graphmart:** Optional field that specifies the graphmart or graphmarts to annotate.

#### Note

If you want the annotator to run against a linked dataset or Anzo knowledgebase instead of a data layer or graphmart, leave the Backed Layer and Backed Graphmart fields blank. After saving the pipeline, you can edit the pipeline and specify a **Backed Dataset** at that time.



- **Backing Layer:** Optional field that specifies the data layer or layers to annotate.

#### Note

The Backing Layer and Backing Graphmart fields are treated independently. Layers that you select do not have to be part of the graphmart that you specify in **Backing Graphmart**. And specifying a layer does not mean that you must select a Backing Graphmart. However, any layers or graphmarts that you select must contain classes and properties from the **Backing Ontology** or the data will not be annotated.

- **Backing Ontology:** Required field that specifies the model for the backing data layers and/or graphmart. Click the field and select a model from the drop-down list.
- **Term Class:** Required field that specifies the class of data for the annotations.
- **Term Label Property:** Required field that lists the primary name or label property of the resources.
- **Term Identifying Properties:** Required field that specifies the properties that contain names, aliases, or other identifiers to use for identifying the resources.

---

## Regex Annotator

This section describes the settings that are available on the Create Regex Annotator screen.

#### Tip

For information about the options that are presented when you edit a Regex Annotator, see [Annotator Settings Reference](#).

### Create Regex Annotator

Title \*

Description

Regular Expression Rule \*

The regular expression rule(s) that this annotator will use

CANCEL

SAVE

- **Title:** Required field that specifies the unique name for this annotator.
- **Description:** Optional field that provides a description of this annotator.
- **Regular Expression Rule:** Required field that lists the regular expression rules for this annotator. To add a rule, click drop-down field and select **Create New**. Anzo opens the Create Regular Expression Rule dialog box where you can define the rule:

### Create Regular Expression Rule

Title \*

Class Structure \*

The class structure used for annotations created from matches of the corresponding regular expression. Example syntax: "0:Person;1:Company"

Description

Regular Expression \*

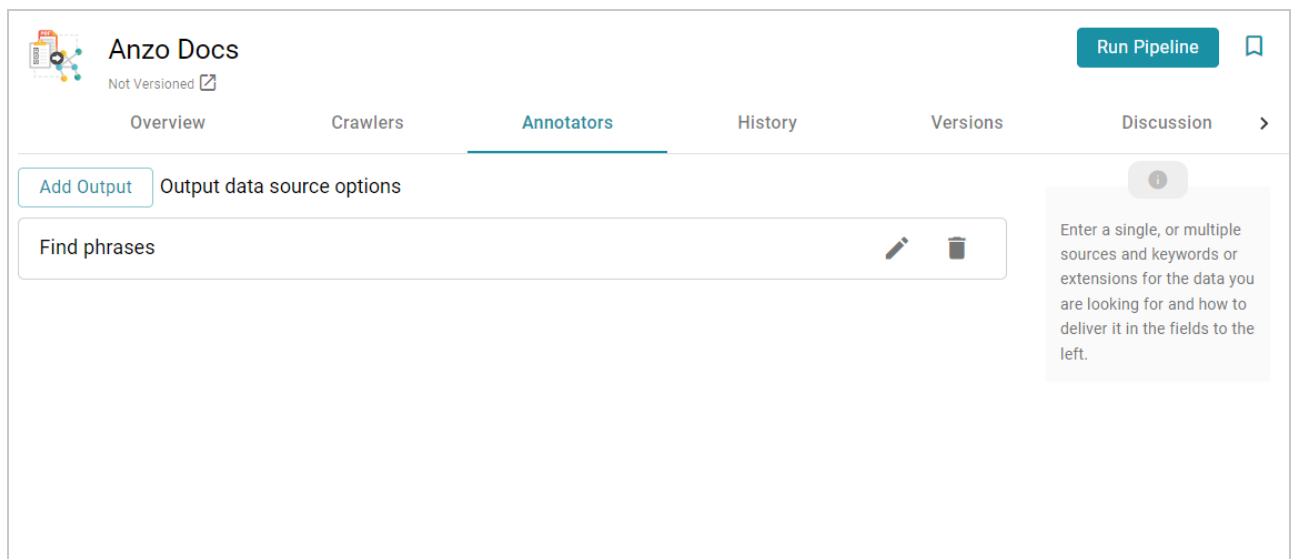
The regular expression to look for in the text

CANCEL

SAVE

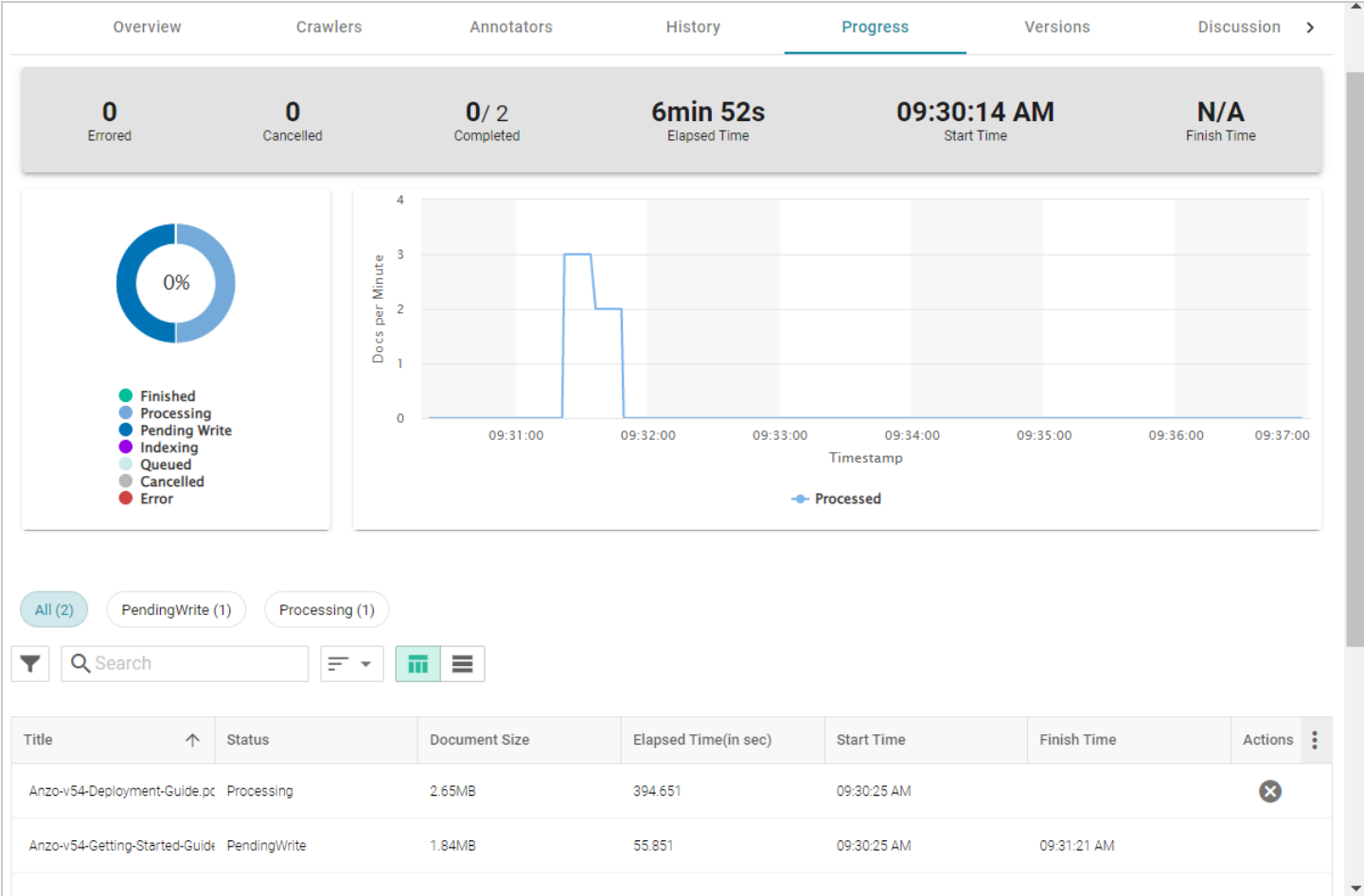
- **Title:** Required field that specifies the name of the rule.
- **Class Structure:** Required field that specifies the class in the model that should be created for this rule. The value should be in the format `group_number:class_name`, where `group_number` corresponds to a group in the regex capture. Each rule should start with group 0. Include groups 1 and higher if needed to represent parts of the expression that are contained in parentheses. The `class_name` is a label that describes the type of data the rule will find. For example, for a rule that finds hyphenated words `0:Hyphens`.
- **Description:** Optional field that describes the rule.
- **Regular Expression:** Required field that specifies the regular expression to use for finding matching entities.

- d. When you have finished configuring the annotator, click **Save**. Anzo adds the annotator to the pipeline and returns to the Annotators screen. For example:



- e. If you want to change the annotator configuration, click the Edit icon (✎) for the annotator and modify the settings as needed (see [Annotator Settings Reference](#) for information about settings). If you want to add another annotator to the pipeline, repeat substeps a – d.
12. When you have finished adding crawlers and annotators to the pipeline, click the **Run Pipeline** button to run the pipeline.

The process can take several minutes to complete. You can click the **Progress** tab to view details such as the pipeline status, runtime, number of documents processed, and errors. For example:



When the pipeline finishes, a new dataset becomes available in the Datasets catalog. From the catalog, you can create a graphmart from the dataset so that you can explore and analyze the data. For instructions, see [Creating a Graphmart from a Dataset](#). You can also add the dataset to an existing graphmart by following the steps in [Adding a Dataset to an Existing Graphmart](#).

# Running an Unstructured Pipeline

This page provides instructions for running an unstructured pipeline.

1. In the Anzo application, expand the **Onboard** menu and click **Unstructured Data**. Anzo displays the Pipelines screen, which lists any existing unstructured pipelines. For example:

Components

Pipelines


Search

Add Unstructured Pipeline


<div></div>	Title		Description	Updated Date	Tags	Actions
	<div>Docs</div>			Nov 14, 2022		<div></div>
	<div>Guide</div>			Nov 14, 2022		<div></div>

Rows per page: 25 1-2 of 2


2. Click the name of the pipeline that you want to run. Anzo displays the pipeline Overview screen. For example:



Anzo Docs

Not Versioned 

Run Pipeline



Overview

Crawlers

Annotators

History

Versions

Discussion >

Description

PDF versions of the documentation

Target Anzo Data Store

Store

☐ Deploy Unstructured Infrastructure Dynamically

Static Elastic Search Config

ES

+ Advanced

General

Type

Unstructured Pipeline

Creator

System Administrator


Updated

a few seconds ago

Released

an hour ago

http://cambridgesemantics.com/DistributedU



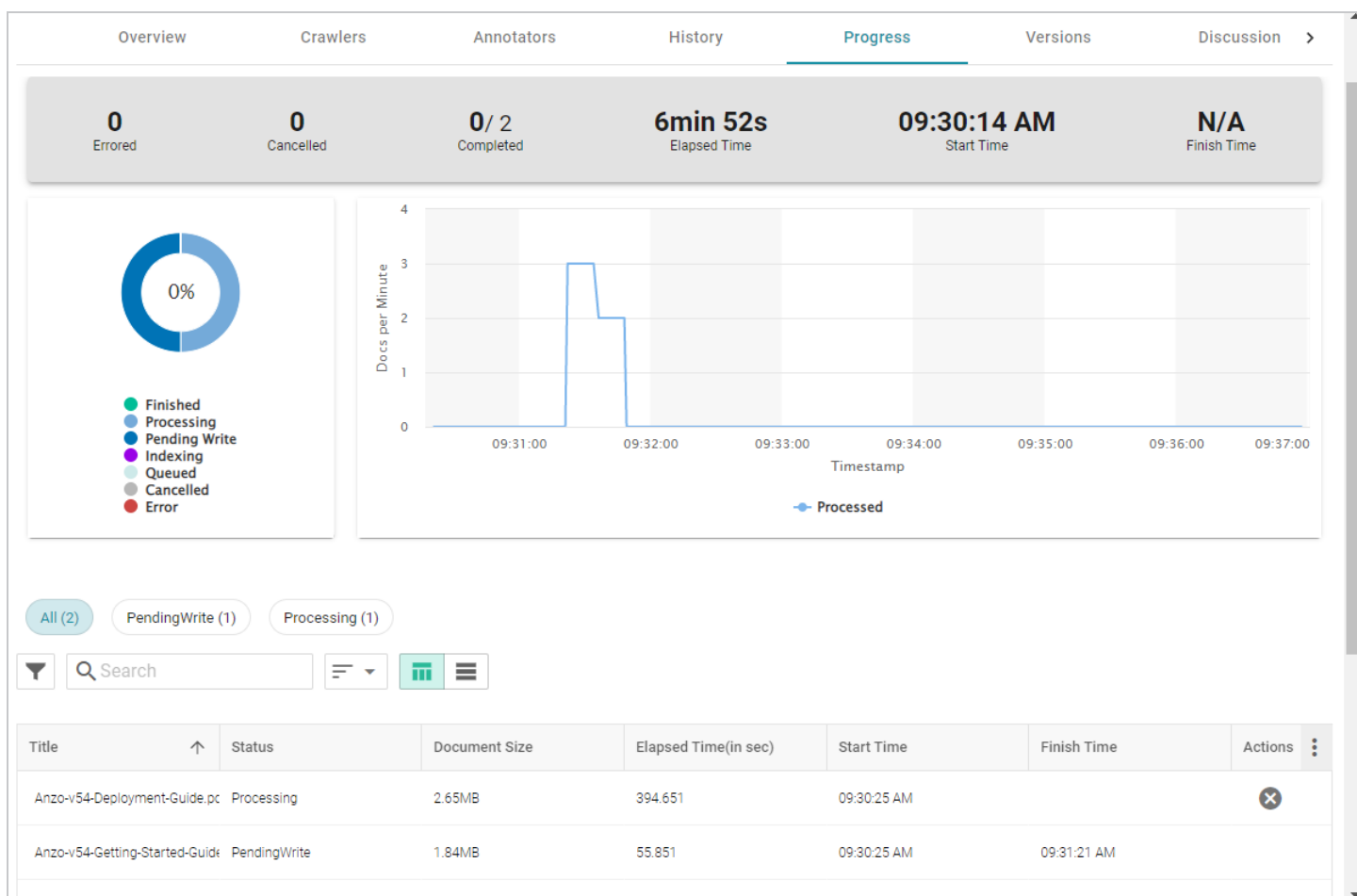
Dynamic Resource Log

Tags

None

3. Click **Run Pipeline** to run the pipeline.

The process can take several minutes to complete. You can click the **Progress** tab to view details such as the pipeline status, runtime, number of documents processed, and errors. For example:



If this is the first time the pipeline was run, a new dataset becomes available in the Datasets catalog. From the catalog, you can create a graphmart from the dataset so that you can explore and analyze the data. For instructions, see [Creating a Graphmart from a Dataset](#). If the pipeline was run previously, the existing dataset was updated and you can refresh or reload the graphmart that contains the dataset to make the new data available for analytics.

## Related Topics

[Creating an Unstructured Pipeline](#)

## Pipeline Settings Reference

The table below defines the Advanced settings that are available on the Overview tab when viewing an unstructured pipeline.

Setting	Description
<b>Append Timestamp</b>	Controls whether to add a timestamp to unstructured document URIs. This setting is enabled by default.
<b>Diagnostic Logging</b>	Controls whether verbose diagnostic logging is enabled for the pipeline. This setting is disabled by default. When enabled, debug-level logging is performed for the duration of the pipeline.
<b>Current Pipeline Run</b>	This setting is a pointer to the pipeline run object that tracks the ongoing execution of the pipeline.
<b>Pipeline Network Connection</b>	This setting specifies the network connection configuration to be used by the pipeline's worker nodes to connect to the Anzo server. If not specified, this setting defaults to the Unstructured Cluster connection configuration.
<b>Persist Extracted Text</b>	Controls whether to persist the extracted text from documents. This setting is enabled by default.
<b>Persist HTML</b>	Controls whether to persist the extracted highlighted/annotated HTML from documents. This setting is enabled by default.
<b>Persist Original Binary</b>	Controls whether to persist the binary from the original documents. This setting is enabled by default.
<b>Persist Hit Spans</b>	Controls whether to persist the hit spans for the annotations of unstructured documents. This setting is disabled by default.
<b>Persist Nothing</b>	Controls whether RDF data about the documents or annotations are saved or persisted. This setting is disabled by default.



Setting	Description
<b>Skip Elastic Search Indexing</b>	Controls whether to skip Elasticsearch indexing. This setting is disabled by default.
<b>Skip Elastic Search JSON creation</b>	Controls whether to skip creating Elasticsearch JSON. This setting is disabled by default.
<b>Is Corpus Cumulative</b>	Controls whether to add the components of each pipeline run to the working edition of the dataset. This setting is disabled by default.
<b>Skip Text Extraction</b>	Controls whether to skip text extraction. This setting is disabled by default.
<b>Delete Elastic Search JSON files</b>	Controls whether to delete the Elasticsearch JSON files after they are indexed. This setting is enabled by default.
<b>Allow Empty Documents</b>	Controls whether to allow documents that have no text to proceed through the pipeline. This setting is disabled by default.
<b>Archive and Host Content</b>	Controls whether to download, cleanse, encapsulate, archive, and host complete document content with inline artifacts. This setting is enabled by default.
<b>HTTP Fetch in Archive</b>	Controls whether the archiving process should resolve and download HTTP URLs that are specified in documents. This setting is disabled by default.
<b>Corpus Linked Dataset</b>	Specifies the FLDS used for documents and annotations from this pipeline. This setting defaults to the name of the pipeline.
<b>Corpus Name</b>	Specifies the name of the corpus (collection of documents) for the pipeline.
<b>Phase Status Persistence</b>	Specifies how phase status metadata is persisted for each document in the pipeline.

Setting	Description
<b>Write Status Updates to Jnl</b>	Controls whether status updates for pipeline runs are written to the journal. This setting is enabled by default.
<b>Write Status Updates to FLDS</b>	Controls whether status updates for pipeline runs are written to an FLDS. This setting is disabled by default.
<b>Write Original Binary On Timeout</b>	Controls whether the original binary is written if the pipeline times out or errors. This setting is disabled by default.
<b>RamDisk Directory Location</b>	Specifies an optional RamDisk base directory to create temporary files under. Using a RamDisk may speed up the pipeline.
<b>Use File Name as Document Title</b>	Controls whether to use the file's name on disk as the document title. This setting is disabled by default.
<b>RDF Statement Buffer Size</b>	Specifies the maximum number of statements to buffer before writing. The default value is 10,000.
<b>RDF File Statement Count</b>	Specifies the maximum number of statements to include in each RDF output file.
<b>Batch Size</b>	Specifies the number of documents to include in one batch.
<b>Maximum Allowed Session Issues</b>	Specifies the maximum number of issues that can be encountered in a run of this pipeline before failing the pipeline.
<b>UI Update Interval (in milliseconds)</b>	The interval of time to wait between running queries to update the data on the pipeline Progress screen. The default value is 30,000 milliseconds (30 seconds).
<b>Document</b>	Specifies the timeout in milliseconds for each document batch to be processed.

Setting	Description
<b>Processing Timeout</b>	Leave this value unset (or set it to 0) to use the microservice cluster's default timeout value.
<b>Error On No Documents Found</b>	Controls whether to fail the pipeline if no documents are found. This setting is enabled by default.
<b>Maximum Pipeline Run Status Journals</b>	Specifies the maximum number of pipeline run status journals to keep before aging them off to an FLDS. By default, only the status of the most recent run of a pipeline remains stored in a status journal. All previous reports are automatically converted to an FLDS and the original status journal is deleted.
<b>Elastic Search Bulk Actions</b>	Specifies the maximum number of indexing actions to queue during Elasticsearch indexing. The default value is 2,000.
<b>Elastic Search Bulk Size</b>	Specifies the maximum size of the document queue during Elasticsearch indexing. The default value is 5.
<b>Elastic Search Bulk Concurrent Requests</b>	This setting specifies the maximum number of concurrent bulk requests to allow during Elasticsearch indexing. The default value is 1.
<b>Elastic Search Bulk Max Threads</b>	This setting specifies the maximum number of threads to use for Elasticsearch indexing. The default value is 1.
<b>Elastic Search Mapping</b>	This setting specifies (in JSON format) the mapping to use when indexing unstructured documents in Elasticsearch.
<b>Elastic Search Pipeline Configuration</b>	This setting specifies (in JSON format) the Elasticsearch pipeline configuration to use when indexing unstructured documents.

Setting	Description
<b>Elastic Search Directory Write-all</b>	Controls whether to give write-all permission to the <code>esi</code> directory in the output corpus FLDS.
<b>Elasticsearch Index Settings</b>	This setting specifies (in JSON format) the index settings to use when indexing unstructured documents in Elasticsearch.
<b>Skip Teardown Of Dynamic Resources</b>	Controls whether dynamic K8s-based resources associated with the pipeline are left running after the pipeline is complete. This setting is disabled by default. Enabling it can result in increased cloud resource usage.
<b>Default Finish Pending Writes On Pipeline Cancellation</b>	Controls whether to finish any pending writes for documents during pipeline cancellation. There is a flag in the cancellation request that can be used to override this setting. This setting is enabled by default.
<b>Post-persist Postprocessor</b>	Specifies any post-persist semantic postprocessors in the pipeline.
<b>Rich Text Extractor</b>	Lists the HTML extractors to use in the pipeline.
<b>Post Worker Service</b>	Specifies a service to invoke on documents after they are successfully processed by the pipeline worker processes.
<b>Pre-persist Postprocessor</b>	Specifies any pre-persist semantic postprocessors in the pipeline.
<b>Status Journal Base Path</b>	Specifies the base path for storage of the status journal. By default, status journals are written to a <code>status_journals</code> subdirectory in the Anzo Data Store that is specified for the pipeline.
<b>Content</b>	Specifies any content transformation and metadata extraction components to use in

Setting	Description
Transformer	the pipeline.
Document Crawler Thread Count	Specifies the number of threads to use for document crawling. The default value is 4.
Worker Service ID	Specifies the worker service ID to send requests to. If not specified, the default is pipelineWorkerService.

## Annotator Settings Reference

When you edit an existing annotator, additional options become available for refining the annotation criteria or customizing the generated model or instance data. This topic describes the advanced settings that are available when editing each type of annotator.

- [External Service Annotator](#)
- [Keyword and Phrase Annotator](#)
- [Knowledgebase Annotator](#)
- [Regex Annotator](#)

### External Service Annotator

The table below defines the settings that are displayed when an External Service Annotator is edited.

Setting	Description
Title	Required field that specifies the unique name for the annotator.
Description	Optional field that provides a description of the annotator.
HTTP Request Config	Required field that specifies the HTTP source object that contains the URL and method to use when sending data for annotations.
Document ID Response Path	Required field that specifies where to find the document ID in the response.
Entity Name Path	Required field that specifies the annotation object name path.
Entity Class Path	Required field that specifies the base class URI for an annotation.
Result Path Root	The path to the object that contains the annotation results.
Store NLP Service Response	Controls whether the service's response is stored in the binary store.

Setting	Description
Result Field Path	The external NLP-specific result configuration for returned entities.
Socket Timeout	Specifies the socket timeout (in milliseconds) to use for requests against the source.
Entity Snippet Path	The snippet path for entities returned in the service response.
Entity End Offset Path	The end text offset location in the document for entities returned in the service response.
Entity Begin Offset Path	The start text offset location in the document for entities returned in the service response.
Entity Span Path	The text offset location in the document for entities returned in the service response.
Entity Text Path	The text path for entities returned in the service response.
Entity ID Path	The ID path for entities returned in the service response.
Document ID Request Field	The Document ID parameter for the external service.
Class Name Property	Specifies an annotation property whose value you want to map to the name of the class. For example, if a <code>Category</code> property has the value <code>Disease</code> and you want the name of the class to be "Disease," add <code>Category</code> to this field. When Class Name Property is not defined, the class name is auto-generated.
Unintended Property Names	A list of any property names to filter out. Type a name in the field and then click <b>Add</b> to add the value.
Unintended Classes	A list of any classes to filter out. Type a class in the field and then click <b>Add</b> to

Setting	Description
	add the value.
<b>Unintended Instances</b>	A list of any entities or instances of the class to filter out. Type an instance in the field and then click <b>Add</b> to add the value.
<b>Is Combine Annotation Instances</b>	Controls whether to combine multiple instances of an extraction into one annotation.
<b>Create Additional General Annotation Type</b>	Controls whether the annotator creates a general shared annotation type in addition to the specific annotation types that are created.
<b>Explicit Property Datatypes/Objecttypes</b>	A list of keys that map property names to a particular object property type.
<b>Output the Detections</b>	Controls whether to include specific detections as an annotation property.
<b>Unintended Property Values</b>	A list of any property values to filter out. Type a value in the field and then click <b>Add</b> to add the value.
<b>Entity URI Property</b>	Specifies an annotation property whose value you want to map to the URIs for instances of the class. For example, if a <code>Disease_ID</code> property has the value <code>http://example.com/Asthma</code> and you want to use <code>http://example.com/Asthma</code> as the base URI for instances of the class, add <code>Disease_ID</code> to this field. When Entity URI Property is not defined, the URI is auto-generated based on the name.
<b>Entity Name Property</b>	Specifies an annotation property whose value you want to map to the names for instances of the class. For example, if a <code>Preferred_Label</code> property includes disease names and you want to use those label values as the names for instances of the Disease class, add <code>Preferred_Label</code> to this field. When Entity Name Property is not defined, the name is auto-generated.
<b>Domain Object Base</b>	If creating a general annotation type (Create Additional General Annotation



Setting	Description
<b>Class URI</b>	Type is enabled), this setting specifies the class to use as the base type for the annotator's domain objects.
<b>Class URI Property</b>	Specifies an annotation property whose value you want to map to the class URI in the model. For example, if a <code>Category_ID</code> property has the value <code>http://example.com/Disease</code> and you want to use <code>http://example.com/Disease</code> as the base class URI, add <code>Category_ID</code> to this field. When Class URI Property is not defined, the URI is auto-generated based on the name.
<b>Is Error Fatal</b>	Controls whether to fail the pipeline if this annotator fails to create annotations.

## Keyword and Phrase Annotator

The table below defines the settings that are displayed when a Keyword and Phrase Annotator is edited.

Setting	Description
<b>Only Consider Text</b>	Controls whether to find phrases via a simplified format of the document. Enabling this setting can be beneficial for a document such as a rich HTML file. Enabling this option is less ideal for documents with multibyte characters.
<b>Require Nonstandard Word Boundaries</b>	Indicates whether the specified phrase can be present with or without surrounding character breaks (e.g. for Chinese) or with regex-nonstandard word boundaries (e.g. for Tagalog).
<b>Title</b>	Required field that specifies the unique name for the annotator.
<b>Description</b>	Optional field that provides a description of the annotator.
<b>Phrase</b>	Required field that specifies the terms or phrases to annotate. Type a word or phrase in the field and then click <b>Add</b> to add the phrase. You can add any number of phrases.

Setting	Description
Unintended Property Names	A list of any property names to filter out. Type a name in the field and then click <b>Add</b> to add the value.
Create Additional General Annotation Type	Controls whether the annotator creates a general shared annotation type in addition to the specific annotation types that are created.
Entity URI Property	Specifies an annotation property whose value you want to map to the URIs for instances of the class. For example, if a <code>Disease_ID</code> property has the value <code>http://example.com/Asthma</code> and you want to use <code>http://example.com/Asthma</code> as the base URI for instances of the class, add <code>Disease_ID</code> to this field. When Entity URI Property is not defined, the URI is auto-generated based on the name.
Explicit Property Datatypes/Objecttypes	A list of keys that map property names to a particular object property type.
Entity Name Property	Specifies an annotation property whose value you want to map to the names for instances of the class. For example, if a <code>Preferred_Label</code> property includes disease names and you want to use those label values as the names for instances of the Disease class, add <code>Preferred_Label</code> to this field. When Entity Name Property is not defined, the name is auto-generated.
Domain Object Base Class URI	If creating a general annotation type (Create Additional General Annotation Type is enabled), this setting specifies the class to use as the base type for the annotator's domain objects.
Unintended Classes	A list of any classes to filter out. Type a class in the field and then click <b>Add</b> to add the value.
Class Name Property	Specifies an annotation property whose value you want to map to the name of the class. For example, if a <code>Category</code> property has the value <code>Disease</code> and you want the name of the class to be "Disease," add <code>Category</code> to this field. When Class Name Property is not defined, the class name is auto-

Setting	Description
	generated.
Unintended Instances	A list of any entities or instances of the class to filter out. Type an instance in the field and then click <b>Add</b> to add the value.
Is Combine Annotation Instances	Controls whether to combine multiple instances of an extraction into one annotation.
Class URI Property	Specifies an annotation property whose value you want to map to the class URI in the model. For example, if a <code>Category_ID</code> property has the value <code>http://example.com/Disease</code> and you want to use <code>http://example.com/Disease</code> as the base class URI, add <code>Category_ID</code> to this field. When Class URI Property is not defined, the URI is auto-generated based on the name.
Output the Detections	Controls whether to include specific detections as an annotation property.
Unintended Property Values	A list of any property values to filter out. Type a value in the field and then click <b>Add</b> to add the value.
Is Error Fatal	Controls whether to fail the pipeline if this annotator fails to create annotations.

## Knowledgebase Annotator

The table below defines the settings that are displayed when a Knowledgebase Annotator is edited.

Setting	Description
Title	Required field that specifies the unique name for the annotator.
Description	Optional field that provides a description of the annotator.
Backing Graphmart	Optional field that specifies the graphmart or graphmarts to annotate.

Setting	Description
Backing Layer	<p>Optional field that specifies the data layer or layers to annotate.</p> <div> <p><b>Note</b></p> <p>The Backing Layer and Backing Graphmart fields are treated independently. Layers that you select do not have to be part of the graphmart that you specify in Backing Graphmart. And specifying a layer does not mean that you must select a Backing Graphmart. However, any layers or graphmarts that you select must contain classes and properties from the Backing Ontology or the data will not be annotated.</p> </div>
Backing Ontology	Required field that specifies the model for the backing data layers and/or graphmart.
Term Class	Required field that specifies the class of data for the annotations.
Term Label Property	Required field that lists the primary name or label property of the resources.
Term Identifying Properties	Required field that specifies the properties that contain names, aliases, or other identifiers to use for identifying the resources.
Backing Dataset	Optional field that specifies the dataset or datasets to annotate.
Case Sensitive	Controls whether matches must be case-sensitive.
Invalidating Properties	A list of any properties for which you do not want to find matching resources.
Discard Matches Of Common Words	Controls whether to discard matches of the most common words.
Discard Matches of Substrings	A list of the substrings for which you want matches to be discarded. Type a string in the field and then click <b>Add</b> to add the value.

Setting	Description
<b>Text Search Query Pattern Precedence</b>	When text search query properties are specified, this setting controls whether resource names or aliases are included as matches. When enabled, resource names and aliases will not be matched.
<b>Lucene Pattern Properties</b>	A list of properties that contain Lucene query syntax for document categorization.
<b>Approximate Label Properties</b>	A list of properties that contain phrases that may be matched only approximately, i.e., fault-tolerantly, via slightly alternate spellings or misspellings.
<b>Simplified Regex Pattern Properties</b>	A list of properties that contain simplified regular expressions.
<b>Regex Pattern Properties</b>	A list of properties that contain regular expressions.
<b>Strip Characters for Match</b>	Characters to strip out before determining if there is a match.
<b>Clear Caches</b>	Controls whether to clear any existing caches when the pipeline is run.
<b>Rows Per Query</b>	The maximum number of rows to query at a time when paging through the knowledgebase.
<b>Minimum Hit Length</b>	The minimum span length that can count as a match.
<b>Domain Object Base Class URI</b>	If creating a general annotation type (Create Additional General Annotation Type is enabled), this setting specifies the class to use as the base type for the annotator's domain objects.
<b>Class URI Property</b>	Specifies an annotation property whose value you want to map to the class URI in the model. For example, if a <code>Category_ID</code> property has the value <code>http://example.com/Disease</code> and you want to use

Setting	Description
Entity URI Property	<p><code>http://example.com/Disease</code> as the base class URI, add <code>Category_ID</code> to this field. When Class URI Property is not defined, the URI is auto-generated based on the name.</p> <p>Specifies an annotation property whose value you want to map to the URIs for instances of the class. For example, if a <code>Disease_ID</code> property has the value <code>http://example.com/Asthma</code> and you want to use <code>http://example.com/Asthma</code> as the base URI for instances of the class, add <code>Disease_ID</code> to this field. When Entity URI Property is not defined, the URI is auto-generated based on the name.</p>
Is Combine Annotation Instances	Controls whether to combine multiple instances of an extraction into one annotation.
Unintended Instances	A list of any entities or instances of the class to filter out. Type an instance in the field and then click <b>Add</b> to add the value.
Explicit Property Datatypes/Objecttypes	A list of keys that map property names to a particular object property type.
Unintended Classes	A list of any classes to filter out. Type a class in the field and then click <b>Add</b> to add the value.
Unintended Property Names	A list of any property names to filter out. Type a name in the field and then click <b>Add</b> to add the value.
Create Additional General Annotation Type	Controls whether the annotator creates a general shared annotation type in addition to the specific annotation types that are created.
Output the Detections	Controls whether to include specific detections as an annotation property.
Unintended Property Values	A list of any property values to filter out. Type a value in the field and then click <b>Add</b> to add the value.

Setting	Description
<b>Entity Name Property</b>	Specifies an annotation property whose value you want to map to the names for instances of the class. For example, if a <code>Preferred_Label</code> property includes disease names and you want to use those label values as the names for instances of the Disease class, add <code>Preferred_Label</code> to this field. When Entity Name Property is not defined, the name is auto-generated.
<b>Class Name Property</b>	Specifies an annotation property whose value you want to map to the name of the class. For example, if a <code>Category</code> property has the value <code>Disease</code> and you want the name of the class to be "Disease," add <code>Category</code> to this field. When Class Name Property is not defined, the class name is auto-generated.
<b>Is Error Fatal</b>	Controls whether to fail the pipeline if this annotator fails to create annotations.

## Regex Annotator

The table below defines the settings that are displayed when a Regex Annotator is edited.

Setting	Description
<b>Title</b>	Required field that specifies the unique name for the annotator.
<b>Description</b>	Optional field that provides a description of the annotator.
<b>Regular Expression Rule</b>	Required field that lists the regular expression rules for this annotator.
<b>Case-Insensitive</b>	Enables or disables case-insensitive matching. By default, case-insensitive matching assumes that only characters in the US-ASCII character set are being matched. Unicode-aware case-insensitive matching can be enabled by enabling <a href="#">Unicode Case Folding</a> in conjunction with this option.
<b>Multiline Mode</b>	Enables or disable multiline mode. When multiline mode is enabled, the expressions <code>^</code> and <code>\$</code> match immediately after or before a line terminator or the end of the input sequence. When multiline mode is disabled, these

Setting	Description
	expressions only match at the beginning and end of the entire input sequence.
<b>Allow Comments</b>	Controls whether whitespace and comments are allowed in a pattern. When enabled, whitespace and embedded comments starting with # are ignored until the end of a line.
<b>Canonical Equivalence</b>	Controls whether canonical equivalence is taken into account when finding matches. When enabled, characters are considered a match if and only if their full canonical decompositions match. For example, the expression <code>a\u030A</code> will match the string <code>\u00E5</code> .
<b>Enable Dotall</b>	Controls whether dotall mode is used. When enabled, the expression <code>.</code> matches any character, including a line terminator. When disabled, <code>.</code> does not match line terminators.
<b>Literal Parsing</b>	Controls whether literal parsing is employed. When enabled, the input string that specifies the pattern is treated as a sequence of literal characters and metacharacters and escape sequences have no special meaning.
<b>Unicode Case Folding</b>	Controls whether case-insensitive matching is done in a manner that is consistent with the Unicode Standard. By default, <a href="#">Case-Insensitive</a> matching assumes that only characters in the US-ASCII set are being matched.
<b>Unix Lines</b>	Enables or disables Unix line mode. When enabled, only the <code>\n</code> line terminator is recognized in the behavior of <code>.</code> , <code>^</code> , and <code>\$</code> .
<b>Is Combine Annotation Instances</b>	Controls whether to combine multiple instances of an extraction into one annotation.
<b>Class URI Property</b>	Specifies an annotation property whose value you want to map to the class URI in the model. For example, if a <code>Category_ID</code> property has the value <code>http://example.com/Disease</code> and you want to use



Setting	Description
	<code>http://example.com/Disease</code> as the base class URI, add <code>Category_ID</code> to this field. When Class URI Property is not defined, the URI is auto-generated based on the name.
Entity URI Property	Specifies an annotation property whose value you want to map to the URIs for instances of the class. For example, if a <code>Disease_ID</code> property has the value <code>http://example.com/Asthma</code> and you want to use <code>http://example.com/Asthma</code> as the base URI for instances of the class, add <code>Disease_ID</code> to this field. When Entity URI Property is not defined, the URI is auto-generated based on the name.
Class Name Property	Specifies an annotation property whose value you want to map to the name of the class. For example, if a <code>Category</code> property has the value <code>Disease</code> and you want the name of the class to be "Disease," add <code>Category</code> to this field. When Class Name Property is not defined, the class name is auto-generated.
Unintended Classes	A list of any classes to filter out. Type a class in the field and then click <b>Add</b> to add the value.
Entity Name Property	Specifies an annotation property whose value you want to map to the names for instances of the class. For example, if a <code>Preferred_Label</code> property includes disease names and you want to use those label values as the names for instances of the Disease class, add <code>Preferred_Label</code> to this field. When Entity Name Property is not defined, the name is auto-generated.
Unintended Instances	A list of any entities or instances of the class to filter out. Type an instance in the field and then click <b>Add</b> to add the value.
Unintended Property Values	A list of any property values to filter out. Type a value in the field and then click <b>Add</b> to add the value.
Create Additional General	Controls whether the annotator creates a general shared annotation type in

Setting	Description
<b>Annotation Type</b>	addition to the specific annotation types that are created.
<b>Output the Detections</b>	Controls whether to include specific detections as an annotation property.
<b>Explicit Property Datatypes/Objecttypes</b>	A list of keys that map property names to a particular object property type.
<b>Unintended Property Names</b>	A list of any property names to filter out. Type a name in the field and then click <b>Add</b> to add the value.
<b>Domain Object Base Class URI</b>	If creating a general annotation type (Create Additional General Annotation Type is enabled), this setting specifies the class to use as the base type for the annotator's domain objects.
<b>Is Error Fatal</b>	Controls whether to fail the pipeline if this annotator fails to create annotations.

# Versioning and Migrating Artifacts

The topics in this section provide guidance on creating backup versions of artifacts and migrating artifacts by exporting and importing artifacts and their related entities.

- Creating and Restoring Versions of Artifacts .....464
- Exporting an Artifact .....469
- Making Values Replaceable on Export .....475
- Importing Exported Versions of Artifacts .....476

## Related Topics

[Migration Packages](#) in the Administration Guide

# Creating and Restoring Versions of Artifacts

Before making changes to data sources, schemas, mappings, pipelines, models, graphmarts, etc., users can take a snapshot of the current version of that artifact. When a backup is created, Anzo automatically creates a backup version of each entity that is related to that artifact. For example, backing up a pipeline backs up the same version of any related models, mappings, schemas, and so on. In addition, Anzo backs up the metadata graphs for all of the entities. Metadata graphs store information about the artifacts such as the creator and creation date and the permissions configuration. Changed artifacts can be reverted at any time to any of the saved versions. If an artifact is restored to a previous version, Anzo automatically saves a version of the current state of the artifact and its related entities and metadata. Follow the appropriate instructions create or restore a backup version of an artifact.

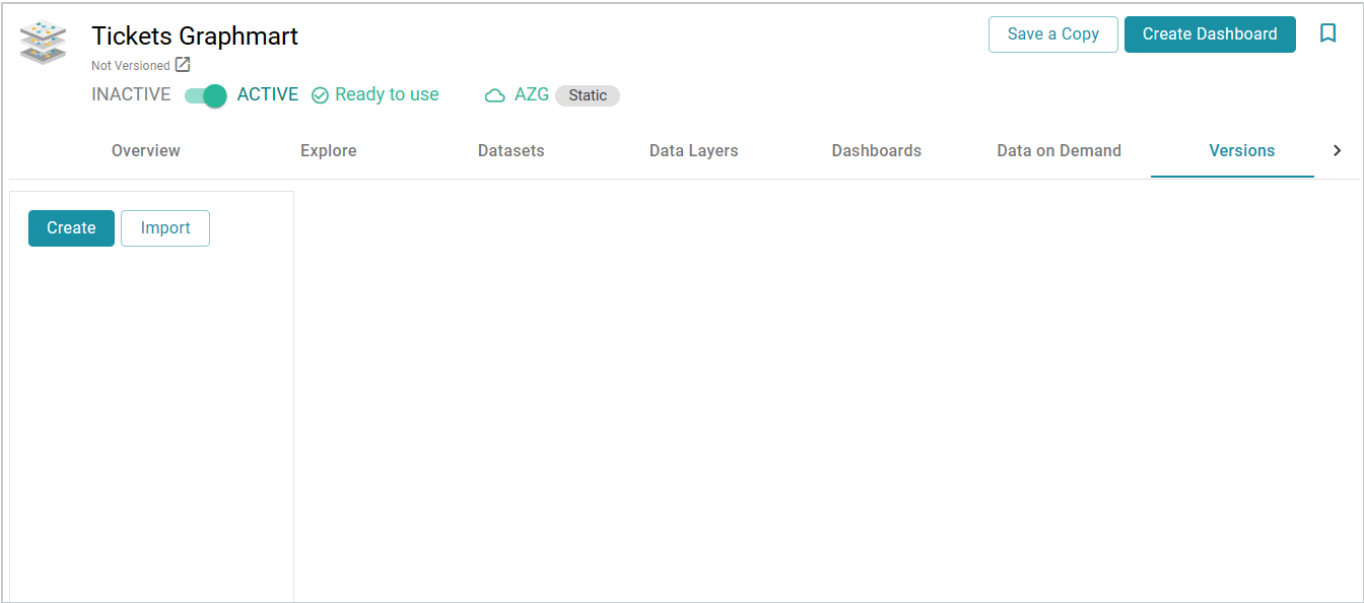
Creating a Backup Version of an Artifact ..... 464

Restoring a Backup Version of an Artifact .....467

## Creating a Backup Version of an Artifact

Follow the instructions below to save a snapshot of an artifact.

1. In the Anzo application, navigate to the artifact that you want to back up, and then click the **Versions** tab. For example, the image below shows the Versions tab for a graphmart. In this example, the graphmart does not have any backup versions:



## Note

For data models, add the model that you want to back up to the Working Set and then open it in the Model editor. Then click the **Versions** tab.

- Click the **Create** button on the left side of the screen. Anzo displays the Create New Version screen.


### Create New Version

Name for New Version \*

Comment for New Version

CANCEL SAVE

- In the **Name for New Version** field, type a name for the backup version. Then type details about the version in the optional **Comment for New Version** field.
- Click **Save**. Anzo takes a snapshot of the artifact as well as its related entities and adds the version to the list on the left side of the screen. Depending on the size and number of related entities, the backup operation can take a few minutes to complete. For example:



### Tickets Graphmart

Derived from: Tickets Backup DEV 1

INACTIVE ACTIVE Ready to use AZG Static

Overview Explore Datasets Data Layers Dashboards Data on Demand **Versions**

Create Import

Tickets Backup DEV 1

Derived from :  
Dec 19, 2022 | System Administrator

Export Restore

Tickets Backup

Description  
Backup of original graphmart

Derived from

Created Dec 19, 2022

Creator System Administrator

Version IDs

Current Env : DEV 1












Related Entities (19)

Type	Title	Version ID	Actions
CSV Data Source	Tickets	DEV 1	
Dataset Pipeline	Load Tickets to store	DEV 1	
File Backed Linked D...	Tickets to store	DEV 1	
File Based Dataset	Tickets to store	DEV 1	
File Connection	Server Shared Filesystem	DEV 1	
File Graph Data Source	store	DEV 1	

The screen displays details such as the version creator and created date and lists each of the related entities that were also backed up. In the list of related entities, the **Actions** column displays a compare icon next to each entity that has changed since the previous version.

## Example

In the image below, the compare icon in the graphmart row indicates that this version of the graphmart includes changes that were not in the previous version:

Related Entities (16)				
Type	Title	Version ID	Actions	
 Anzo Data Store	Store	DEV 1		
 CSV Data Sour...	Flights	DEV 1		
 Dataset Project	Load Flights	DEV 1		
 File Backed Lin...	Flights	DEV 1		
 File Based Dat...	Flights	DEV 1		
 File Connection	Server Filesystem	DEV 1		
 File Connection	sysadmin User Folder	DEV 1		
 Graphmart	Flights Graphmart	DEV 2		
 Layer	Flights	DEV 1		
 Layer	Queries	DEV 1		

Clicking the icon in the Actions column opens the Compare Versions dialog box, which shows a side-by-side comparison of the TriG files for the two versions:

Compare Versions of Graphmart "Flights Graphmart"	
<p>Ver: BackupV2 <span>DEV 2</span></p> <p>Derived from: BackupV1 Jan 29, 2020   System Administrator</p> <pre> 1 &lt;http://cambridgesemantics.com/Graphmart/bf8ad6f71cde47b4bfe8b48a209caf7d&gt; { 2   &lt;http://cambridgesemantics.com/Graphmart/bf8ad6f71cde47b4bfe8b48a209caf7d&gt; a &lt;http:// 3     &lt;http://cambridgesemantics.com/ontologies/Graphmarts#graphQueryEngineUri&gt; &lt;http 4       &lt;http://cambridgesemantics.com/ontologies/Graphmarts#ignoreErrors&gt; "true"^^&lt;ht 5         &lt;http://cambridgesemantics.com/ontologies/Graphmarts#layer&gt; &lt;http://cambridgesem 6           &lt;http://cambridgesemantics.com/ontologies/Graphmarts#manualRefresh&gt; "true"^^&lt;ht 7             &lt;http://purl.org/dc/elements/1.1/title&gt; "Flights Graphmart" ; 8             &lt;http://purl.org/dc/terms/created&gt; "2020-01-29T18:14:00.000Z"^^&lt;http://www.w3.o 9             &lt;http://purl.org/dc/terms/creator&gt; &lt;http://openanzo.org/system/internal/sysadm 10            &lt;http://purl.org/dc/terms/modified&gt; "2020-01-30T03:34:34.000Z"^^&lt;http://www.w3.o 11            &lt;http://purl.org/dc/terms/modified&gt; &lt;http://openanzo.org/system/internal/sysadm 12 13 14       &lt;http://cambridgesemantics.com/OrderedItem/520d3244d451435831ef10b059ba34e&gt; a &lt;ht 15       &lt;http://openanzo.org/ontologies/2008/07/AnzoOrderValue&gt; "1"^^&lt;http://www.w3.org/200 16       &lt;http://openanzo.org/ontologies/2008/07/AnzoOrderValue&gt; &lt;http://cambridgesem 17       &lt;http://purl.org/dc/terms/created&gt; "2020-01-30T03:34:34.000Z"^^&lt;http://www.w3.o 18       &lt;http://purl.org/dc/terms/creator&gt; &lt;http://openanzo.org/system/internal/sysadm 19       &lt;http://purl.org/dc/terms/modified&gt; "2020-01-30T03:34:34.000Z"^^&lt;http://www.w3.o 20       &lt;http://purl.org/dc/terms/modified&gt; &lt;http://openanzo.org/system/internal/sysadm 21 22       &lt;http://cambridgesemantics.com/OrderedItem/647719c11678494f9032b6288203cb51&gt; a &lt;ht 23 24 </pre>	<p>Ver: BackupV1 <span>DEV 1</span></p> <p>Derived from: Jan 29, 2020   System Administrator</p> <pre> 1 &lt;http://cambridgesemantics.com/Graphmart/bf8ad6f71cde47b4bfe8b48a209caf7d&gt; { 2   &lt;http://cambridgesemantics.com/Graphmart/bf8ad6f71cde47b4bfe8b48a209caf7d&gt; a &lt;http:// 3     &lt;http://cambridgesemantics.com/ontologies/Graphmarts#graphQueryEngineUri&gt; &lt;http 4       &lt;http://cambridgesemantics.com/ontologies/Graphmarts#ignoreErrors&gt; "true"^^&lt;ht 5         &lt;http://cambridgesemantics.com/ontologies/Graphmarts#layer&gt; &lt;http://cambridgesem 6           &lt;http://cambridgesemantics.com/ontologies/Graphmarts#manualRefresh&gt; "true"^^&lt;ht 7             &lt;http://purl.org/dc/elements/1.1/title&gt; "Flights Graphmart" ; 8             &lt;http://purl.org/dc/terms/created&gt; "2020-01-29T18:14:00.000Z"^^&lt;http://www.w3.o 9             &lt;http://purl.org/dc/terms/creator&gt; &lt;http://openanzo.org/system/internal/sysadm 10            &lt;http://purl.org/dc/terms/modified&gt; "2020-01-29T18:14:00.000Z"^^&lt;http://www.w3.o 11 12 13 14       &lt;http://purl.org/dc/terms/modified&gt; &lt;http://openanzo.org/system/internal/sysadm 15 16       &lt;http://cambridgesemantics.com/OrderedItem/647719c11678494f9032b6288203cb51&gt; a &lt;ht 17 18 </pre> <p>OK</p>

Users can now make changes to the current version of the backed up artifacts, and the new changes can be reverted to a backup version at any time. For information about restoring a version, see [Restoring a Backup Version of an Artifact](#).

## Restoring a Backup Version of an Artifact

Follow the instructions below to restore an artifact and its related entities to a previous backup version.

1. In the Anzo application, go to **Versions** tab for the artifact that you want to restore.

### Note

For data models, add the model that you want to restore to the Working Set and then open it in the Model editor. Then click the **Versions** tab.

2. On the Versions screen, select the backup version that you want to restore. For example:

Tickets Graphmart

Derived from: Tickets Backup DEV 1

INACTIVE ACTIVE Ready to use AZG Static

Overview

Explore

Datasets

Data Layers

Dashboards

Data on Demand

Versions

Create Import

Tickets Backup DEV 1

Derived from :  
Dec 19, 2022 | System Administrator

Tickets Backup

Description  
Backup of original graphmart

Derived from  
Created Dec 19, 2022  
Creator System Administrator

Version IDs  
Current Env : DEV 1

Export Restore

Related Entities (19)

Type	Title	Version ID	Actions
CSV Data Source	Tickets	DEV 1	
Dataset Pipeline	Load Tickets to store	DEV 1	
File Backed Linked D...	Tickets to store	DEV 1	
File Based Dataset	Tickets to store	DEV 1	
File Connection	Server Shared Filesystem	DEV 1	
File Graph Data Source	store	DEV 1	

3. Click the **Restore** button to restore the artifact to the version that you selected. Since Anzo automatically creates a snapshot of the current version before you restore an artifact, Anzo displays the Revert to Version dialog box so that you can specify a label for the new version.

**Restore Version: DEV 1 (Tickets Backup)**

Before restoring version "DEV 1 (Tickets Backup)", the current state will be saved as a new version.

Name for New Version \*

Comment for New Version

☐ Restore metadata graphs

☒ Create version of current state before restoring

CANCEL SAVE

4. In the Restore to Version dialog box, type a name for the version in the **Name for New Version** field. You can also add an optional comment in the **Comment for New Version** field.
5. Specify whether you want to revert to the backed up version's metadata graphs for this artifact and its related entities:
  - If you want the restored version to use the metadata, such as permission configuration and last created date, that was saved at the time of the backup, select the **Revert metadata graphs** checkbox. Anzo will revert the metadata to the saved version.
  - If changes were made to the metadata for the current version of the artifact and you want to preserve those changes, such as if the permissions were modified to further restrict or allow access, leave the **Revert metadata graphs** checkbox blank. Anzo will preserve the current metadata graphs instead of reverting the metadata to the saved version.
6. Specify whether you would like Anzo to create a backup version of the current version of the artifact before restoring it:
  - If you want Anzo to make a backup copy of the current version before restoring it, leave the **Create version of current state before restoring** checkbox selected.
  - If you want do not want Anzo to create a backup copy of the current version before restoring it, clear the **Create version of current state before restoring** checkbox.
7. Click **Save**. Anzo saves the current version and restores the current files to the backup version. The new version is added to the list of available backups.



## Exporting an Artifact

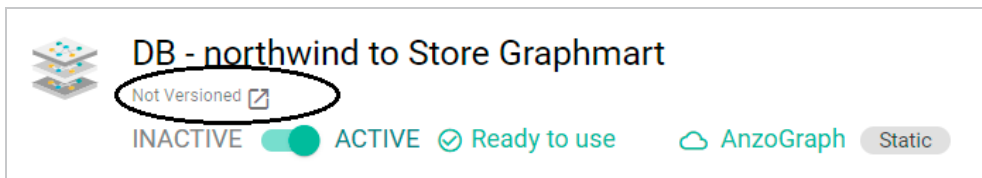
You can export the current version of an artifact and its related entities or any backup version. Follow the instructions below to export an artifact. For instructions on creating a backup version, see [Creating a Backup Version of an Artifact](#).

1. In the Anzo application, navigate to the artifact that you want to export.

### Note

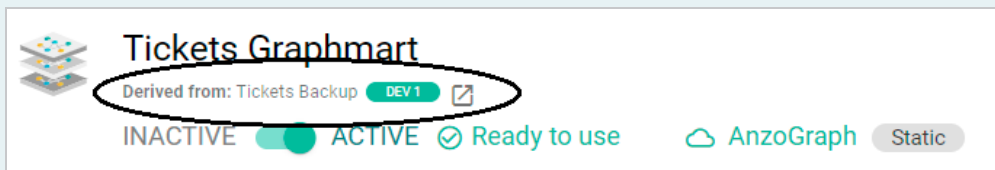
For data models, add the model that you want to export to the Working Set and then open it in the Model editor.

2. Follow one of the options below, depending on whether you want to export the current version of the artifact or a backup version:
  - If you want to export the current, working version of the artifact, click the Export icon (📄) under the artifact name, as shown in the image below.



### Tip

The image below shows an example of the Export icon for an artifact that has a backup version. Clicking the Export icon (📄) exports the current version of the artifact, not the backup version that is listed.



- If you want to export a backup version of the artifact, click the **Versions** tab, which lists the backups that exist for the artifact. Select the version that you want to export and then click the

**Export** button. For example, the image below shows the Versions tab for a graphmart:

OverviewExploreDatasetsData LayersDashboardsData on Demand**Versions**

Create VersionImport Version

Added Airports

Added airport data to graphmart

Derived from : Original Backup  
Jun 24, 2020 | System Administrator

Original Backup

Derived from :  
Jun 24, 2020 | System Administrator

Added Airports

Description

Added airport data to graphmart

Derived from : Original Backup  
Created : Jun 24, 2020  
Creator : System Administrator

Version IDs

Current Env : DEV 2

Related Entities (12)

Type	Title	Version ID	Actions
< > File Backed Linked D...	Airports	DEV 1	
< > File Backed Linked D...	Flights	DEV 1	
< > File Based Dataset	Airports	DEV 1	
< > File Based Dataset	Flights	DEV 1	
File Connection	Server Filesystem	DEV 1	
Graphmart	Flights Graphmart	DEV 2	
Layer	Airports	DEV 1	
Layer	Flights	DEV 1	
Load Data Step	Load Airports	DEV 1	
Load Data Step	Load Flights	DEV 1	
Ontology	Airports - Auto	DEV 1	

The Export dialog box is displayed. For example:

Export Current State

☒ Include Related Entities

☒ Include Registry Statements

☒ Include Metadata

☒ Include Dataset Editions And Components

Advanced

CANCELEXPORT

- On the Export dialog box, configure the following export options as needed. The list below describes each option:

- **Include Related Entities:** Indicates whether to export the artifact's related entities. Since most artifacts have dependencies with other artifacts, Cambridge Semantics recommends that you enable **Include Related Entities** (selected by default) and export all related entities. The number and type of related entities that are included varies by the type of artifact that is being exported.

#### Example

When exporting a pipeline, there are several artifacts that contribute to that pipeline besides the ETL jobs. Since the pipeline reads the source data, it requires the data source connection and schema artifacts. It also depends on the ontology and mapping artifacts for instructions on mapping and/or transforming the source data to the graph data model. And it requires the file store and Anzo data store artifacts to be able to write the resulting RDF data files to the appropriate location. Capturing all of the related entities ensures that the exported package includes all of the artifacts that the pipeline depends on to run successfully.

- **Include Registry Statements:** Indicates whether to export the registry statements for the artifact and each of its related entities.
  - **Include Metadata:** Indicates whether to export the metadata graph for the artifact and its related entities, such as the permissions configuration and last modified date. If you exclude the metadata, the artifacts in this export will follow the metadata configuration on the destination server when they are imported. Select **Include Metadata** if you want to migrate the existing metadata to the destination server. Enabling this setting also gives you the option to change the permissions configuration for the exported entities.
  - **Include Dataset Editions and Components:** If a dataset is included in the export package, this setting controls whether to include all of the editions and components with that dataset.
4. If you want to change permissions or replace the values for certain properties in the exported version of an entity, such as the user name and password for a data source, the base folder location for a File Store connection, or the file path for an Anzo Data Store, expand the **Advanced** option to view the Included Entities list. For example:

☒ Include Related Entities
 ☒ Include Registry Statements
 ☒ Include Metadata
 ☒ Include Dataset Editions And Components

Advanced

Included Entities

Sharing

<input checked="" type="checkbox"/>	Type	Title	
<input checked="" type="checkbox"/>	Configurable lens	Movies	
<input checked="" type="checkbox"/>	Direct Load Step	Load Movies	
<input checked="" type="checkbox"/>	Export Step	Export Movies	
<input checked="" type="checkbox"/>	< > File Backed Linked Data Set	Movies	^
<input checked="" type="checkbox"/>	< > File Based Dataset	Movies	
<input checked="" type="checkbox"/>	File Connection	Server Shared Filesystem	^

CANCEL

EXPORT

The entities with replaceable values are expandable. Click the ^ character to the right of an entity name to expand the options and view the editable properties. Replace any of the existing values with the new values that you want to define for the exported version of the artifact. For example:

Included Entities

Sharing


<input checked="" type="checkbox"/>	Type	Title	
<input checked="" type="checkbox"/>	CSV Data Source	Flights	▼
Variables		Replace	
File Path		<u>/flights10k.csv</u>	
<input checked="" type="checkbox"/>	Dataset Project	Load Flights	
<input checked="" type="checkbox"/>	File Connection	Server Filesystem	▼
Variables		Replace	
Base Folder		<u>/</u>	
<input checked="" type="checkbox"/>	File Connection	sysadmin User Folder	^

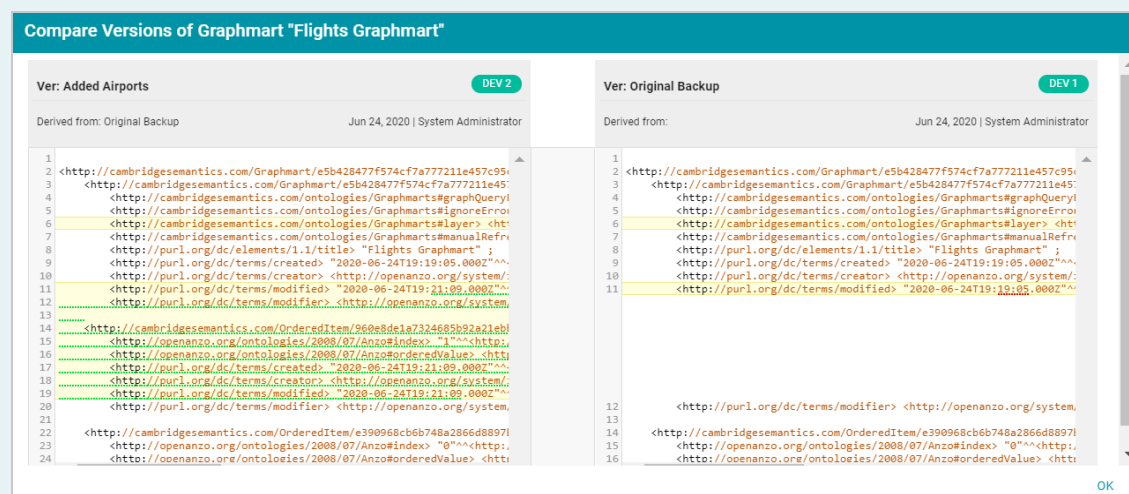
## Tip

For information about configuring additional properties so that their values are replaceable on export, see [Making Values Replaceable on Export](#).

If you specified **Include Metadata** and want modify permissions for the exported entities, click the **Sharing** tab. For information about changing permissions on the Sharing tab, see [Sharing Access to Artifacts](#).

## Tip

When a row in the Related Entities list includes the compare versions icon () in the Actions column, you can click the icon to open a side-by-side comparison of the TriG files for the two versions. For example:



## 5. Click **Export** to export the artifacts.

Anzo packages the files into a .zip file and downloads it to your computer. You do not need to extract the files in order to import the artifacts to another Anzo server. See [Exported ZIP File Contents](#) below for a description of the files that are included in the .zip file.

## Exported ZIP File Contents

Depending on the options configured for the export, the resulting .zip file contains one or more of the following files:

- **<artifact\_name>\_export.trig** contains statements about the type of artifact that was exported and the export settings that were configured.
- **<artifact\_name>\_graph.trig** contains the Model, Data Source, Schema, and Mapping definitions.

- **<artifact\_name>\_metadata.trig** contains metadata statements such as the access control configuration and last modified date for the exported entities.
- **<artifact\_name>\_registry.trig** contains registry statements such as the named graph information for the Data Source, Schema, Model, and instance data.

## Related Topics

[Importing Exported Versions of Artifacts](#)

## Making Values Replaceable on Export

When exporting artifacts, Anzo enables users to replace the existing values for properties like the user name and password for database data sources, the base folder location for file connections, and the file path for Anzo Data Stores. This topic provides instructions for configuring additional properties so that their values can be modified in the exported version of an artifact.

To configure a property so that its value is replaceable on export, add the following statement to the <http://cambridgesemantics.com/annotations/replaceStatements> graph:

```
<class_URI>
http://cambridgesemantics.com/ontologies/2018/06/Export#replaceStatement
<property_URI>
```

Where **<class\_URI>** is the URI for the class that defines the property whose value should be replaceable. And **<property\_URI>** is the URI of the property.

### Important

The specified property must be a datatype property that contains a literal value.

You can use the following TriG contents as a template for defining properties with replaceable values. The contents show the default replaceable properties. You can add your statements to the `ann:replaceStatements` list and then import the file.

```
@prefix ds: <http://cambridgesemantics.com/ontologies/DataSources#> .
@prefix exp: <http://cambridgesemantics.com/ontologies/2018/06/Export#> .
@prefix ann: <http://cambridgesemantics.com/annotations/> .

#Mode:ADD

ann:replaceStatements {
  ds:PathConnection exp:replaceStatement ds:filePath .
  ds:FileConnection exp:replaceStatement ds:fileConnectionBaseFolder .
  ds:DbDataSource exp:replaceStatement ds:dbUser , ds:dbDatabase, ds:dbPassword
  .
}
```

## Related Topics

[Exporting an Artifact](#)

# Importing Exported Versions of Artifacts

Follow the instructions below to import an exported version of an artifact and its related entities.

Note

To import a model that was created outside of Anzo or was downloaded from Anzo (as described in [Downloading a Model](#)), see [Uploading a Model to Anzo](#).

To import a version of a model that was exported from Anzo (as described in [Exporting an Artifact](#)), follow the instructions below.

1. In the Anzo application, go to the resource selection screen for the type of artifact that you want to import. For example, the image below shows the Graphmarts screen:

Search

Add Graphmart

<div></div>	Title		Description	Status	Statements	Actions	
	<div></div> Flights to Store Graphmart			<div></div> Ready to use	530,724	<div></div>	<div></div>
	<div></div> Movies to Store Graphmart			<div></div> Ready to use	176,754	<div></div>	<div></div>
	<div></div> MySQL DB - emrdbsmall to Store Graphma			<div></div> Ready to use	1,000,277	<div></div>	<div></div>
	<div></div> MySQL DB - northwind to Store Graphmart			<div></div> Ready to use	140,456	<div></div>	<div></div>

2. Click the **Add ...** button on the top of the screen and select **Import ...**. Anzo opens the Import dialog box.



Import

Version Location \*

BROWSE

Please select the zip file to import

☒ Include Related Entities
 ☒ Include Registry Statements
 ☐ Include Metadata
 ☒ Include Dataset Editions And Components

CANCEL

PROCEED WITH ADVANCED

PROCEED

- On the Import screen, click the **Version Location** field to open the File Location dialog box.

File Location

Source


☒ From Your Computer
 ☐ From File Store

Upload to

sysadmin User Folder

Change

The location where your uploaded files will be put on the Anzo Server. Defaults to your user folder.



Drag and drop files or [browse](#) from computer.
   
Please select the .zip file to import

Selected: None

CLEAR ALL

CREATE NEW FOLDER

CANCEL

OK

- If the exported .zip file is on your computer, drag and drop the file onto the screen or click **browse** to navigate to the file and select it. If the .zip file is on a File Store, select the **From File Store** radio button and select the file on the File Store.
- Click **OK** to save the file location value and close the File Location dialog box.
- Enable or disable the following options as needed, depending on the data that the import file contains:
  - Include Related Entities:** Indicates whether to import the artifact's related entities. Since most artifacts have dependencies with other artifacts, the **Include Related Entities** option is selected

by default when an artifact is exported. Capturing all related entities on export ensures that all of an artifact's dependencies are included when that artifact is migrated. For example, an exported pipeline has the data source, schema, mapping, and model artifacts that it relies on when the pipeline is published. If the exported package includes related entities, Cambridge Semantics recommends that you enable **Include Related Entities** on import.

- **Include Registry Statements:** Indicates whether to import the registry statements for the artifact and its related entities. This option is selected by default. Registry statements should be included in imports. When registry statements are not included, the imported artifacts are not displayed in Anzo. For example, if a data source artifact is imported without registry statements, it would not be added to the Data Sources registry and therefore not be displayed in the list of data sources in the Anzo application.
- **Include Metadata:** Indicates whether to import the metadata graph for the artifact and its related entities, such as the access control configuration and last modified date. If you select **Include Metadata**, you have the option to edit the permission configuration before importing the artifact.
- **Include Dataset Editions And Components:** This option specifies whether to import all of the editions and components for each dataset included in the import package.

7. Choose one of the following options to proceed with the import:

- If you want to import the files as alternate versions of artifacts and not as the current, working version, and you do not want to replace any values or change permissions, click **Proceed**. Anzo imports the data and the imported files become available as versions on the relevant Version screens for the imported artifacts.
- If you want to import these files as the current working version, and/or you want to change values or modify the permissions, click **Proceed With Advanced**. Anzo opens the Import Advanced Options dialog box. For example:

Import Advanced Options

Included Entities

Sharing

	Type	Title	
<input checked="" type="checkbox"/>	CSV Data Source	Movies	^
<input checked="" type="checkbox"/>	Dataset Pipeline	Load Movies to Store	
<input checked="" type="checkbox"/>	< > File Backed Linked Data Set	Movies to Store	^
<input checked="" type="checkbox"/>	< > File Based Dataset	Movies to Store	
<input checked="" type="checkbox"/>	File Graph Data Source	Store	^
<input checked="" type="checkbox"/>	Graphmart	Movies Graphmart	
<input checked="" type="checkbox"/>	Layer	Movies	
<input checked="" type="checkbox"/>	Load Data Step	Load Movies	
<input checked="" type="checkbox"/>	< > Local File Connection	Server Filesystem	^
<input checked="" type="checkbox"/>	Mapping Definition	Movies - MovieActors	
<input checked="" type="checkbox"/>	Mapping Definition	Movies - MovieDirectors	

CANCEL

IMPORT

IMPORT & APPLY

Click the ^ character to the right of an entity name to expand the options and view the editable properties. Replace any of the existing values with the new values that you want to define for the imported version of the entity. If you specified **Include Metadata** and want modify permission settings for the import, click the **Sharing** tab and edit or add permissions for users and groups. For details about the Sharing tab, see [Sharing Access to Artifacts](#).

When you are ready to import the entities, choose one of the following options:

- If you want to import the files as alternate versions and not as the current, working version, click **Import**. Anzo imports the files and the entities become available as versions on the relevant Version screens.
- If you want to import the files so that they become the current, working versions of the artifacts, click **Import & Apply**. Anzo creates a backup version of the existing working versions and then imports the artifacts as the new working versions.

# Model

Models define the business meaning of the source data. They describe the concepts, attributes, and relationships in or across data sets and define the desired structure of the data after it is onboarded to Anzo. Anzo links data to models to provide flexibility for capturing data from various sources and structures and to enable users to search for and visualize data in Hi-Res Analytics dashboards or other applications.

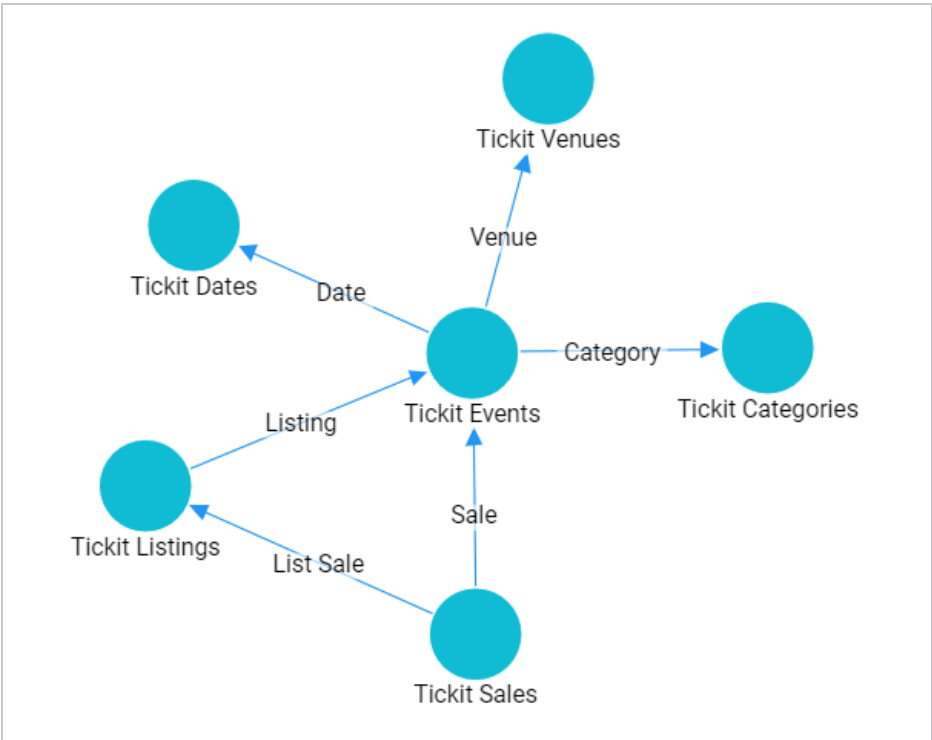
Users can import existing or external models (OWL ontologies) to Anzo, and models can be shared and reused. The topics in this section provide introductory information about data model terminology, describe model requirements and recommendations, and include instructions for creating and editing models.

- [Introduction to Model Concepts](#) .....481
- [Introduction to Managed Models](#) .....484
- [Model Requirements and Recommendations](#) .....486
- [Uploading a Model to Anzo](#) .....491
- [Creating a Model](#) .....494
- [Editing a Model](#) .....497
- [Downloading a Model](#) .....507
- [Defining Resource Templates](#) .....512

# Introduction to Model Concepts

Models define the business meaning of the source data. They describe the concepts, attributes, and relationships in or across data sets. Instead of reflecting the format or schema of the source data, models reflect the desired structure of the data after it is onboarded to Anzo. Anzo links data to models to provide flexibility for capturing data coming from various sources and structures and to enable users to search for and visualize data in Hi-Res Analytics dashboards or other applications. Users can import existing or external models (OWL ontologies) to Anzo, and models can be shared and reused.

The following image shows a portion of the model for a data set that captures sales activity for a fictional website where people buy and sell tickets for sporting events, shows, and concerts.



## Model Vocabulary

The table below defines key terms to know when working with models.

Term	Description
Class	Models are made up of classes. Classes describe a concept or a group of related objects. For example, the model above contains Tickit Events, Tickit Dates, Tickit Categories, Tickit

Term	Description
	Sales, and Tickit Listings classes.
<b>Base Class</b>	A base class is a more general version of a class. For example, a film model might have a Person base class with subclasses that categorize types of people.
<b>Subclass</b>	A subclass is a more specific version of another class. For example, in a film model a Person class might have subclasses of Actors, Directors, and Producers. Subclasses share the properties from the base class and can include additional properties.
<b>Property</b>	<p>Properties are attributes that describe the data in a class. For example the Tickit Events class has properties such as eventName, dateID, and startTime. There are two types of properties:</p> <ul style="list-style-type: none"> <li>• <b>Data property:</b> Relates a class to a simple or literal value. For example, in the events class, the eventName and startTime properties relate to simple values.</li> <li>• <b>Object property:</b> Relates a class to another class. For example, the listID property relates to the Tickit Sales and Tickit Events classes.</li> </ul>
<b>Property Type</b>	A property type is the data type of the values for the property, also known as the Property Range.
<b>Instance</b>	Instances are concrete occurrences of a class. For example, an event's name is an instance of the Tickit Events class.
<b>Simple Value</b>	<p>A simple value is also known as a literal value. The list below describes literal values:</p> <ul style="list-style-type: none"> <li>• Numbers (for example, 15, -9, 10.35)</li> <li>• Text strings (for example, "Jane Doe" or "a long description")</li> <li>• Dates and times (for example, "13-Dec-2021", or "April, 2022")</li> <li>• Boolean (true or false)</li> </ul>
<b>Type</b>	Either a class or a simple value.

## Related Topics



# Introduction to Managed Models

This topic provides an overview of managed models and includes important concepts to know when working with them.

## What is a Managed Model?

When you onboard a data source with the automated direct load workflow (as described in [Directly Loading Data Sources via Graphmarts](#)) or manually onboard a source with a Direct Load Step (as described in [Generating a Graph and Ontology with a Direct Load Step](#)), a *Managed Model* is produced. A managed model is generated, owned, and managed by the data layer that contains the auto-generated or user-created Direct Load Step. If a Direct Load Step query is changed, additional Direct Load Steps are added to the same layer, or the underlying source schema changes, the managed model is automatically updated when the graphmart or layer is reloaded or refreshed.

## Managed Models Cannot be Edited Outside of Direct Load Steps

Though a model that is generated in a Direct Load Step is registered in Anzo and is available for viewing in the Model editor, the model is owned and managed by the layer that contains the Direct Load Step. That means any manual changes made to the model outside of the step, such as from the Model editor, will be overwritten any time the graphmart or layer is refreshed or reloaded. **Do not modify managed models except by editing (or adding) Direct Load Step queries.**

## There is One Managed Model Per Data Layer

If you include multiple Direct Load Steps in the same layer, they will all update the same model. This functionality can be useful if you want to align the data and generated model across multiple steps. If you have multiple sources that are not intended to align or update the same model, create separate data layers.

## Deleting a Layer Deletes the Model

If you delete a layer that includes a managed model, the model is also deleted. Use caution when referencing managed models outside of graphmarts. For example, if you create a dataset and reference a managed model when you select the ontology, the reference will break if the data layer that manages the model is deleted.

## Related Topics



Directly Loading Data Sources via Graphmarts

Directly Loading a Data Source (Direct Load Step)

# Model Requirements and Recommendations

To ensure that data structures are properly defined, Anzo requires that data models include certain information and avoid unsupported information. This topic provides details about the requirements and guidelines to follow when uploading or creating models.

## Requirements

This section lists the requirements or rules to follow when uploading or creating a data model. Models that are generated by Anzo during the auto-ingest process conform to these rules.

- [Define each model as an owl:Ontology](#)
- [Define the model name with rdfs:label](#)
- [The named graph URI must match the ontology URI](#)
- [Define classes and concepts with owl:Class](#)
- [Define taxonomy with rdfs:subClassOf](#)
- [Define properties as owl:DatatypeProperty or owl:ObjectProperty](#)
- [Include rdfs:domain and rdfs:range for all properties](#)
- [Reference only Anzo-stored models](#)

### Define each model as an owl:Ontology

Define each data model as an **owl:Ontology**. To do so, include the following triple in the model:

```
<myOntology> a owl:Ontology
```

Where `myOntology` is the URI that names the model. The URI must be unique. To avoid unexpected results when saving a model, do not include a hash (#) character at the end of the model URI.

### Define the model name with rdfs:label

Use an **rdfs:label** property to define name of the model as a string. Include the following triple:

```
<myOntology> rdfs:label "My Ontology"^^xsd:string .
```

For example, you can use the following statement as a template for inserting owl:Ontology and rdfs:label into the model:

```
<myOntology> a owl:Ontology ;  
  rdfs:label "My ontology"^^xsd:string .
```

## The named graph URI must match the ontology URI

Make sure that the named graph URI for the model matches the ontology URI. For example:

```
<myOntology> { <myOntology> a owl:Ontology . }
```

Like a linked data set, an ontology is a core component that is used throughout the system. The registries that store and track the graphs for core components, such as the ontology registry, expect that each graph contains a resource that matches the graph URI and specifies the type of graph. Having a mismatched graph and ontology URI can break core Anzo functionality.

## Define classes and concepts with owl:Class

Use **owl:Class** for class or concept definitions. Do NOT include `skos:Concept` or `rdfs:Class`. For example, the following statement requires modification to make it valid in an Anzo model:

```
<myConcept> a skos:Concept
```

Changing the statement as follows correctly uses `owl:Class` instead of `skos:Concept`:

```
<myConcept> a owl:Class ;  
  rdfs:label <businessFacingClassLabel> .
```

## Define taxonomy with rdfs:subClassOf

Use **rdfs:subClassOf** for taxonomy. Do NOT use `skos:broader`. For example, the following statement requires modification to make it valid in an Anzo model:

```
<childSkosConcept> skos:broader <parentSkosConcept> .
```

Changing the statement as follows correctly uses `rdfs:subClassOf` instead of `skos:broader`:

```
<childOwlClass> rdfs:subClassOf <parentOwlClass> .
```

## Define properties as owl:DatatypeProperty or owl:ObjectProperty

Define properties using **owl:DatatypeProperty** or **owl:ObjectProperty**. For example:

```
<myObjectProperty> a owl:ObjectProperty .
```

Or

```
<myDatatypeProperty> a owl:DatatypeProperty .
```

### Include **rdfs:domain** and **rdfs:range** for all properties

Define **rdfs:domain** and **rdfs:range** for all properties. For example, the following property definition is incomplete:

```
<myObjectProperty> a owl:ObjectProperty .
```

The statement below completes the definition by adding **rdfs:label**, **rdfs:domain**, and **rdfs:range**:

```
<myObjectProperty> a owl:ObjectProperty ;  
  rdfs:label <businessFacingPropertyLabel> ;  
  rdfs:domain <myClass> ;  
  rdfs:range <myOtherClass> .
```

The example below shows a valid data type definition:

```
<myDatatypeProperty> a owl:DatatypeProperty ;  
  rdfs:label <businessFacingPropertyLabel> ;  
  rdfs:domain <myClass> ;  
  <myDatatypeProperty> rdfs:range <literal> .
```

#### Note

When defining the property range for integer values, use **xsd:int** instead of **xsd:integer**.

### Reference only Anzo-stored models

Models must be self-contained or include references only to models that are stored in Anzo.

## Guidelines

This section lists additional guidelines and important information to know when working with data models in Anzo.

- [Property Range Guidelines](#)
- [TriG is the preferred format for models to upload](#)

- [Load RDFS and OWL vocabularies as graphs](#)
- [Axiomatically defined classes and property hierarchies are not processed](#)

## Property Range Guidelines

When creating or editing properties in the model editor, Anzo offers several RDF property ranges or data types to choose from. Certain types are preferred over others, however, because they are treated consistently and predictably across systems. Cambridge Semantics recommends that you specify one of the following preferred property range values:

- **Boolean:** For true or false values.
- **Byte:** For 1-byte integers from -128 to 127.
- **Date:** For date values that follow a format such as YYYY-MM-DD.
- **Date time:** For 8-byte date and time values that follow a format such as YYYY-MM-DDThh:mm:ss. Note that dateTime values are normalized to GMT.
- **Double:** For up to 8-byte double floating point values.
- **Duration:** For a duration of time expressed as a number of years, months, days, hours, minutes, and seconds in a format such as PnYnMnDTnHnMnS.
- **Float:** For up to 4-byte floating point values with potential decimal places.
- **Int:** For up to 4-byte integers from -2,147,483,648 to 2,147,483,647.
- **Long:** For up to 8-byte integers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807.
- **Short:** For up to 2-byte integers from -32,768 to 32,767.
- **String:** For character values of varying length.
- **Time:** For time values that follow a format such as hh:mm:ss.

## TriG is the preferred format for models to upload

Anzo accepts model files in OWL (.owl), RDF (.rdf), TriG (.trig), TTL (.ttl), and XML (.xml) format. The preferred format for models that will be uploaded to Anzo is **TriG** (.trig) format.

## Load RDFS and OWL vocabularies as graphs

Anzo loads but does not process additional vocabulary data (such as `rdf:subPropertyOf`, `owl:sameAs`, and `owl:intersectionOf`, etc.) if they are encoded in models. Models that contain vocabularies rather than structural information should be loaded as RDF graphs instead. Anzo can load any valid RDF data. Since RDFS, SKOS, and OWL are valid RDF formats, the vocabulary information can be loaded as a graph, and the data can be interpreted with SPARQL in data layers and Hi-Res Analytics.

## **Axiomatically defined classes and property hierarchies are not processed**

When models include axiomatically defined classes or property hierarchies, Anzo loads the information but does not process the data. For example, Anzo does not infer information from axiomatically defined classes.

### **Related Topics**

[Introduction to Model Concepts](#)

[Introduction to Managed Models](#)

# Uploading a Model to Anzo

This topic provides instructions for uploading an existing model to Anzo. Follow these instructions if you have a model that was created outside of Anzo or was downloaded from Anzo as described in [Downloading a Model](#). Anzo accepts model files in OWL (.owl), RDF (.rdf), TriG (.trig), TTL (.ttl), and XML (.xml) format.

## Important

When uploading a model to Anzo, follow the requirements and guidelines defined in [Model Requirements and Recommendations](#).

If you want to import a version of a model that was exported from Anzo (as described in [Exporting an Artifact](#)), follow the instructions in [Importing Exported Versions of Artifacts](#) to import the model.

## Note

One of the following outcomes will occur if two users upload the same data model:

- If the second user does not have permission to modify the model that the first user uploaded, the second user receives an access denied error and cannot upload the model.
- If the second user does have permission to modify the model that the first user uploaded, Anzo overwrites the existing model with the version from user two.

1. In the Anzo application, click **Model**. Anzo displays the Manage Data Model Working Set screen. For example:

Manage Data Model Working Set

Search

Add Model

	Title	Class #	Description	Updated Date ↑	Actions
	Movies Layer Model	4	The managed model for 'Movies Layer' in the 'Movies' graphmart.	Nov 1, 2022	
	Flights Layer Model	1	The managed model for 'Flights Layer' in the 'Flights' graphmart.	Nov 1, 2022	
	DB-QA Layer Model	3	The managed model for 'DB-QA Layer' in the 'Books' graphmart.	Nov 2, 2022	
	DB-QA Layer Model	27	The managed model for 'DB-QA Layer' in the 'Northwind' graphmart.	Nov 2, 2022	
	Tickets Layer Model	6	The managed model for 'Tickets Layer' in the 'Tickets' graphmart.	Nov 2, 2022	
	Data Toolkit	32	Defines the interface to the Data Toolkit service.		
	SKOS Vocabulary	4	An RDF vocabulary for describing the basic structure and content of concept schem...		

Rows per page: 25

1-7 of 7

<

>

UPLOAD MODELS

CANCEL

OK

- On the bottom left corner of the screen, click **Upload Models**. The Upload Data Models dialog box opens.

Upload Data Models

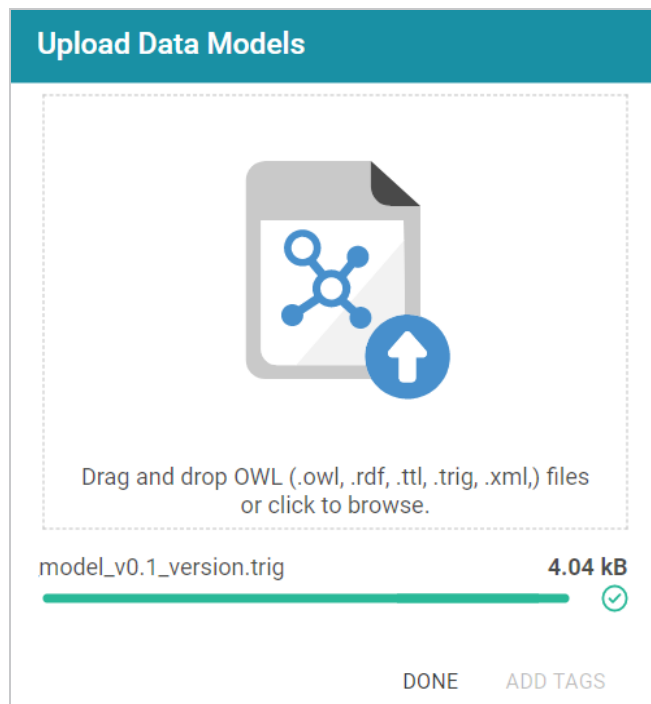
Drag and drop OWL (.owl, .rdf, .ttl, .trig, .xml,) files  
or click to browse.

DONE

ADD TAGS

- To upload a model, drag and drop the file onto the dialog box or click the text to browse and select the file on your computer. Anzo uploads the model that you selected and displays the file name and size. For example:





If you want to upload additional models, you can repeat the process and drag and drop or select files on the Upload Data Models dialog box.

4. If you want to add a tag or edit the tag that was specified in the uploaded model, you can click **Add Tags** and specify the tag in the dialog box. Then click **OK**.
5. Click **Done** when you finish uploading models. The new models become available on the Manage Data Model Working Set screen.

For information about editing models using the model editor, see [Editing a Model](#).

## Related Topics

[Model Requirements and Recommendations](#)

# Creating a Model

This topic provides instructions for creating a new data model in the Anzo application. For instructions on uploading an existing model to Anzo, see [Uploading a Model to Anzo](#).

1. In the Anzo application, click **Model**. Anzo displays the Manage Data Model Working Set screen. For example:

Manage Data Model Working Set

Search

Add Model

	Title	Class #	Description	Updated Date	Actions
	Movies Layer Model	4	The managed model for 'Movies Layer' in the 'Movies' graphmart.	Nov 1, 2022	<div></div> <div></div>
	Flights Layer Model	1	The managed model for 'Flights Layer' in the 'Flights' graphmart.	Nov 1, 2022	<div></div> <div></div>
	DB-QA Layer Model	3	The managed model for 'DB-QA Layer' in the 'Books' graphmart.	Nov 2, 2022	<div></div> <div></div>
	DB-QA Layer Model	27	The managed model for 'DB-QA Layer' in the 'Northwind' graphmart.	Nov 2, 2022	<div></div> <div></div>
	Tickets Layer Model	6	The managed model for 'Tickets Layer' in the 'Tickets' graphmart.	Nov 2, 2022	<div></div> <div></div>
	Data Toolkit	32	Defines the interface to the Data Toolkit service.		<div></div> <div></div>
	SKOS Vocabulary	4	An RDF vocabulary for describing the basic structure and content of concept schem...		<div></div> <div></div>

Rows per page: 25 1-7 of 7

UPLOAD MODELS CANCEL OK

2. Click the **Add Model** button on the top right of the screen and select **New Model**. Anzo displays the Model editor.

The screenshot shows the Anzo application interface for creating a new model. The top bar includes buttons for 'Modify Set', 'Create New Model', and 'Remove From Set', along with a 'Delete' button. The main title is 'Untitled', and it indicates 'Not Versioned'. The interface is divided into two tabs: 'Explore' and 'Details', with 'Details' currently selected. On the left sidebar, there is a search bar for 'Ontology Items' and a list containing 'Untitled Model'. The main content area contains four input fields: 'Label \*' (placeholder: 'The label of the entity'), 'Description' (placeholder: 'A brief description of the entity'), 'Serialization Prefix' (placeholder: 'The prefix that should be used when serializing elements defined in this ontology'), and 'Class Prefix' (placeholder: 'The base URI that should be used when creating classes defined in this ontology'). A 'Save All' button is located at the bottom left of the sidebar.

3. In the **Label** field, type a unique name for the model.
4. Provide the following optional information as needed:
  - **Description:** A brief description of the model.
  - **Serialization Prefix:** The prefix to use for this model when Anzo serializes it. For example, the prefix for the Friend of a Friend (FOAF) model is "foaf," and the prefix for Dublin Core is "dc."

#### Tip

The Prefix value is also used to provide hints when typing queries in the Query Builder. When writing a query against a data source that has this model in scope, typing in the PREFIX clause presents this Prefix value as a suggestion.

- **Class Prefix:** The custom URI template to follow for classes in this model. The value must be a valid URI. When the Class Prefix is set, the URIs for the classes in this model will follow the specified scheme. For example, if Class Prefix is set to **http://cambridgesemantics.com/class/** and a class called **Employees** is created in the model, the URI that is generated for the Employees class will be **http://cambridgesemantics.com/class/Employees**.
- **Property Prefix:** The custom URI template to follow for properties in this model. The value must be a valid URI. When the Property Prefix is set, the URIs for the properties in this model will follow the specified scheme. For example, if Property Prefix is set to

**http://cambridgesemantics.com/property/** and a property called **LastName** is created in the model, the URI that is generated for the LastName property will be **http://cambridgesemantics.com/property/LastName**.

- **Imports:** Lists any definitions that you want to import from another model into this model. To select models to import, click in the **Imports** field and select a model from the drop-down list. Select the field again to select additional models.
- **System Model:** Indicates that the data model is a system model only and not related to business data.
- **Hidden Model:** Hides the data model so that it is not associated with business data.

5. Click **Save** to save the model.

For information about adding classes and properties to the new model, see [Editing a Model](#).

# Editing a Model

This topic provides information about using the Anzo model editor to open a data model and modify it to add, edit, or remove classes, properties, data ranges, and annotations.

## Important

Do not modify auto-generated, layer-managed models. Changes will be overwritten whenever the host graphmart or layer is reloaded or refreshed. For more information, see [Introduction to Managed Models](#).

- [Opening Models in the Editor](#)
- [Changing Model Components](#)
- [Class Editor Reference](#)
- [Property Editor Reference](#)

## Tip

Before editing a model, you have the option to create a backup of the current version. For more information, see [Creating and Restoring Versions of Artifacts](#).






















## Opening Models in the Editor

1. In the Anzo application, click **Model**. Anzo displays the Manage Data Model Working Set screen. For example:

Manage Data Model Working Set

Search

Add Model

<input type="checkbox"/>	Title	Class #	Description	Updated Date	Actions
<input type="checkbox"/>	 Movies Layer Model	4	The managed model for 'Movies Layer' in the 'Movies' graphmart.	Nov 1, 2022	 
<input type="checkbox"/>	 Flights Layer Model	1	The managed model for 'Flights Layer' in the 'Flights' graphmart.	Nov 1, 2022	 
<input type="checkbox"/>	 DB-QA Layer Model	3	The managed model for 'DB-QA Layer' in the 'Books' graphmart.	Nov 2, 2022	 
<input type="checkbox"/>	 DB-QA Layer Model	27	The managed model for 'DB-QA Layer' in the 'Northwind' graphmart.	Nov 2, 2022	 
<input type="checkbox"/>	 Tickets Layer Model	6	The managed model for 'Tickets Layer' in the 'Tickets' graphmart.	Nov 2, 2022	 
<input type="checkbox"/>	 Data Toolkit	32	Defines the interface to the Data Toolkit service.		 
<input type="checkbox"/>	 SKOS Vocabulary	4	An RDF vocabulary for describing the basic structure and content of concept schem...		 

Rows per page: 25 1-7 of 7

UPLOAD MODELS

CANCEL OK

2. On the Manage Working Set screen, select the checkbox next to the model (or models) that you want to add to the working set and edit. Then click **OK**. Anzo opens the selected model in the editor. For example:

Modify Set

Create New Model

DB-QA Layer Model

Not Versioned

Explore

Details

Comments

Sharing

Remove From Set

Delete

Search Ontology Items

DB-QA Layer Model

Alphabetical List Of Products

Categories

Category Sales For 1997

Current Product List

Customer And Suppliers By City

Customers

Employee Territories

Employees

Invoices

Order Details

Order Details Extended

Order Subtotals

Orders

Save All

DB-QA Layer Model

Label \*

DB-QA Layer Model

The label of the entity

http://cambridgesemantics.com/Layer/acfdea8ce55c42ca979fa90ebc6a43e9/Model

Description

The model for the 'Northwind' graphmart.

A brief description of the entity

Serialization Prefix

The prefix that should be used when serializing elements defined in this ontology

Class Prefix

The base URI that should be used when creating classes defined in this ontology

3. You can edit the following model-level settings or view the [Changing Model Components](#) section below for information about working with classes, properties, annotations, and data ranges.

- **Description:** A brief description of the model.
- **Serialization Prefix:** The prefix to use for this model when Anzo serializes it. For example, the prefix for the Friend of a Friend (FOAF) model is "foaf," and the prefix for Dublin Core is "dc."

#### Tip

The Prefix value is also used to provide hints when typing queries in the Query Builder. When writing a query against a data source that has this model in scope, typing in the PREFIX clause presents this Prefix value as a suggestion.

- **Class Prefix:** The custom URI template to follow for classes in this model. The value must be a valid URI. When the Class Prefix is set, the URIs for the classes in this model will follow the specified scheme. For example, if Class Prefix is set to **http://cambridgesemantics.com/class/** and a class called **Employees** is created in the model, the URI that is generated for the Employees class will be **http://cambridgesemantics.com/class/Employees**.
- **Property Prefix:** The custom URI template to follow for properties in this model. The value must be a valid URI. When the Property Prefix is set, the URIs for the properties in this model will follow the specified scheme. For example, if Property Prefix is set to

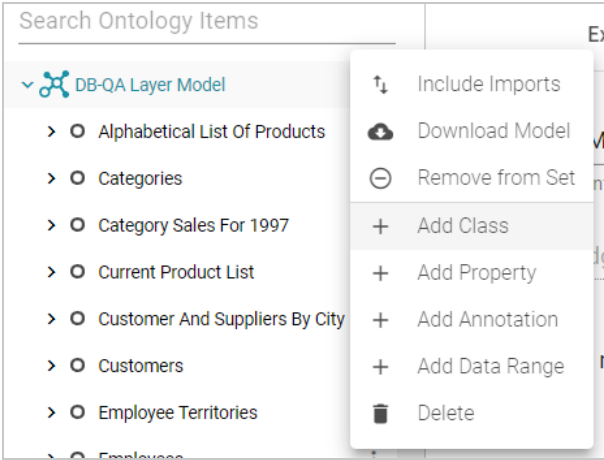
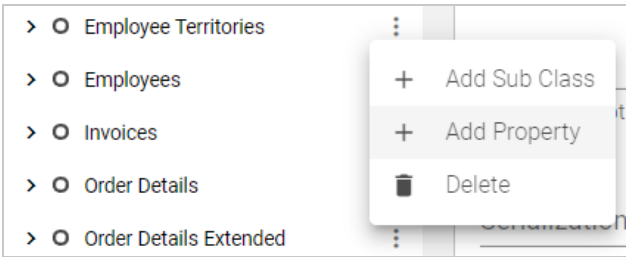
**<http://cambridgesemantics.com/property/>** and a property called **LastName** is created in the model, the URI that is generated for the LastName property will be **<http://cambridgesemantics.com/property/LastName>**.

- **Imports:** Lists any definitions that you want to import from another model into this model. To select models to import, click in the **Imports** field and select a model from the drop-down list. Select the field again to select additional models.
- **System Model:** Indicates that the data model is a system model only and not related to business data.
- **Hidden Model:** Hides the data model so that it is not associated with business data.



# Changing Model Components

The table below provides instructions for working with model components. When modifying models, make sure that you click **Save** periodically to save your changes.

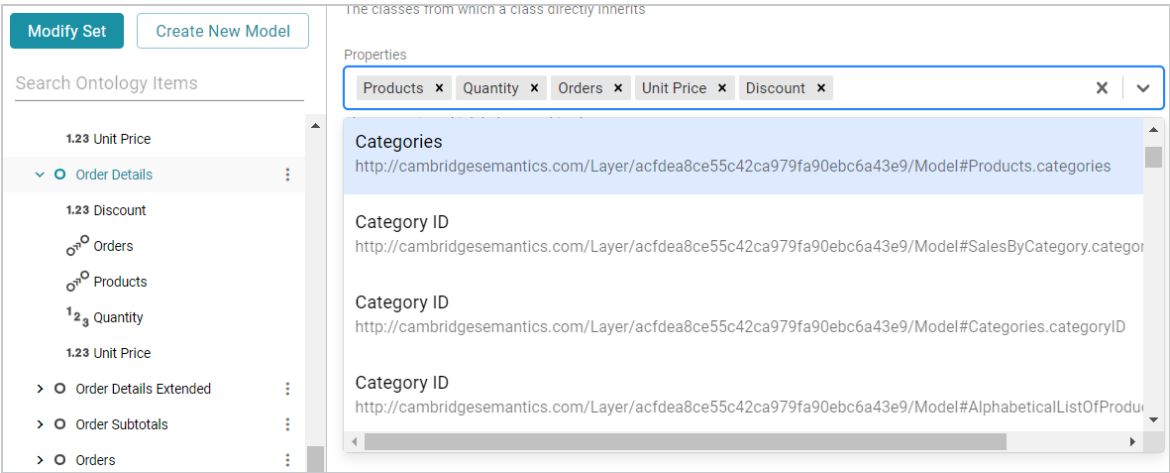
What do you want to do?	Instructions
Create a new class	<p>Open the model menu by clicking the menu icon (⋮) to the right of the model name. Then select <b>Add Class</b>.</p>  <p>Anzo opens the class editor so that you can configure the new class. See <a href="#">Class Editor Reference</a> below for information about class settings.</p>
Create a new property in a class	<p>Open the class menu by clicking the menu icon (⋮) to the right of the class name. Then select <b>Add Property</b>.</p>  <p>Anzo opens the property editor so you can configure the new property. See <a href="#">Property Editor Reference</a> below for information about property settings.</p>
Add an	To add an existing property to a class, click the class in the left pane to display the class

What do you want to do?

Instructions

existing property to a class

details in the editor. In the editor, click in the **Properties** field and select the property that you want to add from the drop-down list. For example:



Edit a class

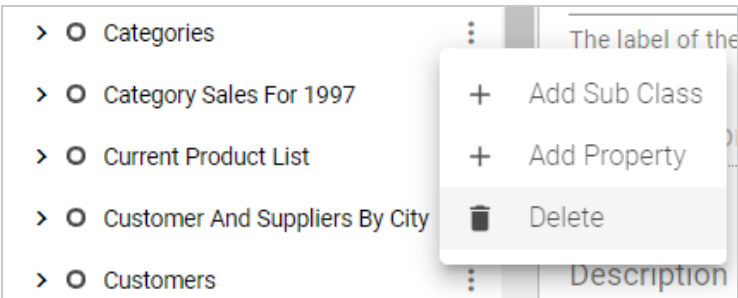
To change an existing class, select the class in the left pane. Anzo expands the class to show its properties and displays the details for that class in the editor. You can make changes in the editor. See [Class Editor Reference](#) below for information about class settings.

Delete a property from a class

In the left pane of the working set, select the property that you want to delete. Anzo opens that property in the editor. To remove the property, click the **Delete** button on the top right of the screen. Then click **Delete** in the dialog box to confirm that you want to delete the property.

Delete a class from the model

Click the menu icon (⋮) to the right of the class that you want to remove from the model.



Click **Delete**. Anzo displays a dialog box that asks if you want to delete only the class or

What do you want to do?	Instructions
	<p>all of the subclasses and properties in the class. Select the appropriate option and then click <b>Delete</b> to confirm that you want to delete the class.</p> <div><b>Important</b> This action cannot be undone. Anzo removes the class and saves the model.</div>
Add a data range	Click the menu icon (⋮) to the right of the model name. Then select <b>Add Data Range</b> . Anzo opens the data range editor so that you can configure the new range.
Add an annotation	Click the menu icon (⋮) to the right of the model name. Then select <b>Add Annotation</b> . Anzo opens the editor so that you can configure the annotation.

## Class Editor Reference

This section describes each of the fields that are available for configuring classes.

Field	Description
Label	The name of the class.
Description	A brief description of the class.
Parent Classes	Lists any parent classes under which this class becomes a child or subclass. Click in the field to select parent classes from the drop-down list. Or click the X to the left of a class name to remove that parent class from the list.
Properties	Lists the properties under this class. Click in the field to a property from the drop-down list. Or click the X to the right of a property name to remove that property from the list.
Inherited Properties	Properties that the class has inherited from a super class or the model.
Preview Property	Defines a property from the class to use as the "name" or entity on default displays. For example, if there is a reference to entity X, and entity X has Name, Title, and Label properties, you could specify that you want Title to display by default instead of "X."
Resource Template	Defines the Uniform Resource Identifier (URI) template to use for instances of the class. You can construct URI templates by typing a value and pressing <b>Enter</b> or by choosing an available property from the drop-down list. For more information, see <a href="#">Defining Resource Templates</a> .
Graph Template	Defines the graph URI template to use for instances of the class. You can construct graph URI templates by typing a value and pressing <b>Enter</b> or by choosing an available property from the drop-down list. You can concatenate the specified graph template value with values of properties in the class. For example, <code>http://cambridgesemantics.com/graph/</code> and <code>Title</code>

## Property Editor Reference

This section describes each of the fields that are available for configuring properties.

Field	Description
Label	The name of the property.
Description	A brief description of the property.
Required	Indicates whether a value is required for this property.
Multi Value	Indicates whether more than one value can exist for this property. <div><b>Note</b> Some business intelligence (BI) applications have limitations on the retrieval of multi-value properties. If you use the Anzo Data on Demand service to query data from BI tools, consider whether your application supports multi-value properties before creating them.</div>
Has Data Range	Indicates whether the property has a single data type or a data range. Selecting this checkbox displays the Data Range field so that you can choose the data range.
Property Range	The data type for the property. See <a href="#">Property Range Guidelines</a> for recommendations on choosing property ranges.
Domain	Lists the class or classes that the property belongs to.
Min Cardinality	The minimum number of distinct values a property can have. When Min Cardinality is blank, the number of values is unrestricted.
Max Cardinality	The maximum number of distinct values a property can have. When Max Cardinality is blank, the number of values is unrestricted.

Field	Description
Value Restriction	Indicates whether to restrict the property's values to certain data types or specific values in a list.

## Related Topics

[Defining Resource Templates](#)

# Downloading a Model






















This topic provides instructions for downloading a data model from Anzo.

1. In the Anzo application, click **Model**. Anzo displays the Manage Data Model Working Set screen. For example:

Manage Data Model Working Set

Search

Add Model

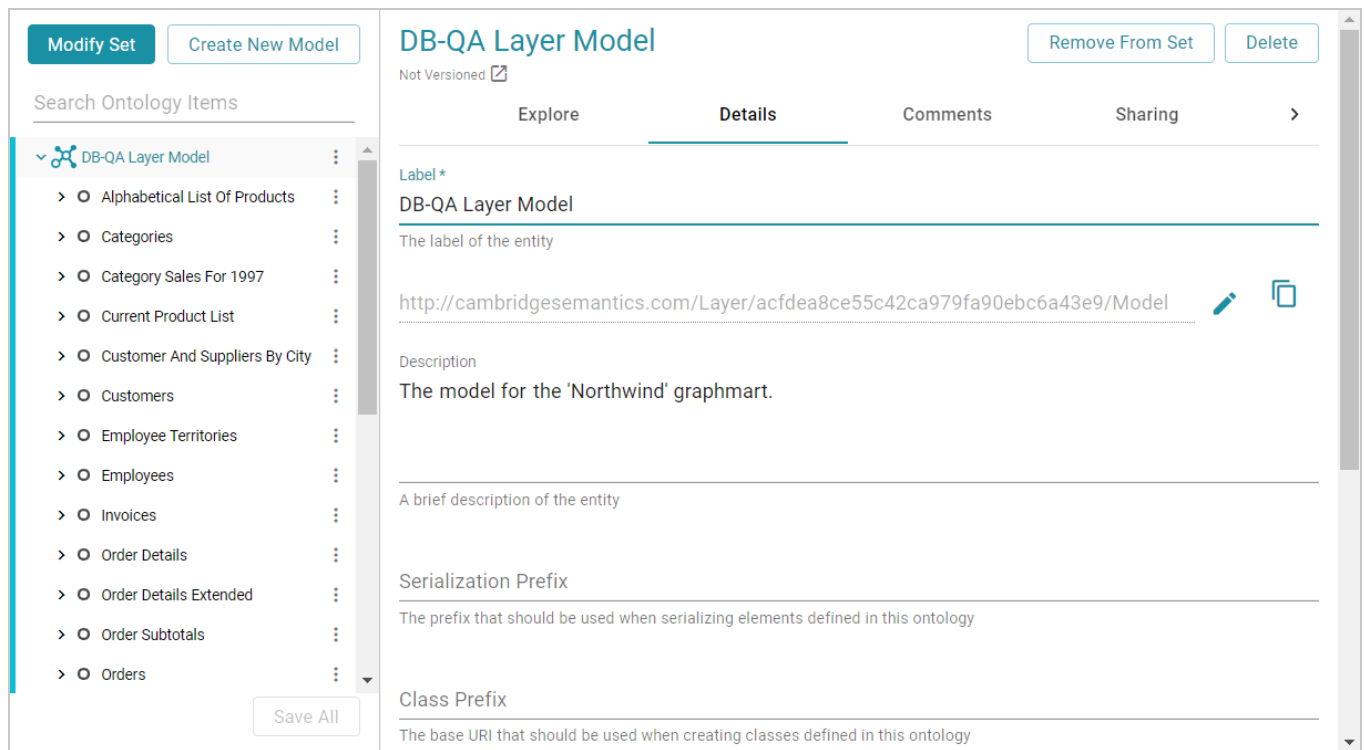
<input type="checkbox"/>	Title	Class #	Description	Updated Date	Actions
<input type="checkbox"/>	 Movies Layer Model	4	The managed model for 'Movies Layer' in the 'Movies' graphmart.	Nov 1, 2022	 
<input type="checkbox"/>	 Flights Layer Model	1	The managed model for 'Flights Layer' in the 'Flights' graphmart.	Nov 1, 2022	 
<input type="checkbox"/>	 DB-QA Layer Model	3	The managed model for 'DB-QA Layer' in the 'Books' graphmart.	Nov 2, 2022	 
<input type="checkbox"/>	 DB-QA Layer Model	27	The managed model for 'DB-QA Layer' in the 'Northwind' graphmart.	Nov 2, 2022	 
<input type="checkbox"/>	 Tickets Layer Model	6	The managed model for 'Tickets Layer' in the 'Tickets' graphmart.	Nov 2, 2022	 
<input type="checkbox"/>	 Data Toolkit	32	Defines the interface to the Data Toolkit service.		 
<input type="checkbox"/>	 SKOS Vocabulary	4	An RDF vocabulary for describing the basic structure and content of concept schem...		 

Rows per page: 25 1-7 of 7

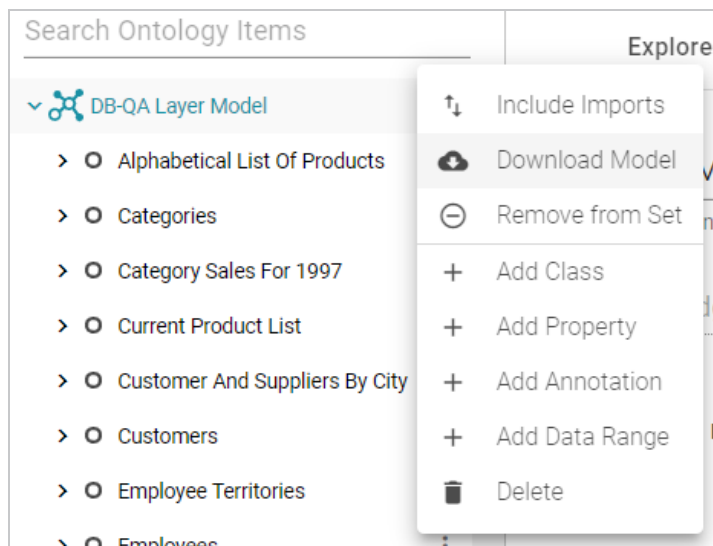
UPLOAD MODELS

CANCEL OK

2. On the Manage Working Set screen, select the checkbox next to the model that you want to download, and then click **OK**. Anzo opens the selected model in the editor. For example:

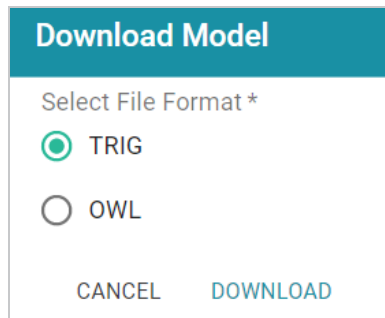


3. Open the model menu by clicking the menu icon (⋮) to the right of the model name. Then select **Download Model**.



Anzo displays the Download Model dialog box:





Download Model

Select File Format \*

☒ TRIG

☐ OWL

CANCEL DOWNLOAD

4. In the Download Model dialog box, select the format to save the model in. By default Anzo saves models in **TRIG** format. If you want to save the file in OWL format, select the **OWL** radio button. Then click **Download**.

Anzo downloads the model to your computer in the selected format.

#### Note

When a model is downloaded from Anzo, the resulting TriG or OWL file size can be significantly larger than the file size of the original data model file that was uploaded. The original model likely includes prefix specifications and abbreviated URIs. When a model is exported, however, Anzo replaces the prefixes with full URIs. In addition, the downloaded model includes the Anzo-generated metadata for the model.

For example, the following simple example TTL content shows part of a data model that uses prefixes:

```
@prefix csi: <http://cambridgesemantics.com/2017/02/ont#> .
csi:testModel a owl:Ontology ;
  rdfs:label "Test Model"^^xsd:string .
csi:DOB a owl:Class;
  rdfs:domain csi:Demographics ;
  rdfs:label "DOB" ;
  rdfs:range xsd:string .
csi:HEIGHT a owl:Class;
  rdfs:domain csi:Demographics ;
  rdfs:label "HEIGHT" ;
  rdfs:range xsd:decimal .
```

After uploading the TTL file and then downloading the model in TriG format, the resulting file includes full URIs as well as the model's metadata:

```

<http://cambridgesemantics.com/2017/02/ont#testModel> {
  <http://cambridgesemantics.com/2017/02/ont#DOB> a
<http://www.w3.org/2002/07/owl#Class> ;
  <http://www.w3.org/2000/01/rdf-schema#domain>
<http://cambridgesemantics.com/2017/02/ont#Demographics> ;
  <http://www.w3.org/2000/01/rdf-schema#label> "DOB" ;
  <http://www.w3.org/2000/01/rdf-schema#range>
<http://www.w3.org/2001/XMLSchema#string> .

  <http://cambridgesemantics.com/2017/02/ont#HEIGHT> a
<http://www.w3.org/2002/07/owl#Class> ;
  <http://www.w3.org/2000/01/rdf-schema#domain>
<http://cambridgesemantics.com/2017/02/ont#Demographics> ;
  <http://www.w3.org/2000/01/rdf-schema#label> "HEIGHT" ;
  <http://www.w3.org/2000/01/rdf-schema#range>
<http://www.w3.org/2001/XMLSchema#decimal> .
  <http://cambridgesemantics.com/2017/02/ont#testModel> a
<http://www.w3.org/2002/07/owl#Ontology> ;
  <http://www.w3.org/2000/01/rdf-schema#label> "Test Model" .
}
<http://cambridgesemantics.com/registries/Ontologies> {
  <http://cambridgesemantics.com/registries/Ontologies>
  <http://openanzo.org/ontologies/2008/07/Anzo#defaultNamedGraph>
  <http://cambridgesemantics.com/2017/02/ont#testModel> ;
  a <http://openanzo.org/ontologies/2008/07/Anzo#Dataset> .
}
<http://openanzo.org/metadataGraphs
(http%3A%2F%2Fcambridgesemantics.com%2F2017%2F02%2Font%23testModel)> {
  <http://cambridgesemantics.com/2017/02/ont#testModel>
  <http://openanzo.org/ontologies/2008/07/Anzo#canBeAddedToBy>
  <http://openanzo.org/system/internal/sysadmin> ;
  <http://openanzo.org/ontologies/2008/07/Anzo#canBeReadBy>
  <http://openanzo.org/Role/everyone> ,
<http://openanzo.org/system/internal/sysadmin> ;
  <http://openanzo.org/ontologies/2008/07/Anzo#canBeRemovedFromBy>
  <http://openanzo.org/system/internal/sysadmin> .
  ...
}

```

## Related Topics

[Uploading a Model to Anzo](#)

# Defining Resource Templates

When you open a data model in the Model editor, there is a **Resource Template** setting for each of the classes in the model. A Resource Template defines the Uniform Resource Identifier (URI) pattern that Anzo should follow when ingesting data and generating the URIs for the instances of each class.

## Tip

For property URIs, the default URI prefix is **<http://cambridgesemantics.com/>**. The value is controlled by the URI Prefix option in server settings. See [Configure URI Prefix and SPARQL Options](#) in the Administration Guide for more information.

Defining a Resource Template for the classes in your models helps link and relate data by using URI patterns that express the meaning of the data and combine similar concepts. Additionally, simpler and more meaningful URIs are easier to read and therefore easier to write in queries.

## Important

Keep the following points in mind when defining class instance URI patterns:

- Do not modify auto-generated, layer-managed models. Changes will be overwritten whenever the host graphmart or layer is reloaded or refreshed. For more information, see [Introduction to Managed Models](#).
- Avoid joining data that should not be joined. For example, using a property such as YearProduced in a movies Resource Template would group all movies from a given year as a single instance.
- Resource Templates with multiple components must have all components present. If a component is missing, Anzo generates random strings for missing Resource Template components.
- Resource templates do not work across different classes. You must define resource templates on individual classes.

1. In the Anzo application, click **Model**. Anzo displays the Manage Data Model Working Set screen. For example:

Manage Data Model Working Set

Search

Add Model

	Title	Class #	Description	Updated Date ↑	Actions
	Movies Layer Model	4	The managed model for 'Movies Layer' in the 'Movies' graphmart.	Nov 1, 2022	
	Flights Layer Model	1	The managed model for 'Flights Layer' in the 'Flights' graphmart.	Nov 1, 2022	
	DB-QA Layer Model	3	The managed model for 'DB-QA Layer' in the 'Books' graphmart.	Nov 2, 2022	
	DB-QA Layer Model	27	The managed model for 'DB-QA Layer' in the 'Northwind' graphmart.	Nov 2, 2022	
	Tickets Layer Model	6	The managed model for 'Tickets Layer' in the 'Tickets' graphmart.	Nov 2, 2022	
	Data Toolkit	32	Defines the interface to the Data Toolkit service.		
	SKOS Vocabulary	4	An RDF vocabulary for describing the basic structure and content of concept schem...		

Rows per page: 25

1-7 of 7

<

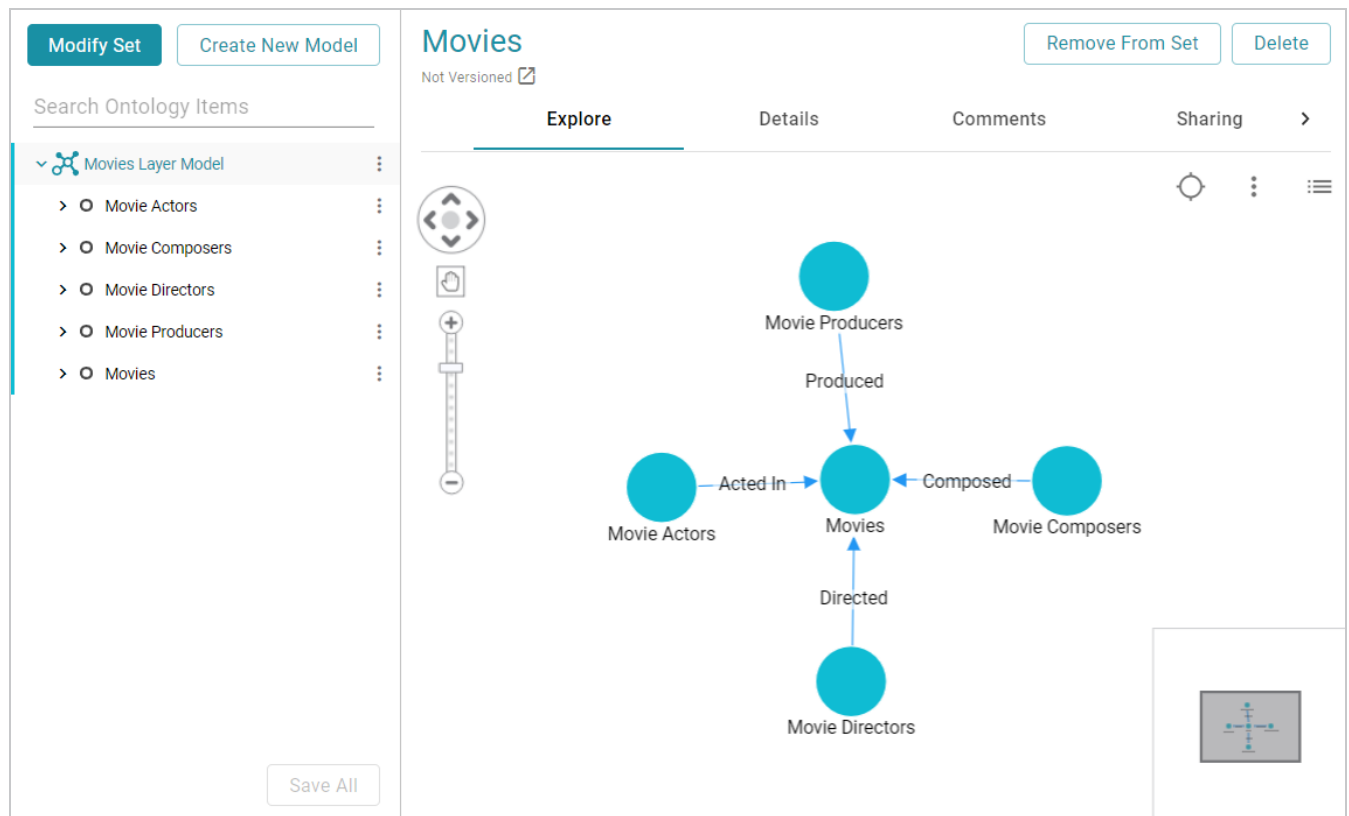
>

UPLOAD MODELS

CANCEL

OK

- On the Manage Working Set screen, select the checkbox next to the model (or models) that you want to add to the working set for editing. Then click **OK**. Anzo opens the selected model in the editor. For example:



3. Click the **Details** tab.
4. Select a class in the model to display the settings for that class. Then scroll down to the **Resource Template** field. For example, the image below shows the Resource Template field for the selected Movie Actors class.

Search Ontology Items

▼ Movies Layer Model

▼ Movie Actors

Acted In

Actor ID

Actor Name

Movie Title

Movie Composers

Movie Directors

Movie Producers

Movies

Properties

Acted In × Movie Title × Actor ID × Actor Name ×

The properties which belong to this class

Inherited Properties

type ×

The properties which belong to super classes of this class or to the model itself

Storage

Actor Name

The property to use as a preview or label for the given class

Resource Template

A template describing the URI for a newly created resource of this Class

Graph Template

A template describing the URI of the graph container for a newly created resource of this Class

Save All

- Click the **Resource Template** field and specify the URI pattern to use for instances of this class. First, type a base value in the field and press **Enter** to add the value to the field. For example, for Movie Actors in the step above: `http://cambridgesemantics.com/Movie/Actor`.

`http://cambridgesemantics.com/Movie/Actor` ✕ ▾

A template describing the URI for a newly created resource of this Class ( Sample:  
`http://cambridgesemantics.com/Movie/Actor` )

Then click the field again and select a property in the class that defines the class, i.e., contains unique values. For example, in the Movie Actors class, Actor ID provides unique values.

`http://cambridgesemantics.com/Movie/Actor` **x** `Actor ID` **x** **x** **▼**

A template describing the URI for a newly created resource of this Class ( Sample:  
`http://cambridgesemantics.com/Movie/Actor{Actor ID}` )

- Click **Save** to save the change, and then select another class for which to set a Resource Template. Repeat the step above for each class in the model.

# Blend

Once your data is onboarded to Anzo, the data sets in the Datasets catalog can be added to Graphmarts. Graphmarts are containers for the data sets that you want to blend and transform to meet the needs of the overall business. Graphmarts enable users to create links between related but previously siloed data as well as apply cleansing, transformation, and validation steps to meet analytic needs. The topics in this section provide information about working with onboarded datasets and graphmarts.

- [Working with Datasets .....517](#)
- [Working with Graphmarts .....558](#)
- [Profiling Data Sources, Datasets, and Graphmarts .....686](#)



# Working with Datasets

The topics in this section provide guidance on working with the Datasets catalog.

- [Adding an Empty Dataset for an Export Step .....518](#)
- [Importing an Existing Dataset \(FLDS\) .....522](#)
- [Creating a Dataset from RDF Files .....528](#)
- [Creating a Graphmart from a Dataset .....534](#)
- [Adding a Dataset to an Existing Graphmart .....540](#)
- [Managing Dataset Editions .....545](#)

# Adding an Empty Dataset for an Export Step

Follow the steps below to create an empty dataset that can be used to create an FLDS from the data that is output from an Export Step.

1. In the Anzo application, expand the **Blend** menu and click **Datasets**. Anzo displays the Datasets screen, which lists the catalog of datasets. For example:

Add Dataset

<input type="checkbox"/>	Title	↑	Description	Updated Date	Actions
	Books		Exported Books dataset		
	Flights		Exported Flights dataset		
	Movies		Exported Movies graphmart		
	Northwind		Exported Northwind graphmart		

Rows per page: 25 1-4 of 4 < >

2. On the Datasets screen, click **Add Dataset**. Anzo opens the Create Dataset dialog box.

### Create Dataset

☐ Empty Dataset☒ From Existing RDF☐ From Existing Dataset

Create a new dataset based on RDF data files from an external source, or create a new empty dataset.  
To create new RDF data from an external source, go to the Data Sources Tab.

Title \*

Description

RDF File Location \* [BROWSE](#)

NOTE: Only ttl is supported. Data should be in a folder named either "rdf.ttl" or "rdf.ttl.gz" depending on the file type, e.g. /path/to/files/rdf.ttl/data.ttl. To upload, click BROWSE and select the folder that contains your dataset.

Ontologies | ▾

Ontologies associated with the file based linked data set

☐ Include System Data

CANCEL [SAVE](#)

3. Select the **Empty Dataset** radio button.

### Create Dataset

☒ Empty Dataset☐ From Existing RDF☐ From Existing Dataset

Create a new dataset based on RDF data files from an external source, or create a new empty dataset.  
To create new RDF data from an external source, go to the Data Sources Tab.

Title \*

Description

RDF File Location \* [BROWSE](#)

Click BROWSE and select the folder where you want to create your dataset.

Ontologies | ▾

Ontologies associated with the file based linked data set

Rdf Format

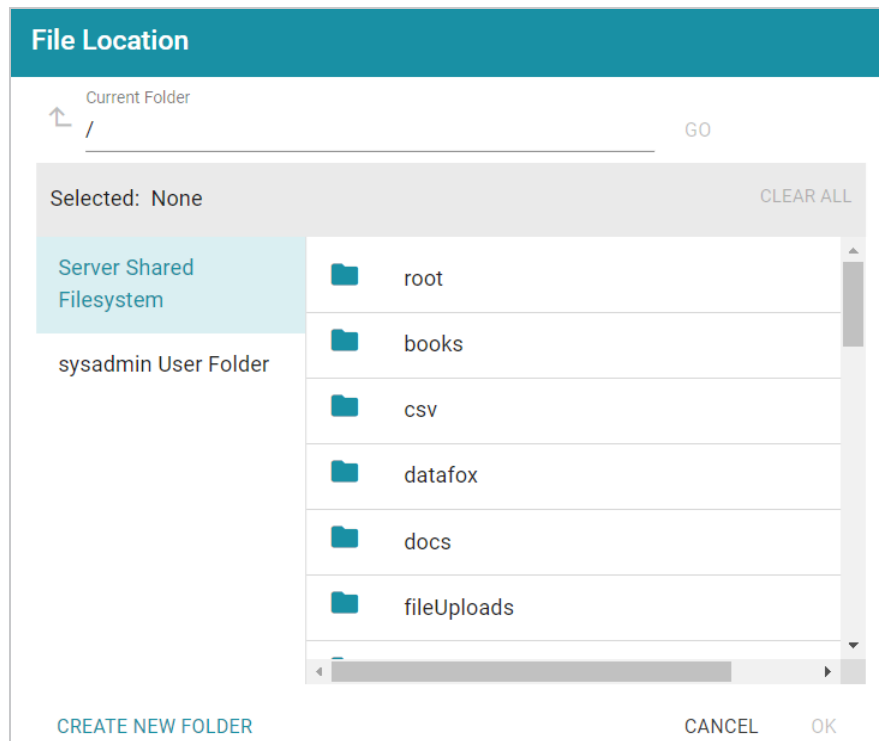
ttl.gz

Set the dataset's default data format, such as ttl or ttl.gz.

CANCEL [SAVE](#)

4. Type a name for the new dataset in the **Title** field and an optional description in the **Description** field.

5. Click the **RDF File Location** field to open the File Location dialog box.



6. Find and select the directory where the FLDS for this dataset should be created. If needed, you can click **Create New Folder** to create a new directory. Then click **OK** to close the dialog box and return to the Create Dataset screen.
7. Next, if you know which model or models are associated with the data that you plan to export, select the models from the **Ontologies** drop-down list. Leave **Ontologies** blank if you do not know which models apply or do not want to select one at this time. When the Export steps runs, Anzo automatically exports any models that are related to the exported data.
8. Lastly, the RDF Format setting defaults to **ttl.gz**, meaning files that are output to this dataset will be in compressed Turtle format. This is the ideal format for preserving space on the file store. If you do not want the files to be compressed, you can change the value to **ttl**.
9. Click **Save** to save the empty dataset and return to the Datasets screen. The new dataset becomes available as a selection when choosing the Target FLDS for an Export Step.

## Related Topics

[Exporting Data to an FLDS \(Export Step\)](#)

## Importing an Existing Dataset (FLDS)

Follow the instructions below to add an existing FLDS to the Datasets catalog. Make sure that the FLDS meets the following [File Requirements](#).

### File Requirements

To add an FLDS to the Datasets catalog, the location of the files, the file format, and the directory structure must meet the following requirements:

- **Supported File Locations:** Files can be staged on a configured file store or they can be uploaded from your computer as a .zip file.
- **Supported Directory Structure:** FLDS directories should contain an **flds.trig** file, an **onts** directory that includes the model .trig file, and an **rdf.ttl** or **rdf.ttl.gz** directory that contains the data files. For example:

#### Note

Models must be in TriG format, regardless of the file type of the data files.

```
LoadEmployees_f7b1f
├─ flds.trig
├─ onts
│   └─ Employees.trig
└─ rdf.ttl.gz
    └─ Loadnew_employees_8be23.ttl.gz
        └─ 20191021034225.ttl.gz
            └─ part-00000.ttl.gz
            └─ part-00001.ttl.gz
            └─ part-00003.ttl.gz
```

### Importing an FLDS

Follow the steps below to import an FLDS from the file store or from a .zip file on your computer.

1. In the Anzo application, expand the **Blend** menu and click **Datasets**. Anzo displays the Datasets screen, which lists the catalog of datasets. For example:

Add Dataset

<input type="checkbox"/>	Title	↑	Description	Updated Date	Actions
	Books		Exported Books dataset		
	Flights		Exported Flights dataset		
	Movies		Exported Movies graphmart		
	Northwind		Exported Northwind graphmart		

Rows per page: 25
1-4 of 4

2. On the Datasets screen, click **Add Dataset**. Anzo opens the Create Dataset dialog box.

### Create Dataset

☐ Empty Dataset☒ From Existing RDF☐ From Existing Dataset


Create a new dataset based on RDF data files from an external source, or create a new empty dataset.  
To create new RDF data from an external source, go to the Data Sources Tab.

Title \*

Description

RDF File Location \* [BROWSE](#)

NOTE: Only ttl is supported. Data should be in a folder named either "rdf.ttl" or "rdf.ttl.gz" depending on the file type, e.g. /path/to/files/rdf.ttl/data.ttl. To upload, click BROWSE and select the folder that contains your dataset.

Ontologies 

Ontologies associated with the file based linked data set

☐ Include System Data

CANCEL [SAVE](#)

3. Select the **From Existing Dataset** radio button.

### Create Dataset

☐ Empty Dataset☐ From Existing RDF☒ From Existing Dataset

Create a new dataset based on RDF data files from an external source, or create a new empty dataset.  
To create new RDF data from an external source, go to the Data Sources Tab.

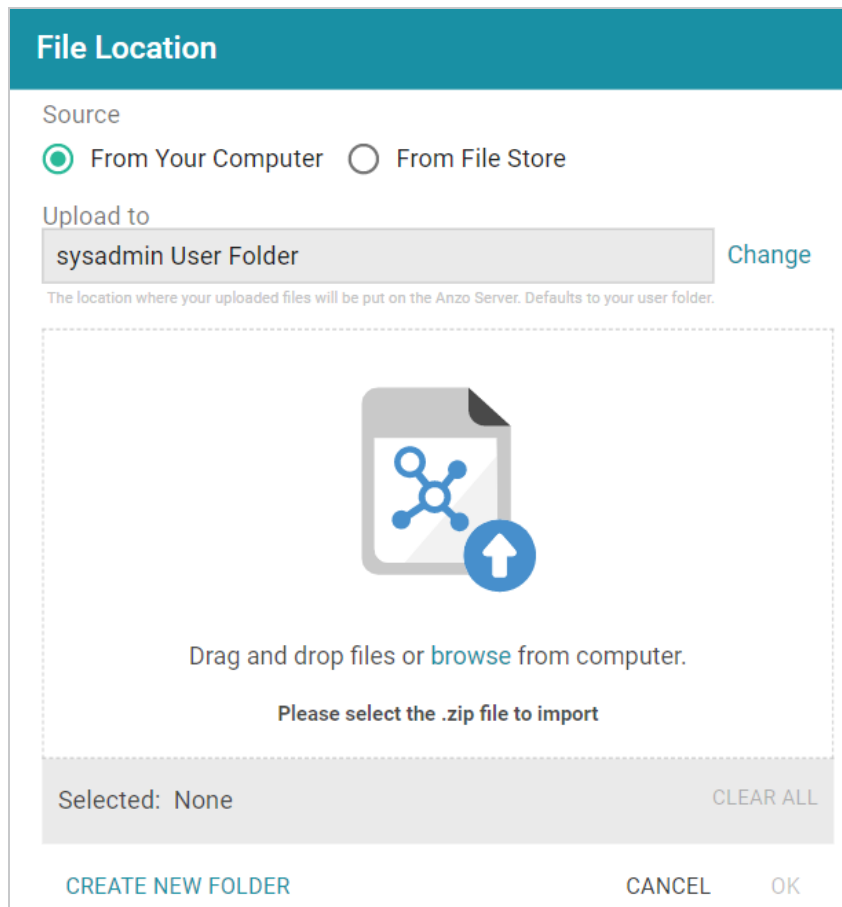
RDF File Location \* [BROWSE](#)

File path should be the root of the existing FLDS

CANCEL [SAVE](#)



4. Click the **RDF File Location** field to open the File Location dialog box.



The dialog box is titled "File Location" in a teal header. It contains two radio buttons under the "Source" label: "From Your Computer" (selected) and "From File Store". Below this is an "Upload to" section with a text field containing "sysadmin User Folder" and a "Change" link. A tooltip below the text field reads: "The location where your uploaded files will be put on the Anzo Server. Defaults to your user folder." The main area of the dialog is a large dashed box containing a file icon with a blue upload arrow and the text: "Drag and drop files or [browse](#) from computer. Please select the .zip file to import". At the bottom, there is a "Selected: None" label, a "CLEAR ALL" link, and three buttons: "CREATE NEW FOLDER", "CANCEL", and "OK".

5. Follow the appropriate steps below depending on whether the FLDS is on your computer or the shared File Store:

**If the file is on your computer:**

- a. As a best practice, check the upload location that is listed in the **Upload To** field by hovering your pointer over the value to view the tooltip. Make sure the upload location is a directory on the shared file store and not in the server installation path. If the file is not uploaded to the shared file store it is not accessible by applications like AnzoGraph. In addition, other users cannot create graphmarts from the data source because they typically do not have access to the file location. For example, viewing the Upload To location for the screen above shows that the file will be uploaded to the server installation path, `/opt/Anzo/Server/data...`

Source

☒ From Your Computer ☐ From File Store

Upload to

sysadmin User Folder Change

/opt/Anzo/Server/data/userUploads/sysadmin\_20235 ... on the Anzo Server. Defaults to your user folder.

If your Upload To location is configured to upload the file to the server installation path, click **Change** and select an upload location that is on the shared file store. For example, the image below shows the Upload Folder Location dialog box that is presented after clicking **Change**. A folder called **fileUploads** is selected on the shared store.

Upload Folder Location

Current Folder

/ GO

Selected: /fileUploads/ CLEAR ALL

Server Shared Filesystem

sysadmin User Folder

docs

fileUploads

flights

fmcsa

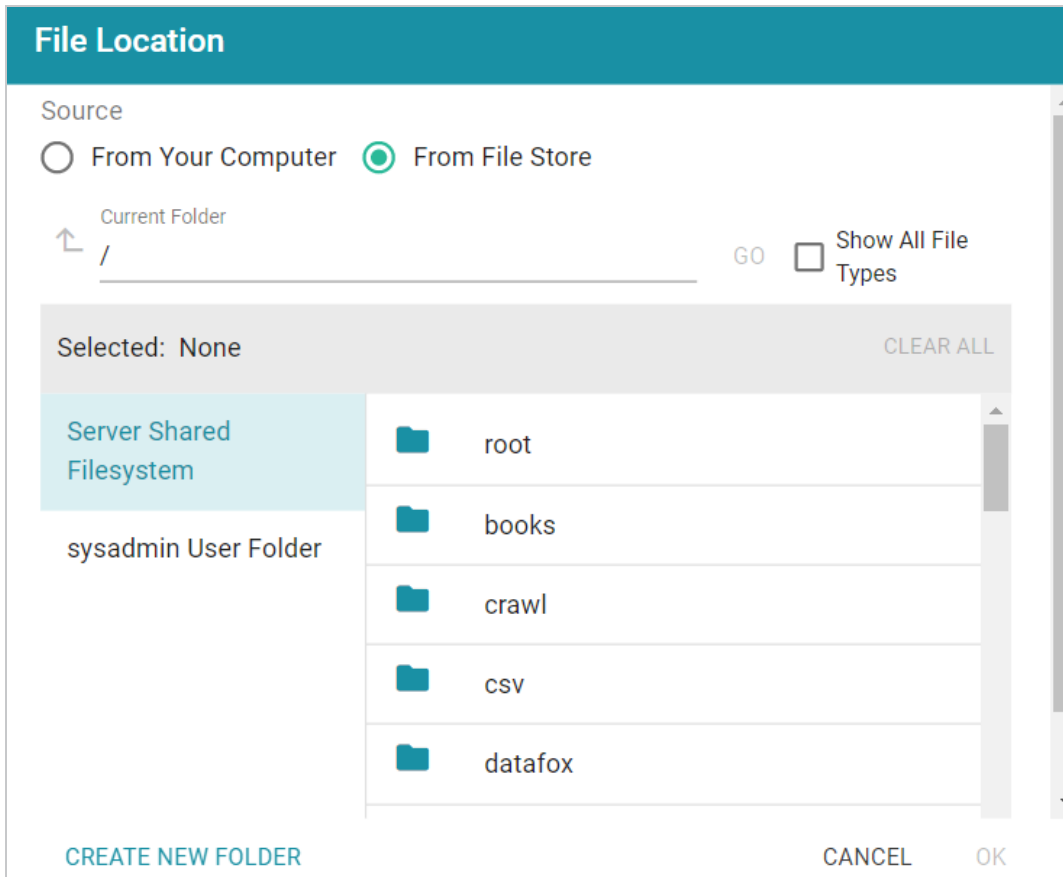
nh1h10

CREATE NEW FOLDER CANCEL OK

- Drag and drop the file onto the screen or click **browse** to navigate to the file and select it. Anzo attaches the file and the **OK** button becomes active.
- Click **OK**. The file is added to the RDF File Location field on the Create Dataset screen.

**If the files are on the File Store:**

- a. Click the **From File Store** radio button. Anzo displays the file selection dialog box. For example:



- b. On the left side of the screen, select the file store that hosts the FLDS. On the right side of the screen, navigate to the root directory for the dataset. This is the directory that contains the **flds.trig** file, the **onts** directory, and the **rdf.ttl** or **rdf.ttl.gz** directory.
- c. Select the root directory for the FLDS and then **click OK**. The location is added to the RDF File Location field on the Create Dataset screen.
6. Click **Save** to import the FLDS and return to the Datasets screen.

You can now select the dataset in the catalog and create a new graphmart or add the dataset to an existing graphmart. See [Creating a Graphmart from a Dataset](#) or [Adding a Dataset to an Existing Graphmart](#) for instructions.

## Creating a Dataset from RDF Files

Source data that is not in RDF format is onboarded through the automated direct data load workflow or unstructured pipelines, where the data is converted to RDF format. If you have data that is already in RDF format in Turtle or N-Triple files, those files can be added to the Datasets catalog directly, making the data available to add to a graphmart for loading and analyzing in AnzoGraph.

### Note

To import data from CSV, JSON, XML, Parquet, or SAS files, follow the processes described in [Adding Data Sources](#).

Follow the instructions below to create a dataset from a directory of Turtle or N-Triple files. Make sure that the files and directory meet the requirements in [File Requirements](#).

### File Requirements

To add data to the Dataset catalog, the location of the files, the file format, and the directory structure must meet the following requirements:

- **Supported File Locations:** Files can be staged on a configured file store, or they can be uploaded from your computer as a .zip file.
- **Supported File Formats:** Files must be in one of the following formats:
  - Turtle (.ttl file type)
  - N-Triple (.n3 and .nt file types)

Either of the file types listed above can be compressed in GZIP format and named as *filename.filetype.gz* files.

- **Supported Directory Structure:** When importing RDF files that are not part of an FLDS, the files must be placed in a directory named **rdf.<filetype>** or **rdf.<filetype>.gz**. Stage uncompressed TTL files in a directory called **rdf.ttl**, and stage compressed TTL files in a directory called **rdf.ttl.gz**. Stage uncompressed N-Triple files in a directory called **rdf.nt** or **rdf.n3**, depending on the file type extension. Place compressed files in an **rdf.nt.gz** or **rdf.n3.gz** directory. For example:

```
External-RDF-Top-Level-Directory
├── rdf.ttl.gz
│   └── external-rdf-file1.ttl.gz
```

```
└─ external-rdf-file2.ttl.gz
└─ external-rdf-file3.ttl.gz
```

### Important

All files inside an `rdf.<filetype>` or `rdf.<filetype>.gz` directory must be the same format and end in the same extension. Data in mixed formats will not load successfully. If you plan to import multiple file types, organize files into separate directories by file extension type, and then import each directory separately.

### Note

To upload files from your computer, use the same directory structure as shown above. Zip the top-level directory so that the upload file is `External-RDF-Top-Level-Directory.zip` and contains the `rdf.ttl.gz` directory.

## Importing RDF Files

Follow the steps below to create a dataset from RDF files.

### Tip

Anzo provides the option to link the files to an existing data model during the import. If the model is not yet available in Anzo, consider uploading it before importing the RDF files. See [Uploading a Model to Anzo](#) for instructions. You are not required to include a model at import time; a model can be associated with a data set at any time. [How do I associate a model with an existing dataset?](#)

1. In the Anzo application, expand the **Blend** menu and click **Datasets**. Anzo displays the Datasets screen, which lists the catalog of datasets. For example:

Search

Add Dataset

<div></div>	Title		Description	Updated Date	Actions	
	<div>Books</div>		Exported Books dataset		<div></div>	<div></div>
	<div>Flights</div>		Exported Flights dataset		<div></div>	<div></div>
	<div>Movies</div>		Exported Movies graphmart		<div></div>	<div></div>
	<div>Northwind</div>		Exported Northwind graphmart		<div></div>	<div></div>

Rows per page: 25

1-4 of 4

<

>

2. On the Datasets screen, click **Add Dataset**. Anzo opens the Create Dataset dialog box.

### Create Dataset

☐ Empty Dataset☒ From Existing RDF☐ From Existing Dataset

Create a new dataset based on RDF data files from an external source, or create a new empty dataset.  
To create new RDF data from an external source, go to the Data Sources Tab.

Title \*

Description

RDF File Location \* [BROWSE](#)

NOTE: Only ttl is supported. Data should be in a folder named either "rdf.ttl" or "rdf.ttl.gz" depending on the file type, e.g. /path/to/files/rdf.ttl/data.ttl. To upload, click BROWSE and select the folder that contains your dataset.

Ontologies

Ontologies associated with the file based linked data set

☐ Include System Data

[CANCEL](#) [SAVE](#)

3. The **From Existing RDF** radio button is selected by default. Type a name for the new dataset in the **Title** field and an optional description in the **Description** field.
4. Click the **RDF File Location** field to open the File Location dialog box and follow the appropriate steps below depending on the location of the files.

#### If you are uploading a .zip file from your computer:

- a. As a best practice, check the upload location that is listed in the **Upload To** field by hovering your pointer over the value to view the tooltip. Make sure the upload location is a directory on the shared file store and not in the server installation path. If the file is not uploaded to the shared file store it is not accessible by applications like AnzoGraph. In addition, other users cannot create graphmarts from the data source because they typically do not have access to the file location.  
  
For example, viewing the Upload To location for the screen above shows that the file will be uploaded to the server installation path, `/opt/Anzo/Server/data...`

Source

☒ From Your Computer ☐ From File Store

Upload to

sysadmin User Folder Change

/opt/Anzo/Server/data/userUploads/sysadmin\_20235

... on the Anzo Server. Defaults to your user folder.

If your Upload To location is configured to upload the file to the server installation path, click **Change** and select an upload location that is on the shared file store. For example, the image below shows the Upload Folder Location dialog box that is presented after clicking **Change**. A folder called **fileUploads** is selected on the shared store.

Upload Folder Location

Current Folder

/ GO

Selected: /fileUploads/ CLEAR ALL

Server Shared Filesystem

sysadmin User Folder

docs

fileUploads

flights

fmcsa

nhih10

CREATE NEW FOLDER CANCEL OK

- Drag and drop the .zip file with the RDF files onto the screen or click **Browse** to navigate to the file on your computer and select it.
- Click **OK** to close the dialog box and return to the Create Dataset screen.



## If the files are on the File Store

- a. Select the **From File Store** radio button.
  - b. Find and select the **rdf.<filetype>** directory that you want to import, and then click **OK** to close the dialog box and return to the Create Dataset screen.
5. If you want to associate a model with this dataset, click the **Ontologies** drop-down list and select the model. To include a system model, select the **Include System Data** checkbox. If you do not want to associate a model with the data at this time, leave the Ontologies field blank.

### Note

Datasets without a model cannot be viewed in Hi-Res Analytics dashboards, but the imported data can still be queried. A model can be associated with the data set at a later time. [How do I associate a model with an existing dataset?](#)

6. Click **Save**. Anzo creates the FLDS and adds the new dataset to the Datasets catalog, and return to the Datasets screen.

### Note

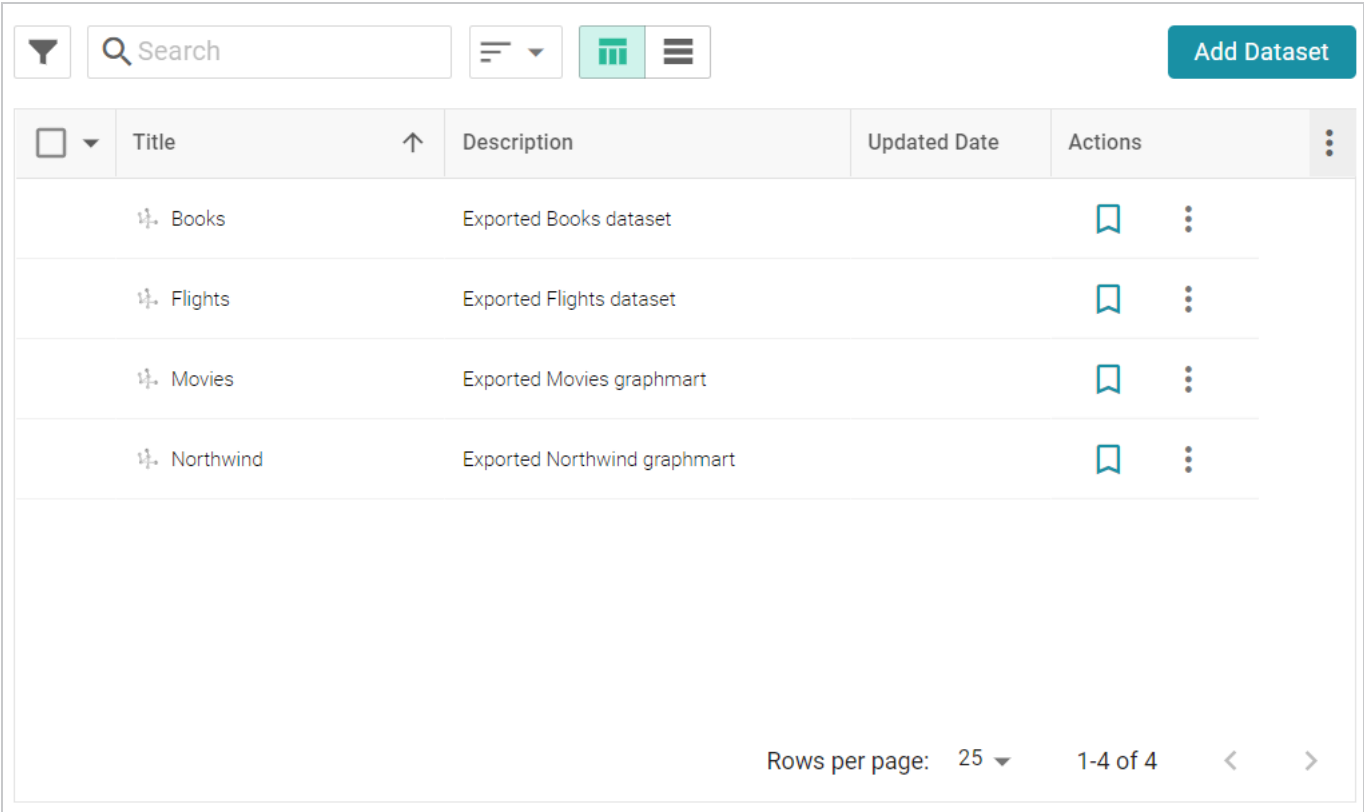
Anzo generates an flds.trig file at the same level as the rdf.<filetype> directory. The file contains metadata about the load files.

You can now select the dataset in the catalog and create a new graphmart or add the dataset to an existing graphmart. See [Creating a Graphmart from a Dataset](#) or [Adding a Dataset to an Existing Graphmart](#) for instructions.

# Creating a Graphmart from a Dataset

Follow the steps below to create a new graphmart from a dataset in the Datasets catalog.

1. In the Anzo application, expand the **Blend** menu and click **Datasets**. Anzo displays the Datasets screen, which lists the catalog of datasets. For example:



2. In the catalog, click the checkbox next to each dataset that you want to add to the new graphmart. Hover the pointer over an item to display the checkbox in the left column. Anzo adds the datasets to the shopping cart and additional icons become available at the bottom of the screen. For example:

Search

Add Dataset

	Title	Description	Updated Date	Actions
<input type="checkbox"/>	Books	Exported Books dataset		
<input type="checkbox"/>	Flights	Exported Flights dataset		
<input checked="" type="checkbox"/>	Movies	Exported Movies graphmart	Nov 2, 2022	
<input type="checkbox"/>	Northwind	Exported Northwind graphmart		
<input type="checkbox"/>	Tickets	Exported Tickets graphmart		

1 Items Selected

Bookmark

Delete

Create Graphmart

Add To Package

Rows per page: 25

1-5 of 5

<

>

- Click the **Create Graphmart** button. Anzo displays the Create Graphmart screen and populates the Title field by appending "Graphmart" to the name of the dataset.

Create Graphmart

Title \*

Movies Graphmart

The title of the graphmart

Description

A brief description of the graphmart

Datasets

Title	Edition	Description	Last Published Date	Actions
Movies	Movies - Edition Cumulative Export ...			

Rows per page: 25

1-1 of 1

<

>

CANCEL

CREATE & ACTIVATE

CREATE

4. On the Create Graphmart screen, you can edit the **Title** and add an optional **Description**.
5. By default the current working edition (Managed Edition) of the dataset is selected. If you want to select a different edition, follow these steps:
  - a. click the Modify Edition (✎) icon in the Actions column. The Modify Edition dialog box is displayed. For example:

Modify Edition

Title

Description

Movies - Edition Cumulativ...

✕

▼

Data Component for selected job included in this Edition

<input checked="" type="checkbox"/>	Title	Created Date	Actions
<input checked="" type="checkbox"/>	Created : 11/02/2022 03:07PM	11/02/2022 03:07PM	⋮
<input checked="" type="checkbox"/>	Created : 11/02/2022 02:13PM	11/02/2022 02:13PM	⋮
<input checked="" type="checkbox"/>	Created : 11/02/2022 12:28PM	11/02/2022 12:28PM	⋮
<input checked="" type="checkbox"/>	Created : 11/02/2022 10:27AM	11/02/2022 10:27AM	⋮

Rows per page: 25

1-4 of 4

<

>

4 Items Selected

SAVE AS NEW EDITION

CANCEL

SAVE CUSTOMIZATIONS


- b. To choose a different edition, click the drop-down list at the top of the screen and select the edition to use.
- c. If you want to make changes to the selected edition, select or clear the Data Component checkboxes on the left side of the screen to include or exclude components.

### Tip

You can also rename a component by clicking the menu icon in the Actions column and selecting **Rename Component**. In addition, you can remove a component from a dataset by clicking the Actions menu and selecting **Remove Component from Dataset**. However, note that choosing this option deletes the component from the edition and **deletes that component's data from the file system**.

- d. When you are finished making changes, choose one of the following options for saving the changes:
  - To save the changes as a new edition, click **Save As New Edition** on the left side of the screen. The Create New Edition dialog box is displayed. Specify a Title and optional

Description and click **Save**.

A dialog box titled "Create New Edition" with a teal header. It contains two text input fields: "Title \*" and "Description". At the bottom right, there are two buttons: "CANCEL" and "SAVE".

Create New Edition

Title \*


Description

CANCEL SAVE


- To save the changes as a copy of the existing edition, click **Save Customizations**. Anzo clones the edition and adds the copy to the list on the Create Graphmart screen.
6. To create the graphmart without activating it, click **Create**. If you want to create the graphmart and activate it, click **Create & Activate**. Anzo creates the graphmart and displays the Overview screen. For example:



#### Note


If you activate the graphmart and have more than one static AnzoGraph engine configured or you have a Cloud Location configured for dynamic AnzoGraph deployments, Anzo displays a **Select an AnzoGraph Query Engine** dialog box. Click the drop-down list to select the engine to load the graphmart to, or select **Spin up new AnzoGraph** (if available) to deploy a new instance. Then click **OK**.




# Movies Graphmart

Not Versioned 

INACTIVE  ACTIVE  Ready to use


 AZG Static


Save a Copy
Create Dashboard



Overview
Explore
Datasets
Data Layers
Dashboards
Dat >


Description  
None


Load Priority  
100

Metrics  
Profile data upon activation 

Disable Load Counts  
Disable populating counts during load operations 


Data Loading Settings  
Load layers that do not fail 

Ignore Source Errors  
Ignore any step or view's source layer if they failed to load 

Leave Graphmart Online During Refresh  
When refreshing layers, leave graphmart and layer online. 

General

Type : Graphmart  
Creator : System Administrat...  
Last Accessed : 3 minutes ago  
Last Updated : 3 minutes ago  
Structure Modified : 3 minutes ago  
Released : 3 minutes ago

<http://cambridgesemantics.com/Graph> 


Dynamic Resource Log

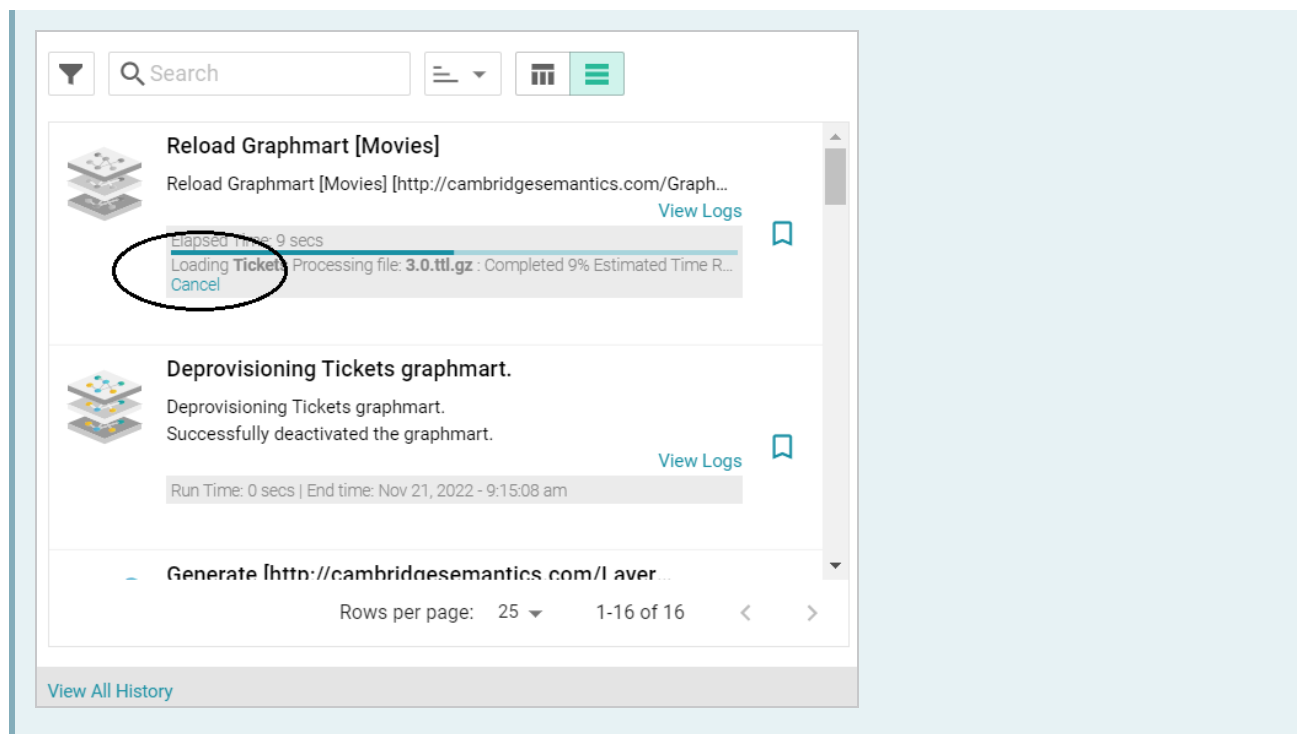
AnzoGraph Server Details

AnzoGraph : AZG  
Status : Ready to use  
Last Accessed : 20 hours ago  
Memory Used : 266.47 MB( 3% )  
Memory Total : 11.7 GB

Inactivity Deactivate Timeout

## Tip

If you want to cancel graphmart activation while data is loading, open the Activity Log by clicking the Activity Log icon () in the main menu bar. Then click **Cancel** for the **Provisioning...graphmart** activity. For example:

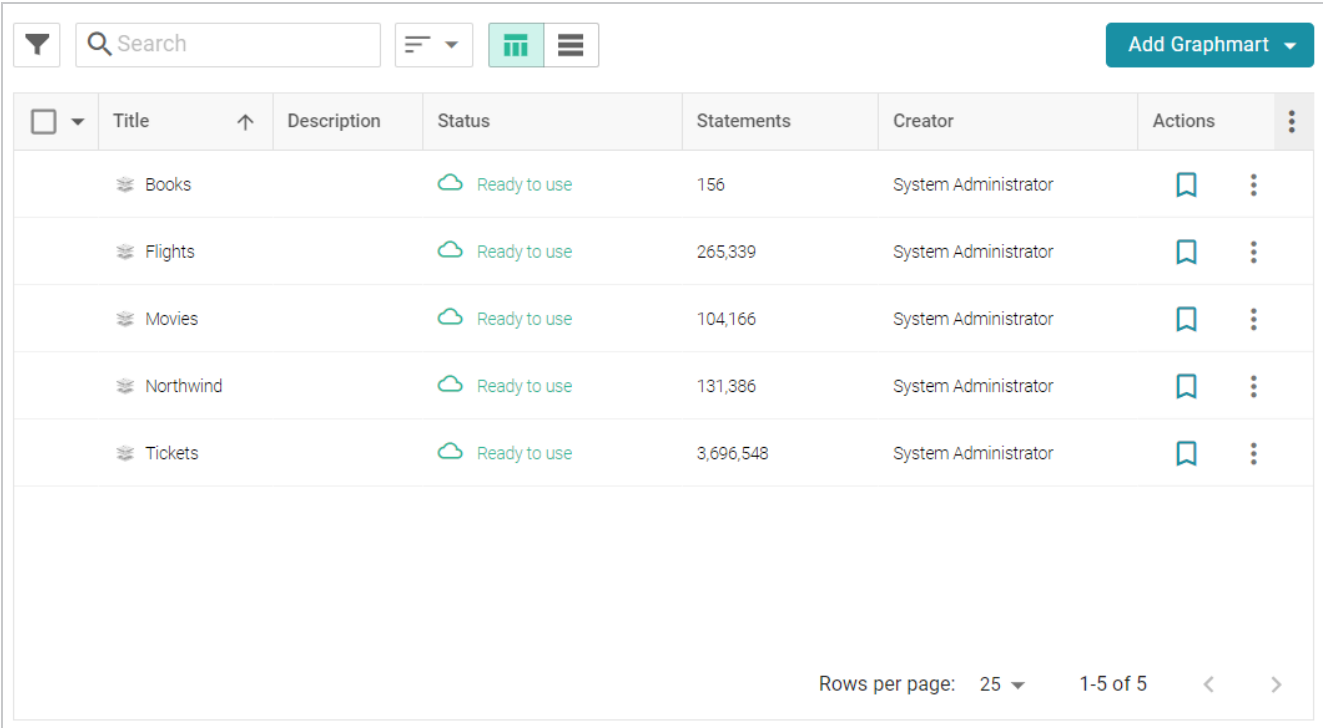


When graphmart creation is complete and the graphmart is activated, the data is available to access and analyze. For more information, see [Access](#). For more information about graphmarts, see [Working with Graphmarts](#).

# Adding a Dataset to an Existing Graphmart

This topic provides instructions for adding a dataset from the Datasets catalog to an existing graphmart.

1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing graphmarts. For example:



The screenshot shows the Anzo Graphmarts interface. At the top, there is a search bar with a magnifying glass icon and the text 'Search'. To the right of the search bar are two icons: a green bar chart and a black bar chart. Further right is a blue button labeled 'Add Graphmart' with a dropdown arrow. Below these elements is a table with the following columns: Title, Description, Status, Statements, Creator, and Actions. The table contains five rows of data, each representing a graphmart. The 'Status' column for all rows is 'Ready to use'. The 'Actions' column contains a bookmark icon and a three-dot menu icon for each row. At the bottom right of the table, there is a pagination control showing 'Rows per page: 25' and '1-5 of 5' with navigation arrows.

<input type="checkbox"/>	Title	Description	Status	Statements	Creator	Actions
	Books		Ready to use	156	System Administrator	
	Flights		Ready to use	265,339	System Administrator	
	Movies		Ready to use	104,166	System Administrator	
	Northwind		Ready to use	131,386	System Administrator	
	Tickets		Ready to use	3,696,548	System Administrator	

Rows per page: 25 1-5 of 5

2. On the Graphmarts screen, click the name of the graphmart that you want to add data to. Anzo displays the Overview.
3. Click the **Datasets** tab. The screen lists the datasets in the graphmart. For example, the image below shows a graphmart without any datasets because this is a graphmart that was auto-generated from a data source:



**Movies**

Derived from: Movies\_20221118182145
DEV 1

Save a Copy
Create Dashboard

INACTIVE
ACTIVE
Ready to use
AZG
Static

Overview
Explore
Datasets
Data Layers
Dashboards
Data on Demand

Member Datasets

Search

Add Dataset

No editions found

- Click the **Add Dataset** button. The Select Dataset dialog box is displayed.

Select Dataset

Search

<input type="checkbox"/>	Title	Description	Updated Date	Tags	Actions
<input type="checkbox"/>	Northwind		Nov 18, 2022		
<input type="checkbox"/>	Tickets		Nov 18, 2022		

Rows per page: 25
1-2 of 2

CANCEL
ADD

- In the dialog box, select the checkbox next to the dataset that you want to add, then click **Add**. Anzo adds the dataset to the graphmart. (A new data layer is generated with a Load Dataset Step that will load the dataset.)

This graphmart is modified and needs refresh.
 REFRESH

**Movies**
Derived from: Movies\_20221118182145 DEV 1
INACTIVE ☐ ACTIVE ☒ Ready to use
 AZG Static
Save a Copy
Create Dashboard

Overview
Explore
**Datasets**
Data Layers
Dashboards
Data on Demand >

**Member Datasets**Add Dataset

Title	↑	Edition	Description	Last Published Date	Actions
Tickets		Tickets - Edition Cumulative Export Starting Wed ...			

6. By default the current working edition (Managed Edition) of the dataset is selected. If you want to select a different edition, follow these steps:
  - a. Click the Edit icon () in the Actions column. The Modify Edition dialog box is displayed. For example:

**Modify Edition**

Title
 

Tickets - Edition Cumulative...

Description

**Data Component for selected job included in this Edition**

<input checked="" type="checkbox"/>	Title	Created Date	Actions
<input checked="" type="checkbox"/>	Created : 11/21/2022 09:13AM	11/21/2022 09:13AM	
<input checked="" type="checkbox"/>	Created : 11/18/2022 04:46PM	11/18/2022 04:46PM	
<input checked="" type="checkbox"/>	Created : 11/18/2022 04:26PM	11/18/2022 04:26PM	
<input checked="" type="checkbox"/>	Created : 11/18/2022 02:25PM	11/18/2022 02:25PM	
<input checked="" type="checkbox"/>	Created : 11/18/2022 09:18AM	11/18/2022 09:18AM	

Rows per page: 25
 1-8 of 8

**8 Items Selected**

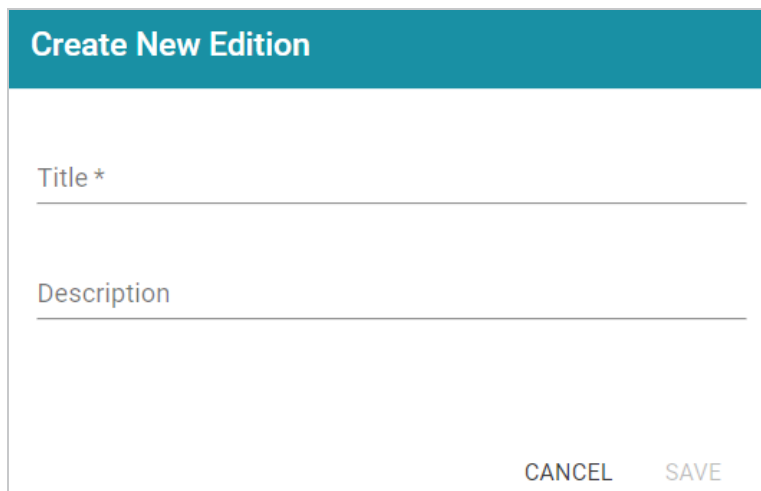
SAVE AS NEW EDITION
CANCEL
SAVE CUSTOMIZATIONS

- b. To choose a different edition, click the drop-down list at the top of the screen and select the edition to use.
- c. If you want to make changes to the selected edition, select (to include) or clear (to exclude) the data component checkboxes on the left side of the screen.

#### Note

When you make changes to an edition while creating or changing a graphmart, Anzo creates a copy of the edition (with the changes) and uses the copy as a dataset in the graphmart. The original published edition remains unchanged.

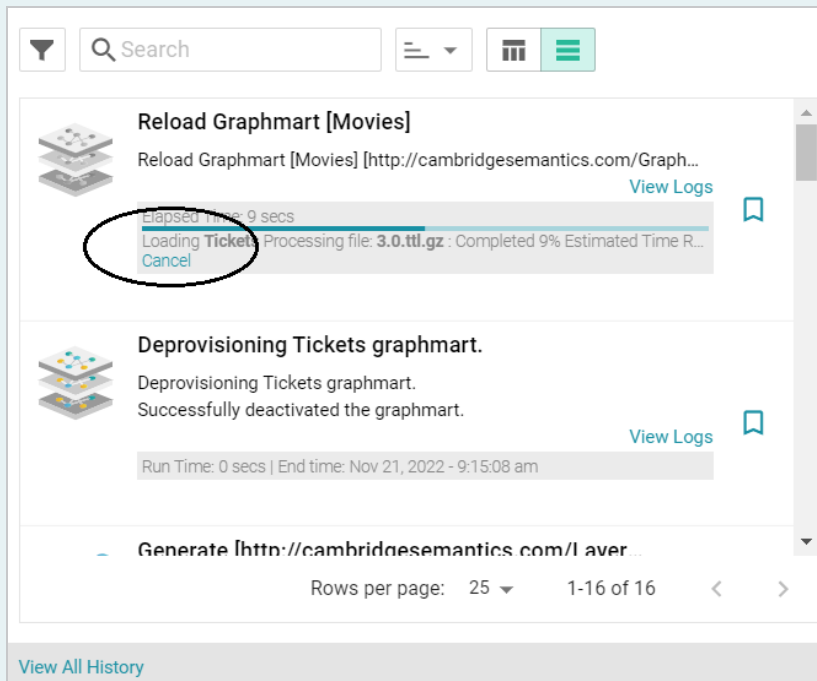
- d. When you are finished making changes, choose one of the following options for saving the changes:
  - If you want to save the changes as a new Saved Edition, click **Save As New Edition**. Anzo displays the Create New Edition dialog box. Specify a Title and optional Description for the edition, and click **Save**. Then click **Save Customizations** on the Modify Edition screen.

A screenshot of the 'Create New Edition' dialog box. It has a teal header with the title 'Create New Edition'. Below the header, there are two text input fields: 'Title \*' and 'Description'. At the bottom right of the dialog, there are two buttons: 'CANCEL' and 'SAVE'.

- If you want to save the changes as a copy of the existing edition, click **Save Customizations**. Anzo clones the edition and adds the copy to the list on the screen.
7. To reload the graphmart and add the new dataset to AnzoGraph, click the **Data Layers** tab, and then click the **Reload** button (↻).

## Tip

If you want to cancel graphmart activation while data is loading, open the Activity Log by clicking the Activity Log icon (🕒) in the main menu bar. Then click **Cancel** for the **Provisioning...graphmart** activity. For example:



Once the graphmart is loaded into AnzoGraph, the data is available to access and analyze. For more information, see [Access](#).

## Managing Dataset Editions

The topics in this section introduce the concepts to know when working with Editions and provide instructions for creating, deleting, and modifying Editions.

[Introduction to Editions](#)

[Creating an Edition](#)

[Modifying an Edition](#)

[Deleting a Saved Edition](#)

### Introduction to Editions

Editions are collections of the data components that are published by a given pipeline. Editions can be assembled by users and can include any subset of the jobs from a pipeline and any version of a job's output. This topic introduces you to the concepts that are helpful to know when working with editions.

- [What is a Data Component?](#)
- [What is the Managed Edition?](#)
- [What is a Saved Edition?](#)

### What is a Data Component?

A **Data Component** is the data that is generated by one successful run of a job in a pipeline. Each time a job runs to completion, a new data component is created that contains the version of the data that was generated by that run. If a job is run 5 times, there are 5 data components. Anzo preserves each version of the data that is output by each job.

For example, the image below shows a list of the jobs in an edition. The right side of the screen shows that the selected job has been successfully published four times:

Title *		Description	
Job Data included in this Edition		Data Component for selected job included in this Edition	
Name ↓	Published Date	Job Type	
Load tickit_venues	12/12/2022 04:07...	Standard	
Load tickit_categories	12/12/2022 04:01...	Standard	
Load tickit_sales	12/12/2022 04:00...	Standard	
Load tickit_listings	12/12/2022 04:00...	Standard	
Load tickit_users	12/12/2022 03:59...	Standard	
Load tickit_events	12/12/2022 03:59...	Standard	
Rows per page: 25 ▼		1-7 of 7	< >
Name ↓	Published Date		
Loadtickit_venues_tickit_venues	12/12/2022 04:07PM		
Loadtickit_venues_tickit_venues	12/12/2022 04:01PM		
Loadtickit_venues_tickit_venues	12/12/2022 03:53PM		
Loadtickit_venues_tickit_venues	12/09/2022 01:06PM		
Rows per page: 25 ▼		1-4 of 4	< >

The data component from the most recent run of the selected job is automatically included in the **Managed Edition** (see [What is the Managed Edition?](#)), and any of the four data components could be added to a **Saved Edition** (see [What is a Saved Edition?](#)).

## What is the Managed Edition?

When a pipeline is published, the result of the most recent run becomes the **Managed Edition**. This edition is managed by Anzo and **always contains the most recent successfully published data components for all of the jobs in the pipeline**. If one or more of the jobs fail, those jobs are excluded from the Managed Edition. If the failed jobs are published later or additional jobs are created and published, the data that results from those jobs gets added to the Managed Edition.

For example, the image below shows the Managed Edition for a dataset. Editions are viewed from a dataset's **Overview** tab. The same view is available on the Overview tab for the pipeline.

Overview

Explore

Graphmarts

Pipelines

Dashboards

Versions

Description

None

Advanced

Pipeline

Load Tickets to store

http://cambridgesemantics.com/Project/c1c694bd-849d-91d0-eb34-bfef8c2d8894/c1c694bd-849d-91d0-eb34-bfef8c2d8894

Managed Editions

Title	↑	Description	Most Recent Published Date	Actions
Default Edition		Contains the latest successfully p...	12/12/2022 04:01PM	

Search

Create New Edition

No editions found

General

Type : Catalog

Creator : System Administrat...

Structure Modified : a minute ago

Released : Dec 9, 2022 1:06 PM

Statistics

Size(Bytes) : 29,119,170

http://csi.com/FileBasedLinkedDataSet

Dynamic Resource Log

Tags

None

## Browse Edition

Title

Default Edition

Description

Contains the latest successfully published data for all jobs

Edit a Copy

Job Data included in this Edition

Name	Published Date	Job Type
Load tickit_venues	12/12/2022 04:07P...	Standard
Load tickit_categories	12/12/2022 04:01P...	Standard
Load tickit_sales	12/12/2022 04:00P...	Standard
Load tickit_listings	12/12/2022 04:00P...	Standard
Load tickit_users	12/12/2022 03:59P...	Standard
Load tickit_events	12/12/2022 03:59P...	Standard

Rows per page: 25
1-7 of 7

Data Component for selected job included in this Edition

Name	Published Date	Actions
Loadtickit_venues_tickit_venues	12/12/2022 04:07PM	

Rows per page: 25
1-1 of 1

OK

The Managed Edition cannot be changed, but it can be cloned (via the **Actions** menu) and saved as a **Saved Edition**. Saved Editions can be modified. See [What is a Saved Edition?](#) below.

### What is a Saved Edition?

A **Saved Edition** is a user-assembled collection of data components from a pipeline. A Saved Edition can contain any combination of jobs and any version of a job's data components. Saved Editions can be created from scratch or can be cloned from the Managed Edition or another Saved Edition.

The Managed Edition or any Saved Edition can be added to a graphmart for analysis.

### Related Topics

[Creating an Edition](#)

[Modifying an Edition](#)

### Creating an Edition

Follow the instructions below to create a new edition.

1. In the Anzo application, expand the **Blend** menu and click **Datasets**. Anzo displays the Datasets screen, which lists the catalog of datasets. For example:



Search

Add Dataset

<div></div>	Title		Description	Updated Date	Actions	
	<div></div> Books		Exported Books dataset		<div></div>	<div></div>
	<div></div> Flights		Exported Flights dataset		<div></div>	<div></div>
	<div></div> Movies		Exported Movies graphmart		<div></div>	<div></div>
	<div></div> Northwind		Exported Northwind graphmart		<div></div>	<div></div>


Rows per page: 25

1-4 of 4


<

>

- Click the dataset for which you want to create an edition. Anzo displays the Explore tab for the Dataset. Click the **Overview** tab, which lists the existing editions. For example:




Tickets to store

Not Versioned 

+ Add to Cart

Create Graphmart



Overview

Explore

Graphmarts

Pipelines

Dashboards

Versions >

Description


None

Advanced ▾


Pipeline

Load Tickets to store

http://cambridgesemantics.com/Project/c1c694bd-849d-91d0-eb34-bfef8c2d8894/c1c694bd-849d-91d0-eb34-bfef8c2d8894



Managed Editions

Title	↑	Description	Most Recent Published Date	Actions
Default Edition		Contains the latest successfully p...	12/12/2022 04:07PM	

Saved Editions

▾

Create New Edition

No editions found

General

Type : Catalog

Creator : System Administrat...


Structure Modified : 44 minutes ago

Released : Dec 9, 2022 1:06 PM

Statistics

Size(Bytes) : 29,119,163

http://csi.com/FileBasedLinkedDataSet




Dynamic Resource Log

Tags

None

- To create a new edition from scratch, click the **Create New Edition** button at the bottom of the screen.

### Tip

If you want to create an edition by cloning the Managed Edition, click the menu icon (  ) in the Actions column for the Managed Edition and select **Clone Edition**. To clone a Saved Edition, click the menu icon for the Saved Edition and select **Clone Edition**.

The Create New Edition (or Clone Edition) screen is displayed. The left side of the screen lists each of the jobs in the dataset's pipeline. For example:

## Create New Edition

Title \*

Description

Job Data included in this Edition

Name	Published Date	Job Type
Load ticket_venue	12/12/2022 0...	Standard
Load ticket_category	12/12/2022 0...	Standard
Load ticket_sales	12/12/2022 0...	Standard
Load ticket_listing	12/12/2022 0...	Standard
Load ticket_users	12/12/2022 0...	Standard
Load ticket_event	12/12/2022 0...	Standard

Rows per page: 25 1-7 of 7

Data Component for selected job included in this Edition

Select a Job in the table at left to include/exclude its available data components for your edition here.

CANCEL SAVE

Selecting a job shows the job's data components on the right side of the screen. For example:

## Create New Edition

Title \*

Description

Job Data included in this Edition

Name	Published Date	Job Type
Load ticket_venue	12/12/2022 0...	Standard
Load ticket_category	12/12/2022 0...	Standard
Load ticket_sales	12/12/2022 0...	Standard
Load ticket_listing	12/12/2022 0...	Standard
Load ticket_users	12/12/2022 0...	Standard
Load ticket_event	12/12/2022 0...	Standard

Rows per page: 25 1-7 of 7

Data Component for selected job included in this Edition

Name	Published Date
Loadticket_venues_ticket_venues	12/12/2022 04:07PM
Loadticket_venues_ticket_venues	12/12/2022 04:01PM
Loadticket_venues_ticket_venues	12/12/2022 03:53PM
Loadticket_venues_ticket_venues	12/09/2022 01:06PM

Rows per page: 25 1-4 of 4

CANCEL SAVE

- Note**
- When multiple jobs are selected in the Job Data list, the Data Component list shows the components for the most recently selected job.

Overview

Explore

Graphmarts

Pipelines

Dashboards

Versions

Description

None

Advanced

Pipeline

Load Tickets to store

<http://cambridgesemantics.com/Project/c1c694bd-849d-91d0-eb34-bfef8c2d8894/c1c694bd-849d-91d0-eb34-bfef8c2d8894>

Managed Editions

Title	↑	Description	Most Recent Published Date	Actions
Default Edition		Contains the latest successfully p...	12/12/2022 04:07PM	⋮

Saved Editions

Create New Edition

Title	↑	Description	Last Modified Date	Most Recent Published Date	Actions
Original components			12/12/2022 05:03PM	12/09/2022 01:08PM	⋮

General

Type : Catalog

Creator : System Administrator

Structure Modified : an hour ago

Released : Dec 9, 2022 1:06 PM

Statistics

Size(Bytes) : 29,119,163

<http://csi.com/FileBasedLinkedDataSet>

Dynamic Resource Log

Tags

None

## Related Topics

Modifying an Edition

You cannot change an existing Edition, but you can clone and edit a copy of an Edition. Follow the steps below to create a new Edition based on an existing version.

- 1. In the Anzo application, expand the **Blend** menu and click **Datasets**. Anzo displays the Datasets screen, which lists the catalog of datasets. For example:


Search

Add Dataset


<div></div>	Title	↑	Description	Updated Date	Actions	
	<div>Books</div>		Exported Books dataset		<div></div>	<div></div>
	<div>Flights</div>		Exported Flights dataset		<div></div>	<div></div>
	<div>Movies</div>		Exported Movies graphmart		<div></div>	<div></div>
	<div>Northwind</div>		Exported Northwind graphmart		<div></div>	<div></div>

Rows per page: 25 1-4 of 4 < >

- 2. Click the dataset for which you want to edit an edition. Anzo displays the Explore tab for the Dataset. Click the **Overview** tab, which lists the existing editions. For example:




Tickets to store

Not Versioned 

+ Add to Cart

Create Graphmart



Overview

Explore

Graphmarts


Pipelines

Dashboards

Versions

Description


None

Advanced 


Pipeline

Load Tickets to store


http://cambridgesemantics.com/Project/c1c694bd-849d-91d0-eb34-bfef8c2d8894/c1c694bd-4




Managed Editions

Title	↑	Description	Most Recent Published Date	Actions
Default Edition		Contains the latest successfully p...	12/12/2022 04:07PM	

Search



Create New Edition

Title	↑	Description	Last Modified Date	Most Recent Published Date	Actions
Original components			12/12/2022 05:03PM	12/09/2022 01:08PM	

General

Type : Catalog


Creator : System Administrator

Structure Modified : an hour ago

Released : Dec 9, 2022 1:06 PM

Statistics

Size(Bytes) : 29,119,163

http://csi.com/FileBasedLinkedDataSet 

Dynamic Resource Log

Tags

None

- Click the menu icon in the Actions column for the edition to copy. Select **Clone Edition** if you want to create a copy to change, or select **Browse Edition** if you want to review the edition before making a copy. When you are ready to make a copy, click the **Edit a Copy** button. Anzo opens the edition for editing. For example:

Clone Edition

Title \*

Description

Job Data included in this Edition

	Name	Published Date	Job Type
<input type="checkbox"/>	Load tickit_venues	12/12/2022 04:...	Standard
<input type="checkbox"/>	Load tickit_catego	12/12/2022 04:...	Standard
<input type="checkbox"/>	Load tickit_sales	12/12/2022 04:...	Standard
<input checked="" type="checkbox"/>	Load tickit_listings	12/12/2022 04:...	Standard
<input type="checkbox"/>	Load tickit_users	12/12/2022 03:...	Standard

Rows per page: 25
1-7 of 7

1 Items Selected

Data Component for selected job included in this Edition

	Name	Published Date
<input type="checkbox"/>	Loadtickit_listings_tickit_listings	12/12/2022 04:00PM
<input type="checkbox"/>	Loadtickit_listings_tickit_listings	12/12/2022 03:52PM
<input checked="" type="checkbox"/>	Loadtickit_listings_tickit_listings	12/09/2022 01:08PM

Rows per page: 25
1-3 of 3

1 Items Selected

CANCEL

SAVE

- Specify a name for the edition in the **Title** field and include an optional description in the **Description** field.
- To make changes to the edition, select or clear the Job Data checkboxes on the left side of the screen. Each time you select a job, the Data Components for that job are displayed on the right side of the screen. Select or clear the data component checkboxes to include or exclude components.
- When you have finished modifying the edition, click **Save**. Anzo creates the edition and adds it to the list of Saved Editions on the Overview screen.

The new edition is now available to add to a new or existing graphmart. To quickly create a new graphmart, you can click the menu icon in the Actions column for the new edition and select **Create Graphmart with this Edition**.

## Deleting a Saved Edition

Follow the steps below to delete a Saved Edition.

### Tip

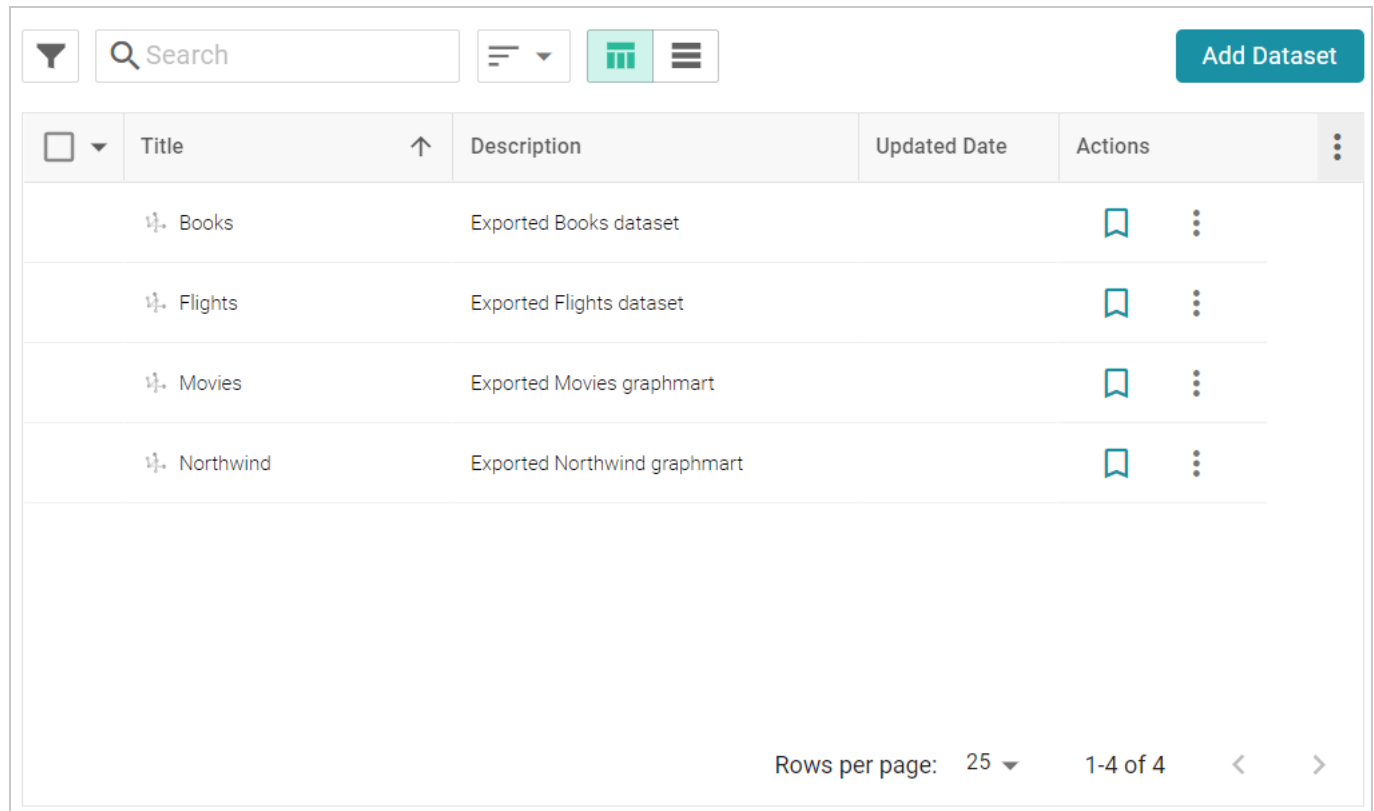
You cannot delete the Managed Edition, but users with administrative privileges can clear out the existing components so that the edition is recreated from scratch the next time the pipeline is published. For instructions, see [How do I clear the data components from the Managed Edition of a](#)

dataset?

### Note

Before deleting an edition, ensure that there are no graphmarts that require that edition.

1. In the Anzo application, expand the **Blend** menu and click **Datasets**. Anzo displays the Datasets screen, which lists the catalog of datasets. For example:



The screenshot shows the 'Datasets' screen in the Anzo application. At the top, there is a search bar with a magnifying glass icon and the text 'Search'. To the right of the search bar are two icons: a funnel and a list icon. Further right is a green button labeled 'Add Dataset'. Below these elements is a table with the following columns: 'Title', 'Description', 'Updated Date', and 'Actions'. The table contains four rows of data:

Title	Description	Updated Date	Actions
Books	Exported Books dataset		
Flights	Exported Flights dataset		
Movies	Exported Movies graphmart		
Northwind	Exported Northwind graphmart		

At the bottom right of the table, there is a pagination control showing 'Rows per page: 25' and '1-4 of 4'.

2. Click the dataset for which you want to delete an Edition. Anzo displays the Explore tab for the dataset. Click the **Overview** tab, which lists the existing editions. For example:



Overview
Explore
Graphmarts
Pipelines
Dashboards
Versions

Description  
None  
Advanced

Pipeline  
Load Tickets to store

Managed Editions

Title	Description	Most Recent Published Date	Actions
Default Edition	Contains the latest successfully p...	12/12/2022 04:07PM	

Saved Editions

Search
Create New Edition

Title	Description	Last Modified Date	Most Recent Published Date	Actions
Exclude Users		12/12/2022 05:11PM		
Original components		12/12/2022 05:03PM	12/09/2022 01:08PM	

General
Type : Catalog  
Creator : System Administrat...  
Structure Modified : an hour ago  
Released : Dec 9, 2022 1:06 PM

Statistics
Size(Bytes) : 29,119,163  
http://csi.com/FileBasedLinkedDataSet  
Dynamic Resource Log

Tags  
None

- In the list of Saved Editions, click the menu icon in the Actions column for the edition that you want to delete and select **Delete**. Anzo displays a confirmation message. Click **OK** to confirm the delete operation and remove the edition.

# Working with Graphmarts

The topics in this section provide guidance on working with graphmarts and data layers.

- [Creating a Graphmart](#) ..... 559
- [Copying a Graphmart](#) ..... 560
- [Graphmart Settings Reference](#) ..... 562
- [Creating an Elasticsearch Index from Graphmart Data](#) ..... 566
- [Adding Data Layers to Graphmarts](#) ..... 572
- [Adding Steps to Data Layers](#) ..... 589
- [Creating Data on Demand Endpoints](#) ..... 654
- [Sharing Access to Graphmarts](#) ..... 675

## Creating a Graphmart

Besides importing an exported graphmart, there are four ways to create a new graphmart:

1. You can create a graphmart directly from a connected data source. For instructions, see [Creating a Graphmart from a Data Source](#).
2. You can create a graphmart from a dataset in the Datasets catalog. For instructions, see [Creating a Graphmart from a Dataset](#).
3. You can create a graphmart by copying an existing one. For instructions, see [Copying a Graphmart](#).
4. Or you can create an empty graphmart and build it from scratch by adding data layers and steps that load and transform data. To create an empty graphmart, go to the Graphmarts screen in the Anzo application, click **Add Graphmart**, and select **Create New Graphmart**. See [Graphmart Settings Reference](#) for information about graphmart settings, and see [Adding Data Layers to Graphmarts](#) for information about adding layers and steps.

# Copying a Graphmart

Follow the instructions below if you want to create a graphmart by making a copy of an existing graphmart and its data layers and steps. For instructions on creating a new graphmart from scratch, see [Creating a Graphmart](#).

- 1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing graphmarts. For example:

Search

Add Graphmart

<input type="checkbox"/>	Title	Description	Status	Statements	Creator	Actions
<input type="checkbox"/>	Books		Ready to use	156	System Administrator	
<input type="checkbox"/>	Flights		Ready to use	265,339	System Administrator	
<input type="checkbox"/>	Movies		Ready to use	104,166	System Administrator	
<input type="checkbox"/>	Northwind		Ready to use	131,386	System Administrator	
<input type="checkbox"/>	Tickets		Ready to use	3,696,548	System Administrator	

Rows per page: 25 1-5 of 5 < >

- 2. On the Graphmarts screen, click the name of the graphmart that you want to copy. Anzo displays the Overview. At the top of the screen, click the **Save a Copy** button.

Tickets Graphmart

Not Versioned

INACTIVE

☒ ACTIVE

Ready to use

AZG

Static

Save a Copy

Create Dashboard

The Clone dialog box is displayed. For example:

**Clone**

Clone Title

Tickets Graphmart (Cloned 12/09/2022 02:08 PM)

Clone Description

(Cloned 12/09/2022 02:08 PM)

CANCEL

SAVE A COPY

3. In the Clone dialog box you have the option to edit the new graphmart name in the **Clone Title** field and modify the description in the **Clone Description** field.
4. When you are ready to save the copy, click **Save a Copy**.

Anzo creates and displays the new graphmart. You can proceed with modifying the graphmart settings, changing and adding datasets, data layers, or Data on Demand endpoints, or activating the new graphmart.

## Related Topics

[Graphmart Settings Reference](#)

[Creating a Graphmart](#)

[Sharing Access to Graphmarts](#)

[Creating Data on Demand Endpoints](#)

[Adding Data Layers to Graphmarts](#)

# Graphmart Settings Reference

This topic describes the graphmart configuration settings that are available on the graphmart Overview tab.

Overview

Explore

Datasets

Data Layers

Description

None

Load Priority

100

Metrics

Profile data upon activation

Disable Load Counts

Disable populating counts during load operations

Data Loading Settings

Load layers that do not fail

Ignore Source Errors

Ignore any step or view's source layer if they failed to load

Leave Graphmart Online During Refresh

When refreshing layers, leave graphmart and layer online.

Manual Refresh Graphmart

Once loaded, changes only pushed to AnzoGraph manually.(Only affects Journal Based Data)

Generate Statistics After Each Layer

Call generate statistics after loading each layer, instead of only at the end.

Require Elasticsearch

This Graphmart requires Elasticsearch.

Setting	Description
Load Priority	If you want Anzo to prioritize the order in which graphmarts are activated when reconnecting to AnzoGraph or resetting and reloading AnzoGraph, you can designate a Load Priority for each graphmart. When reloading AnzoGraph, Anzo activates the graphmarts in sequence, starting with the lowest Load Priority number. The default value is <b>100</b> .

Setting	Description
Metrics	<p><b>Profile Data Upon Activation</b></p> <p>This setting is disabled by default and controls whether a data profile is automatically generated each time the graphmart is activated. For information about data profiles, see <a href="#">Generating a Graphmart Data Profile</a>.</p>
Disable Load Counts	<p><b>Disable Populating Counts During Load Operations</b></p> <p>This setting is disabled by default controls whether Anzo periodically sends <code>select (count(*) as ?count) . . .</code> queries to AnzoGraph to count the total number of statements that are being loaded in each data layer. Disabling the load counts may increase load performance as it decreases the number of queries that run during graphmart activation.</p>
Data Loading Settings	<p><b>Load Layers that Do Not Fail</b></p> <p>This setting is enabled by default and controls what to do if a data layer fails during graphmart activation. When enabled (the default setting), the graphmart is configured to load all layers that succeed and skip any layers that fail. When disabled, the entire graphmart activation is aborted if any layer fails.</p>
Ignore Source Errors	<p><b>Ignore any Step or View's Source Layer if they Failed to Load</b></p> <p>This setting is enabled by default and controls what to do if a source that is referenced by a step or view fails to load. For example, if the source for a Query Step is set to "All Previous Layers Within Graphmart" and one of the previous layers fails to load, this setting controls whether to run the Query Step but ignore the failed layer or fail the step since one of the sources failed.</p> <p>If <b>Ignore Source Errors</b> is enabled (the default setting), Anzo ignores the failed source and runs the step against the sources that did not fail. For example, if <code>\${usingSources}</code> in a Query Step translates to</p> <pre> USING &lt;layer1&gt; USING &lt;layer2&gt; USING &lt;layer3&gt; </pre>

Setting	Description
	<p>And layer1 failed to load, Anzo runs the Query Step but ignores layer1 and automatically changes the query to</p> <pre> USING &lt;layer2&gt; USING &lt;layer3&gt; </pre> <p>If <b>Ignore Source Errors</b> is disabled and a source layer fails, any steps or views that use that source will also fail since the source is not available.</p>
<b>Leave Graphmart Online During Refresh</b>	<p><b>When Refreshing Layers, Leave Graphmart and Layer Online</b></p> <p>This setting is disabled by default and controls whether a graphmart remains online while it is being refreshed in AnzoGraph. When this option is enabled, if a user clicks the <b>Refresh</b> button to refresh a graphmart (or the Refresh icon on a data layer), Anzo copies the existing layers into temporary graphs so that the data remains online while the original graphs are refreshed. When the refresh is complete, the temporary graphs are deleted.</p> <div> <p><b>Note</b></p> <p>This setting applies only to <b>Refresh</b> operations. If <b>Leave Graphmart Online During Refresh</b> is enabled and a user clicks <b>Reload</b>, the data layers will not remain online. During reloads all of the data is dropped and then loaded again.</p> </div>
<b>Manual Refresh Graphmart</b>	<p><b>Once Loaded, Changes only Pushed to AnzoGraph Manually (Only Affects Journal Based Data)</b></p> <p>This setting is enabled by default and controls whether changes to a dataset in this graphmart are automatically deployed to AnzoGraph without requiring a manual refresh or reload of the graphmart. This setting only applies to graphmarts with Load Dataset Steps that load a journal-based data set, such as a system metadata graph. When this option is enabled, changes to the journal-based data set are only deployed to AnzoGraph when the graphmart is manually reloaded or refreshed. When this option is disabled, changes to the dataset are automatically loaded to AnzoGraph without requiring a manual refresh.</p>



Setting	Description
<b>Generate Statistics After Each Layer</b>	<p><b>Call Generate Statistics after Loading Each Layer, Instead of Only at the End</b></p> <p>Typically the AnzoGraph connection is configured to automatically initiate AnzoGraph's internal statistics gathering queries after loading a graphmart. However, if a user refreshes individual layers rather than the entire graphmart, those queries are not triggered. Enabling this setting initiates the statistics gathering queries each time a layer is loaded. This helps the AnzoGraph query planner generate ideal query execution plans for queries that are run against the refreshed data layers.</p>
<b>Require Elasticsearch</b>	<p><b>This Graphmart Requires Elasticsearch</b></p> <p>This setting is disabled by default. If you plan to include unstructured datasets in this graphmart or configure a data layer that creates an Elasticsearch index, you can enable this option to ensure that Anzo validates the connection to Elasticsearch whenever this graphmart is activated, reloaded, or refreshed.</p>

## Related Topics

[Sharing Access to Graphmarts](#)

## Creating an Elasticsearch Index from Graphmart Data

By associating an Elasticsearch index with a data layer, you can load data from a graphmart to an Elasticsearch index, enabling you to perform free-text and pattern searches on your knowledge graphs. This topic gives instructions for adding a layer that creates an Elasticsearch index for a graphmart.

### Prerequisites

Before configuring a graphmart to create an Elasticsearch index, make sure that the following requirements are met:

1. A supported version of Elasticsearch is installed and configured. For more information, see the [Deployment Guide](#).
2. The Elasticsearch instance is connected to Anzo. See [Connecting to Elasticsearch](#) in the Administration Guide for more information.
3. The AnzoGraph instance that you will load the graphmart to is also connected to Elasticsearch. Configure the connection by selecting the Elasticsearch instance in the **Elasticsearch Configuration** field in the AnzoGraph configuration. For more information, see [Connecting to AnzoGraph](#) in the Administration Guide.

### Adding a Data Layer that Creates an Elasticsearch Index

The steps below guide you through configuring a workflow that generates an Elasticsearch index and snapshot from graphs in a graphmart.

1. If a file-based linked data set (FLDS) has not been generated for this Graphmart, add an Export Step to the last layer in the Graphmart so that all of the graph data is exported to an FLDS on the file store. For instructions on adding steps, see [Adding Steps to Data Layers](#). For information about configuring an Export Step, see [Exporting Data to an FLDS \(Export Step\)](#).
2. Add a new Data Layer to the Graphmart. This layer will be associated with the Elasticsearch index and contain steps to generate the index and snapshot. For instructions on adding a layer, see [Adding Data Layers to Graphmarts](#).
3. In the new layer, click the **Elasticsearch Indexing** tab. Then select the **Associate an Elasticsearch index with this layer** checkbox. When this setting is enabled, Anzo creates an Elasticsearch index on the Elasticsearch server and links this layer to that index.

Details	Execution Condition	Data Access	Query Context	Elasticsearch Indexing
---------	---------------------	-------------	---------------	------------------------

☒ Associate an Elasticsearch index with this layer

**Warning:** These are advanced settings. Proceed with caution. Changes may significantly affect Elasticsearch behavior.

Advanced ▼

#### Note

Though the index will be populated with data from other layers in the Graphmart, you do not need to modify those layers to associate them with the index. **Associate an Elasticsearch index with this layer** should be disabled on all layers that do not contain Elasticsearch Indexing Steps.

4. The Advanced settings on this screen relate to managing the index's life cycle. It is not necessary to modify the settings to successfully generate an index for the Graphmart. If you have an advanced use case, such as a case that requires linking this layer to an existing index or needing to add custom Elasticsearch-specific index and/or mapping configurations, expand **Advanced** to access the options. The list below describes the available settings.

Details
Execution Condition
Data Access
Query Context
Elasticsearch Indexing

☒ Associate an Elasticsearch index with this layer

**Warning:** These are advanced settings. Proceed with caution. Changes may significantly affect Elasticsearch behavior.

Advanced ▲

Elasticsearch Index Name

Name will be system generated if left blank

Name will be system generated if left blank

☒ Clear Elasticsearch index before this layer executes
☒ Delete Elasticsearch Index on Layer Unload

Elasticsearch Index Settings

Settings (JSON format) for Elasticsearch index

Elasticsearch Index Mapping

Mapping (JSON format) for Elasticsearch index

CANCEL
SAVE

- **Elasticsearch Index Name:** By default, the new index is given a system-generated name. However, if you are linking this layer to an existing index, add the existing index name to this field. Or, if you plan to reference this index elsewhere and want to give it a human-readable name, you can specify a custom name.
- **Clear Elasticsearch index before this layer executes:** This option is enabled by default and means that the index is cleared and recreated each time this layer is run.
- **Delete Elasticsearch Index on Layer Unload:** This option is enabled by default and means that the index is deleted any time the Graphmart is deactivated or reloaded. The index is recreated during the reload or when the Graphmart is activated again. If you do not want the index to be deleted when the Graphmart is offline, clear the checkbox.
- **Elasticsearch Index Settings:** You can use this field to add, in JSON format, any Elasticsearch-specific index settings.
- **Elasticsearch Index Mapping:** You can use this field to add, in JSON format, any Elasticsearch-specific mapping settings.

5. When you have finished configuring the layer, click **Save** to add it to the Graphmart.

- Next, add an **Elasticsearch Indexing Step** to the new layer.
- On the **Details** tab for the Elasticsearch Indexing Step, complete the required fields and enter any optional details.

The screenshot shows a configuration form for an Elasticsearch Indexing Step. At the top, there are four tabs: 'Details' (selected), 'Query', 'Execution Condition', and 'Query Context'. The 'Details' tab contains the following fields:

- Title \***: A required text field with a placeholder 'The title of the step'.
- Description**: A text area with a placeholder 'A brief description of the step'.
- Enabled**: A checkbox that is checked, with the label 'Enabled'.
- Source**: A dropdown menu with 'Self' selected, a close button (x), and a dropdown arrow.
- Source data to act upon**: A text field with a placeholder 'Source data to act upon'.
- Data models**: A dropdown menu with 'Data models' selected, a close button (x), and a dropdown arrow.
- Associated data models**: A text field with a placeholder 'Associated data models'.

At the bottom right of the form, there are two buttons: 'CANCEL' and 'SAVE'.

The required field is described below. For more information about configuring Elasticsearch Indexing Steps, see [Creating an Elasticsearch Index \(Elasticsearch Indexing Step\)](#).

- **Title**: Specify the name to give the step.

- When you have finished configuring the Details tab, click the **Query** tab. The tab includes a template for writing a SPARQL SELECT query that incorporates the Graph Data Interface (GDI) service to generate an index on the Elasticsearch server.

Details	Query	Execution Condition	Query Context
Elasticsearch Indexing Query *			
1	PREFIX rdf:	<http://www.w3.org/1999/02/22-rdf-syntax-ns#>	
2	PREFIX rdfs:	<http://www.w3.org/2000/01/rdf-schema#>	
3	PREFIX xsd:	<http://www.w3.org/2001/XMLSchema#>	
4	PREFIX foaf:	<http://xmlns.com/foaf/0.1/>	
5	PREFIX s:	<http://cambridgesemantics.com/ontologies/DataToolkit#>	
6	PREFIX es:	<http://elastic.co/search/>	
7			
8	# Select clause should not be modified		
9	SELECT *		
10	# fromSources is replaced with the URIs of the Layer's Sources at runtime		
11	{fromSources}		
12	WHERE {		
13			
14	# above the SERVICE clause, execute a "standard" SPARQL query against the graph		
15	# bind any data you'd like to index in ES to variables		
16	# that will be passed TOPDOWN into the SERVICE clause below		
17			
18	### ----- EXAMPLE -----		
19	### ?person a foaf:Person ;		
20	### foaf:name ?name ;		
21	### foaf:age ?age ;		
22	### .		
23	### -----		
24			
25	SERVICE TOPDOWN	<http://cambridgesemantics.com/services/DataToolkit> {	
26			
27	# you should NOT define any ES connection or index information within the query		
28	# as it will be automatically populated when the step executes		
29			
30	?data a es:ElasticSource ;		
31			

CANCEL
SAVE

Unlike queries for other steps, this step does not run an INSERT query because the data is not being inserted into AnzoGraph. Edit the template as needed. You can click the **Preview in Query Builder** button to open the query in the Query Builder, where you can perform practice runs to see results without having to refresh the Graphmart or layer. For general information about writing GDI queries, see [Getting Started with GDI Queries](#).

#### Note

Do not include Elasticsearch connection or index parameters in the query. Anzo automatically populates that information from the AnzoGraph configuration when the step is run.

- When you have completed the indexing query, click **Save** to save the configuration and add the step to the layer.

10. If you want to create a snapshot for the index in this layer, add an **Elasticsearch Snapshot Step** to the same layer. On the **Details** tab for the Elasticsearch Snapshot Step, complete the required fields and enter any optional details.

The screenshot shows the configuration form for an Elasticsearch Snapshot Step, with the 'Details' tab selected. The form includes the following fields and options:

- Title \***: A text input field with the placeholder text 'The title of the step'.
- Description**: A text input field with the placeholder text 'A brief description of the step'.
- Enabled**: A checkbox that is checked, indicating the step is enabled.
- Source**: A dropdown menu showing 'Self' and 'All Previous Layers Within Graphmart' as options. Below the dropdown, it states 'This option cannot be changed.'
- Target FLDS \***: A dropdown menu for selecting the target FLDS, with the placeholder text 'FLDS only.'

At the bottom right of the form are 'CANCEL' and 'SAVE' buttons.

The list below describes the required fields. For more information about configuring Elasticsearch Indexing Steps, see [Taking a Snapshot of an Elasticsearch Index \(Elasticsearch Snapshot Step\)](#).

- **Title**: Specify the name to give the step.
- **Target FLDS**: Select the FLDS to save the snapshot to. Typically you select the FLDS that was either created by an Export Step in a previous layer or generated automatically based on the workflow that was followed to create the graphmart.

11. When you have finished configuring the snapshot step, click **Save** to save the configuration and add the step to the layer.

Once the new Elasticsearch-related layer is complete, reload the graphmart to generate an FLDS (if one did not exist), create an index, and save a snapshot to the FLDS.

## Adding Data Layers to Graphmarts

The topics in this section introduce the concepts to know when working with data layers and provide instructions for creating and configuring layers.

[Introduction to Data Layers](#)

[Creating a New Data Layer](#)

[Cloning an Existing Data Layer](#)

[Using Query Contexts in Queries](#)

[Defining Execution Conditions](#)

[Advanced Data Access Settings](#)

### Related Topics

[Adding Steps to Data Layers](#)

### Introduction to Data Layers

Data layers enhance graphmarts dynamically by loading additional data sets, inferring new data automatically, or running SPARQL queries to create, clean, conform, transform, or validate data. This topic introduces the fundamental concepts to know when working with data layers.

- [What is a Data Layer?](#)
- [What is a Step?](#)

### What is a Data Layer?

A data layer is a container for one or more steps. The steps in a layer perform the data loads or creation and transformation.

- You can create any number of layers in a graphmart and control which users and groups have access to the layers and the data within the layers.
- Users can toggle data layers on and off in Hi-Res Analytics dashboards.
- You can configure Execution Conditions on layers so that the layer is processed only when certain conditions are met.
- You cannot share data layers between graphmarts, but you can clone layers and include a copy in multiple graphmarts.
- You control the source data for steps in a layer. Layers can build upon the data generated by steps in previous layers or can be self-contained, applying changes only to the data defined in the layer.



## What is a Step?

The steps in a layer perform all of the data operations, such as loading a data set, creating new data, or transforming the existing data. You can add any number of steps to a layer and can create the following types of steps:

- **Direct Load Step:** Runs a SPARQL query that invokes the Graph Data Interface (GDI) RDF and Ontology Generators to directly load a data source to a graphmart. For more information, see [Generating a Graph and Ontology with a Direct Load Step](#).
- **Elasticsearch Indexing Step:** Creates an Elasticsearch index to associate with a layer.
- **Elasticsearch Snapshot Step:** Creates an Elasticsearch snapshot of the index that is associated with a layer.
- **Export Step:** Exports the Graphmart data in memory to a file-based linked data set (FLDS).
- **Load Dataset Step:** Loads a dataset from the Datasets catalog into a data layer graph. When you create a graphmart from the catalog, Anzo automatically creates a layer with a Load Dataset Step.
- **Pre-compile Query Step:** Runs the included query immediately after a graphmart is loaded so that the query is pre-compiled by AnzoGraph. Pre-compiling a query reduces execution time when a user runs that query for the first time.
- **Query Driven Templated Step:** Enables users to create reusable query-driven templates for quickly creating additional query steps. Unlike the Templated Step, where users define each key-value pair, this step runs a query to identify all of the key-value pairs. Then the template query is run for each key-value solution from the first query.
- **Query Step:** Provides a SPARQL query template that you can use for writing a query that creates, cleans, conforms, or transforms data in the data layer.
- **RDFS+ Inference Step:** Uses RDFS and OWL rules to generate new data in a layer based on the vocabularies in the existing data.
- **Templated Step:** Enables users to create reusable templates for quickly creating additional query steps. The query in a Templated Step uses parameters to represent key-value pairs. When reusing the step, users modify the values for the keys rather than rewriting the query.
- **Validation Step:** Enables users to write a query that validates the data in a data layer and is a required component when setting up an execution condition.
- **View:** Enables users to write a SPARQL CONSTRUCT query that defines a view of the data but does not alter the source data or create new data unless you choose to materialize the view.

For information about creating steps, see [Adding Steps to Data Layers](#).

## Creating a New Data Layer

Follow the steps below to create a new data layer in a graphmart. For instructions on creating a new layer by copying an existing one, see [Cloning an Existing Data Layer](#).

1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing graphmarts. For example:

Search

Add Graphmart

<div></div>	Title	Description	Status	Statements	Creator	Actions
	Books		<div>Ready to use</div>	156	System Administrator	<div></div>
	Flights		<div>Ready to use</div>	265,339	System Administrator	<div></div>
	Movies		<div>Ready to use</div>	104,166	System Administrator	<div></div>
	Northwind		<div>Ready to use</div>	131,386	System Administrator	<div></div>
	Tickets		<div>Ready to use</div>	3,696,548	System Administrator	<div></div>

Rows per page: 25

1-5 of 5

2. Click the name of the graphmart that you want to add a layer to. The Overview tab is displayed. For example:

Northwind

Not Versioned

INACTIVE

ACTIVE

Ready to use

AZG

Static

Save a Copy

Create Dashboard

Overview

Explore

Datasets

Data Layers

Dashboards

Data on Demand

Description

None

Load Priority

None

Metrics

Profile data upon activation

Disable Load Counts

Disable populating counts during load operations

Data Loading Settings

Load layers that do not fail

Ignore Source Errors

Ignore any step or view's source layer if they failed to load

Leave Graphmart Online During Refresh

When refreshing layers, leave graphmart and layer online.

Manual Refresh Graphmart

Once loaded, changes only pushed to AnzoGraph manually.(Only affects Journal Based Data)

General

Type : Graphmart

Creator : System Administrat...

Last Accessed : 5 hours ago

Last Updated : 5 hours ago

Structure Modified : 6 hours ago

Released : 6 hours ago

http://cambridgesemantics.com/Graph

Dynamic Resource Log

AnzoGraph Server Details

AnzoGraph : AZG

Status : Ready to use

Last Accessed : 3 hours ago

Memory Used : 280.5 MB( 3% )

Memory Total : 11.7 GB

Inactivity Deactivate Timeout

None

3. Click the **Data Layers** tab. Anzo displays the existing data layers. For example:

Northwind

Not Versioned

INACTIVE

ACTIVE

Ready to use

AZG

Static

Save a Copy

Create Dashboard

Overview

Explore

Datasets

Data Layers

Dashboards

Data on Demand

RELOAD

HIDE SIMPLE LOAD LAYERS

EXPAND ALL

ADD

1. DB-QA Layer

Load data from DB-QA

2. Create Dataset

4. Click **Add** and select **New Layer**. Anzo displays the Create dialog box.

**Create**

Details Execution Condition Data Access Query Context Elasticsearch >

Title \*

The title of the layer

Description

A brief description of the layer

☐ Auto Deploy Ontology Changes ☐ Disable Loading Counts

CANCEL SAVE

5. Specify a name for the layer in the **Title** field and an optional description in the **Description** field.
6. Determine how you want to control changes to this layer's dependent data models:

- If you want Anzo to automatically deploy to AnzoGraph any changes to the related models without having to manually refresh the layer or graphmart, select the **Auto Deploy Ontology Changes** checkbox.

#### Note

The **Manual Refresh Graphmart** setting on the graphmart must be **disabled** for automatic deployment of models to work. See [Graphmart Settings Reference](#) for information about graphmart settings.

- If you want model changes to be deployed to AnzoGraph only when this layer (or entire graphmart) is refreshed or reloaded, leave the **Auto Deploy Ontology Changes** checkbox empty (disabled).
7. Determine whether to **Disable Loading Counts** for this layer. This setting controls whether Anzo periodically sends COUNT queries to AnzoGraph while this layer is reloading or refreshing. Disabling

the load counts may increase load performance as it decreases the number of queries that are executed during loads.

8. Click **Save** to add the new layer to the graphmart and return to the Data Layers screen.

The new layer becomes the last layer in the graphmart. If you want to change the order of the layers, you can click the black bar on the left side of a layer and drag the layer up or down. Data layers are processed from top to bottom.

#### Tip




















The Refresh icon (🔄) on the new layer indicates that the layer is out of sync with the data that is in AnzoGraph. Once you configure the new layer and add data processing steps, you can click the **Reload** button (🔄) at the top of the screen to reload the entire graphmart, or you can click the **Refresh** icon (🔄) on the layer to reload only that layer.

See [Adding Steps to Data Layers](#) for instructions on creating steps.

### Cloning an Existing Data Layer

Follow the steps below to create a new data layer by copying an existing one from any graphmart. For instructions on creating a layer from scratch, see [Creating a New Data Layer](#).

1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing graphmarts. For example:

	<input type="text" value="Search"/>				<button>Add Graphmart</button>		
<input type="checkbox"/>	Title	↑	Description	Status	Statements	Creator	Actions
	Books			 Ready to use	156	System Administrator	 ⋮
	Flights			 Ready to use	265,339	System Administrator	 ⋮
	Movies			 Ready to use	104,166	System Administrator	 ⋮
	Northwind			 Ready to use	131,386	System Administrator	 ⋮
	Tickets			 Ready to use	3,696,548	System Administrator	 ⋮
Rows per page: 25 1-5 of 5 < >							

2. Click the name of the graphmart that you want to add a layer to. The Overview tab is displayed. For example:

The screenshot shows the 'Northwind' graphmart interface. At the top, there's a header with the graphmart name, status (Inactive/Active), and a 'Ready to use' indicator. Below this is a navigation bar with tabs: Overview, Explore, Datasets, Data Layers, Dashboards, and Data on Demand. The 'Overview' tab is selected. The main content area is divided into two columns. The left column contains settings for the graphmart, including 'Description' (None), 'Load Priority' (None), 'Metrics' (Profile data upon activation), 'Disable Load Counts' (Disable populating counts during load operations), 'Data Loading Settings' (Load layers that do not fail), 'Ignore Source Errors' (Ignore any step or view's source layer if they failed to load), 'Leave Graphmart Online During Refresh' (When refreshing layers, leave graphmart and layer online), and 'Manual Refresh Graphmart' (Once loaded, changes only pushed to AnzoGraph manually). The right column contains a 'General' section with metadata (Type: Graphmart, Creator: System Administrator, Last Accessed: 5 hours ago, Last Updated: 5 hours ago, Structure Modified: 6 hours ago, Released: 6 hours ago) and a 'Dynamic Resource Log' section with a URL (http://cambridgesemantics.com/Graph). Below the log is an 'AnzoGraph Server Details' section showing 'AnzoGraph: AZG', 'Status: Ready to use', 'Last Accessed: 3 hours ago', 'Memory Used: 280.5 MB (3%)', and 'Memory Total: 11.7 GB'. At the bottom, there's an 'Inactivity Deactivate Timeout' section set to 'None'.

3. Click the **Data Layers** tab. Anzo displays the existing data layers. For example:

The screenshot shows the 'Northwind' graphmart interface with the 'Data Layers' tab selected. The header and navigation bar are the same as in the previous screenshot. The main content area shows a list of data layers. At the top of the list, there are controls: 'RELOAD', 'HIDE SIMPLE LOAD LAYERS', 'EXPAND ALL', and an 'ADD' button. The list contains two layers: '1. DB-QA Layer' (with a checkmark icon and description 'Load data from DB-QA') and '2. Create Dataset'. Each layer has a toggle switch and a three-dot menu icon to its right. The 'DB-QA Layer' toggle is turned on, and the 'Create Dataset' toggle is turned off.

- click **Add** and select **Add Existing**. Anzo opens the Select a value to add dialog box, which lists the existing layers for all graphmarts. For example:

Select a value to add

Search

Hide simple load layers

Title	Description	Updated Date	Tags	Actions
Create Dataset		Nov 16, 2022		<div></div> <div></div>
Create Dataset		Nov 16, 2022		<div></div> <div></div>
DB-QA Layer	Load data from DB-QA	Nov 16, 2022	Auto-Gen	<div></div> <div></div>
Flights Layer	Load data from Flights	Nov 16, 2022	Auto-Gen	<div></div> <div></div>
Movies Layer	Load data from Movies	Nov 16, 2022	Auto-Gen	<div></div> <div></div>
Movies Layer	Load data from Movies	Nov 16, 2022	Auto-Gen	<div></div> <div></div>
Tickets Layer	Load data from Tickets	Nov 16, 2022	Auto-Gen	<div></div> <div></div>

Rows per page: 25 1-8 of 8

CANCEL OK

- Select the layer that you want to copy and click **OK**. The Clone dialog box is displayed. For example:

Clone

Clone Title

Flights Layer (Cloned 11/16/2022 05:31 PM)

Clone Description

(Cloned 11/16/2022 05:31 PM)

CANCEL SAVE AS

- In the Clone dialog box, you have the option to edit the **Clone Title** and/or **Clone Description**.
- Click **Save As** to add the cloned layer and any steps that the layer contains to the graphmart.

The new layer becomes the last layer in the graphmart. If you want to change the order of the layers, you can click the black bar on the left side of a layer and drag the layer up or down. Data layers are processed from top to bottom.

### Tip

The Refresh icon (🔄) on the new layer indicates that the layer is out of sync with the data that is in AnzoGraph. Once you configure the new layer and add data processing steps, you can click the **Reload** button (🔄) at the top of the screen to reload the entire graphmart, or you can click the **Refresh** icon (🔄) on the layer to reload only that layer.

See [Adding Steps to Data Layers](#) for instructions on creating steps.

## Using Query Contexts in Queries

When you use the Graph Data Interface (GDI) for querying data sources, you may connect to sources that require input of sensitive connection and authorization information such as keys, tokens, and user credentials. When configuring a step that runs a GDI query, Cambridge Semantics recommends that you refer to a Query Context whenever possible. Contexts store sensitive information as key-value pairs. Queries reference only the keys from the context and the sensitive values are abstracted from the requests that are sent to the data source and AnzoGraph. This topic provides information on configuring Query Contexts and referring to Context Variables in a query.

- [Overview of Query Contexts](#)
- [Referencing Context Variables in a Query](#)

## Overview of Query Contexts

Query contexts are accessible from the **Query Context** tab that is available when creating or editing a data layer or step. The image below shows the Query Context tab for a layer.



Details

Execution Condition

Data Access

Query Context

Elasticsearch Indexing

Select the Context Providers that will be made available to queries via Context Variables.

Context Providers

Reference existing database connections to provide connection information for services accessed during the query.

Search to add Context Providers

Context Variables

Access Context Variables for each Context Provider added above.

Context Provider

Custom Context

Add Key

Name	Value
None	

CANCEL

SAVE

## Context Providers

Connections in Anzo implement the **Context Provider** interface. For example, file store connections and data source connections provide contexts (in the form of JSON objects) that contain key-value pairs. The contexts contain the source connection details such as URLs, database names, user names, passwords, and tokens. A context is passed to the data source when a request is made against that source. To use one of the Anzo-generated Context Providers that was created for a pre-existing connection, select that provider from the drop-down list. When you select a provider, the variables from that context are displayed under Context Variables, as shown in the image below:

Details
Execution Condition
Data Access
**Query Context**
Elasticsearch Indexing

Select the Context Providers that will be made available to queries via Context Variables.

### Context Providers

Reference existing database connections to provide connection information for services accessed during the query.

Context Providers

DB-QA x

### Context Variables

Access Context Variables for each Context Provider added above.

Context Provider  
DB-QA ▼

```
#DB-QA
db.718cc8ab3e9fb2b22345ad4dc31e685c.label
db.718cc8ab3e9fb2b22345ad4dc31e685c.password
db.718cc8ab3e9fb2b22345ad4dc31e685c.server
db.718cc8ab3e9fb2b22345ad4dc31e685c.url
db.718cc8ab3e9fb2b22345ad4dc31e685c.user
db.http://cambridgesemantics.com/MySQLDatabaseDataSource/30eb8be931f9440c8e570625709d1377.label
db.http://cambridgesemantics.com/MySQLDatabaseDataSource/30eb8be931f9440c8e570625709d1377.password
db.http://cambridgesemantics.com/MySQLDatabaseDataSource/30eb8be931f9440c8e570625709d1377.server
db.http://cambridgesemantics.com/MySQLDatabaseDataSource/30eb8be931f9440c8e570625709d1377.url
db.http://cambridgesemantics.com/MySQLDatabaseDataSource/30eb8be931f9440c8e570625709d1377.user
```

Custom Context
Add Key

CANCEL
SAVE

## Context Variables

When you select a Context Provider, the variables from the selected context are displayed under Context Variables. You can copy the text and then use any key in the list as a variable in any query that connects to this data source. Typically there are two versions of each variable. Either version can be used in a query. The short versions are generated from the long versions for ease of use and readability in queries. For information about referencing variables in a query, see [Referencing Context Variables in a Query](#) below.

## Custom Context

Custom Contexts are user-defined key-value pairs that are not associated with a particular Context Provider. To add a key and define its value, click the **Add Key** button. Then specify the **Key Name** and **Key Value** in the Create Context Key dialog box. Click **Create** to add the key-value pair to the context.

Create Context Key

Key Name \*

Key Value \*

CANCEL







CREATE

**Tip**

When defining custom keys at the layer level, make sure that the key names are unique. A layer may have different steps that connect to different sources but reference the same custom context. The image below, for example, creates URL, username, and password Context Keys for a MySQL database.

Context Keys

Add Key

Name	Value	
mysql-password	admin123	 
mysql-username	sysadmin	 
mysql-northwind-url	jdbc:mysql://10.111.4.9:3306/NORTHWIND	 

Referencing Context Variables in a Query

The format that you use for referencing a Context Variable in a query depends on the type of AnzoGraph plugin or extension that is being called by the query. Generally, contexts are only used in steps that contain GDI queries. When referencing context keys in GDI queries, use the following format:

```
{{@context_key_name}}
```

For example, the following GDI query references three variables from the Context Provider shown above: url, user, and password.

```
PREFIX s:      <http://cambridgesemantics.com/ontologies/DataToolkit#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
```

```

SELECT DISTINCT *
WHERE
{
  SERVICE <http://cambridgesemantics.com/services/DataToolkit>
  {
    ?data a s:DbSource ;
    s:url "{{@db.718cc8ab3e9fb2b22345ad4dc31e685c.url}}" ;
    s:username "{{@db.718cc8ab3e9fb2b22345ad4dc31e685c.user}}" ;
    s:password "{{@db.718cc8ab3e9fb2b22345ad4dc31e685c.password}}" ;
    s:selector "[dbo].[FILM]" ;
    ?year ("[YEAR]" xsd:int);
    ?length (xsd:int) ;
    ?title (xsd:string) ;
    ?subject ("[dbo].[FILM].[SUBJECT]" xsd:string) ;
    ?actor ("[ACTOR]" xsd:string) ;
    ?actress (xsd:string) ;
    ?director (xsd:string) ;
    ?popularity (xsd:int) ;
    ?awards (xsd:string) ;
    ?image (xsd:string) .
    FILTER(?year >= 1990 && ?year < 2000)
    FILTER(?subject = "Drama" || ?subject = "Action")
    FILTER(?length <= 90)
  }
}

```

## Related Topics

[Adding Steps to Data Layers](#)

## Defining Execution Conditions

Execution Conditions can be defined at the Data Layer or Step level and are used to conditionalize the execution of the Data Layer or Step based on the result of a specified *Validation Condition*.

### Note

Validation Conditions are defined in Validation Steps. In order to set up an Execution Condition, the Graphmart needs to have at least one **Validation Step** that defines a **Condition Variable**.

Validation Conditions can be used across all Data Layers in the same Graphmart. For guidance on

configuring a Validation Step, see [Validating the Data \(Validation Step\)](#).

This topic focuses on configuring an Execution Condition at the Data Layer level.

## Configuring an Execution Condition

Execution Conditions are configured from the **Execution Condition** tab that is available when creating or editing a Data Layer or Step. The image below shows the Execution Condition tab for a Data Layer.

The screenshot shows a configuration window with four tabs: Details, Execution Condition (selected), Data Access, and Masking. Under the Execution Condition tab, there is a checkbox labeled "Enable layer based on boolean condition". Below this is a dropdown menu labeled "Select condition variable" with a downward arrow. Underneath the dropdown is the text "Conditional Variable". Below that is another dropdown menu labeled "Execute only if returned value is" with a downward arrow. Underneath this dropdown is the text "Conditional Variable If Result". At the bottom right of the window are two buttons: "CANCEL" and "SAVE".

### Enable Layer Based on Boolean Condition

This setting indicates whether to enable this Data Layer only if the returned value from the Validation Condition is either **true** or **false**. You specify true or false in the **Conditional Variable If Result** field. If the Validation Condition fails, the Data Layer is disabled.

### Conditional Variable

This field specifies the variable that you want to base this Execution Condition on. If the list is empty, that means either there are no Validation Steps in the Graphmart or a Validation Step exists but it does not include a Condition Variable (defined in the Options tab). Refer to [Validating the Data \(Validation Step\)](#) for more information.

### Conditional Variable If Result

If you enabled the **Enable Layer Based on Boolean Condition** setting, select **true** or **false** from the drop-down list. The Data Layer will be enabled only if the result of the Validation Step Query matches the value that you specified. If Enable Layer Based on Boolean Condition is disabled, leave this field blank.

## Related Topics

Advanced Data Access Settings

This topic provides reference information about the Advanced Data Layer Hi-Res Analytics settings that control how a layer is exposed to and affects Hi-Res Analytic dashboards.

**Important**  
Changing these settings can have unexpected consequences.


The Advanced Hi-Res Analytics settings are available on the **Data Access** tab when you create or edit a data layer:

DetailsExecution ConditionData AccessQuery ContextElasticsearch Indexing

**View Permissions**  

☒ No View Acls Set (Inherits From Layer's Configuration Graph)  
☐ Inherit from Graphmart  
☐ Custom view permissions

**Advanced**  

 **Warning:** These are advanced settings, please change wisely.

☒ Display data outputted by the layer in Hi-Res Analytics dashboards

☒ Display Layer in Data Layers on Hi-res Analytics

☒ User can both view and hide the data outputted by the layer

☒ By default, the data outputted by the layer is visible

☐ Hide From Other Layers

CANCELSAVE

The sections below describe each of the available settings:

- [Display data outputted by the layer in Hi-Res Analytics dashboards](#)
- [Display Layer in Data Layers in Hi-Res Analytics](#)
- [User can both view and hide the data outputted by the layer](#)

- [By default, the data outputted by the layer is visible](#)
- [Hide from Other Layers](#)

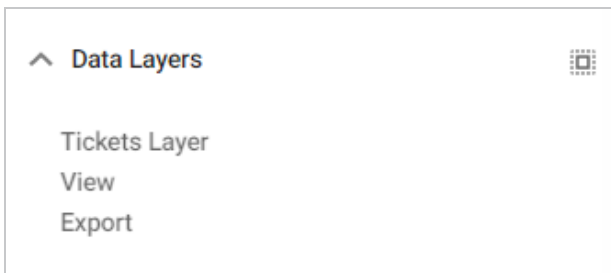
## Display data outputted by the layer in Hi-Res Analytics dashboards

This setting controls whether the data generated by the steps is available to query and display in Hi-Res Analytics dashboards:

- When the setting is **enabled** (the default value), the layer's data is available to dashboards.
- When the setting is **disabled**, other data layers in the graphmart can use the layer's data, but the data is not available to dashboards.

## Display Layer in Data Layers in Hi-Res Analytics

This setting controls whether Anzo displays the layer in the Data Layers panel on Hi-Res Analytics dashboards. The image below shows an example Data Layers panel:



- When the setting is **enabled** (the default value), the layer is listed in the Data Layers panel in dashboards.
- When the setting is **disabled**, the layer's data is always used in dashboards but users do not see the layer listed in the Data Layers panel.

## User can both view and hide the data outputted by the layer

This setting controls whether users have the option to show and hide the layer in the Data Layers panel on dashboards:

- When the setting is **enabled** (the default value), the layer is listed in the Data Layers panel in dashboards and users have the option to show and hide the layer.
- When the setting is **disabled**, whether the layer shows up in the Data Layers panel depends on the **By default, the data outputted by the layer is visible** setting. If the layer is visible in the Data Layers panel ("By default, the data outputted by the layer is visible" is enabled), users cannot toggle it on and off.

## By default, the data outputted by the layer is visible

This setting controls whether the data that is generated by the steps in the layer is visible in queries and Hi-Res Analytics dashboards:

- When the setting is **enabled** (the default value), the layer is listed in the Data Layers panel in dashboards and is selected by default. The layer's data is also included by default when queries are run against the graphmart.
- When the setting is **disabled**, the layer shows up in the Data Layers panel but is not selected. To include the layer's data in Hi-Res Analytic queries, users must select the layer. In addition, the layer's data is automatically excluded from queries that are run against the graphmart. To include data from the layer in results, the queries must explicitly list the layer's URI.

## Hide from Other Layers

This setting controls whether the other layers in the graphmart can act upon the source data in this layer.

- When the setting is **disabled** (the default value), this layer is available as a choice in the **Source data to act upon** drop-down list when a step's source is configured.
- When the setting is **enabled**, this layer is not listed as a choice in the **Source data to act upon** list when the a step's source configured.



## Adding Steps to Data Layers

The steps in a data layer perform the data operations, such as loading, creating, deleting, changing, or exporting data. You can add any number of steps to a layer. The topics in this section provide information about configuring each type of step.

[Directly Loading a Data Source \(Direct Load Step\)](#)

[Creating an Elasticsearch Index \(Elasticsearch Indexing Step\)](#)

[Taking a Snapshot of an Elasticsearch Index \(Elasticsearch Snapshot Step\)](#)

[Exporting Data to an FLDS \(Export Step\)](#)

[Loading a Dataset from the Catalog \(Load Dataset Step\)](#)

[Pre-Compiling a Query \(Pre-Compile Query Step\)](#)

[Creating a Reusable Query Template](#)

[Running a Transformation Query \(Query Step\)](#)

[Inferring New Data \(RDFS+ Inference Step\)](#)

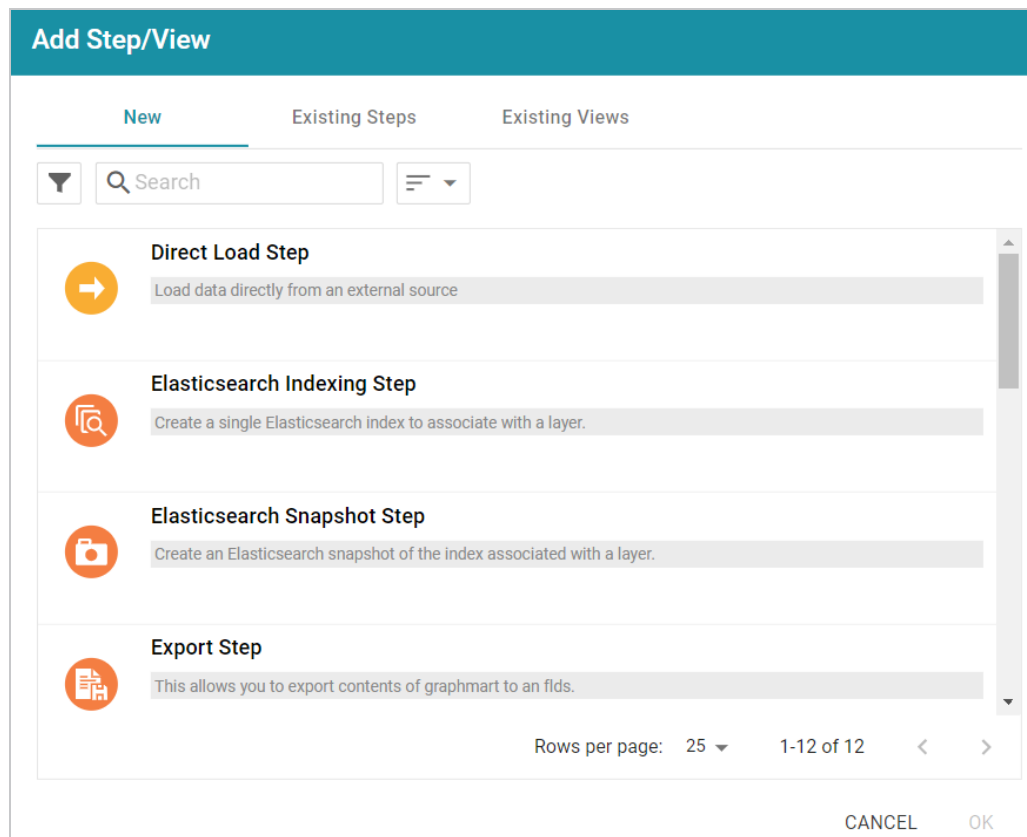
[Validating the Data \(Validation Step\)](#)

[Constructing a View of the Data \(View Step\)](#)

### Directly Loading a Data Source (Direct Load Step)

With no mapping required, a Direct Load Step can be used to automatically generate a graph and model for a data source. The Direct Load Step is the only type of step with the ability to manage generated models. A model that is generated by a Direct Load Step is automatically registered in Anzo is linked to and managed by the layer that contains the step. If a query is changed, additional Direct Load Steps are added to the same layer, or the underlying source schema changes, the managed model is automatically updated when the graphmart is reloaded or refreshed. Follow the steps below to create a Direct Load Step.

1. Go to the graphmart for which you want to add a step and then click the **Data Layers** tab.
2. On the Data Layers tab, find the layer that you want to add the step to. Click the menu icon (⋮) for that layer and select **Add Step/View**. The Add Step/View dialog box is displayed with the New tab selected.



3. To create a new Direct Load step, select **Direct Load Step** and then click **OK**. If you want to clone an existing step, click the **Existing Steps** tab, select the step that you want to clone, and then click **OK**. Anzo creates or clones the step and displays the Details tab:

The screenshot shows a configuration window with four tabs: **Details**, **Query**, **Execution Condition**, and **Query Context**. The **Details** tab is active and contains the following elements:

- Title \***: A text input field with the placeholder text "The title of the step".
- Description**: A text input field with the placeholder text "A brief description of the step".
- Enabled**: A checkbox that is currently checked.
- Source**: A dropdown menu showing "Self" as the selected option.
- Data models**: A dropdown menu showing "Managed" as the selected option.
- Associated data models**: A label below the "Data models" dropdown.
- Buttons**: "CANCEL" and "SAVE" buttons located at the bottom right of the window.

4. On the Details tab, configure the following options as needed:

- **Title:** The required name of the step.
- **Description:** An optional short description of the step.
- **Enabled:** When creating a new step, the Enabled option is selected by default, indicating that the step is enabled and will run when the layer is loaded or refreshed. If you want to disable the step so that it is not processed, clear the **Enabled** checkbox.
- **Source:** The source data that this step should act upon. Steps can build upon the data generated by steps in other layers or can be self-contained, applying changes that relate only to the data defined in the layer that contains this step. You can select any number of the following options:
  - **Self:** This option is selected by default and means that the step runs against only the data that is generated in the layer this step belongs to.
  - **All Previous Layers Within Graphmart:** This option means that the step runs against the data that is generated by all of the successful layers that precede the layer this step is in. Any failed layers are ignored.
  - **Previous Layer Within Graphmart:** This option means that the query runs against only the data that is generated by the one layer that precedes the layer this step is in.

- **Layer Name:** The Source drop-down list also includes options for specific layer names.

You can choose a specific layer to act upon the data in that layer only.

- **Data models:** This optional field specifies the model or models to associate with this step. By default, **Managed** is selected. If you are onboarding a source that does not have a model, make sure Managed remains specified so that the step generates a model. See [Important Notes about Managed Models](#) below for more information about managed models.

The Data Models list displays all of the available models. By default, the field is set to **Exclude System Data** (☺). If you want to choose a system model, click the toggle button on the right side of the field to change it to **Include System Data** (☹). When system data is included, the drop-down list displays the system models in addition to the user-generated models.

- **Pre-Run Generate Statistics:** This option controls whether to initiate AnzoGraph's internal statistics gathering queries before running the query to pre-compile. The statistics gathering helps ensure that the AnzoGraph query planner generates ideal query execution plans for queries that are run against the graphmart.

5. When you have finished configuring the Details tab, click the **Query** tab. This tab defines the query that this step should run.

Details

Query

Execution Condition

Query Context

Transformation query \*

1

PREFIX rdf:

<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

2

PREFIX rdfs:

<http://www.w3.org/2000/01/rdf-schema#>

3

PREFIX xsd:

<http://www.w3.org/2001/XMLSchema#>

4

PREFIX s:

<http://cambridgesemantics.com/ontologies/DataToolkit#>

5

6

# targetGraph is replaced with the Layers URI at runtime

7

# usingSources is replaced with the URIs of the Layer's Sources at runtime

8

DELETE {

9

GRAPH \${targetGraph} {

10

11

}

12

}

13

INSERT {

14

GRAPH \${targetGraph} {

15

16

}

17

}

18

\${usingSources}

19

WHERE {

20

SERVICE <http://cambridgesemantics.com/services/DataToolkit> {

21

.

CANCEL

SAVE

- Typically Direct Load Step queries are GDI RDF and Ontology Generator queries. Using a relatively simple SPARQL query, the GDI Generators recognize the structure of a data source and automatically generate the necessary statements. Invoking the Generators is preferable when the structure of the data is very complex, such as a JSON data source with many inner repeating structures or a database with many tables and keys. When the source contains complex structures, only the required statements are generated, avoiding cross-products and optimizing query execution and memory usage. For details about writing GDI Generator queries, see [GDI Generator Query Syntax](#).

#### Note

If your query connects to a source that requires input of connection and authorization information, Cambridge Semantics recommends that you do not include the connection and authorization values directly in the query. Instead, replace those values with Context Variables from a Query Context. You can access Context Providers for each data source from the step's Query Context tab. For detailed information about query contexts and referencing variables in a query, see [Using Query Contexts in Queries](#).

- When you have finished writing the query, click **Save** to save the step configuration.

Once the Details tab is configured and the query is written, the step can be run. For information about running this step conditionally by setting up an execution condition, see [Defining Execution Conditions](#).

### Important Notes about Managed Models

Though an ontology that is generated in a Direct Load Step is registered in Anzo and is available for viewing in the Model editor, **the model is owned and managed by the data layer that contains the Direct Load Step**. That means any manual changes made to the model outside of the step, such as from the Model editor, will be overwritten any time the graphmart or layer is refreshed or reloaded. **Do not modify generated managed models except by editing (or adding) Direct Load Step queries.**

**There is only one managed model per layer.** If you include multiple Direct Load Steps in the same layer, they will all update the same ontology. This functionality can be useful if you want to align the data and generated model across multiple steps. If you have multiple sources that are not intended to align or update the same model, create separate layers.

**If you delete a layer that includes a managed model, the model is also deleted.** Use caution when referencing a managed model outside of a graphmart. For example, if you create a dataset and reference a managed model when you select the ontology, the reference will break if the data layer that manages the model is deleted.

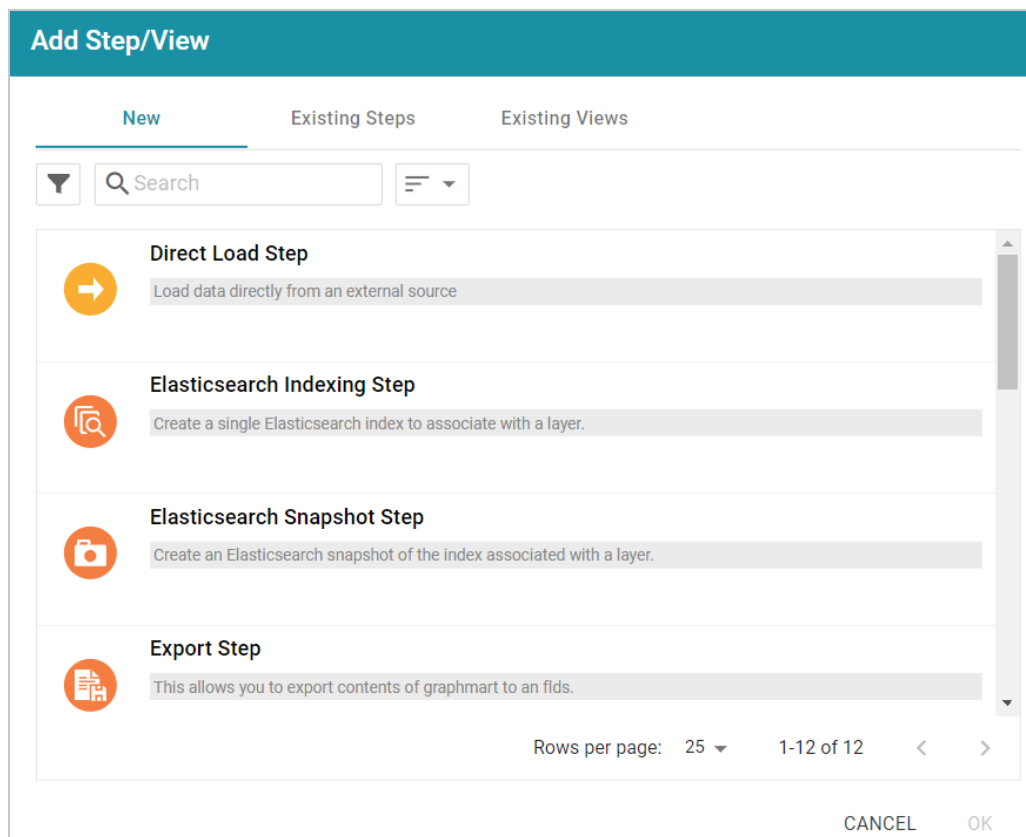
## Creating an Elasticsearch Index (Elasticsearch Indexing Step)

This topic provides guidance on configuring an Elasticsearch Indexing Step that generates an Elasticsearch index from data in a graphmart. Follow the steps below to create an Elasticsearch Indexing Step.

### Tip

For an overview on creating an Elasticsearch index and snapshot from a graphmart, see [Creating an Elasticsearch Index from Graphmart Data](#).

1. Go to the graphmart for which you want to add a step and then click the **Data Layers** tab.
2. On the Data Layers tab, find the layer that you want to add the step to. Click the menu icon (⋮) for that layer and select **Add Step/View**. The Add Step/View dialog box is displayed with the New tab selected.



3. To create a new Elasticsearch Indexing step, select **Elasticsearch Indexing Step** and then click **OK**. If you want to clone an existing step, click the **Existing Steps** tab, select the step that you want to clone, and then click **OK**. Anzo creates or clones the step.

If this is the first Elasticsearch-related step in the layer, the following warning is displayed:

## Warning

Adding this step to this layer will enable this Layer's ES Indexing with a default configuration. You can change that configuration at the layer level.

OK

The warning is to inform you that the Elasticsearch index-related configuration settings at the layer level are automatically adjusted to the default settings for layers that contain Elasticsearch Indexing or Snapshot Steps. Those settings are on the layer's Elasticsearch Indexing tab. The image below shows the tab with the default settings:

Details

Execution Condition

Data Access

Query Context

Elasticsearch Indexing

☒ Associate an Elasticsearch index with this layer

**Warning:** These are advanced settings. Proceed with caution. Changes may significantly affect Elasticsearch behavior.

Advanced ▲

Elasticsearch Index Name

Name will be system generated if left blank

Name will be system generated if left blank

☒ Clear Elasticsearch index before this layer executes ☒ Delete Elasticsearch Index on Layer Unload

Elasticsearch Index Settings

Settings (JSON format) for Elasticsearch index

Elasticsearch Index Mapping

Mapping (JSON format) for Elasticsearch index

CANCEL

SAVE

4. Click **OK** to close the warning, if necessary. The Details tab is displayed:

The screenshot shows a configuration window with four tabs: 'Details', 'Query', 'Execution Condition', and 'Query Context'. The 'Details' tab is active. It contains the following elements:

- Title \***: A text input field with the placeholder text 'The title of the step'.
- Description**: A text input field with the placeholder text 'A brief description of the step'.
- Enabled**: A checkbox that is checked, with the label 'Enabled'.
- Source**: A dropdown menu showing 'Self' with a close button (x) and a dropdown arrow (v). Below it is the text 'Source data to act upon'.
- Data models**: A dropdown menu showing 'Data models' with a dropdown arrow (v). Below it is the text 'Associated data models'.
- Buttons**: 'CANCEL' and 'SAVE' buttons at the bottom right.

5. On the Details tab, configure the following options as needed:

- **Title**: The required name of the step.
- **Description**: An optional short description of the step.
- **Enabled**: When creating a new step, the Enabled option is selected by default, indicating that the step is enabled and will run when the layer is loaded or refreshed. If you want to disable the step so that it is not processed, clear the **Enabled** checkbox.
- **Source**: The source data that this step should act upon. Steps can build upon the data generated by steps in other layers or can be self-contained, applying changes that relate only to the data defined in the layer that contains this step. You can select any number of the following options:
  - **Self**: This option is selected by default and means that the step runs against only the data that is generated in the layer this step belongs to.
  - **All Previous Layers Within Graphmart**: This option means that the step runs against the data that is generated by all of the successful layers that precede the layer this step is in. Any failed layers are ignored.
  - **Previous Layer Within Graphmart**: This option means that the query runs against only the data that is generated by the one layer that precedes the layer this step is in.



- **Layer Name:** The Source drop-down list also includes options for specific layer names.

You can choose a specific layer to act upon the data in that layer only.

- **Data models:** This optional field specifies the model or models to associate with this step. The list displays all of the available models. By default, the field is set to **Exclude System Data** (↕). If you want to choose a system model, click the toggle button on the right side of the field to change it to **Include System Data** (×). When system data is included, the drop-down list displays the system models in addition to the user-generated models.

6. When you have finished configuring the Details tab, click the **Query** tab. This tab contains the Elasticsearch indexing query to run. The template includes the syntax for writing a SPARQL SELECT query that uses the Graph Data Interface (GDI) to generate the index on the Elasticsearch server. Unlike queries for other steps, this step does not run an INSERT query because the data is not being inserted into AnzoGraph. For general information about writing GDI queries, see [Getting Started with GDI Queries](#).

#### Note

Do not include Elasticsearch connection or index parameters in the query. Anzo automatically populates that information from the AnzoGraph configuration when the step is run.



7. Click **Save** to save the step configuration.

Once the Details tab is configured and the query is written, the step can be run. For information about running this step conditionally by setting up an execution condition, see [Defining Execution Conditions](#).

## Related Topics

[Creating an Elasticsearch Index from Graphmart Data](#)

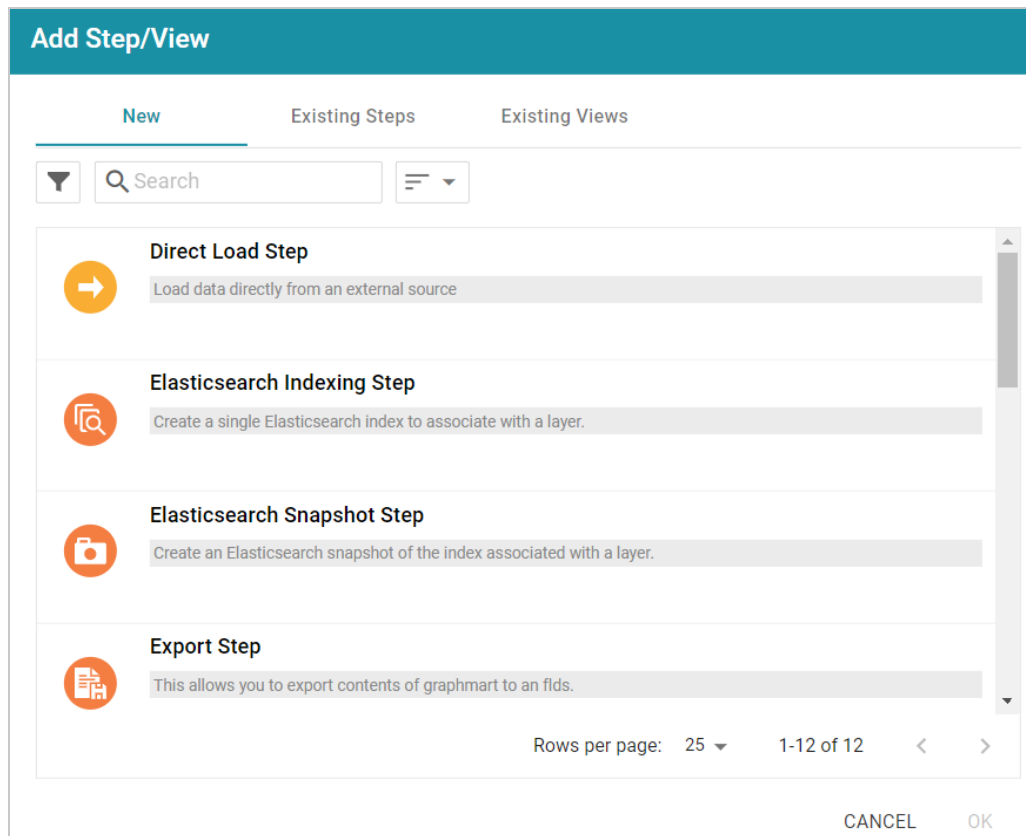
### Taking a Snapshot of an Elasticsearch Index (Elasticsearch Snapshot Step)

This topic provides guidance on configuring an Elasticsearch Snapshot Step that generates a snapshot of an Elasticsearch index that was created from a graphmart. Follow the steps below to create an Elasticsearch Snapshot Step.

### Tip

For an overview on creating an Elasticsearch index and snapshot from a graphmart, see [Creating an Elasticsearch Index from Graphmart Data](#).

1. Go to the graphmart for which you want to add a step and then click the **Data Layers** tab.
2. On the Data Layers tab, find the layer that you want to add the step to. Click the menu icon (⋮) for that layer and select **Add Step/View**. The Add Step/View dialog box is displayed with the New tab selected.



3. To create a new Elasticsearch Snapshot step, select **Elasticsearch Snapshot Step** and then click **OK**. If you want to clone an existing step, click the **Existing Steps** tab, select the step that you want to clone, and then click **OK**. Anzo creates or clones the step.

If this is the first Elasticsearch-related step in the layer, the following warning is displayed:

### Warning

Adding this step to this layer will enable this Layer's ES Indexing with a default configuration. You can change that configuration at the layer level.

OK

The warning is to inform you that the Elasticsearch index-related configuration settings at the layer level are automatically adjusted to the default settings for layers that contain Elasticsearch Indexing or Snapshot Steps. Those settings are on the layer's Elasticsearch Indexing tab. The image below shows the tab with the default settings:

Details

Execution Condition

Data Access

Query Context

Elasticsearch Indexing

☒ Associate an Elasticsearch index with this layer

**Warning:** These are advanced settings. Proceed with caution. Changes may significantly affect Elasticsearch behavior.

Advanced ▲

Elasticsearch Index Name

Name will be system generated if left blank

Name will be system generated if left blank

☒ Clear Elasticsearch index before this layer executes ☒ Delete Elasticsearch Index on Layer Unload

Elasticsearch Index Settings

Settings (JSON format) for Elasticsearch index

Elasticsearch Index Mapping

Mapping (JSON format) for Elasticsearch index

CANCEL

SAVE

4. Click **OK** to close the warning, if necessary. The Details tab is displayed:

Details Execution Condition Query Context

Title \*

The title of the step

Description

A brief description of the step

☒ Enabled

Source

Self x All Previous Layers Within Graphmart x

This option cannot be changed.

Target FLDS \*

FLDS only.

CANCEL SAVE

5. On the Details tab, configure the following options as needed:

- **Title:** The required name of the step.
- **Description:** An optional short description of the step.
- **Enabled:** When creating a new step, the Enabled option is selected by default, indicating that the step is enabled and will run when the layer is loaded or refreshed. If you want to disable the step so that it is not processed, clear the **Enabled** checkbox.
- **Source:** The source data that this step should act upon. Steps can build upon the data generated by steps in other layers or can be self-contained, applying changes that relate only to the data defined in the layer that contains this step. You can select any number of the following options:
  - **Self:** This option is selected by default and means that the step runs against only the data that is generated in the layer this step belongs to.
  - **All Previous Layers Within Graphmart:** This option means that the step runs against the data that is generated by all of the successful layers that precede the layer this step is in. Any failed layers are ignored.
  - **Previous Layer Within Graphmart:** This option means that the query runs against only the data that is generated by the one layer that precedes the layer this step is in.

- **Layer Name:** The Source drop-down list also includes options for specific layer names. You can choose a specific layer to act upon the data in that layer only.
- **Target FLDS:** This is the target FLDS to save the snapshot to. Typically this is the FLDS that was created by an Export Step in a previous layer. If an FLDS does not exist, you can select - **Create New**- to create an empty dataset. See [Adding an Empty Dataset for an Export Step](#) for instructions.

6. Click **Save** to save the step configuration.

Once the Details tab is configured, the step can be run. For information about running this step conditionally by setting up an execution condition, see [Defining Execution Conditions](#).

## Related Topics

[Creating an Elasticsearch Index from Graphmart Data](#)

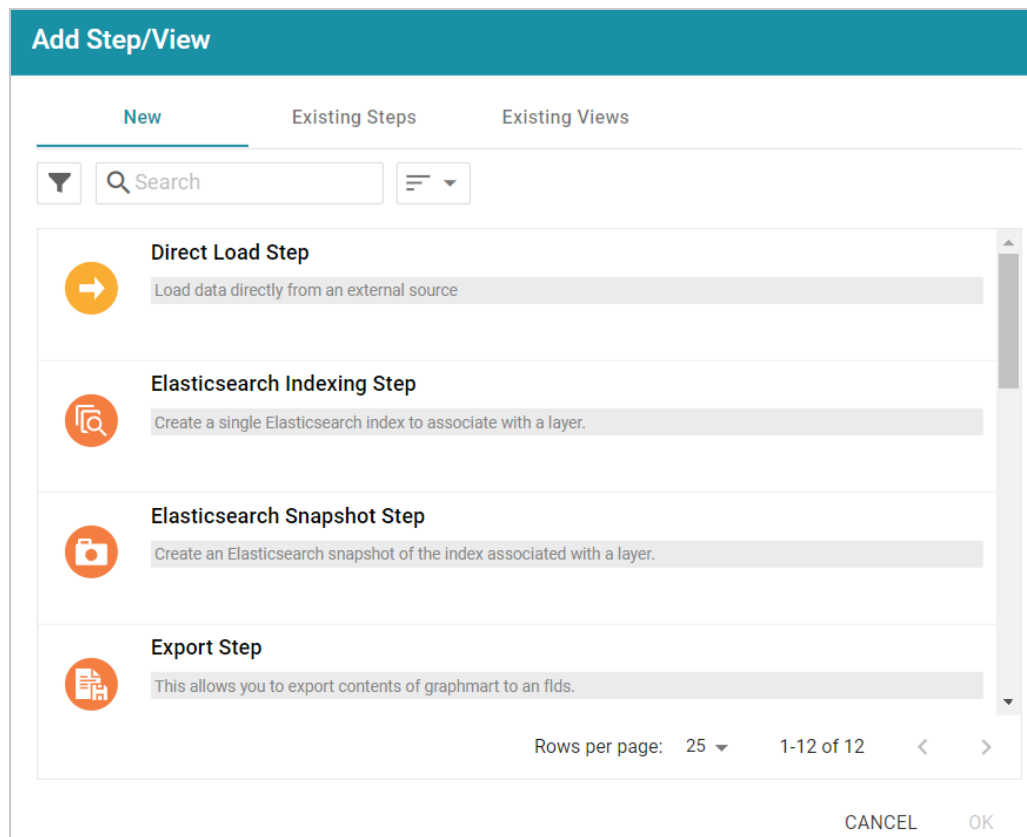
### Exporting Data to an FLDS (Export Step)

This topic provides guidance on configuring an Export Step to use for exporting the knowledge graphs in memory to a file-based linked data set (FLDS) on the shared file store. Follow the steps below to create an Export Step.

#### Important

If you add an Export Step to a graphmart that has been activated, you must reload the entire graphmart after adding the step. Simply refreshing the layer or graphmart after adding the step does not create the ontology graph that the Export Step requires.

1. Go to the graphmart for which you want to add a step and then click the **Data Layers** tab.
2. On the Data Layers tab, find the layer that you want to add the step to. Click the menu icon (⋮) for that layer and select **Add Step/View**. The Add Step/View dialog box is displayed with the New tab selected.



3. To create a new Export step, select **Export Step** and then click **OK**. If you want to clone an existing step, click the **Existing Steps** tab, select the step that you want to clone, and then click **OK**. Anzo creates or clones the step and displays the Details tab:

Details
Elastic Search
Execution Condition
Query Context

Title \*
The title of the step

Description
A brief description of the step

☒ Enabled

Source
Self x
Source data to act upon

Data models
Associated data models

CANCEL
SAVE

4. On the Details tab, configure the following options as needed:

- **Title:** The required name of the step.
- **Description:** An optional short description of the step.
- **Enabled:** When creating a new step, the Enabled option is selected by default, indicating that the step is enabled and will run when the layer is loaded or refreshed. If you want to disable the step so that it is not processed, clear the **Enabled** checkbox.
- **Source:** The source data that this step should act upon. Steps can build upon the data generated by steps in other layers or can be self-contained, applying changes that relate only to the data defined in the layer that contains this step. You can select any number of the following options:
  - **Self:** This option is selected by default and means that the step runs against only the data that is generated in the layer this step belongs to.
  - **All Previous Layers Within Graphmart:** This option means that the step runs against the data that is generated by all of the successful layers that precede the layer this step is in. Any failed layers are ignored.
  - **Previous Layer Within Graphmart:** This option means that the query runs against only the data that is generated by the one layer that precedes the layer this step is in.



- **Layer Name:** The Source drop-down list also includes options for specific layer names.  
You can choose a specific layer to act upon the data in that layer only.
- **Data models:** This optional field specifies the model or models to associate with this step. The list displays all of the available models. By default, the field is set to **Exclude System Data** (↕). If you want to choose a system model, click the toggle button on the right side of the field to change it to **Include System Data** (×). When system data is included, the drop-down list displays the system models in addition to the user-generated models.
- **Target FLDS:** This is the target FLDS for this export. If an FLDS does not exist, you can select - **Create New**- to create an empty dataset. See [Adding an Empty Dataset for an Export Step](#) for instructions. If you select an existing target FLDS, you also have the option to specify whether or not to overwrite the existing dataset.
- **Overwrite FLDS:** This setting controls whether the existing FLDS is replaced with the exported files each time the Export Step runs or whether the exported files are added to the existing FLDS:
  - If you want Anzo to replace the current edition of the dataset, select the **Overwrite FLDS** checkbox. When Overwrite FLDS is enabled, Anzo archives the existing files in a new timestamped **export** subdirectory directory under the Target FLDS directory. Each time the Export step runs, Anzo archives the current edition, and creates a new export directory. If you add this dataset to a graphmart, only the latest version of the exported data will be loaded to AnzoGraph.
  - If you want Anzo to add the exported files to the existing FLDS, leave the **Overwrite FLDS** checkbox unchecked. When Overwrite FLDS is disabled, Anzo adds all of the exported components to a **cumulative export** directory under the Target FLDS directory. The dataset will contain the original files as well as all cumulative working editions. If you subsequently add this dataset to a graphmart, all of the data from all of the subdirectories will be loaded into AnzoGraph.
- **Export Binary Store Contents:** This option applies to exports of unstructured graphmarts and controls whether the binary store is exported along with the data.
- **Always Move Binary Store:** This option also applies to exports of unstructured graphmarts and controls whether the binary store is moved or copied during the export. Since the binary store can be large and have a nested structure, copying the data can take a very long time. Since moving the binary store is almost instantaneous, however, enabling **Always Move Binary Store** can reduce the time it takes to complete the export.

- **Export Elasticsearch Contents:** This option is enabled by default and controls whether Elasticsearch contents, such as associated indexes, are copied to disk in the Target FLDS. If you do not want Elasticsearch contents to be exported to the FLDS, clear the checkbox.

#### Note

When exporting data from an AnzoGraph instance that does not have an associated Elasticsearch connection, clear the **Export Elasticsearch Contents** checkbox.

- **Keep Elasticsearch Index Online:** This option controls whether the Elasticsearch index that is associated with the dataset remains stored in Elasticsearch or is removed from Elasticsearch once it is exported.

#### Tip

For information about advanced Elasticsearch options that are available for Export Steps, see [Elastic Search Tab](#) below.

- **Generate Metrics:** This option controls whether a data profile is generated before the data is exported. Enabling this option increases the time it takes to run the step, but enabling it ensures that the information on the Dataset Explore is complete if the dataset is viewed. If you load the exported files in the future, the data profile is also loaded.
- **Do Not Create New Edition in Dataset on Export:** This option controls whether or not a new edition is created for the dataset each time the Export Step is run. By default this option is disabled, which means each export results in a new edition. If you do not want a new edition to be created on each export, select the **Do Not Create New Edition in Dataset on Export** checkbox.
- **Maximum Number of Components in Edition:** This option controls the maximum number of components to retain in an edition. The default value is blank, which means unlimited. If you specify a number in this field and the limit is reached, Anzo ages off the oldest components as new ones are created.

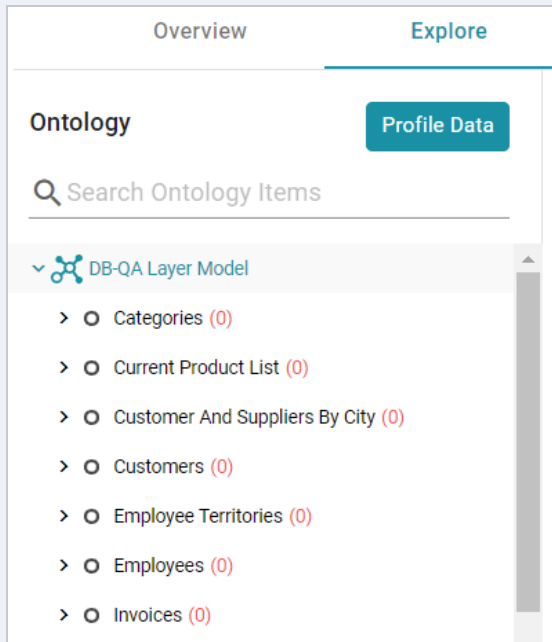
5. Click **Save** to save the step configuration.

Once the Details tab is configured, the step can be run. If you added the step to an active graphmart, make sure that you deactivate and then reactivate the graphmart. Simply refreshing the layer or graphmart after adding the step does not create the ontology graph that the Export Step requires. For information about running this step conditionally by setting up an execution condition, see [Defining Execution Conditions](#).

Once the graphmart is reactivated, the new dataset becomes available in the Datasets catalog.

## Note

Unless you enable **Generate Metrics**, when you view in the Datasets catalog a dataset that was created by an Export Step, the counts in the Ontology panel on the Explore tab remain **0** until a Data Profile is generated for the dataset. For information about generating a profile, see [Generating a Dataset Data Profile](#). For example, the image below shows the Explore tab for a dataset that was created by an Export Step and has not been profiled.



## Elastic Search Tab

The **Elastic Search** tab contains optional settings that you can use to set any desired limits on Elasticsearch indexing processes.

Details

Elastic Search

Execution Condition

Query Context

Elasticsearch Index Settings

Elasticsearch Index Settings

Elasticsearch Bulk Size

Elasticsearch Bulk Size (max batch size in MB)

Elasticsearch Bulk Actions

Elasticsearch Bulk Actions (max number of documents per batch)

Elasticsearch Bulk Concurrent Requests

Elasticsearch Bulk Concurrent Requests (max number of concurrent bulk requests)

Elasticsearch Bulk Max Threads Per FLDS

Elasticsearch Bulk Max Threads Per FLDS (max number threads to use for indexing per FLDS)

Elasticsearch Bulk Max FLDS Threads

Elasticsearch Bulk Max FLDS Threads (max number FLDSes to index concurrently)

CANCEL

SAVE

## Elasticsearch Index Settings

A custom JSON list of any Elasticsearch index settings that you want to apply to the export.

## Elasticsearch Bulk Size

The maximum batch size in MB.

## Elasticsearch Bulk Actions

The maximum number of documents to include in each batch.

## Elasticsearch Bulk Concurrent Requests

The maximum number of bulk requests that can run concurrently.

## Elasticsearch Bulk Max Threads Per FLDS

The maximum number of threads to use for indexing per file-backed linked data set (FLDS).

## Elasticsearch Bulk Max FLDS Threads

The maximum number of FLDSes to index concurrently.

### Loading a Dataset from the Catalog (Load Dataset Step)

This topic provides guidance on configuring a Load Dataset Step to use for adding a dataset from the Datasets catalog to a graphmart. Follow the steps below to create a Load Dataset Step.

1. Go to the graphmart for which you want to add a step and then click the **Data Layers** tab.
2. On the Data Layers tab, find the layer that you want to add the step to. Click the menu icon (⋮) for that layer and select **Add Step/View**. The Add Step/View dialog box is displayed with the New tab selected.

Add Step/View

NewExisting StepsExisting Views

⌵

Q Search

⌵

➔

Direct Load Step

Load data directly from an external source

🔍

Elasticsearch Indexing Step

Create a single Elasticsearch index to associate with a layer.

📷

Elasticsearch Snapshot Step

Create an Elasticsearch snapshot of the index associated with a layer.

📄

Export Step

This allows you to export contents of graphmart to an flds.

Rows per page: 251-12 of 12<>

CANCELOK

Loading a Dataset from the Catalog (Load Dataset Step)

609

3. Select **Load Dataset Step** and then click **OK**. Anzo creates the step and displays the Details tab:

**Create**

Details Execution Condition Filter Query Context

Enabled

Title \*

The title of the step

Description

A brief description of the step

Linked Dataset \*

Linked Dataset to load

☐ Watch Flds Directory

☐ Ignore Missing File or Directory

☐ Skip Elastic Search Snapshot Restoration if Index Already Exists

CANCEL SAVE

4. On the Details tab, configure the following options as needed:

- **Title:** The required name of the step.
- **Description:** An optional short description of the step.
- **Enabled:** When creating a new step, the Enabled option is selected by default, indicating that the step is enabled and will run when the layer is loaded or refreshed. If you want to disable the step so that it is not processed, slide the **Enabled** slider to the left.
- **Linked Dataset:** This field specifies the dataset to load. The list displays all of the datasets in the Dataset catalog. By default, the field is set to **Exclude System Data** (🔒). If you want to choose a system dataset, click the toggle button on the right side of the field to change it to **Include System Data** (🔓). When you select a dataset, the current working edition (Managed Edition) of the dataset is selected as the data to load. If you want to change the edition, you can click **Modify Edition** and follow the steps in [Modifying an Edition](#).
- **Watch FLDS Directory:** This option controls whether the FLDS directory is monitored for changes. If Watch FLDS Directory is enabled and changes to the files in the FLDS directory are detected, Anzo will mark this step (and layer) as needing a refresh.

- **Ignore Missing File or Directory:** This option controls whether to ignore missing files or subdirectories in the FLDS directory and proceed with the load or fail the step if files or directories are missing.
- **Skip Elastic Search Snapshot Restoration if Index Already Exists:** This option applies to graphmarts with Elasticsearch indexes and controls whether Anzo first checks to see if an index with the alias for the dataset already exists in Elasticsearch. If this setting is enabled and the index does exist, Anzo will not reload the index snapshot into Elasticsearch.

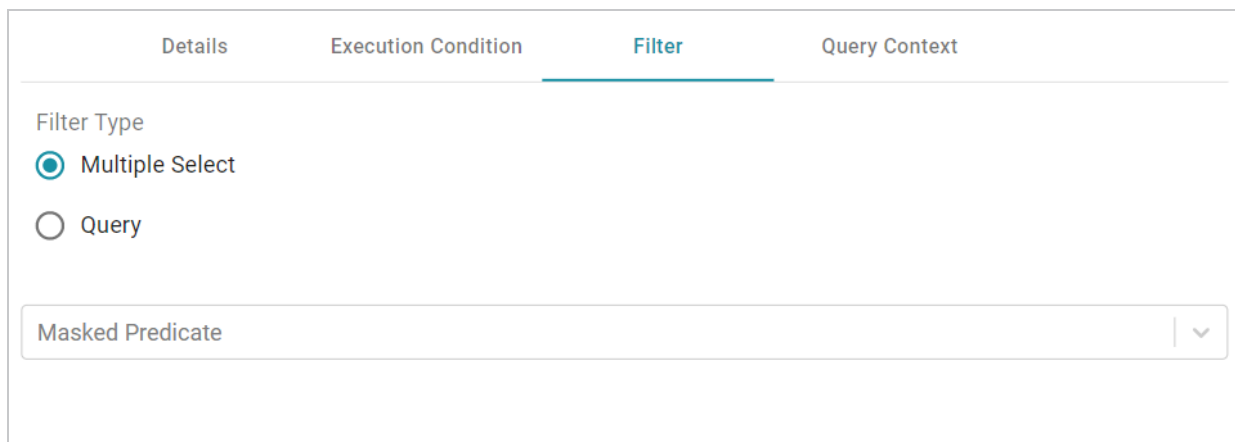
5. Typically when users add a dataset to a graphmart, they want to load the entire dataset. However, if you are familiar with the data and want to exclude certain predicates from the dataset or write an INSERT query that filters the data, you can configure filtering options on the Filter tab. For information, see [Filter Tab](#) below.

6. Click **Save** to save the step configuration.

Once the Details tab is configured, the step can be run. For information about running this step conditionally by setting up an execution condition, see [Defining Execution Conditions](#).

## Filter Tab

The **Filter** tab includes options for filtering out some of the data in the dataset. If you want to load all of the statements in the dataset, do not configure Filter options. If you want to exclude some statements, configure the Filter options.



The screenshot shows the 'Filter' tab selected in a configuration interface. At the top, there are four tabs: 'Details', 'Execution Condition', 'Filter' (which is active and underlined), and 'Query Context'. Below the tabs, the 'Filter Type' section has two radio buttons: 'Multiple Select' (which is selected) and 'Query'. Below this is a 'Masked Predicate' dropdown menu with a downward arrow on the right.

## Multiple Select

This option enables you to exclude certain triples from the load by selecting the predicates to filter out.

These are known as **Masked Predicates**. To exclude predicates, select the **Multiple Select** radio button,

then click the **Masked Predicate** drop-down list and select a predicate to add it to the Masked Predicate field. Click the field again to select additional predicates. You can remove a property from the masked list by clicking the X next to the predicate name.

## Query

If you want to hand-pick the data to load, you can use this option to write a SPARQL query that inserts specific values or filters out certain values. To write a query, select the **Query** radio button, and then type an INSERT query in the text box. For example, you can use the following format to filter out properties from the files:

```
INSERT {  
  GRAPH ${targetGraph}{  
    ?s ?p ?o.  
  }  
}  
${usingSources}  
WHERE {  
  ?s ?p ?o .  
  FILTER EXISTS { ?s a ?type . }  
  FILTER(?type = <URI>)  
}
```

### Note

Including the **\${targetGraph}** and **\${usingSources}** parameters are required. Anzo automatically populates the query with the appropriate graph URIs when the step is run.

### Important

In load filter queries, URIs are not supported in the object position. To specify a URI as an object, include the standard `?s ?p ?o` triple pattern in the WHERE clause and then apply FILTER statements with URIs as needed. URIs are supported in the subject or predicate position.

For example, the following query filters the data in a sample dataset that includes information about people and the events they buy tickets for. The WHERE clause filters the data to load only the triples that are related to person1 (personid=1):

```
INSERT { GRAPH ${targetGraph} {  
  ?s ?p ?o  
}
```

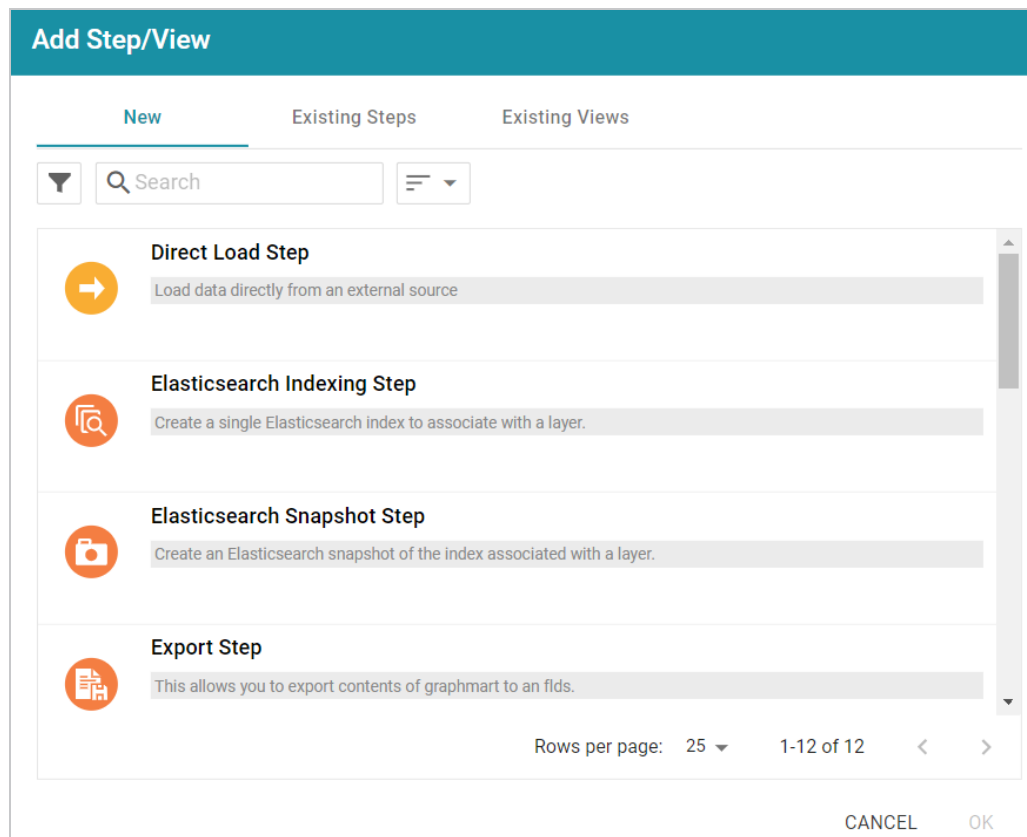


```
}
${usingSources}
WHERE {
  ?s ?p ?o ;
  <http://cambridgesemantics.com/ont/autogen/c89d/Tickets#ticket_users_
personid> ?id .
  FILTER (?id=1)
}
```

## Pre-Compiling a Query (Pre-Compile Query Step)

The first time a user runs an analytic query against AnzoGraph, AnzoGraph performs a code compilation process to generate the code for running that query. It then executes the query using that compiled code, and the same code is reused for subsequent runs of the query. If you determine that a particular query has a long code compilation time, you can add that query to a Pre-Compile Query Step. That way the query is run during the graphmart load and the compiled code is available before an end-user runs that query. Follow the steps below to create a Pre-Compile Query Step.

1. Go to the graphmart for which you want to add a step and then click the **Data Layers** tab.
2. On the Data Layers tab, find the layer that you want to add the step to. Click the menu icon (⋮) for that layer and select **Add Step/View**. The Add Step/View dialog box is displayed with the New tab selected.



3. To create a new Pre-Compile Query Step, select **Pre-Compile Query Step** and then click **OK**. If you want to clone an existing step, click the **Existing Steps** tab, select the step that you want to clone, and then click **OK**. Anzo creates or clones the step and displays the Details tab:

Details
Query
Execution Condition
Query Context

Title \*

The title of the step

Description

A brief description of the step

☒ Enabled
☐ If the precompile query fails, the layer will be marked as failed.

☐ If the precompile query fails, the whole graphmart will be marked as failed.

Source

Self x

Source data to act upon

☐ Pre-Run Generate Statistics

CANCEL
SAVE

4. On the Details tab, configure the following options as needed:

- **Title:** The required name of the step.
- **Description:** An optional short description of the step.
- **Enabled:** When creating a new step, the Enabled option is selected by default, indicating that the step is enabled and will run when the layer is loaded or refreshed. If you want to disable the step so that it is not processed, clear the **Enabled** checkbox.
- **If the precompile query fails, the layer will be marked as failed:** This option controls whether loading the data layer that contains this step is completed or aborted if this step fails. Select this option if you want Anzo to fail the layer if this step fails.
- **If the precompile query fails, the whole graphmart will be marked as failed:** This option controls whether loading the graphmart is completed or aborted if this step fails. Select this option if you want Anzo to fail the entire graphmart if this step fails.
- **Source:** The source data that this step should act upon. Steps can build upon the data generated by steps in other layers or can be self-contained, applying changes that relate only to the data defined in the layer that contains this step. You can select any number of the following options:
  - **Self:** This option is selected by default and means that the step runs against only the data that is generated in the layer this step belongs to.

- **All Previous Layers Within Graphmart:** This option means that the step runs against the data that is generated by all of the successful layers that precede the layer this step is in. Any failed layers are ignored.
  - **Previous Layer Within Graphmart:** This option means that the query runs against only the data that is generated by the one layer that precedes the layer this step is in.
  - **Layer Name:** The Source drop-down list also includes options for specific layer names. You can choose a specific layer to act upon the data in that layer only.
  - **Pre-Run Generate Statistics:** This option controls whether to initiate AnzoGraph's internal statistics gathering queries before running the query to pre-compile. The statistics gathering helps ensure that the AnzoGraph query planner generates ideal query execution plans for queries that are run against the graphmart.
5. When you have finished configuring the Details tab, click the **Query** tab. This tab contains the query to pre-compile. The tab provides a template for writing a SPARQL SELECT query. You can edit the template to write the query, or you can paste in query contents that you copied from a log file, dashboard, the Query Builder, etc.

#### Note

Make sure that you include the `${fromSources}` parameter in the query. Anzo automatically populates the query with the appropriate source graph URIs according to the Source configured from the Details tab.

Details
Query
Execution Condition
Query Context

Precompile query \*

```

1
2 #fromSources is replaced with the URIs of the Layer's Sources at runtime
3 SELECT *
4 ${fromSources}
5 WHERE{
6 }

```

Query used to pre-compile

CANCEL
SAVE

### Note

If your query connects to a source that requires input of connection and authorization information, Cambridge Semantics recommends that you do not include the connection and authorization values directly in the query. Instead, replace those values with Context Variables from a Query Context. You can access Context Providers for each data source from the step's Query Context tab. For detailed information about query contexts and referencing variables in a query, see [Using Query Contexts in Queries](#).

- Click **Save** to save the step configuration.

Once the Details tab is configured and the query is written, the step can be run. For information about running this step conditionally by setting up an execution condition, see [Defining Execution Conditions](#).

## Related Topics

[SPARQL Best Practices and Query Templates](#)

### Creating a Reusable Query Template

There are two types of steps that enable you to create a query template that is reusable across data layers and graphmarts:

## Templated Step

Templated Steps use user-defined key-value pairs. The keys are represented by parameters in the query. Creating the key-value pairs requires familiarity with the data and properties defined in the model. When this step is reused, users do not need to rewrite the query; they modify the values for the keys.

### Query-Driven Templated Step

Query-Driven Templated Steps are similar to Templated Steps in that they provide a way to create query templates that use parameters to represent key-value pairs. The queries are reusable across datasets because the existing parameters can be substituted for alternate key-value pairs. The difference between the two types of steps is that the key-value pairs for Templated Steps are user-defined. In Query-Driven Templated Steps, a parameter query is run that automatically generates the key-value pairs. Then the defined template query is run for each key-value solution from the parameter query.

This section includes instructions for creating both types of templated steps:

- [Creating a Template with a Templated Step](#)
- [Creating a Template with a Query-Driven Templated Step](#)

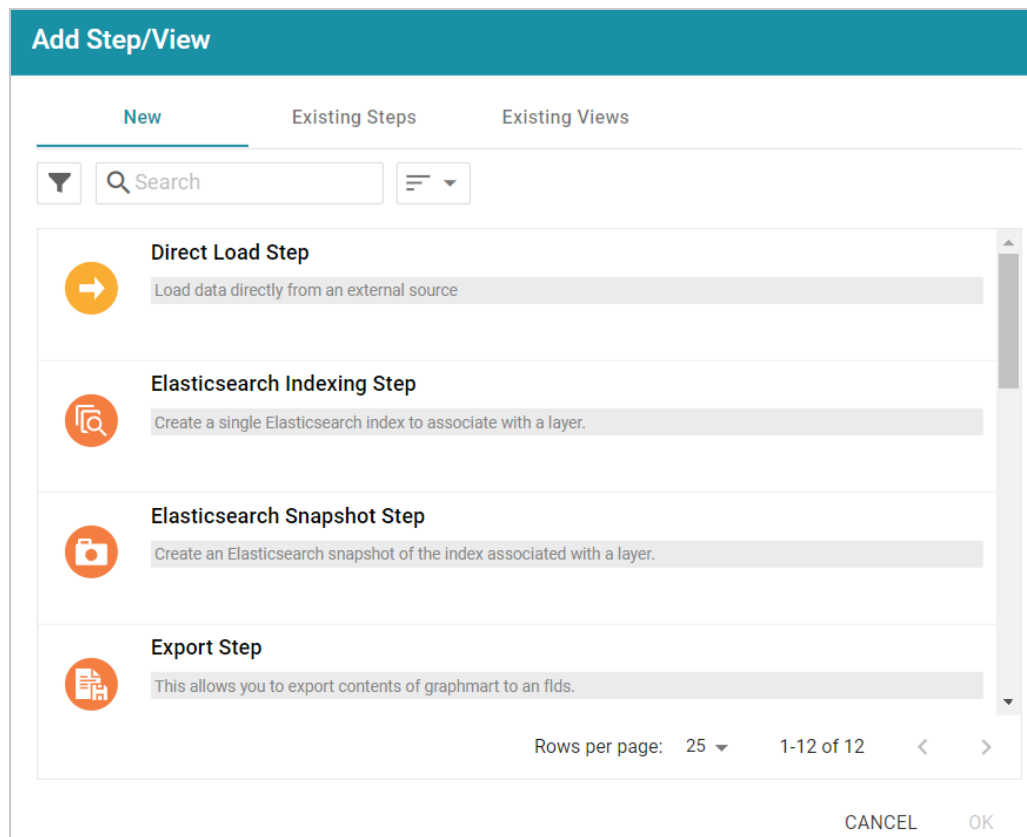
### Creating a Template with a Templated Step

Templated Steps use user-defined key-value pairs. The keys are represented by parameters in the query. When this step is reused, users do not need to rewrite the query; they modify the values for the keys. Follow the steps below to create a Templated Step.

#### Tip

This type of template step uses key-value pairs that are user-defined. Creating the key-value pairs requires familiarity with the data and properties defined in the model. To create a query template that enables you to run a query and automatically generate the key-value pairs, see [Creating a Template with a Query-Driven Templated Step](#).

1. Go to the graphmart for which you want to add a step and then click the **Data Layers** tab.
2. On the Data Layers tab, find the layer that you want to add the step to. Click the menu icon (⋮) for that layer and select **Add Step/View**. The Add Step/View dialog box is displayed with the New tab selected.



3. To create a new Templated step, select **Templated Step** and then click **OK**. If you want to clone an existing step, click the **Existing Steps** tab, select the step that you want to clone, and then click **OK**. Anzo creates or clones the step and displays the Details tab:

Details
Template
Execution Condition
Query Context

Title \*

The title of the step

Description

A brief description of the step

☒ Enabled

Source

Self x v

Source data to act upon

Data models \*

Associated data models

CANCEL
SAVE

4. On the Details tab, configure the following options as needed:
- **Title:** The required name of the step.
  - **Description:** An optional short description of the step.
  - **Enabled:** When creating a new step, the Enabled option is selected by default, indicating that the step is enabled and will run when the layer is loaded or refreshed. If you want to disable the step so that it is not processed, clear the **Enabled** checkbox.
  - **Source:** The source data that this step should act upon. Steps can build upon the data generated by steps in other layers or can be self-contained, applying changes that relate only to the data defined in the layer that contains this step. You can select any number of the following options:
    - **Self:** This option is selected by default and means that the step runs against only the data that is generated in the layer this step belongs to.
    - **All Previous Layers Within Graphmart:** This option means that the step runs against the data that is generated by all of the successful layers that precede the layer this step is in. Any failed layers are ignored.
    - **Previous Layer Within Graphmart:** This option means that the query runs against only the data that is generated by the one layer that precedes the layer this step is in.



- **Layer Name:** The Source drop-down list also includes options for specific layer names.



You can choose a specific layer to act upon the data in that layer only.

- **Data models:** This required field specifies the model or models to associate with this step. The list displays all of the available models. By default, the field is set to **Exclude System Data** (↕). If you want to choose a system model, click the toggle button on the right side of the field to change it to **Include System Data** (×). When system data is included, the drop-down list displays the system models in addition to the user-generated models.
- **Pre-Run Generate Statistics:** This option controls whether to initiate AnzoGraph's internal statistics gathering queries before running the query to pre-compile. The statistics gathering helps ensure that the AnzoGraph query planner generates ideal query execution plans for queries that are run against the graphmart.

5. When you have finished configuring the Details tab, click the **Template** tab. This tab defines the template query and the key-value pairs.

Details   **Template**   Execution Condition   Query Context



How to add a key to the template:  
 1. Add the key to your query using the syntax: `${key_name}`  
 2. Fill in the associated value in the key/value table.

Query    Key/Value Table    Preview   Refresh Keys

Transformation Template \*

```

1
2 #targetGraph is replaced with the Layers
3 #usingSources is replaced with the URIs
4 of the Layer's Sources at runtime
5 DELETE{
6     GRAPH ${targetGraph}{
7     }
8 }
9 INSERT{
10    GRAPH ${targetGraph}{
11    }
12 }
13 }
14 ${usingSources}
  
```

Search keys...    Search values... 

No parameters found

CANCEL   SAVE

6. On the left side of the screen create the query template. The default template includes the syntax for writing SPARQL INSERT and DELETE queries and includes source and target graph parameters that Anzo replaces at runtime. In the query, include the parameters in the format `${key_name}`. Each parameter will become a key in the Key/Value Table will you click the **Refresh Keys** button. For example, the following INSERT query includes several parameters that represent properties and

functions:

```
INSERT {  
  GRAPH ${targetGraph}{  
    ?lsubj ${linkProperty} ?rsubj  
  }  
}  
${usingSources}  
WHERE {  
  ?lsubj ${sourceProperty} ?lobj .  
  ?rsubj ${targetProperty} ?robj .  
  FILTER (${lFunction} (?lobj) ${operator} ${rFunction} (?robj))  
}
```

#### Note

If your query connects to a source that requires input of connection and authorization information, Cambridge Semantics recommends that you do not include the connection and authorization values directly in the query. Instead, replace those values with Context Variables from a Query Context. You can access Context Providers for each data source from the step's Query Context tab. For detailed information about query contexts and referencing variables in a query, see [Using Query Contexts in Queries](#).

7. Once the template query has been defined, populate the Key/Value Table with the keys from the query by clicking the **Refresh Keys** button. For example, using the example query above, the Key/Value table is populated with the following keys:

Key/Value Table

Preview

Refresh Keys

Search keys...

Search values...

linkProperty	Value	
lFunction	Value	
targetProperty	Value	
rFunction	Value	
operator	Value	
sourceProperty	Value	

8. In each row, specify the desired **Value** for the key. For example, in the image below, the property URI `http://cambridgesemantics.com/ont/autogen/c89d/Tickets#tickit_events_eventid` is specified as the Value for the `linkProperty` key.

Search keys...

Search values...

lFunction	Value	
rFunction	Value	
linkProperty	<code>http://cambridgesemantics.com/ont/autogen/c89d/Tickets#tickit_events_eventid</code>	
operator	Value	

9. Click **Save** to save the step configuration.

Once the Details tab is configured and the template and key-value pairs are defined, the step can be run. For information about running this step conditionally by setting up an execution condition, see [Defining Execution Conditions](#).

## Related Topics

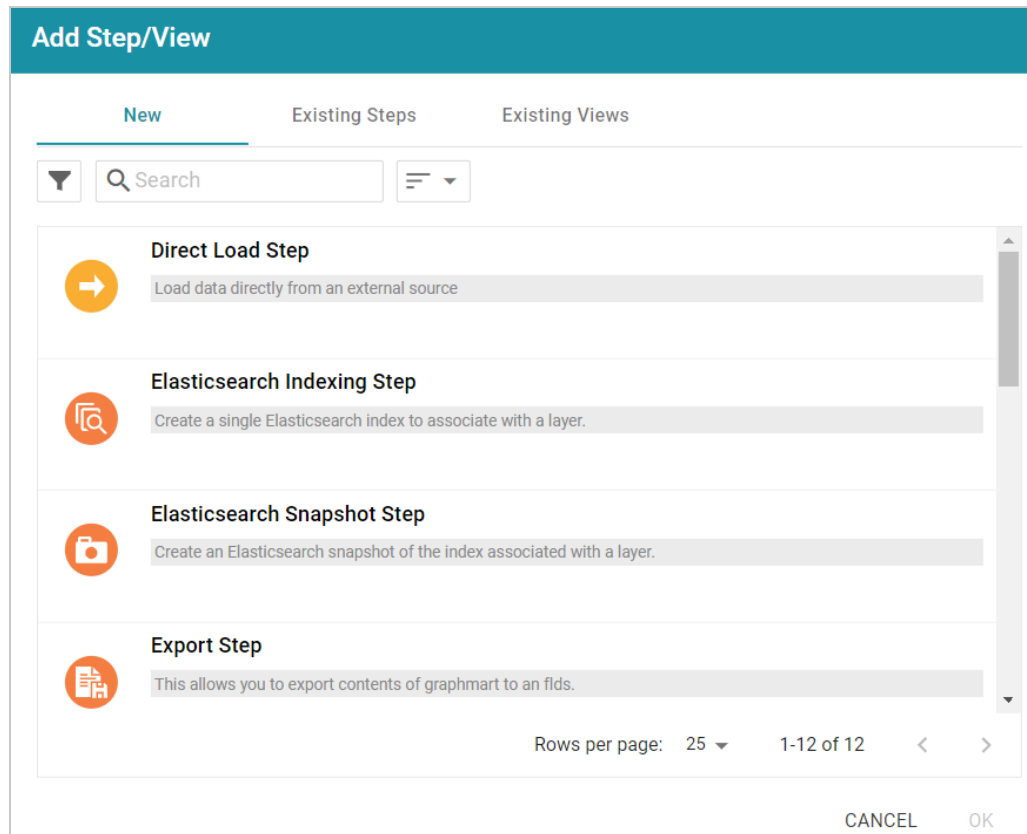
[SPARQL Best Practices and Query Templates](#)

## Creating a Template with a Query-Driven Templated Step

Query-Driven Templated Steps are similar to Templated Steps in that they provide a way to create query templates that use parameters to represent key-value pairs. The queries are reusable across datasets because the existing parameters can be substituted for alternate key-value pairs. The difference between

the two types of steps is that the key-value pairs for Templated Steps are user-defined. In Query-Driven Templated Steps, a parameter query is run that automatically generates the key-value pairs. Then the defined template query is run for each key-value solution from the parameter query. Follow the steps below to create a Query-Driven Templated Step.

1. Go to the graphmart for which you want to add a step and then click the **Data Layers** tab.
2. On the Data Layers tab, find the layer that you want to add the step to. Click the menu icon (⋮) for that layer and select **Add Step/View**. The Add Step/View dialog box is displayed with the New tab selected.



3. To create a new Query-Driven Templated step, select **Query Driven Templated Step** and then click **OK**. If you want to clone an existing step, click the **Existing Steps** tab, select the step that you want to clone, and then click **OK**. Anzo creates or clones the step and displays the Details tab:

The screenshot shows a configuration window with five tabs: Details, Parameters Query, Template, Execution Condition, and Query Context. The 'Details' tab is selected. It contains the following elements:

- Title \***: A text input field with the placeholder text 'The title of the step'.
- Description**: A text input field with the placeholder text 'A brief description of the step'.
- Enabled**: A checkbox that is checked, with the label 'Enabled'.
- Source**: A dropdown menu showing 'Self' as the selected option. There is a small 'x' icon to the right of the dropdown.
- Data models \***: A dropdown menu that is currently empty.
- Associated data models**: A label below the 'Data models' dropdown.
- Buttons**: 'CANCEL' and 'SAVE' buttons at the bottom right.

4. On the Details tab, configure the following options as needed:

- **Title**: The required name of the step.
- **Description**: An optional short description of the step.
- **Enabled**: When creating a new step, the Enabled option is selected by default, indicating that the step is enabled and will run when the layer is loaded or refreshed. If you want to disable the step so that it is not processed, clear the **Enabled** checkbox.
- **Source**: The source data that this step should act upon. Steps can build upon the data generated by steps in other layers or can be self-contained, applying changes that relate only to the data defined in the layer that contains this step. You can select any number of the following options:
  - **Self**: This option is selected by default and means that the step runs against only the data that is generated in the layer this step belongs to.
  - **All Previous Layers Within Graphmart**: This option means that the step runs against the data that is generated by all of the successful layers that precede the layer this step is in. Any failed layers are ignored.
  - **Previous Layer Within Graphmart**: This option means that the query runs against only the data that is generated by the one layer that precedes the layer this step is in.

- **Layer Name:** The Source drop-down list also includes options for specific layer names.

You can choose a specific layer to act upon the data in that layer only.

- **Data models:** This required field specifies the model or models to associate with this step. The list displays all of the available models. By default, the field is set to **Exclude System Data** (↕). If you want to choose a system model, click the toggle button on the right side of the field to change it to **Include System Data** (×). When system data is included, the drop-down list displays the system models in addition to the user-generated models.
- **Pre-Run Generate Statistics:** This option controls whether to initiate AnzoGraph's internal statistics gathering queries before running the query to pre-compile. The statistics gathering helps ensure that the AnzoGraph query planner generates ideal query execution plans for queries that are run against the graphmart.

5. When you have finished configuring the Details tab, click the **Parameters Query** tab. This tab defines the query to use for determining the key-value pairs for the Source that was selected on the Details tab. The tab provides a template for writing a SPARQL SELECT DISTINCT query. Edit the template as needed.

#### Note

Make sure that you include the `${fromSources}` parameter in the query. Anzo automatically populates the query with the appropriate source graph URIs according to the Source configured on the Details tab.

Details	Parameters Query	Template	Execution Condition	Query Context
<p>Template Parameter *</p> <pre> 1  #fromSources is replaced with the URIs of the Layer's Sources at runtime 2  #For each result row, the template query is run with the selected variables as the input bindings 3  SELECT DISTINCT ?param1 ?param2 ?param3 4  \${fromSources} 5  WHERE{ 6      ?param1 ?param2 ?param3. 7  }</pre> <p>Query query used to populate the parameters</p>				
				<p>CANCEL    SAVE</p>

### Note

If your query connects to a source that requires input of connection and authorization information, Cambridge Semantics recommends that you do not include the connection and authorization values directly in the query. Instead, replace those values with Context Variables from a Query Context. You can access Context Providers for each data source from the step's Query Context tab. For detailed information about query contexts and referencing variables in a query, see [Using Query Contexts in Queries](#).

- When you have finished configuring the Parameters Query tab, click the **Template** tab. This tab contains the query to run for each of the key-value pairs identified by the Parameters Query. The template includes the syntax for writing SPARQL DELETE and INSERT queries and includes source and target graph parameters as well as the placeholder parameters from the Parameters Query.

### Note

By default, Anzo uses RDF encoding for parameters, meaning a parameter specified as `${param}` is translated as `${rdf.param}`. If you do not want to use RDF encoding, you can specify plain text by adding `text.` before the parameter name, for example, `${text.param}`.

Details
Parameters Query
**Template**
Execution Condition
Query Context

Transformation Template \*

```

1  #targetGraph is replaced with the Layers URI at runtime
2  #usingSources is replaced with the URIs of the Layer's Sources at runtime
3  #Query is run once per result from the parameters query
4  DELETE{
5      GRAPH ${targetGraph}{
6
7      }
8  }
9  INSERT{
10     GRAPH ${targetGraph}{
11
12     }
13 }
14 ${usingSources}
15 WHERE{
16     ${param1} ${param2} ${param3} .
17 }

```

Query template used to perform the transformation

CANCEL
SAVE

7. Click **Save** to save the step configuration.

Anzo replaces the parameters at runtime. The query is executed  $n$  times, where  $n$  is the number of rows returned by the Parameters Query.

Once the Details tab is configured, the parameters query is in place, and the template is completed, the step can be run. Anzo replaces the parameters at runtime, and the query is executed  $n$  times, where  $n$  is the number of rows returned by the parameters query. For information about running this step conditionally by setting up an execution condition, see [Defining Execution Conditions](#).

## Related Topics

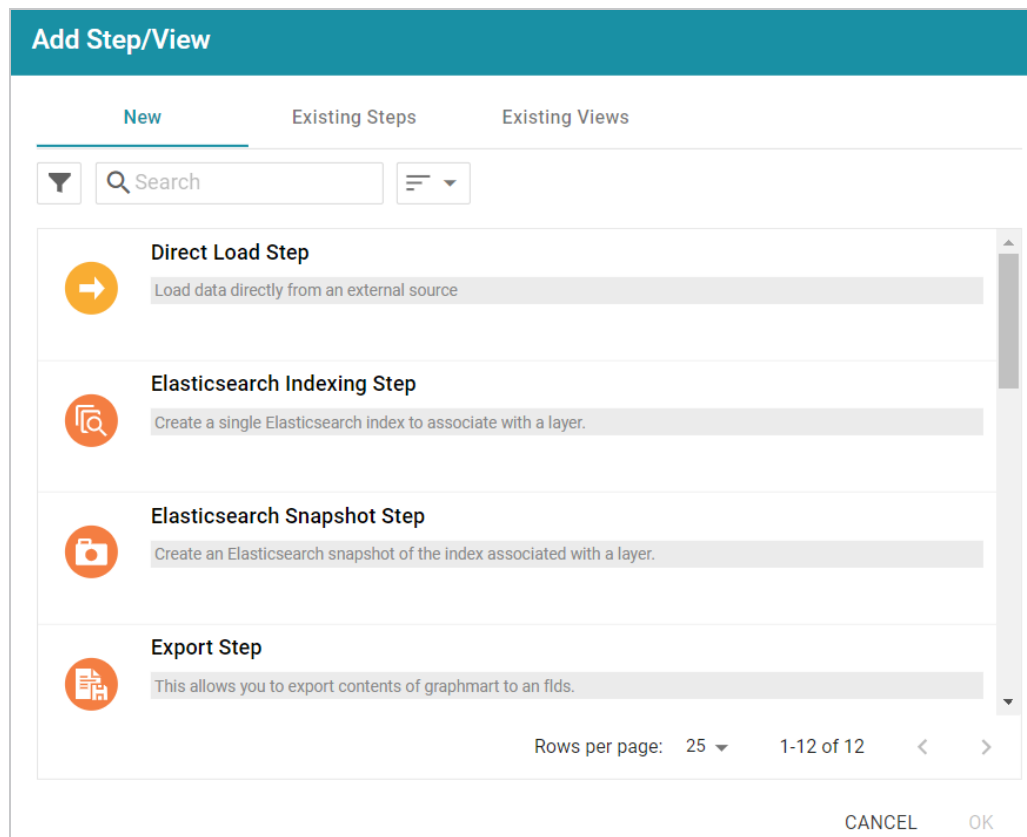
[SPARQL Best Practices and Query Templates](#)

## Running a Transformation Query (Query Step)

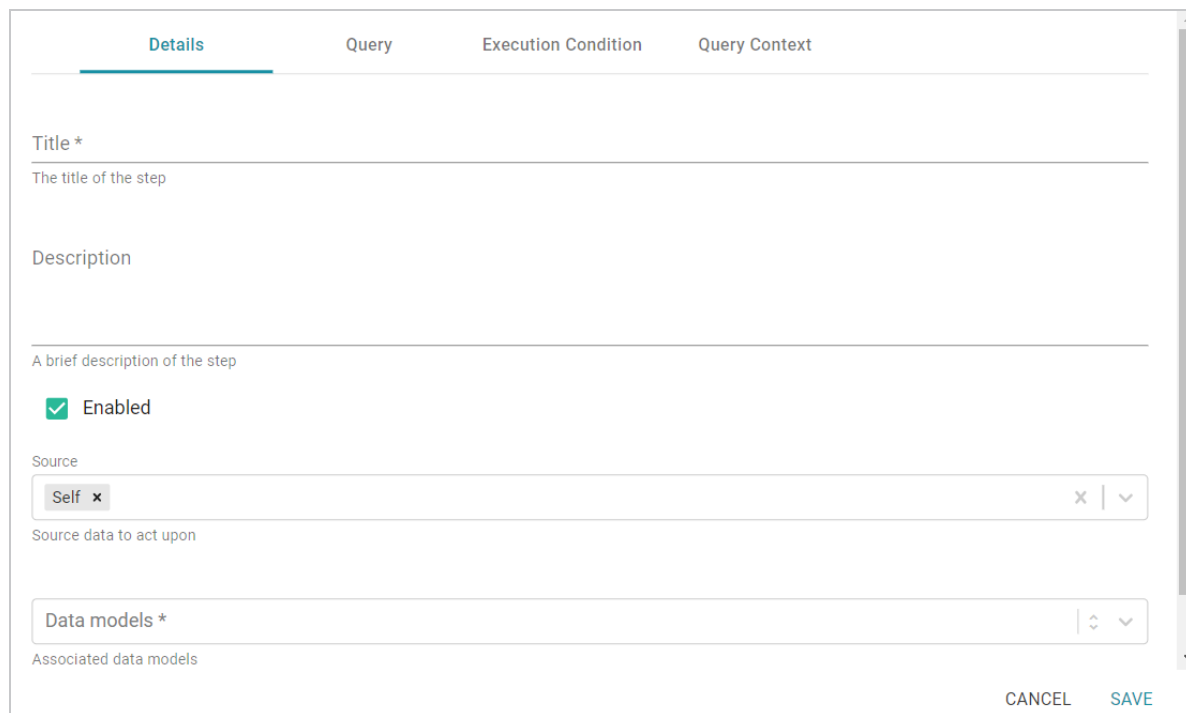
This topic provides guidance on configuring a Query Step that you can use for creating, cleaning, conforming, or transforming the data in a layer. Follow the steps below to create a Query Step.

1. Go to the graphmart for which you want to add a step and then click the **Data Layers** tab.
2. On the Data Layers tab, find the layer that you want to add the step to. Click the menu icon (⋮) for that layer and select **Add Step/View**. The Add Step/View dialog box is displayed with the New tab selected.





- To create a new Query step, select **Query Step** and then click **OK**. If you want to clone an existing step, click the **Existing Steps** tab, select the step that you want to clone, and then click **OK**. Anzo creates or clones the step and displays the Details tab:



4. On the Details tab, configure the following options as needed:
  - **Title:** The required name of the step.
  - **Description:** An optional short description of the step.
  - **Enabled:** When creating a new step, the Enabled option is selected by default, indicating that the step is enabled and will run when the layer is loaded or refreshed. If you want to disable the step so that it is not processed, clear the **Enabled** checkbox.
  - **Source:** The source data that this step should act upon. Steps can build upon the data generated by steps in other layers or can be self-contained, applying changes that relate only to the data defined in the layer that contains this step. You can select any number of the following options:
    - **Self:** This option is selected by default and means that the step runs against only the data that is generated in the layer this step belongs to.
    - **All Previous Layers Within Graphmart:** This option means that the step runs against the data that is generated by all of the successful layers that precede the layer this step is in. Any failed layers are ignored.
    - **Previous Layer Within Graphmart:** This option means that the query runs against only the data that is generated by the one layer that precedes the layer this step is in.
    - **Layer Name:** The Source drop-down list also includes options for specific layer names. You can choose a specific layer to act upon the data in that layer only.
  - **Data models:** This required field specifies the model or models to associate with this step. The list displays all of the available models. By default, the field is set to **Exclude System Data** (↕). If you want to choose a system model, click the toggle button on the right side of the field to change it to **Include System Data** (×). When system data is included, the drop-down list displays the system models in addition to the user-generated models.
  - **Pre-Run Generate Statistics:** This option controls whether to initiate AnzoGraph's internal statistics gathering queries before running the query to pre-compile. The statistics gathering helps ensure that the AnzoGraph query planner generates ideal query execution plans for queries that are run against the graphmart.
5. When you have finished configuring the Details tab, click the **Query** tab. This tab defines the query that this step should run. The template includes the syntax for writing SPARQL INSERT and DELETE queries and includes the target and source graph parameters (`${targetGraph}` and `${usingSources}`). Anzo replaces the parameters with the appropriate URIs when the step runs. Edit the template as needed. You can click the **Preview in Query Builder** button to open the query in

the Query Builder, where you can perform practice runs to see results without having to refresh the graphmart or layer.

### Note

If your query connects to a source that requires input of connection and authorization information, Cambridge Semantics recommends that you do not include the connection and authorization values directly in the query. Instead, replace those values with Context Variables from a Query Context. You can access Context Providers for each data source from the step's Query Context tab. For detailed information about query contexts and referencing variables in a query, see [Using Query Contexts in Queries](#).

The screenshot shows the Query Builder interface with four tabs: Details, Query (selected), Execution Condition, and Query Context. The Query tab displays a SPARQL transformation query with line numbers 1 through 20. The query includes PREFIX statements for rdf, rdfs, and xsd, followed by comments about runtime replacements for targetGraph and usingSources, and then DELETE and INSERT blocks with GRAPH clauses. At the bottom, there is a label 'Query used to perform the transformation' and two buttons: CANCEL and SAVE.

```
1 PREFIX rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs:     <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX xsd:      <http://www.w3.org/2001/XMLSchema#>
4
5 # targetGraph is replaced with the Layers URI at runtime
6 # usingSources is replaced with the URIs of the Layer's Sources at runtime
7 DELETE {
8   GRAPH ${targetGraph} {
9
10  }
11 }
12 INSERT {
13   GRAPH ${targetGraph} {
14
15  }
16 }
17 ${usingSources}
18 WHERE {
19
20 }
```

Query used to perform the transformation

CANCEL SAVE

6. Click **Save** to save the step configuration.

Once the Details tab is configured and the query is defined, the step can be run. For information about running this step conditionally by setting up an execution condition, see [Defining Execution Conditions](#).

## Related Topics

[SPARQL Best Practices and Query Templates](#)

## Inferring New Data (RDFS+ Inference Step)

This topic provides guidance on configuring an RDFS+ Inference Step that uses RDFS+ and OWL rules to create new relationships based on the vocabularies in the existing data. Follow the steps below to create an RDFS+ Inference Step.

1. Go to the graphmart for which you want to add a step and then click the **Data Layers** tab.
2. On the Data Layers tab, find the layer that you want to add the step to. Click the menu icon (⋮) for that layer and select **Add Step/View**. The Add Step/View dialog box is displayed with the New tab selected.

**Add Step/View**

New Existing Steps Existing Views

Search

**Direct Load Step**  
Load data directly from an external source

**Elasticsearch Indexing Step**  
Create a single Elasticsearch index to associate with a layer.

**Elasticsearch Snapshot Step**  
Create an Elasticsearch snapshot of the index associated with a layer.

**Export Step**  
This allows you to export contents of graphmart to an fids.

Rows per page: 25 1-12 of 12

CANCEL OK

3. To create a new RDFS+ Inference step, select **RDFS+ Inference Step** and then click **OK**. If you want to clone an existing step, click the **Existing Steps** tab, select the step that you want to clone, and then click **OK**. Anzo creates or clones the step and displays the Details tab:

Details
Execution Condition
Query Context

Title \*

The title of the step

Description

A brief description of the step

☒ Enabled

Inference Rules To Run

Comma separated rules: 'all' - for all set of rules, 'rdfsplus' - for RDFS+ set of rules, {rulenames} - from owl2rl specific spec rules, use prefix '-' to exclude a rule

Source

Self x

CANCEL SAVE

4. On the Details tab, configure the following options as needed:

- **Title:** The required name of the step.
- **Description:** An optional short description of the step.
- **Enabled:** When creating a new step, the Enabled option is selected by default, indicating that the step is enabled and will run when the layer is loaded or refreshed. If you want to disable the step so that it is not processed, clear the **Enabled** checkbox.
- **Inference Rules to Run:** By default the step runs all of the RDFS-plus inference rules and a subset of the OWL 2 RL rules (see [Inference Rule Reference](#) below for specifics). If you want to customize the step to include or exclude certain rules, specify any combination of the following options in the **Inference Rules To Run** field. Specify multiple options in a comma-separated list:
  - **all:** Run all rules.
  - **rdfsplus:** Run only the RDFS-plus rules.
  - **rule\_names:** List specific rules to run only those rules. See [Inference Rule Reference](#).
  - **-rule\_name:** Specify a hyphen (-) in front of a rule name to exclude that rule. For example, **-scm-svf2** excludes the scm-svf2 rule.

For example, the following value runs all of the inference rules except prp-fp and prp-ifp:

all, -prp-fp, -prp-ifp

### Note

Certain inference rules are coupled. Specifying either of the rules in the pair automatically runs the coupled rule. The list below describes the paired rules:

- scm-dom1 and scm-rng1
- scm-dom2 and scm-rng2
- prp-inv1 and prp-inv2

In addition, running scm-eqc1 and cax-sco also runs cax-eqc1 and cax-eqc2. And running scm-eqp1 and prp-spo1 also runs prp-eqp1 and prp-eqp2.

- **Source:** The source data that this step should act upon. Steps can build upon the data generated by steps in other layers or can be self-contained, applying changes that relate only to the data defined in the layer that contains this step. You can select any number of the following options:
  - **Self:** This option is selected by default and means that the step runs against only the data that is generated in the layer this step belongs to.
  - **All Previous Layers Within Graphmart:** This option means that the step runs against the data that is generated by all of the successful layers that precede the layer this step is in. Any failed layers are ignored.
  - **Previous Layer Within Graphmart:** This option means that the query runs against only the data that is generated by the one layer that precedes the layer this step is in.
  - **Layer Name:** The Source drop-down list also includes options for specific layer names. You can choose a specific layer to act upon the data in that layer only.
- **Data models:** This required field specifies the model or models to associate with this step. The list displays all of the available models. By default, the field is set to **Exclude System Data** (🚫). If you want to choose a system model, click the toggle button on the right side of the field to change it to **Include System Data** (🔲). When system data is included, the drop-down list displays the system models in addition to the user-generated models.
- **Pre-Run Generate Statistics:** This option controls whether to initiate AnzoGraph's internal statistics gathering queries before running this step. The statistics gathering helps ensure that the AnzoGraph query planner generates ideal query execution plans for queries that are run against the graphmart.

5. Click **Save** to save the step configuration.

Once the Details tab is configured, the step can be run. For information about running this step conditionally by setting up an execution condition, see [Defining Execution Conditions](#).

## Inference Rule Reference

This section provides reference information for the RDFS-plus rules and the subset of OWL 2 RL rules that inference steps run.

- [RDFS+ Rules](#)
- [OWL 2 RL Rules](#)

## RDFS+ Rules

The tables below define the RDFS-plus inference rules.

- [Semantics of Class Axioms](#)
- [Semantics of Axioms about Properties](#)
- [Semantics of Schema Vocabulary](#)

## Semantics of Class Axioms

### Note

Because **cax-eqc1** and **cax-eqc2** (described in the table below) are implied rules that are coupled with scm-eqc1 and cax-sco, including **cax-eqc1** or **cax-eqc2** in the Inference Rules to Run field will result in an `invalid inference rule name` error. To run the cax-eqc1 and cax-eqc2 rules, specify **scm-eqc1** and **cax-sco** (**scm-eqc1,cax-sco**).

Rule	Description	IF	THEN
cax-eqc1	Two classes are synonymous.	T(?c1, owl:equivalentClass, ?c2) T(?x, rdf:type, ?c1)	T(?x, rdf:type, ?c2)
cax-eqc2	Two classes are synonymous.	T(?c1, owl:equivalentClass, ?c2) T(?x, rdf:type, ?c2)	T(?x, rdf:type, ?c1)

Rule	Description	IF	THEN
cax-sco	Members of a subclass are also members of the superclass.	T(?c1, rdfs:subClassOf, ?c2) T(?x, rdf:type, ?c1)	T(?x, rdf:type, ?c2)

## Semantics of Axioms about Properties

### Note

Because **prp-eqp1** and **prp-eqp2** (described in the table below) are implied rules that are coupled with **scm-eqp1** and **prp-spo1**, including **prp-eqp1** or **prp-eqp2** in the Inference Rules to Run field will result in an `invalid inference rule name error`. To run the **prp-eqp1** and **prp-eqp2** rules, specify **scm-eqp1** and **prp-spo1** (**scm-eqp1,prp-spo1**).

Rule	Description	IF	THEN
prp-dom	Infer the subject's type from the predicate's domain.	T(?p, rdfs:domain, ?c) T(?x, ?p, ?y)	T(?x, rdf:type, ?c)
prp-eqp1	Two properties are synonymous.	T(?p1, owl:equivalentProperty, ?p2) T(?x, ?p1, ?y)	T(?x, ?p2, ?y)
prp-eqp2	Two properties are synonymous.	T(?p1, owl:equivalentProperty, ?p2) T(?x, ?p2, ?y)	T(?x, ?p1, ?y)
prp-fp	If predicate p is a functional property, then a subject can be related to only one specific object by p.	T(?p, rdf:type, owl:FunctionalProperty) T(?x, ?p, ?y1) T(?x, ?p, ?y2)	T(?y1, owl:sameAs, ?y2)
prp-ifp	If predicate p is an inverse functional property, then a specific object can be related to only one subject by p.	T(?p, rdf:type, owl:InverseFunctionalProperty) T(?x1, ?p, ?y) T(?x2, ?p, ?y)	T(?x1, owl:sameAs, ?x2)



Rule	Description	IF	THEN
prp-inv1	Two properties are the inverse of each other.	T(?p1, owl:inverseOf, ?p2) T(?x, ?p1, ?y)	T(?y, ?p2, ?x)
prp-inv2	Two properties are the inverse of each other.	T(?p1, owl:inverseOf, ?p2) T(?x, ?p2, ?y)	T(?y, ?p1, ?x)
prp-rng	Infer the object's type from the predicate's range.	T(?p, rdfs:range, ?c) T(?x, ?p, ?y)	T(?y, rdf:type, ?c)
prp-spo1	Relationships that are described by a subproperty also hold for the superproperty.	T(?p1, rdfs:subPropertyOf, ?p2) T(?x, ?p1, ?y)	T(?x, ?p2, ?y)
prp-symp	The inverse is true for a property.	T(?p, rdf:type, owl:SymmetricProperty) T(?x, ?p, ?y)	T(?y, ?p, ?x)
prp-trp	Chains of relationships collapse into a single relationship.	T(?p, rdf:type, owl:TransitiveProperty) T(?x, ?p, ?y) T(?y, ?p, ?z)	T(?x, ?p, ?z)

## Semantics of Schema Vocabulary

Rule	Description	IF	THEN
scm-cls	Every class is its own subclass and equivalent class, and it is a subclass of owl:Thing.	T(?c, rdf:type, owl:Class)	T(?c, rdfs:subClassOf, ?c) T(?c, owl:equivalentClass, ?c) T(?c, rdfs:subClassOf, owl:Thing) T(owl:Nothing, rdfs:subClassOf, ?c)

Rule	Description	IF	THEN
scm-dom1	A property with domain c also has domain c's superclasses.	T(?p, rdfs:domain, ?c1) T(?c1, rdfs:subClassOf, ?c2)	T(?p, rdfs:domain, ?c2)
scm-dom2	A subproperty inherits the domains of the superproperties.	T(?p2, rdfs:domain, ?c) T(?p1, rdfs:subPropertyOf, ?p2)	T(?p1, rdfs:domain, ?c)
scm-eqc1	Equivalent classes are subclasses of each other.	T(?c1, owl:equivalentClass, ?c2)	T(?c1, rdfs:subClassOf, ?c2) T(?c2, rdfs:subClassOf, ?c1)
scm-eqc2	If two classes are subclasses, they are also equivalent classes.	T(?c1, rdfs:subClassOf, ?c2) T(?c2, rdfs:subClassOf, ?c1)	T(?c1, owl:equivalentClass, ?c2)
scm-eqp1	Equivalent properties are subproperties of each other.	T(?p1, owl:equivalentProperty, ?p2)	T(?p1, rdfs:subPropertyOf, ?p2) T(?p2, rdfs:subPropertyOf, ?p1)
scm-eqp2	If two properties are subproperties, they are also equivalent properties.	T(?p1, rdfs:subPropertyOf, ?p2) T(?p2, rdfs:subPropertyOf, ?p1)	T(?p1, owl:equivalentProperty, ?p2)
scm-rng1	A property with range c also has range c's superclasses.	T(?p, rdfs:range, ?c1) T(?c1, rdfs:subClassOf, ?c2)	T(?p, rdfs:range, ?c2)
scm-rng2	A subproperty inherits the ranges of its superproperties.	T(?p2, rdfs:range, ?c) T(?p1, rdfs:subPropertyOf, ?p2)	T(?p1, rdfs:range, ?c)

Rule	Description	IF	THEN
		?p2)	
scm-sco	owl:subClassOf relationships are transitive	T(?c1, rdfs:subClassOf, ?c2) T(?c2, rdfs:subClassOf, ?c3)	T(?c1, rdfs:subClassOf, ?c3)
scm-spo	owl:subPropertyOf relationships are transitive.	T(?p1, rdfs:subPropertyOf, ?p2) T(?p2, rdfs:subPropertyOf, ?p3)	T(?p1, rdfs:subPropertyOf, ?p3)

### Note

The scm-dp and scm-op schema vocabulary rules are not run. Those rules add significant compute overhead but do not result in meaningful inference results.

## OWL 2 RL Rules

The tables below define the subset of OWL 2 RL inference rules that inference steps run.

- [Semantics of Equality](#)
- [Semantics of Schema Vocabulary](#)
- [Semantics of Classes](#)

### Semantics of Equality

Rule	Description	IF	THEN
eq-rep-o	Describes the replacement property of the owl:sameAs axiom.	T(?o, owl:sameAs, ?o') T(?s, ?p, ?o)	T(?s, ?p, ?o')
eq-rep-p	Describes the replacement property of the owl:sameAs axiom.	T(?p, owl:sameAs, ?p') T(?s, ?p, ?o)	T(?s, ?p', ?o)
eq-rep-s	Describes the replacement property of the owl:sameAs axiom.	T(?s, owl:sameAs, ?s') T(?s, ?p, ?o)	T(?s', ?p, ?o)

Rule	Description	IF	THEN
eq-sym	Describes the symmetric property of the owl:sameAs axiom.	$T(?x, \text{owl:sameAs}, ?y)$	$T(?y, \text{owl:sameAs}, ?x)$
eq-trans	Describes the transitive property of the owl:sameAs axiom.	$T(?x, \text{owl:sameAs}, ?y)$ $T(?y, \text{owl:sameAs}, ?z)$	$T(?x, \text{owl:sameAs}, ?z)$

## Semantics of Schema Vocabulary

Rule	Description	IF	THEN
scm-svf1	A property restriction c1 is a subclass of c2 if they are both someValuesFrom restrictions on the same property and c1's target class is a subclass of c2's target class.	$T(?c1, \text{owl:someValuesFrom}, ?y1)$ $T(?c1, \text{owl:onProperty}, ?p)$ $T(?c2, \text{owl:someValuesFrom}, ?y2)$ $T(?c2, \text{owl:onProperty}, ?p)$ $T(?y1, \text{rdfs:subClassOf}, ?y2)$	$T(?c1, \text{rdfs:subClassOf}, ?c2)$
scm-svf2	A property restriction c1 is a subclass of c2 if they are both someValuesFrom restrictions on the same class where c1's target property is a subproperty of c2's target property.	$T(?c1, \text{owl:someValuesFrom}, ?y)$ $T(?c1, \text{owl:onProperty}, ?p1)$ $T(?c2, \text{owl:someValuesFrom}, ?y)$ $T(?c2, \text{owl:onProperty}, ?p2)$ $T(?p1, \text{owl:subPropertyOf}, ?p2)$	$T(?c1, \text{rdfs:subClassOf}, ?c2)$

Rule	Description	IF	THEN
		rdfs:subPropertyOf, ?p2)	
scm-int		T(?c, owl:intersectionOf, ?x) LIST[?x, ?c1, ..., ?cn]	T(?c, rdfs:subClassOf, ?c1) T(?c, rdfs:subClassOf, ?c2) ... T(?c, rdfs:subClassOf, ?cn)

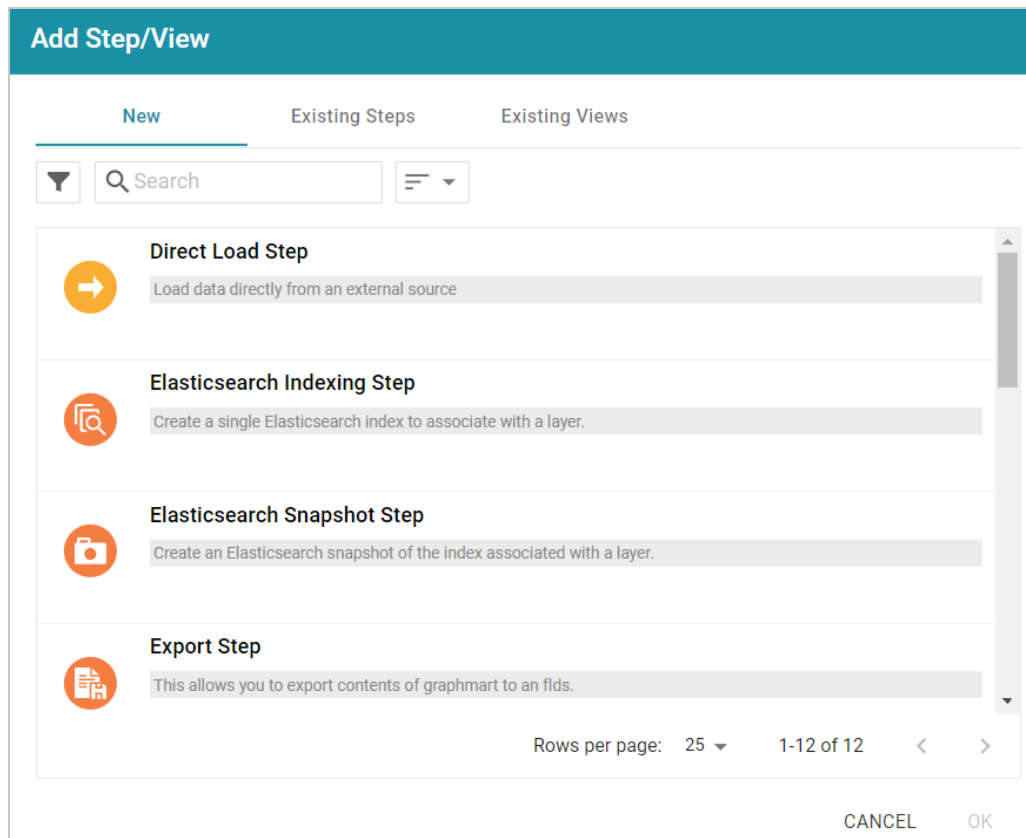
## Semantics of Classes

Rule	Description	IF	THEN
cls-svf1	At least one object of a property is a member of the specified class.	T(?x, owl:someValuesFrom, ?y) T(?x, owl:onProperty, ?p) T(?u, ?p, ?v) T(?v, rdf:type, ?y)	T(?u, rdf:type, ?x)
cls-int1	An instance belongs to every one of the specified classes.	T(?c, owl:intersectionOf, ?x) LIST[?x, ?c1, ..., ?cn] T(?y, rdf:type, ?c1) T(?y, rdf:type, ?c2) ... T(?y, rdf:type, ?cn)	T(?y, rdf:type, ?c)

## Validating the Data (Validation Step)

This topic provides guidance on configuring a Validation Step to use for validating the data in a layer and optionally setting up execution conditions. Follow the steps below to create a Validation Step.

1. Go to the graphmart for which you want to add a step and then click the **Data Layers** tab.
2. On the Data Layers tab, find the layer that you want to add the step to. Click the menu icon (⋮) for that layer and select **Add Step/View**. The Add Step/View dialog box is displayed with the New tab selected.



3. To create a new Validation step, select **Validation Step** and then click **OK**. If you want to clone an existing step, click the **Existing Steps** tab, select the step that you want to clone, and then click **OK**. Anzo creates or clones the step and displays the Details tab:

The screenshot shows a configuration window with four tabs: **Details**, **Options**, **Query**, and **Query Context**. The **Details** tab is active and contains the following elements:

- Title \***: A text input field with the placeholder text "The title of the step".
- Description**: A text input field with the placeholder text "A brief description of the step".
- Enabled**: A checkbox that is checked, indicating the step is enabled.
- Validation Source Datasource**: A dropdown menu with a downward arrow.
- Source**: A dropdown menu currently showing "Self" with a close button (x) and a downward arrow.
- Buttons**: "CANCEL" and "SAVE" buttons at the bottom right.

4. On the Details tab, configure the following options as needed:

- **Title**: The required name of the step.
- **Description**: An optional short description of the step.
- **Enabled**: When creating a new step, the Enabled option is selected by default, indicating that the step is enabled and will run when the layer is loaded or refreshed. If you want to disable the step so that it is not processed, clear the **Enabled** checkbox.
- **Validation Source Datasource**: This optional field enables you to select a data source (such as a system data source) to perform the validation against if you do not want the query to run against the graphmart that the step is in. When Validation Source Datasource is unset, the validation is performed against the graphmart.
- **Source**: The source data that this step should act upon. Steps can build upon the data generated by steps in other layers or can be self-contained, applying changes that relate only to the data defined in the layer that contains this step. You can select any number of the following options:
  - **Self**: This option is selected by default and means that the step runs against only the data that is generated in the layer this step belongs to.
  - **All Previous Layers Within Graphmart**: This option means that the step runs against the data that is generated by all of the successful layers that precede the layer this step is in. Any failed layers are ignored.

- **Previous Layer Within Graphmart:** This option means that the query runs against only the data that is generated by the one layer that precedes the layer this step is in.
- **Layer Name:** The Source drop-down list also includes options for specific layer names. You can choose a specific layer to act upon the data in that layer only.
- **Pre-Run Generate Statistics:** This option controls whether to initiate AnzoGraph's internal statistics gathering queries before running the query to pre-compile. The statistics gathering helps ensure that the AnzoGraph query planner generates ideal query execution plans for queries that are run against the graphmart.

5. When you have finished configuring the Details tab, click the **Options** tab. This tab includes the settings that specify the type of check to perform on the data as well as instructions for what to do if the validation fails.

The screenshot shows the 'Options' tab in a software interface. At the top, there are four tabs: 'Details', 'Options' (which is active and highlighted with a blue underline), 'Query', and 'Query Context'. Below the tabs, the 'Check Type' section has two radio buttons: 'Validation' (which is selected) and 'Condition'. Below these are two checkboxes: 'If the validation query fails, the layer will be marked as failed.' and 'If the validation query fails, the whole graphmart will be marked as failed.'. Below the checkboxes is a text input field with the placeholder text 'Optional variable name to which results are placed.' and the label 'Validation Result Variable Name'. At the bottom right of the form are two buttons: 'CANCEL' and 'SAVE'.

6. On the Options tab, determine which type of check to perform and select the appropriate radio button. There are two check types:

- **Validation:** A Validation check validates the data according to the defined query (on the Query tab) and can be configured to take action depending on whether the validation passes or fails.
- **Condition:** A Condition check takes the results of the query and associates it with the specified variable. That variable can then be used for setting up an execution condition at the layer or step level.



7. If you selected the Validation check type, you have the option to configure what to do if the validation query fails:
  - **If the validation query fails, the layer will be marked as failed:** Select this option if you want Anzo to abort the load of the layer if this step fails.
  - **If the validation query fails, the whole graphmart will be marked as failed:** Select this option if you want Anzo to abort the load of the entire graphmart if this step fails.
8. If you selected the Condition check type, you are required to specify the variable name that you want to use to store the result from the query. This variable becomes available as a choice when configuring an execution condition.
9. When you have finished configuring the Options tab, click the **Query** tab and compose the validation query that the step should run. The tab includes the syntax for writing a SPARQL ASK query, which is useful for determining whether a certain pattern exists in the data. ASK queries return "true" or "false" to indicate whether a solution exists. The template includes a source graph parameter (`${fromSources}`). Using the configured Source options from the Details tab, Anzo automatically populates the query with the appropriate source graph URIs when the query runs.

#### Note

If your query connects to a source that requires input of connection and authorization information, Cambridge Semantics recommends that you do not include the connection and authorization values directly in the query. Instead, replace those values with Context Variables from a Query Context. You can access Context Providers for each data source from the step's Query Context tab. For detailed information about query contexts and referencing variables in a query, see [Using Query Contexts in Queries](#).

DetailsOptionsQueryQuery Context

Validation query \*

```

1
2  #fromSources is replaced with the URIs of the Layer's Sources at runtime
3  ASK
4  ${fromSources}
5  WHERE{
6    {
7      SELECT (COUNT(*) as ?count)
8      WHERE
9        {
10         ?s ?p ?o.
11        }
12      }
13    FILTER(?count > 0)
14  }

```

Query used to perform the validation

CANCELSAVE

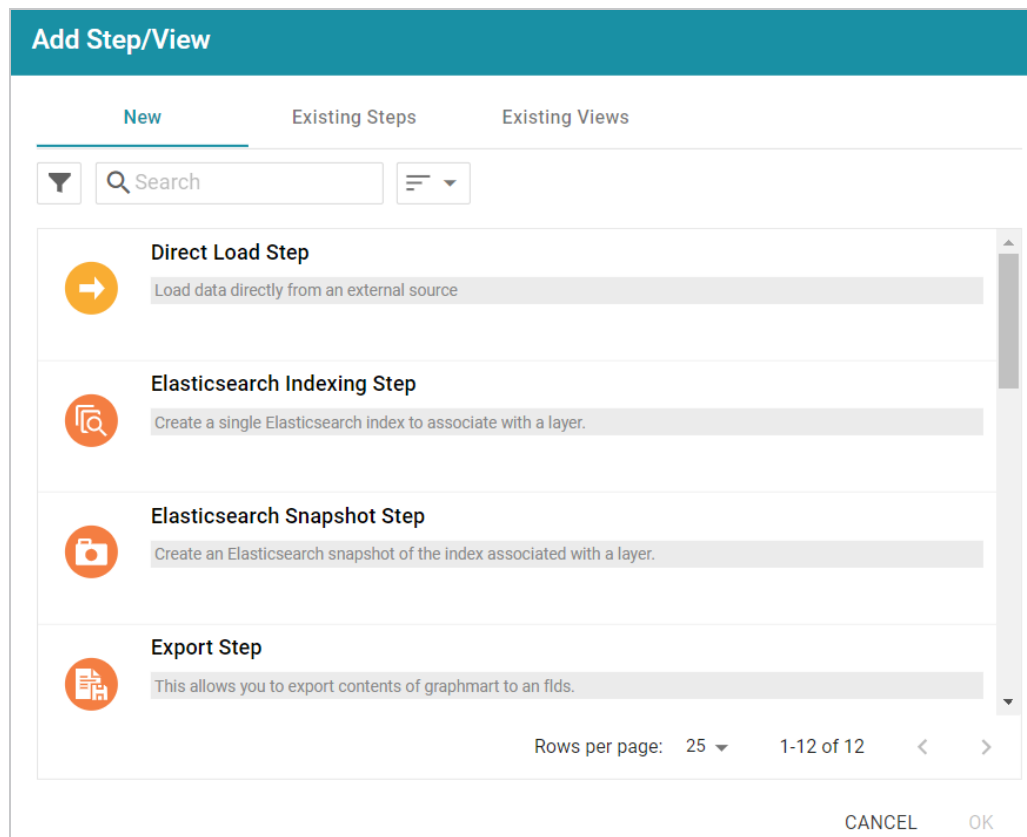
10. Click **Save** to save the step configuration.

Once the Details tab is configured and the validation options and query are defined, the step can be run. For information about setting up an execution condition that uses this step, see [Defining Execution Conditions](#).

## Constructing a View of the Data (View Step)

This topic provides guidance on configuring a View Step to create a custom view of the data that does not change the graphmart or necessarily materialize any new data. View steps contain SPARQL CONSTRUCT queries to create a view definition in AnzoGraph. Follow the steps below to create a View.

1. Go to the graphmart for which you want to add a step and then click the **Data Layers** tab.
2. On the Data Layers tab, find the layer that you want to add the step to. Click the menu icon (⋮) for that layer and select **Add Step/View**. The Add Step/View dialog box is displayed with the New tab selected.



3. To create a new View step, select **View** and then click **OK**. If you want to clone an existing view, click the **Existing Views** tab, select the view that you want to clone, and then click **OK**. Anzo creates or clones the view and displays the Details tab:

Details
Query
Execution Condition
Hi-Res Analytics
>

Title \*
The title of the view

Description
A brief description of the view

☐ Materialize the view when activated, otherwise at runtime.
☒ Enabled

Source
Source data to act upon

Data models \*
Associated data models

CANCEL
SAVE

4. On the Details tab, configure the following options as needed:

- **Title:** The required name of the step.
- **Description:** An optional short description of the step.
- **Materialize the view when activated, otherwise at runtime:** This option controls whether a copy of the data the view creates is saved in the graphmart (materialized) or whether this is a virtual view where the data is recreated each time the view runs. If you are creating a view against an extremely large data source or a source that changes often, typically the view should not be materialized. If you want to store a copy of the data that the view creates, select the **Materialize the view when activated...** check box. When this option is disabled Anzo creates a virtual view where only the view definition is stored in memory and not a copy of the data.
- **Enabled:** When creating a new step, the Enabled option is selected by default, indicating that the step is enabled and will run when the layer is loaded or refreshed. If you want to disable the step so that it is not processed, clear the **Enabled** checkbox.
- **Source:** The source data that this step should act upon. Steps can build upon the data generated by steps in other layers or can be self-contained, applying changes that relate only to the data defined in the layer that contains this step. You can select any number of the following

options:

- **Self:** This option is selected by default and means that the step runs against only the data that is generated in the layer this step belongs to.
- **All Previous Views Within Layer:** This options means that the step runs against the data that is generated by all of the previous views in the same layer.
- **Previous View Within Layer:** This options means that the step runs against the data that is generated by the previous view in the same layer.
- **All Previous Layers Within Graphmart:** This option means that the step runs against the data that is generated by all of the successful layers that precede the layer this step is in. Any failed layers are ignored.
- **Previous Layer Within Graphmart:** This option means that the query runs against only the data that is generated by the one layer that precedes the layer this step is in.
- **Data models:** This required field specifies the model or models that you want to create this view against. The list displays all of the available models. By default, the field is set to **Exclude System Data** (☺). If you want to choose a system model, click the toggle button on the right side of the field to change it to **Include System Data** (☹). When system data is included, the drop-down list displays the system models in addition to the user-generated models.

5. When you have finished configuring the Details tab, click the **Query** tab. This tab contains the query to use to create the view.

Details

Query

Execution Condition

Hi-Res Analytics

Query Context

Query \*

1

#fromSources is replaced with the URIs of the Layer's Sources at runtime

2

CONSTRUCT{

3

4

}

5

\${fromSources}

6

WHERE{

7

8

}

Query for the view

Open in query builder

CANCEL

SAVE

6. Edit the provided template to compose the CONSTRUCT query the step should run. For information about CONSTRUCT queries, see [CONSTRUCT](#) in the W3C SPARQL 1.1 Query Language specification.

#### Note

Do not include a GRAPH keyword in the CONSTRUCT clause. Anzo uses the view's URI as the graph URI for the constructed triples. In addition, Anzo uses the configured Source options from the Details tab to automatically replace the `${fromSources}` parameter with the appropriate FROM clauses when the query runs.

You can click the **Open in Query Builder** button to open the query in the Query Builder, where you can perform practice runs to see results without having to refresh the graphmart or layer.

#### Note

If your query connects to a source that requires input of connection and authorization information, Cambridge Semantics recommends that you do not include the connection and authorization values directly in the query. Instead, replace those values with Context Variables from a Query Context. You can access Context Providers for each data source from the step's Query Context tab. For detailed information about query contexts and referencing variables in a query, see [Using Query Contexts in Queries](#).

#### Tip

If your view query employs the Graph Data Interface, be sure to use the following DataToolkitView service call in the query:

```
SERVICE <http://cambridgesemantics.com/services/DataToolkitView>
  (${targetGraph})
```

For more information, see [GDI Query Syntax](#).

7. Click **Save** to save the step configuration.

Once the Details tab is configured and the query is written, the step can be run. For information about running this step conditionally by setting up an execution condition, see [Defining Execution Conditions](#). For details about the advanced Hi-Res Analytics settings that control how the view affects dashboards, see [Hi-Res Analytics Tab](#) below.

## Hi-Res Analytics Tab

The **Hi-Res Analytics** tab contains advanced settings that control how the layer is exposed to and affects Hi-Res Analytic dashboards.

### Note

Changing these settings can have unexpected consequences, and Cambridge Semantics recommends that you do not modify them unless you understand the repercussions.

The screenshot shows a settings window with five tabs: Details, Query, Execution Condition, Hi-Res Analytics (selected), and Query Context. An orange warning banner at the top reads: "Warning: These are Advanced Settings. Change Wisely." Below the banner are four settings, each with a green checkmark icon: "Display data outputted by the view in Hi-Res Analytics dashboards", "Display View in Data Layers Facet in Hi-res Analytics", "User can both view and hide the data outputted by the view", and "By default, the data outputted by the view is visible". There is also an unchecked checkbox for "Is Dynamic". A section titled "Dynamic Refresh Interval" includes a description: "The interval in milli-seconds between when server will send out data updated events." and an unchecked checkbox for "Hide From Other Layers/Views". At the bottom right are "CANCEL" and "SAVE" buttons.

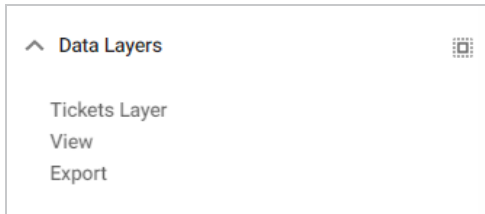
### Display data outputted by the view in Hi-Res Analytics dashboards

This setting controls whether the data accessed by the view is available to query and display in dashboards:

- When the setting is **enabled** (the default value), the view data is available to dashboards.
- When the setting is **disabled**, other data layers in the graphmart can use the view's data, but the data is not available to use Hi-Res Analytics dashboards.

### Display View in Data Layers Facet in Hi-res Analytics

This setting controls whether the view name is displayed in the Data Layers panel on dashboards. The image below shows an example Data Layers panel:



- When the setting is **enabled** (the default value), the view is listed in the Data Layers panel.
- When the setting is **disabled**, the view's data is always used in dashboards for this graphmart but users do not see the view listed in the Data Layers panel.

### User can both view and hide the data outputted by the view

This setting controls whether users have the option to show and hide the view in the Data Layers panel on dashboards:

- When the setting is **enabled** (the default value), the view is listed in the Data Layers panel and users have the option to show and hide the layer.
- When the setting is **disabled**, whether the view shows up in the Data Layers panel depends on the **By default, the data outputted by the view is visible** setting. If the view is visible in the Data Layers panel ("By default, the data outputted by the view is visible" is enabled), users cannot toggle it on and off.

### By default, the data outputted by the view is visible

This setting controls whether the data generated by the view is visible in dashboards:

- When the setting is **enabled** (the default value), the view is listed in the Data Layers panel in dashboards and is selected by default.
- When the setting is **disabled**, the view shows up in the Data Layers panel but is not selected. To include the view's data in a dashboard, the user must select the view.

### Is Dynamic

Typically this option is used only for Graph Data Interface (GDI) connections where a remote data source is accessed and that source data changes dynamically. If the source is dynamic and you want Anzo to automatically refresh the view of the data at certain intervals, select the **Is Dynamic** checkbox. Then set the Dynamic Refresh Interval (described below).



## Dynamic Refresh Interval

If the **Is Dynamic** option is enabled, this setting configures the interval at which Anzo queries the data source to retrieve any updated view data. Specify the number of milliseconds to wait between refreshes of the data.

## Hide from Other Layers/Views

This setting controls whether the other layers in the graphmart can act upon the data in this view.

- When the setting is **disabled** (the default value), this view is available as a choice in the **Source** drop-down list when a step is configured.
- When the setting is **enabled**, this view is not listed as a choice in the **Source** list.

## Related Topics

[SPARQL Best Practices and Query Templates](#)

## Creating Data on Demand Endpoints

With the Anzo Data on Demand service you can generate Open Data Protocol (OData)-based feeds that can be used to access Graphmarts programmatically via a RESTful API or from third-party business intelligence applications such as TIBCO Spotfire, Tableau, and Microsoft Power BI. The OData protocol enables web clients to use simple HTTP messages to access resources that are identified using URLs. OData shares some similarities with JDBC and ODBC. Like ODBC, OData is not limited to relational databases. The Anzo Data on Demand service follows the OData Version 4.0 specification, which defines the standard URL conventions, query options, and metadata schema.

Anzo supports two types of Data on Demand endpoints. The first type is called an **Auto-Generated** endpoint. Auto-Generated endpoints are the quickest type to create. They simply make available as-is all of the data in the selected Data Layers. Any joins, filters, and other operations must be performed by the consumer of the data outside of Anzo and AnzoGraph.

The second type of endpoint is called a **Custom** endpoint (sometimes called a **Table** endpoint). Since queries that join data often perform very poorly when run in BI applications with a JDBC driver, Custom endpoints let you assemble custom queries that join classes and apply filters and formulas. The endpoint becomes a view in AnzoGraph and AnzoGraph executes the custom queries in memory. Results can then be viewed from the endpoint without having to run the complex analytic queries over JDBC.

### Note

The Custom Data on Demand Endpoint feature is available as a **Preview** release, which means the implementation has recently been completed but is not yet thoroughly tested and could be unstable. The feature is available for trial usage, but Cambridge Semantics recommends that you do not rely on Preview features in production environments.

The topics in this section provide instructions for creating both types of endpoints:

[Creating an Auto-Generated Endpoint](#)

[Creating a Custom Endpoint \(Preview\)](#)

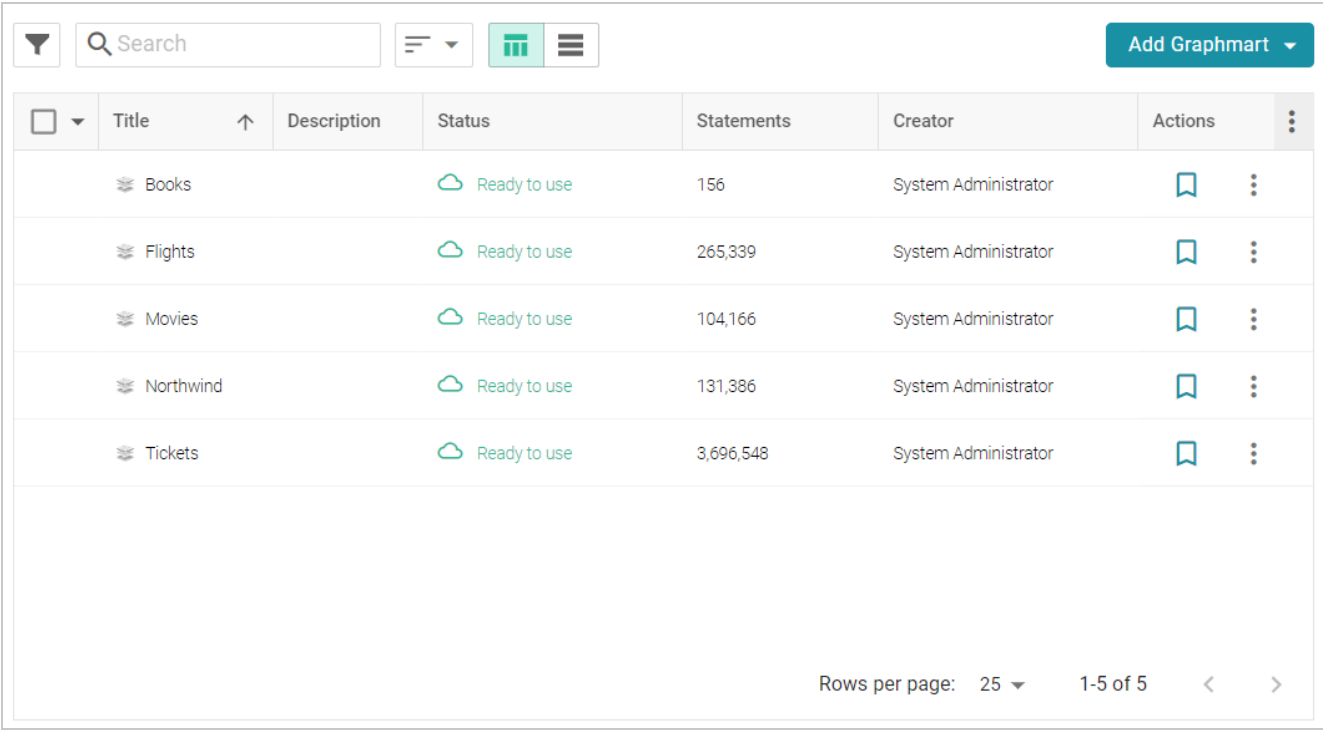
## Related Topics

[Accessing Data on Demand Endpoints](#)

## Creating an Auto-Generated Endpoint

Follow the instructions below to create an Auto-Generated Data on Demand endpoint. Auto-Generated endpoints can quickly be created to make available all of the data in the selected Data Layers. The data cannot be customized to exclude certain classes, join data across classes, or apply functions and formulas to properties. For instructions on creating an endpoint that can be customized, see [Creating a Custom Endpoint \(Preview\)](#).

1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing graphmarts. For example:

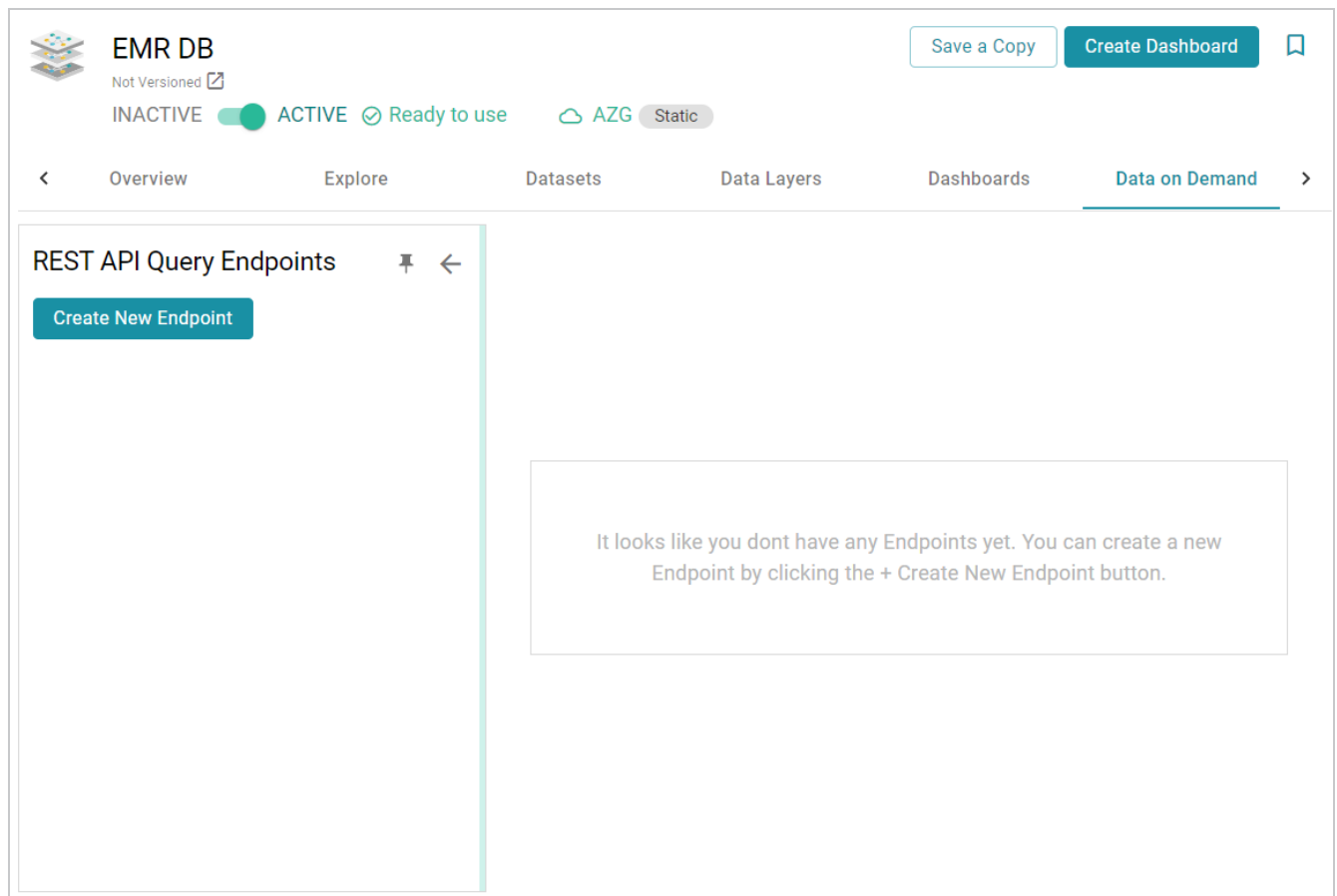


The screenshot shows the Anzo Graphmarts interface. At the top, there is a search bar with a magnifying glass icon and the text "Search". To the right of the search bar are two icons: a green bar chart and a grey bar chart. Further right is a blue button labeled "Add Graphmart" with a downward arrow. Below these elements is a table with the following columns: a checkbox, "Title", "Description", "Status", "Statements", "Creator", "Actions", and a vertical ellipsis. The table contains five rows of data:

<input type="checkbox"/>	Title	Description	Status	Statements	Creator	Actions	
<input type="checkbox"/>	Books		Ready to use	156	System Administrator		
<input type="checkbox"/>	Flights		Ready to use	265,339	System Administrator		
<input type="checkbox"/>	Movies		Ready to use	104,166	System Administrator		
<input type="checkbox"/>	Northwind		Ready to use	131,386	System Administrator		
<input type="checkbox"/>	Tickets		Ready to use	3,696,548	System Administrator		

At the bottom right of the table, there is a pagination control showing "Rows per page: 25" with a dropdown arrow, followed by "1-5 of 5" and navigation arrows.

2. On the Graphmarts screen, click the name of the graphmart for which you want to create an endpoint.
3. Click the **Data on Demand** tab. Anzo displays the Data on Demand screen, which lists any existing endpoints. For example, the image below shows a graphmart that does not have any endpoints configured:



4. Click the **Create New Endpoint** button on the left side of the screen. Anzo displays the Create REST API Query Endpoint screen.

### Create REST API Query Endpoint

Endpoint Name \*

Name of Endpoint

Endpoint Description

Description of Endpoint

Endpoint Creation

☒ Auto-Generated

☐ Custom

☐ Denormalize Results

Some BI tools work best with denormalized results.

CANCEL

SAVE

5. Configure the endpoint options on the screen as needed. The list below describes each setting:
- **Endpoint Name:** Specify a name for the endpoint in this field. The endpoint name must be unique.
  - **Endpoint Description:** You can add an optional description for the endpoint in this field.
  - **Endpoint Creation:** This field specifies the type of endpoint to create. By default, the type is set to **Auto-Generated**. Leave the Auto-Generated radio button selected.
  - **Denormalize Results:** By default (when Denormalize Results is not selected), OData returns multi-valued properties as arrays. Certain BI tools, however, do not support arrays or multi-valued properties. If your data includes multi-valued properties and you plan to view the endpoint using a BI tool that does not support them, you can select the **Denormalize Results** setting to denormalize all multi-valued properties that are exposed in the endpoint. For JSON, XML, and CSV output formats, denormalization expands the properties into new rows so that they can be viewed in BI tools.

#### Example

The following image shows an example of CSV output of multi-valued properties when **Denormalize Results** is disabled:

And the example below shows the output of multi-valued properties when **Denormalize Results** is enabled:

### Note

## Creating an Auto-Generated Endpoint

6. Click **Save** to create the endpoint and view the configuration details. For example:

The screenshot shows the 'Data on Demand' tab in the Anzo configuration interface. On the left, a sidebar titled 'REST API Query Endpoints' contains a 'Create New Endpoint' button and a list with 'EMR data'. The main panel displays the configuration for the 'EMR data' endpoint. Fields include 'Endpoint Name' (EMR data), 'Endpoint Description' (None), and an 'Enabled' toggle switch (checked). Under 'Included Data layers', the 'All' radio button is selected. There are buttons for 'DB-QA Layer' and 'Create EMR dataset'. A checked checkbox indicates that the system should look up names using 'endPointNamePredicate'. Below this, a text field for the predicate is set to 'None'. There are also unchecked checkboxes for 'Cache Enabled' and 'Denormalize Results'. The 'Connection String' section is empty, with a note about service root URLs. At the bottom, 'ODBC (SQL)' and 'JDBC (SQL)' options are visible.

7. You can make changes to any of the following additional configuration options that become available after the endpoint is created:

- **Enabled:** By default the endpoint is set to **Enabled**, indicating that the endpoint is active. If you want to disable the endpoint, slide the Enabled slider to the left.

#### Note

If a request is sent to a disabled endpoint, Anzo displays a `503: Service Unavailable` error with a message indicating that the endpoint is disabled. For example, "Unable to process request. The endpoint '<name>' is DISABLED."

- **Included Data Layers:** By default the Included Data Layers option is set to **All**, indicating that all of the layers in the graphmart are available from the endpoint. The included layers are listed below the radio buttons.

If you do not want to include all layers, click the **Selected** radio button. An **Edit Selections** link becomes available under the list of layers. Click **Edit Selections** to open the Select Data Layers dialog box. For example:

Select Data Layers

<input checked="" type="checkbox"/>	Title	Description	Updated Date	Tags
<input checked="" type="checkbox"/>	Create EMR dataset		Nov 21, 2022	
<input checked="" type="checkbox"/>	DB-QA Layer	Load data from DB-QA	Nov 21, 2022	Auto-Gen

2 Items Selected

CANCEL SAVE SELECTIONS

Clear the checkbox for any layer that you want to exclude from the endpoint, and then click **Save Selections** to save the change and return to the configuration screen.

- Controls whether or not to look up name using endPointNamePredicate:** This setting controls which predicate value from the related model is used for the class and property display names in the endpoint. By default, the setting is enabled and the **Predicate used to retrieve value for name from class or property** is blank. That means Anzo uses the **rdfs:label** (<http://www.w3.org/2000/01/rdf-schema#label>) value for each class and property name.
  - If you want the endpoint to use a different value for class and property names, you can edit **Predicate used to retrieve value for name from class or property** to specify the URI for another predicate from the model. For example, specifying `http://purl.org/dc/elements/1.1/description` would use each entity's **Description** value.
  - If you disable the **Controls whether or not to look up name using endPointNamePredicate** setting, each entity's local name is used.



- **Cache Enabled:** When the endpoint is accessed, Anzo translates the OData query to a SPARQL query and sends it to AnzoGraph for execution. The Cache Enabled setting controls whether the results of that AnzoGraph query are cached in Anzo so that subsequent endpoint requests can run against the cache in Anzo. When **Cache Enabled** is disabled (the default setting), Anzo does not store the cache, and endpoint requests are sent to AnzoGraph. When **Cache Enabled** is selected, Anzo stores the cached results and AnzoGraph only gets queried if the cached results are invalidated and need to be refreshed.

Once you are satisfied with the configuration, this Data on Demand endpoint is ready for access via OData/ODBC or JDBC. At the bottom of the screen, retrieve the ODBC or JDBC service URL to use to access the endpoint. For example:


### Connection String

Service root URLs used to retrieve class data from this graphmart.

ODBC (SQL)

JDBC (SQL)

---

<https://10.102.0.17/dataondemand/EMR-DB/EMR-data> 

To test whether the endpoint is active, you can copy the ODBC service URL and paste it into a web browser. If the endpoint is active, the browser shows an XML feed of the schema. For example:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<app:service xmlns:atom="http://www.w3.org/2005/Atom" xmlns:app="http://www.w3.org/2007/app" xmlns:metadata="http://docs.oasis-open.org/odata/ns/metadata"
  metadata:context="https://10.102.0.17/dataondemand/EMR-DB/EMR-data/$metadata">
  <app:workspace>
    <atom:title>Feeds.Default</atom:title>
    <app:collection href="Emr_Observationdescription" metadata:name="Emr_Observationdescription">
      <atom:title>Emr_Observationdescription</atom:title>
    </app:collection>
    <app:collection href="Emr_Observation" metadata:name="Emr_Observation">
      <atom:title>Emr_Observation</atom:title>
    </app:collection>
    <app:collection href="Emr_Medication" metadata:name="Emr_Medication">
      <atom:title>Emr_Medication</atom:title>
    </app:collection>
    <app:collection href="Emr_Specialty" metadata:name="Emr_Specialty">
      <atom:title>Emr_Specialty</atom:title>
    </app:collection>
    <app:collection href="Emr_Complaint" metadata:name="Emr_Complaint">
      <atom:title>Emr_Complaint</atom:title>
    </app:collection>
    <app:collection href="Emr_Complaintdescription" metadata:name="Emr_Complaintdescription">
      <atom:title>Emr_Complaintdescription</atom:title>
    </app:collection>
    <app:collection href="Emr_Medicationdescription" metadata:name="Emr_Medicationdescription">
      <atom:title>Emr_Medicationdescription</atom:title>
    </app:collection>
    <app:collection href="Emr_Activity" metadata:name="Emr_Activity">
      <atom:title>Emr_Activity</atom:title>
    </app:collection>
    <app:collection href="Emr_Study" metadata:name="Emr_Study">
      <atom:title>Emr_Study</atom:title>
    </app:collection>
    <app:collection href="Emr_Patient" metadata:name="Emr_Patient">
      <atom:title>Emr_Patient</atom:title>
    </app:collection>
    <app:collection href="Emr_Signal" metadata:name="Emr_Signal">
      <atom:title>Emr_Signal</atom:title>
    </app:collection>
  </app:workspace>
</app:service>
```

## Note

The endpoint is accessible only when it is **Enabled** and the associated graphmart is **Active**.

For information about accessing endpoints programmatically, see [Accessing an Endpoint Programmatically](#). For information about accessing endpoints with third-party analytics tools, see [Accessing an Endpoint from an Application](#). For information about the supported OData operators, output format, and query examples, see [OData Reference](#).

## Related Topics

### [Creating a Custom Endpoint \(Preview\)](#)

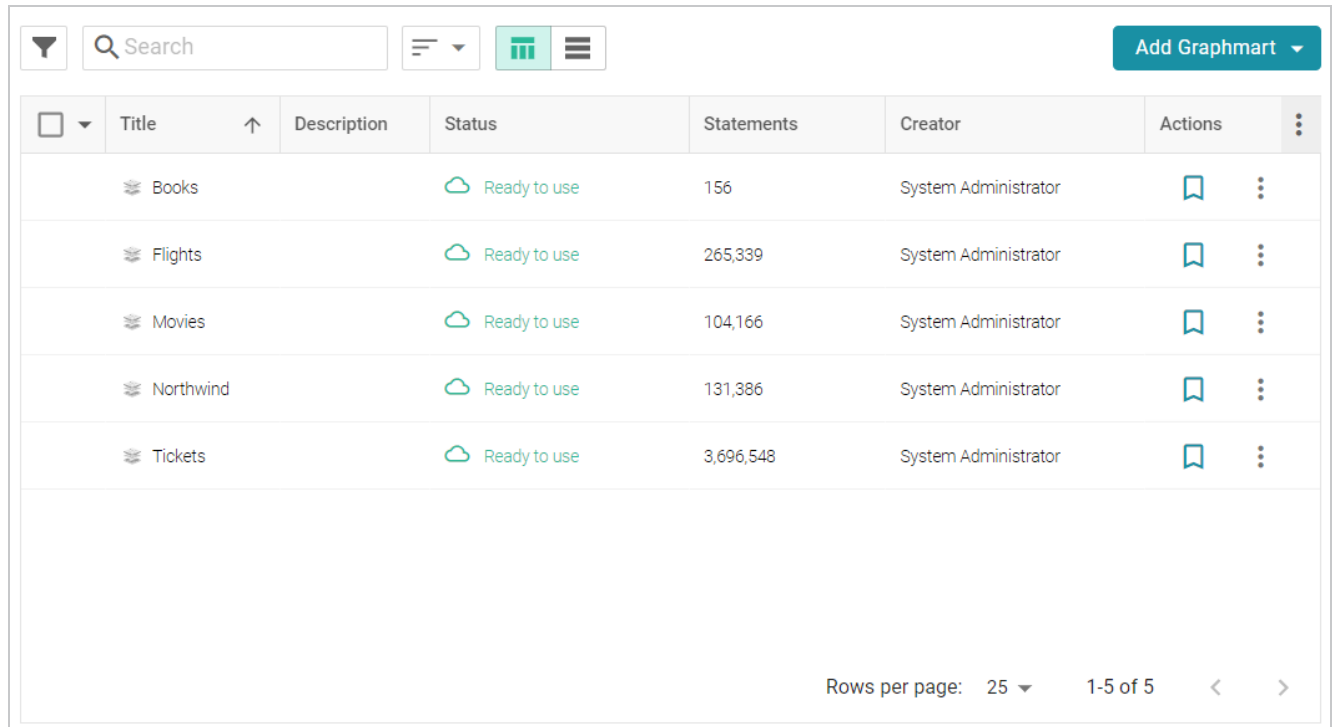
### Creating a Custom Endpoint (Preview)

Follow the instructions below to create a Custom Data on Demand endpoint (sometimes called a Table endpoint). Creating a custom endpoint is similar to creating a dashboard Table lens in that you build a table with the columns that you want to see. You can traverse the relationships and join classes, add filters, and apply functions to properties. The tables are translated to SPARQL queries that create views in AnzoGraph, allowing you to interact with the graph for complex analytics but generate results in the tabular format that BI tools expect.

## Note

The Custom Data on Demand Endpoint feature is available as a **Preview** release, which means the implementation has recently been completed but is not yet thoroughly tested and could be unstable. The feature is available for trial usage, but Cambridge Semantics recommends that you do not rely on Preview features in production environments.

1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing graphmarts. For example:

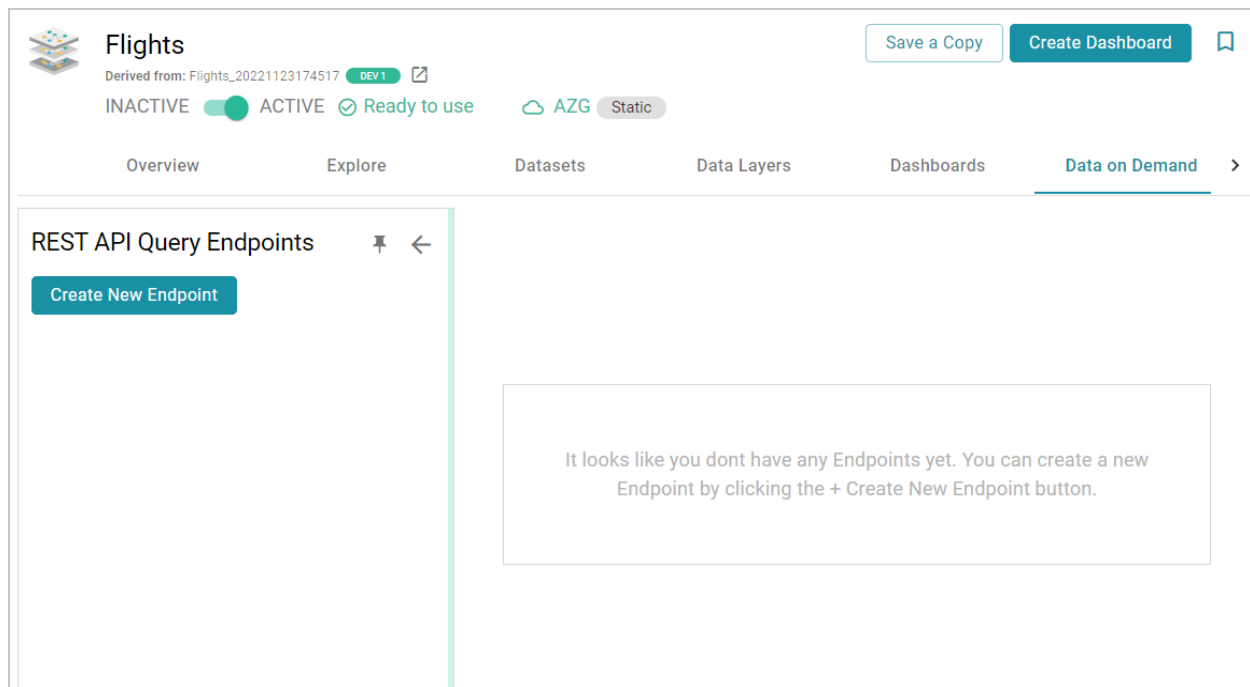


The screenshot shows the Anzo Graphmarts interface. At the top, there is a search bar, a filter icon, and a button to 'Add Graphmart'. Below this is a table with the following columns: Title, Description, Status, Statements, Creator, and Actions. The table lists five graphmarts: Books, Flights, Movies, Northwind, and Tickets. Each row shows the graphmart's name, a description, a status of 'Ready to use', the number of statements, the creator (System Administrator), and action icons (bookmark and menu).

Title	Description	Status	Statements	Creator	Actions
Books		Ready to use	156	System Administrator	
Flights		Ready to use	265,339	System Administrator	
Movies		Ready to use	104,166	System Administrator	
Northwind		Ready to use	131,386	System Administrator	
Tickets		Ready to use	3,696,548	System Administrator	

Rows per page: 25 1-5 of 5

2. On the Graphmarts screen, click the name of the graphmart for which you want to create an endpoint.
3. Click the **Data on Demand** tab. Anzo displays the Data on Demand screen, which lists any existing endpoints. For example, the image below shows a graphmart without any endpoints:



4. Click the **Create New Endpoint** button on the left side of the screen. Anzo displays the Create REST API Query Endpoint screen:

The screenshot shows the 'Create REST API Query Endpoint' form. It has a teal header with the title 'Create REST API Query Endpoint'. Below the header, there are two text input fields: 'Endpoint Name \*' with a placeholder 'Name of Endpoint' and 'Endpoint Description' with a placeholder 'Description of Endpoint'. Below these fields, there's a section for 'Endpoint Creation' with two radio buttons: 'Auto-Generated' (which is selected) and 'Custom'. Below the radio buttons, there's a checkbox labeled 'Denormalize Results' with a note: 'Some BI tools work best with denormalized results.' At the bottom right of the form, there are two buttons: 'CANCEL' and 'SAVE'.

5. Configure the endpoint options on the screen as needed. The list below describes each setting:
  - **Endpoint Name:** Specify a name for the endpoint in this field. The endpoint name must be unique.

- **Endpoint Description:** You can add an optional description for the endpoint in this field.
- **Endpoint Creation:** This field specifies the type of endpoint to create. By default, the type is set to **Auto-Generated**. Select the **Custom** radio button.
- **Denormalize Results:** By default (when Denormalize Results is not selected), OData returns multi-valued properties as arrays. Certain BI tools, however, do not support arrays or multi-valued properties. If your data includes multi-valued properties and you plan to view the endpoint using a BI tool that does not support them, you can select the **Denormalize Results** setting to denormalize all multi-valued properties that are exposed in the endpoint. For JSON, XML, and CSV output formats, denormalization expands the properties into new rows so that they can be viewed in BI tools.

### Example

The following image shows an example of CSV output of multi-valued properties when **Denormalize Results** is disabled:

```
TeamId,TeamName,TeamToPlayer_PlayerName,league_tbal44_key
1,Al Thomas,"[Fred Wynn,Steve Jones,James Smith]",aHR0cDovL2NzaS5jb20vVGh0bXMvMQ
2,Black Sox,"[Tim Hooper,Jared Bonds,Matt Butler]",aHR0cDovL2NzaS5jb20vVGh0bXMvMg
3,Braves,"[Billy Roper,Alex Granderson,Ted Sale]",aHR0cDovL2NzaS5jb20vVGh0bXMvMw
4,Somerville,"[Chris Underwood,Mike Magazine,Ted James]",aHR0cDovL2NzaS5jb20vVGh0bXMvNA
```

And the example below shows the output of multi-valued properties when **Denormalize Results** is enabled:

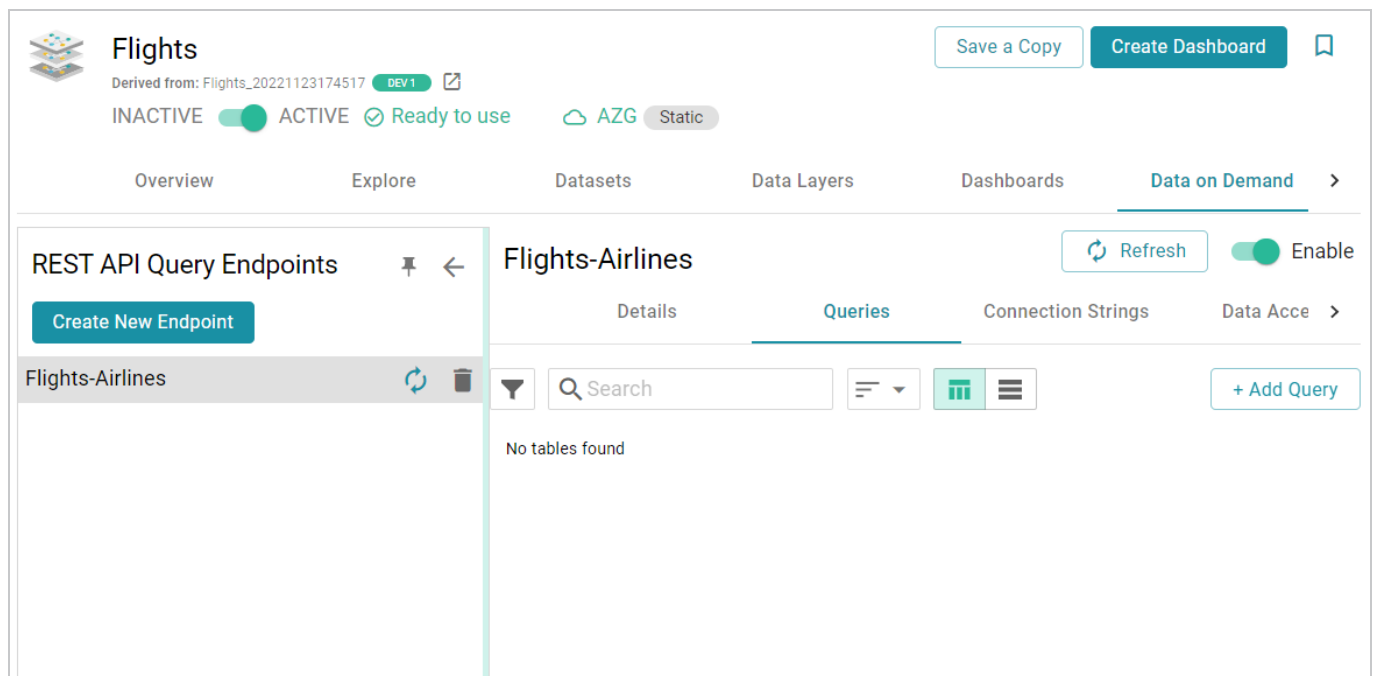
```
TeamId,TeamName,PlayerName,table51_key
1,Al Thomas,James Smith,aHR0cDovL2NzaS5jb20vVGh0bXMvMQ
1,Al Thomas,Fred Wynn,aHR0cDovL2NzaS5jb20vVGh0bXMvMQ
1,Al Thomas,Steve Jones,aHR0cDovL2NzaS5jb20vVGh0bXMvMQ
2,Black Sox,Matt Butler,aHR0cDovL2NzaS5jb20vVGh0bXMvMg
2,Black Sox,Jared Bonds,aHR0cDovL2NzaS5jb20vVGh0bXMvMg
2,Black Sox,Tim Hooper,aHR0cDovL2NzaS5jb20vVGh0bXMvMg
3,Braves,Billy Roper,aHR0cDovL2NzaS5jb20vVGh0bXMvMw
3,Braves,Alex Granderson,aHR0cDovL2NzaS5jb20vVGh0bXMvMw
3,Braves,Ted Sale,aHR0cDovL2NzaS5jb20vVGh0bXMvMw
4,Somerville,Mike Magazine,aHR0cDovL2NzaS5jb20vVGh0bXMvNA
4,Somerville,Chris Underwood,aHR0cDovL2NzaS5jb20vVGh0bXMvNA
4,Somerville,Ted James,aHR0cDovL2NzaS5jb20vVGh0bXMvNA
```

### Note

For Custom endpoints, you also have the option to denormalize data on a per-column basis. If you do not want to denormalize all multi-valued properties, you can leave **Denormalize Results** disabled and then enable denormalization for specific columns when you build the endpoint views.

Also note that denormalizing all results means that more rows are created and transferred. Depending on the number of multi-valued properties and how the data is set up, you may see slower performance when creating and querying denormalized endpoints.

- When you have finished configuring the endpoint options, click **Save**. The endpoint is created and is empty until you create queries. For example, the image below shows a new endpoint:



By default the endpoint is set to **Enable**, indicating that the endpoint is active. If you want to disable the endpoint, slide the **Enable** slider to the left.

### Note

If a request is sent to a disabled endpoint, Anzo displays a 503: Service Unavailable error with a message indicating that the endpoint is disabled. For example, "Unable to process request. The endpoint '<name>' is DISABLED."

7. (Optional) You can click the **Details** tab to make changes to any of the following additional configuration options that become available after the endpoint is initially created:

Details

Queries

Endpoint Description

flights by airline

EndPoint Query Priority

Extraction

☒ Cache Enabled

☐ Denormalize Results

- **Endpoint Query Priority:** This setting controls where in Anzo's query queue the generated SPARQL queries for this endpoint are placed for processing, i.e., the priority for executing queries against this endpoint versus other types of queued queries like data layer and dashboard queries. By default, Endpoint Query Priority is set to **Extraction**, which is priority level 3 out of 8. The options are Interactive = 1, Extraction = 3, and Batch Process = 8.
  - **Cache Enabled:** When the endpoint is accessed, Anzo translates the OData query to a SPARQL query and sends it to AnzoGraph for execution. The Cache Enabled setting controls whether the results of that AnzoGraph query are cached in Anzo so that subsequent endpoint requests can run against the cache in Anzo. When Cache Enabled is selected (the default setting), Anzo stores the cached results and AnzoGraph only gets queried if the cached results are invalidated and need to be refreshed. When Cache Enabled is disabled, Anzo does not store the cache, and endpoint requests are sent to AnzoGraph.
8. To start building a view, click the **Queries** tab (if necessary), and then click the **Add Query** button. The New Query dialog box is displayed.

The screenshot shows a 'New Query' dialog box with a teal header. Below the header are two tabs: 'Details' (active) and 'Query'. The 'Details' tab contains the following fields: 'Title \*' with a placeholder 'The title of the view', 'Description' with a placeholder 'A brief description of the view', a 'Materialize' checkbox, and an 'Include Data Layers' dropdown menu currently set to 'All layers'. At the bottom right are 'CANCEL' and 'SAVE' buttons.

9. On the **Details** tab, configure the following options as needed:
- **Title:** Type a name for the table in the Title field. The name must be unique for the endpoint.
  - **Description:** You can add an optional description for the view in this field.
  - **Materialize:** If you want to store a copy of the data that this view creates (materialize the data), select the **Materialize** check box. When this option is disabled AnzoGraph creates a virtual view where only the view definition is stored in memory and not a copy of the data. If a request is made against this view, AnzoGraph temporarily materializes the data in memory, performs the query operations, and then drops the temporary data.
  - **Include Data Layers:** By default, the Include Data Layers option is set to include **All Layers**. If you do not want the query to target the source data in all layers, select the field and choose alternate layers from the drop-down list.



10. Click the **Query** tab to start building a table.

The screenshot shows a 'New Query' dialog box. At the top, there's a teal header with the text 'New Query' and a maximize icon. Below the header, there are two tabs: 'Details' and 'Query'. The 'Query' tab is currently selected. Under the 'Query' tab, there's a 'Starting point:' label followed by a dropdown menu that says 'Select starting point...'. Below this is a large gray rectangular area containing the text 'Set a starting point to begin defining your query'. At the bottom right of the dialog, there are two buttons: 'CANCEL' and 'SAVE'.

11. On the Query screen, get started by selecting a class to use as the starting point. Click the **Select Starting Point** drop-down list and select a class. Once you select the class, the **Add Column** button is displayed.

12. Click **Add Column** to create a column in the table. The Add Column dialog box is displayed:

**Edit Column**

Formula Value

Column Header Label

☐ Denormalize column data

Sample Data

No expression specified

CANCEL SAVE

13. On the Add Column screen, configure the following options as needed:
- **Formula Value:** Click this field to choose the column header. Like building columns in a dashboard table, you can navigate the relationships to join data from different classes, and you can apply functions to the values. When you click the field and start to select paths and properties, sample data is shown on the right side of the screen. For example:

### Edit Column

Formula Value

/{Flights 10k}Airline(string)

/

ROOT

Returns values of the root type.

123

Air System Delay

int 123

123

Air Time

int 123

T

Airline

string T

123

Airline Delay

int 123

123

Arrival Delay

int 123

Sample Data

AS

AA

US

AA

AS

DL

NK

US

AA

DL

DL

AA

DL

DL

DL

AS

CANCEL

SAVE

- **Column Header Label:** This is the label that you want to use for displaying the Formula Value.
- **Denormalize column data:** If you enabled **Denormalize Results** at the endpoint level, leave this setting disabled. If Denormalize Results is disabled at the endpoint level, you can enable this setting to denormalize the values for this column only.

#### Tip

If you do not denormalize the data, you can use the **Format** tab to specify the character to use for separating the values in the arrays that are returned for multi-valued properties. By default, the Value Separator is set to comma (,). The Format tab becomes available after saving the column.

14. Click **Save** to add the column to the table.
15. Add new columns to the table by clicking **Add Column**. You can change the order of columns by dragging a column up or down, and you can enable or disable columns by sliding the slider for the column. Click the menu icon (⋮) for a column to edit or delete that column.

New Query

Details

Query

Starting point: Flights10k

Airline	Flight Number	Distance	Elapsed Time
AS	98	1448	194
AA	2336	2330	279
US	840	2296	293
AA	258	2342	281
AS	135	1448	215
DL	806	1580	230

Rows per page: 25

1-25 of 9999

Manage Configuration

Columns

Group by

Filters

Format

+ Add Column

Airline	ON
Flight Number	ON
Distance	ON
Elapsed Time	ON

CANCEL

SAVE

- If you would like to group by clause, you can click the Group By tab and select the column or columns to group.
- If you would like to filter out some data, you can add one or more filters to the overall query. To add a filter, click the **Filters** tab and then click **Add Filter**. In the **Formula Value** field, specify the formula to use to determine which values should be included in the results. Then specify a name for the column in the **Column Header Label**. For example, the filter below is configured to include only the results where the Player ID is less than or equal to 5.

Add Filter

Formula Value

LE( /{Players}PlayerId{int} , 5)

Column Header Label

LE(PlayerId, 5)

- Once you are satisfied with the configuration, click **Save** to create the endpoint.

This endpoint is ready for access via OData/ODBC or JDBC. On the **Connection Strings** tab for the endpoint, retrieve the ODBC or JDBC service URL to use to access the endpoint. For example:

The screenshot shows a web interface with a top navigation bar containing tabs: Overview, Explore, Datasets, Data Layers, Dashboards, and Data on Demand (which is active). On the left, there's a sidebar with 'REST API Query Endpoints' and a 'Create New Endpoint' button. Below this, 'Flights-Airlines' is listed with a refresh icon and a trash icon. The main panel is titled 'Flights-Airlines' and has a 'Refresh' button and an 'Enable' toggle switch. Below the title are tabs: Details, Queries, Connection Strings (active), and Data Access. Under the 'Connection Strings' tab, there's a section for 'Connection String' with the text 'Service root URLs used to retrieve class data from this graphmart.' Below this are two sub-tabs: 'ODBC (SQL)' (active) and 'JDBC (SQL)'. The ODBC (SQL) section displays the URL 'https://10.102.0.17/dataondemand/Flights/Flights-Airlines' with a copy icon.

To test whether the endpoint is active, you can copy the ODBC service URL and paste it into a web browser.

### Important

After pasting the URL into the browser, **add the table name to the end of the string**. The URL for a Custom endpoint is `<ODBC_or_JDBC_URL>/<table_name>`. For example, if the ODBC URL is `https://10.10.0.11/dataondemand/Flights/Flights-Airlines`, and the table name is Airlines, the connection string is `https://10.10.0.11/dataondemand/Flights/Flights-Airlines/Airlines`.

If the endpoint is active, the browser shows an XML feed of the data. For example:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<a:feed xmlns:a="http://www.w3.org/2005/Atom" xmlns:m="http://docs.oasis-open.org/odata/ns/metadata"
xmlns:d="http://docs.oasis-open.org/odata/ns/data"
m:context="https://10.102.0.17/dataondemand/Flights/Flights-Airlines/$metadata#Airlines">
  <a:id>https://10.102.0.17/dataondemand/Flights/Flights-Airlines/Airlines</a:id>
</a:feed>
```

### Note

The endpoint is accessible only when it is **Enabled** and the associated graphmart is **Active**.

For information about accessing endpoints programmatically, see [Accessing an Endpoint Programmatically](#). For information about accessing endpoints with third-party analytics tools, see [Accessing an Endpoint from an Application](#). For information about the supported OData operators, output format, and query examples, see [OData Reference](#).

### Related Topics

[Creating an Auto-Generated Endpoint](#)

## Sharing Access to Graphmarts

The topics in this section introduce the concepts to know when working with graphmart and data layer access configuration and provide instructions for configuring permissions.

[Introduction to Graphmart and Data Layer Sharing](#)

[Configuring Graphmart and Data Layer Permissions](#)

### Introduction to Graphmart and Data Layer Sharing

This topic introduces you to the concepts that are helpful to know when working with graphmart and data layer permissions. It also gives of overview of the graphmart sharing settings in the user interface and describes the predefined permission sets and associated privileges for users and groups.

- [Default Access Configuration](#)
- [Configuration vs. Data Access Control](#)
- [Permission Inheritance](#)
- [User and Group Configuration Permissions](#)

### Default Access Configuration

When a new graphmart is created, the access control configuration of that Graphmart is defined by the **Graphmarts Registry** Default Access Policy that is configured by your administrator (see [Managing Default Access Policies](#) in the Administration Guide for information). The graphmart also inherits permissions from other artifacts in the onboarding workflow. For example, when a graphmart is created from a data source, the graphmart inherits permissions from the source schema (which inherits permissions from the data source). Users who have permission to modify graphmart access can share that graphmart with other users and groups.

### Configuration vs. Data Access Control

Graphmart and data layer sharing is managed on two levels: **Configuration** and **Data Access**. When managing access at the **Configuration** level, you are controlling who can view or modify the *configuration* of the graphmart, such as who can edit the graphmart settings on the Overview tab, who can enable, disable, modify, or add layers, and who can view or modify the graphmart permissions. The **Data Access** configuration controls who can view the data that is contained within the graphmart.

## Permission Inheritance

When assigning Configuration and Data Access permissions at the graphmart level, you can configure the graphmart to inherit the permissions from another artifact and/or pass on its permissions to additional artifacts. For example, you can configure one graphmart to pass its permissions to other graphmarts. Inheritance transmits all of the artifact's permissions for all users and groups.

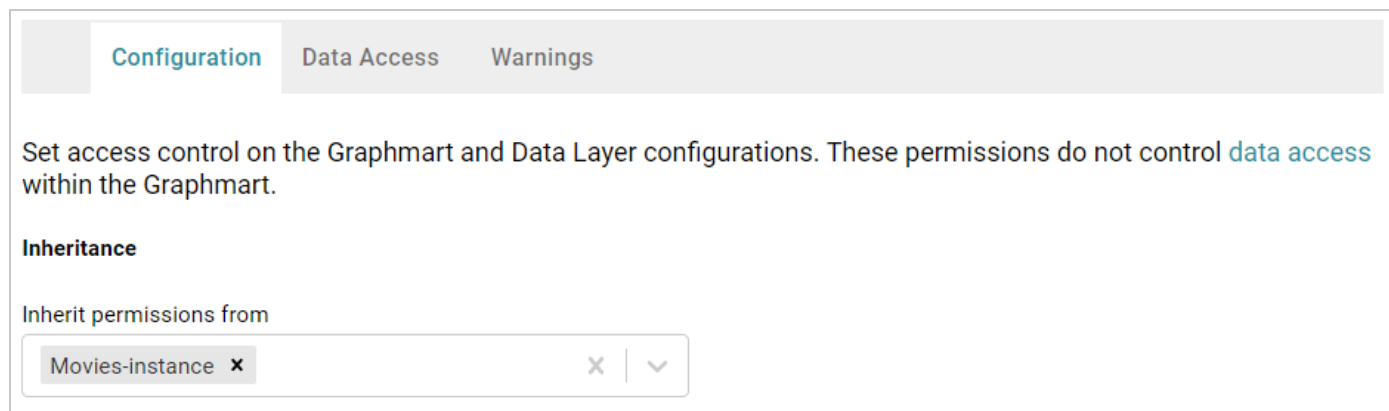
### Note

Since data layers are created in graphmarts, they inherit their permissions from the graphmart by default—with one exception: Layers with Load Dataset Steps inherit their Data Access permissions from the dataset. Data on Demand endpoints also inherit their permissions from the parent graphmart by default.

The inheritance settings are displayed at the top of the Configuration and Data Access tabs on the graphmart Sharing screen.

## Configuration Inheritance

The image below shows a graphmart Configuration tab with the default inheritance settings. The **Inherit permissions from** field shows that the graphmart inherits permissions from the schema instance that the graphmart was created from.



The screenshot shows the 'Configuration' tab of a graphmart's sharing screen. At the top, there are three tabs: 'Configuration' (selected), 'Data Access', and 'Warnings'. Below the tabs, a text block states: 'Set access control on the Graphmart and Data Layer configurations. These permissions do not control data access within the Graphmart.' Below this, the 'Inheritance' section is visible. It contains the label 'Inherit permissions from' and a dropdown menu. The dropdown menu is open, showing 'Movies-instance' as the selected option, with a close button (x) and a dropdown arrow (v) to its right.

## Data Access Inheritance

The image below shows the Data Access tab for the same graphmart. The **Graphmart Level View Permissions** are set to **Inherit from Graphmart** by default. And **Default Layer View Permissions (for new Layers)** is also set to **Inherit from Graphmart**.

Below the inheritance settings, the **Permissions Overview** provides a detailed view of the permission inheritance for each layer, view, and Data on Demand endpoint in the graphmart.



ConfigurationData AccessWarnings

Set access control to the data within the Graphmart. Data Access settings are inherited by default from either the Graphmart or the included Datasets. Inheritance and custom permissions may be set for all Data Layers. Warnings are displayed whenever the Data Access permissions are changed from the original permissions inherited from the Dataset. These permissions do not control the permissions to edit the [configuration](#) of the Graphmart or Data Layers.

**Viewing Data in Layers**

Graphmart-Level View Permissions

Inherit from Graphmart [CHANGE](#)

Default Layer View Permissions (for new Layers)

Inherit from Graphmart [CHANGE](#)

**Permissions Overview**

Title	Type	Warnings	Inherits ACLs from	Roles that can view	Actions	
Movies	Graphmart		Graphmart Configuration	Everyone, sysadmin,		
Movies Layer	Layer		Layer Configuration	Everyone, sysadmin,		
Create dataset	Layer		Layer Configuration	Everyone, sysadmin,		

### User and Group Configuration Permissions

Graphmart Configuration permissions control who can view or modify the graphmart settings, who can enable, disable, modify, or add data layers, and who can view or modify the graphmart permissions. There are three predefined permission sets that can be applied to a user or group. The permission sets include a combination of six permissions. You also have the option to customize the set of permissions that are applied to a user or group.

The table below lists the predefined permission sets and describes the privileges that are granted for each permission that is part of the set:

Set	Permission	Allows a user to:
View	View	<ul style="list-style-type: none"><li>• See the graphmart in the Anzo application.</li><li>• Copy the graphmart URI from the Overview tab.</li><li>• Copy data layer URIs from the data layers tab.</li><li>• See the existing Data on Demand endpoints on</li></ul>

Set	Permission	Allows a user to:
		<p>the Data on Demand tab.</p> <ul style="list-style-type: none"> <li>• View and clone the dataset editions that are included in the graphmart.</li> <li>• Reload and refresh the graphmart.</li> <li>• Create and import graphmart versions.</li> </ul>
	<b>Meta View</b>	<ul style="list-style-type: none"> <li>• This permission relates only to the graphmart Sharing tab. A user with this permission can see the Sharing tab, but they cannot modify, add, or remove permissions.</li> </ul>
<b>Modify</b>	<p>In addition to the <b>View</b> and <b>Meta View</b> permissions described above, the <b>Modify</b> set includes the <b>Add/Edit</b> and <b>Delete</b> permissions described below.</p>	
	<b>Add/Edit</b>	<ul style="list-style-type: none"> <li>• Rename the graphmart and edit the description.</li> <li>• Create Data on Demand endpoints.</li> <li>• Add datasets and data sources to the graphmart.</li> <li>• Enable, disable, add, or edit layers and steps.</li> <li>• Activate and deactivate the graphmart.</li> </ul>
	<b>Delete</b>	<ul style="list-style-type: none"> <li>• Remove datasets from the graphmart.</li> <li>• Delete data layers and steps from the graphmart.</li> <li>• Cannot delete the graphmart.</li> </ul>
<b>Admin</b>	<p>In addition to the <b>View</b>, <b>Meta View</b>, <b>Add/Edit</b>, and <b>Delete</b> permissions described above, the <b>Admin</b> set includes the <b>Meta Add/Edit</b> and <b>Meta Delete</b> permissions described below.</p>	
	<b>Meta Add/Edit</b>	<ul style="list-style-type: none"> <li>• This permission relates only to the graphmart Sharing tab. A user with this permission can modify the sharing settings by adding permissions to a user or group.</li> </ul>

Set	Permission	Allows a user to:
	<b>Meta Delete</b>	<ul style="list-style-type: none"> <li>• Modify the sharing settings by removing permissions from a user or group.</li> <li>• Delete the graphmart.</li> </ul>

## Related Topics

[Configuring Graphmart and Data Layer Permissions](#)

### Configuring Graphmart and Data Layer Permissions

Follow the instructions below to configure permissions at the graphmart and data layer levels. Graphmart and layer sharing is managed at the **Configuration** and **Data Access** levels. When managing access at the **Configuration** level, you are controlling who can view or modify the *configuration* of the graphmart, such as who can edit the graphmart settings and enable, disable, modify, or add data layers. The **Data Access** configuration controls who can view the data that is contained within the graphmart.

#### Tip

See [Introduction to Graphmart and Data Layer Sharing](#) for background information on graphmart sharing concepts and an introduction to the Sharing user interface.

### Sharing Configuration Access

Follow the steps below if you want to modify the configuration-level access for a graphmart.

1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing graphmarts. For example:

Search

Add Graphmart

<div></div>	Title	Description	Status	Statements	Creator	Actions
	<div></div> Books		<div></div> Ready to use	156	System Administrator	<div></div> <div></div>
	<div></div> Flights		<div></div> Ready to use	265,339	System Administrator	<div></div> <div></div>
	<div></div> Movies		<div></div> Ready to use	104,166	System Administrator	<div></div> <div></div>
	<div></div> Northwind		<div></div> Ready to use	131,386	System Administrator	<div></div> <div></div>
	<div></div> Tickets		<div></div> Ready to use	3,696,548	System Administrator	<div></div> <div></div>

Rows per page: 251-5 of 5

- Click the name of the graphmart for which you want to configure permissions. Then click the **Sharing** tab. The Sharing screen is displayed and the **Configuration** tab is selected. For example:

**Northwind**  
Derived from: Northwind\_20221123174519 DEV 1

INACTIVE ACTIVE Ready to use AZG Static

< Data Layers Dashboards Data on Demand Versions Discussion **Sharing** >

**Configuration** Data Access Warnings

Set access control on the Graphmart and Data Layer configurations. These permissions do not control [data access](#) within the Graphmart.

**Inheritance**

Inherit permissions from

northwind-instance

**User and Group Permissions**

Permissions defined for specific users or groups are applied in addition to any inherited permissions.

Search users, roles or groups

Everyone View

System Administrator Admin

Define the permissions of this artifact.  
Start by selecting a user or group on the left.

- If you want to change how the Configuration permissions are inherited, use the **Inherit permissions from** field at the top of the screen. To apply all of the permissions from another artifact to this one, select the artifact to inherit from in the **Inherit permissions from** field.

#### Tip

For more information about permission inheritance at the graphmart level, see [Permission Inheritance](#).

- To modify Configuration access to this graphmart with a particular user or group, type a value in the **Search users, roles or groups** field to find and display the user or group. The resulting list shows the current permission level that is set for each user or group in the search results. For example, the image below shows the current permissions for the IT group (None):

### User and Group Permissions

Permissions defined for specific users or groups are applied in addition to any inherited permissions.

Information Technology (IT)
None

Define the permissions of this artifact.  
Start by selecting a user or group on the left.

5. Select the user or group for which you want to configure permissions. The permissions settings are displayed on the right side of the screen. For example:

### User and Group Permissions

Permissions defined for specific users or groups are applied in addition to any inherited permissions.

Information Technology (IT)
None

Information Technology (IT)
Set permission for Information Technology (IT)

Permissions
☐ View
☐ Modify
☐ Admin
☐ Custom

Add/Edit		✓	✓	✓
View	✓	✓	✓	✓
Delete		✓	✓	✓
Meta Add/Edit			✓	✓
Meta View	✓	✓	✓	✓
Meta Delete			✓	✓

6. To assign a predefined set of permissions, click the **View**, **Modify**, or **Admin** radio button to assign that level of access to the selected user or group. Refer to [User and Group Configuration Permissions](#) for details about the permissions sets. For example, the image below gives **Admin** permissions to users in the IT group:

### User and Group Permissions

Permissions defined for specific users or groups are applied in addition to any inherited permissions.

Information Technology (IT)
Admin

#### Information Technology (IT)

Set permission for Information Technology (IT)

Permissions

☐ View
☐ Modify
☒ Admin
☐ Custom

Add/Edit		✓	✓	<input checked="" type="checkbox"/>
View	✓	✓	✓	<input checked="" type="checkbox"/>
Delete		✓	✓	<input checked="" type="checkbox"/>
Meta Add/Edit			✓	<input checked="" type="checkbox"/>
Meta View	✓	✓	✓	<input checked="" type="checkbox"/>
Meta Delete			✓	<input checked="" type="checkbox"/>

If you want to customize the permissions, click the **Custom** radio button and then select or deselect the permissions checkboxes. To clear permissions for a user or group, click the trashcan icon () next to the name.

## Sharing Data Access

Follow the steps below if you want to modify permissions at the Data Access level for a Graphmart.

1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing graphmarts. For example:

Search

Add Graphmart

<div></div>	Title	Description	Status	Statements	Creator	Actions
	<div></div> Books		<div></div> Ready to use	156	System Administrator	<div></div> <div></div>
	<div></div> Flights		<div></div> Ready to use	265,339	System Administrator	<div></div> <div></div>
	<div></div> Movies		<div></div> Ready to use	104,166	System Administrator	<div></div> <div></div>
	<div></div> Northwind		<div></div> Ready to use	131,386	System Administrator	<div></div> <div></div>
	<div></div> Tickets		<div></div> Ready to use	3,696,548	System Administrator	<div></div> <div></div>

Rows per page: 251-5 of 5

- Click the name of the graphmart for which you want to configure permissions. Then click the **Sharing** tab. The Sharing screen is displayed and the **Configuration** tab is selected. Click the **Data Access** tab. For example:



ConfigurationData AccessWarnings

Set access control to the data within the Graphmart. Data Access settings are inherited by default from either the Graphmart or the included Datasets. Inheritance and custom permissions may be set for all Data Layers. Warnings are displayed whenever the Data Access permissions are changed from the original permissions inherited from the Dataset. These permissions do not control the permissions to edit the [configuration](#) of the Graphmart or Data Layers.

### Viewing Data in Layers

Graphmart-Level View Permissions

Inherit from Graphmart [CHANGE](#)

Default Layer View Permissions (for new Layers)

Inherit from Graphmart [CHANGE](#)

### Permissions Overview

Q Search

Title	Type	Warnings	Inherits ACLs from	Roles that can view	Actions	
Movies	Graphmart		Graphmart Configuration	Everyone, sysadmin,		
Movies Layer	Layer		Layer Configuration	Everyone, sysadmin,		
Create dataset	Layer		Layer Configuration	Everyone, sysadmin,		

- If you want to change how the Data Access permissions are inherited, use the fields at the top of the screen:
  - Graphmart-Level View Permissions** controls who can view the data within the entire graphmart.
  - Default Layer View Permissions (for new Layers)** controls who can view the data within the data layers.

#### Tip

For more information about permission inheritance at the graphmart level, see [Permission Inheritance](#).

- To change the permissions for an individual layer, Data on Demand endpoint, or another graphmart component that is listed in the Permissions Overview, click the Edit icon () in the Actions column in the in the row for that component.

Changes to graphmart and layer permissions take effect immediately. Users do not need to log out and log back in, and affected graphmarts do not need to be reloaded or refreshed.

# Profiling Data Sources, Datasets, and Graphmarts

To help you explore your data and assess its quality, Anzo provides the option to generate a data profile for data sources, datasets, and graphmarts. Creating a profile runs several metrics against the data and reports statistics at the class, property, and instance levels. Data profile metrics measure data quality, perform data discovery, and can help you decide on the types of analytics to run. The topics in this section provide instructions for generating data profiles and describe each of the metrics that Anzo runs.

- Generating a Source Data Profile ..... 687
- Generating a Dataset Data Profile ..... 692
- Generating a Graphmart Data Profile ..... 697
- Data Profiling Metrics ..... 702

## Generating a Source Data Profile

To help users assess the quality of the data coming from a data source, Anzo provides the ability to generate a Data Profile for each source. Creating a profile runs several metrics against the source data and reports statistics to help you assess data quality and perform data discovery. In addition to running the data quality analytics, you also have the option to configure the profile to generate a list of foreign key suggestions between schema tables that include the same column.

### Note

AnzoGraph uses the Graph Data Interface (GDI) Java plugin to connect directly to the data sources that you profile. The GDI plugin is included in the AnzoGraph installation. For file-based data sources, make sure the source files are available to AnzoGraph on the shared file store. For database sources, if you have configured custom drivers to access those sources in Anzo, the same drivers need to be added to AnzoGraph. For instructions, see [Deploy Optional Drivers for Accessing Custom Database Sources](#) in the Deployment Guide.

Follow the instructions below to generate and review data profile metrics for a data source.

1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing sources. For example:

Data Sources   Schemas   Mappings   Pipelines										
<div> <div> <div> <div></div> <div>Search</div> </div> <div> <div></div> <div></div> </div> </div> <div>Add Data Source</div> </div>										
<input type="checkbox"/>	Title		Description	Type	Schema	Creator	Created Date	Updated Date	Actions	
<input type="checkbox"/>	DB			Database Data...	emrddb, new_...	System Admin...	Dec 8, 2022	Dec 8, 2022		
<input type="checkbox"/>	Movies			CSV Data Sour...	Movies	System Admin...	Dec 8, 2022	Dec 8, 2022		
<input type="checkbox"/>	Tickets			CSV Data Sour...	Tickets	System Admin...	Dec 9, 2022	Dec 12, 2022		
<div> <div>Rows per page: 25</div> <div>1-3 of 3</div> <div> <div></div> <div></div> </div> </div>										

- Click the source that you want to profile. Anzo displays the Tables tab for the source, which lists the schema and table details. For example, the images below show the Tables tab for a database source:

DB-QA  
Not Versioned

Create Graphmart Profile Data Add To Dictionary Ingest

Overview **Tables** Versions Discussion Sharing

Schemas

Create Schemas From Query Import Schemas

Search

<input type="checkbox"/>	Schema Name	Table Name	Type	Actions
	northwind			
		Alphabetical list ...		
		Categories		
		Category Sales f...		
		Current Product ...		

Schema Details  
Select a table on the left to see details here.

- Click the **Profile Data** button at the top of the screen. If the source is a database with multiple schemas, you are prompted to select the schema to profile. Select a schema and click **OK**. The Data Profile dialog box is displayed:

**Profile Data**

Generate metrics for data quality and discovery at the class, property, and instance level.  
*Data Profiling can be a time-consuming operation for larger volumes of data.*

Profiling method

☒ **Primary**  
Executes a subset of metrics focused on data quality.

☐ **Advanced**  
Executes all available metrics (including Primary ones) for further data discovery and analytics.

☐ Suggest Foreign keys

[LEARN MORE](#)

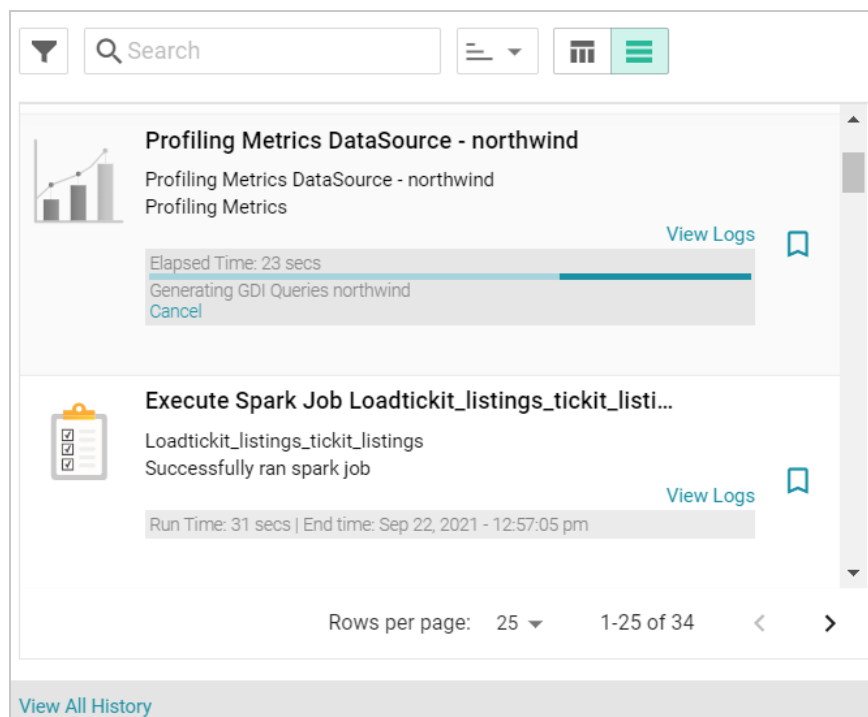
CANCEL RUN PROFILING

- On the Data Profile screen, choose the Profiling Method to use. The **Primary** method focuses on data quality type analysis. The **Advanced** method includes the Primary data quality analytics plus several advanced metrics for deeper data discovery and analysis. For details about the Primary and Advanced metrics, see [Data Profiling Metrics](#).

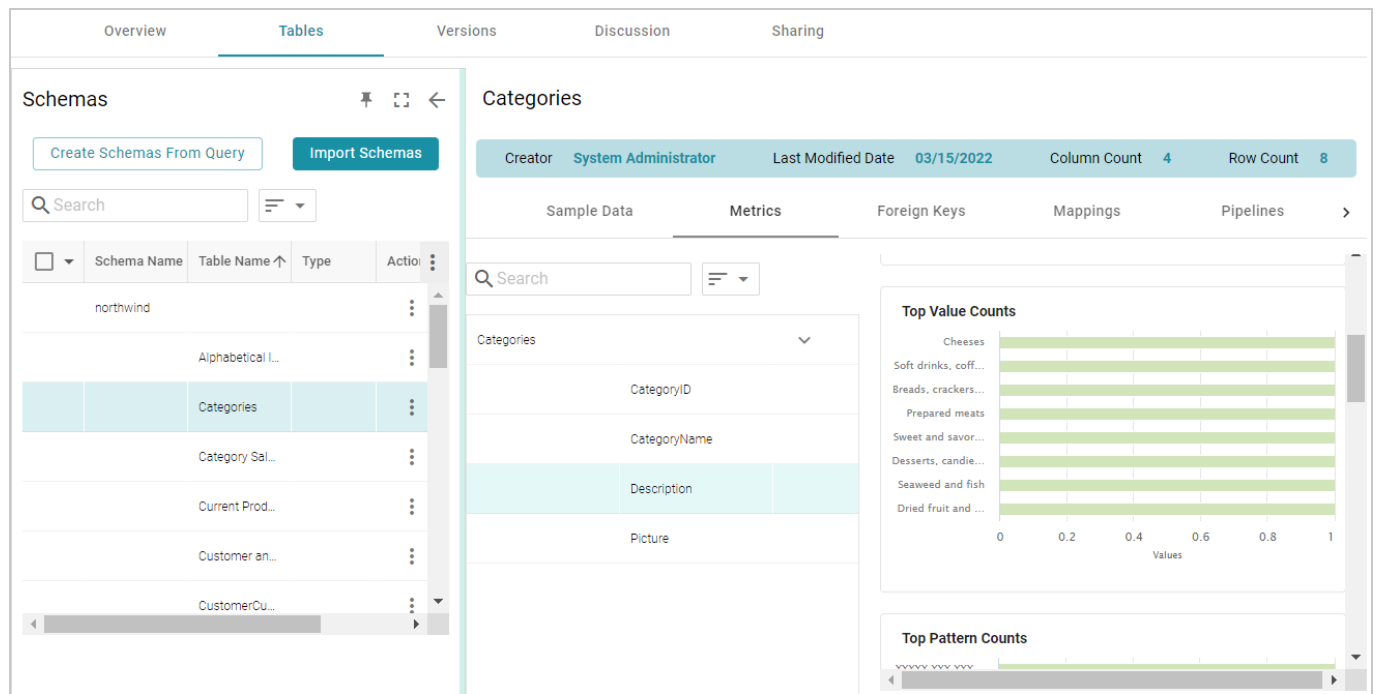
## Note

To run the Advanced metrics, the optional AnzoGraph C++ extensions and dependencies must be installed. If you use dynamic, K8s-based deployments of AnzoGraph, the extensions are included. If you installed a static AnzoGraph instance with the installer, the C++ extensions are optional and are only installed if **yes** was specified for the `Do you want to install C++ UDXs packaged with AnzoGraph DB?` prompt. For information about the C++ dependencies, see [Install the Optional C++ Extension Dependencies](#) in the Deployment Guide.

5. If you would like the profiling process to generate a list of Foreign Key suggestions that you can review and activate later, select the **Suggest Foreign Keys** checkbox.
6. Click **Run Profiling** at the bottom of the screen. Anzo provisions the source data into a temporary graphmart so that the metrics can be run by AnzoGraph. The profiling process may take several minutes, especially for large volumes of data. You can check the status of the process in the Activity Log. The Activity Log also presents the option to stop the profiling process by clicking **Cancel** under the progress bar for the task. For example:



7. When the process is complete, select a row in the Schema for the profiled data source. Then click the **Metrics** tab on the right side of the screen. Anzo displays the metrics for the selected table. For example:



To view metrics for a property in the table, select the property in the middle of the screen.

For more information about the metrics that are displayed, see [Data Profiling Metrics](#). For information about data profiling for datasets and graphmarts, see [Generating a Dataset Data Profile](#) and [Generating a Graphmart Data Profile](#).

# Generating a Dataset Data Profile

Similar to generating a profile for a graphmart, you can generate a data profile for a dataset in the Datasets catalog. Generating a dataset profile helps users perform data discovery, assess the quality of the onboarded data, and decide whether to use the dataset in a graphmart. The reports can also assist users in determining the types of data layer steps to create and writing the queries to include in the steps.

Important

To generate a dataset profile, AnzoGraph must be online. If you have dynamic AnzoGraph deployments enabled, AnzoGraph will be provisioned automatically when the profile is generated.

1. In the Anzo application, expand the **Blend** menu and click **Datasets**. Anzo displays the Datasets screen, which lists the catalog of datasets. For example:

Search

Add Dataset

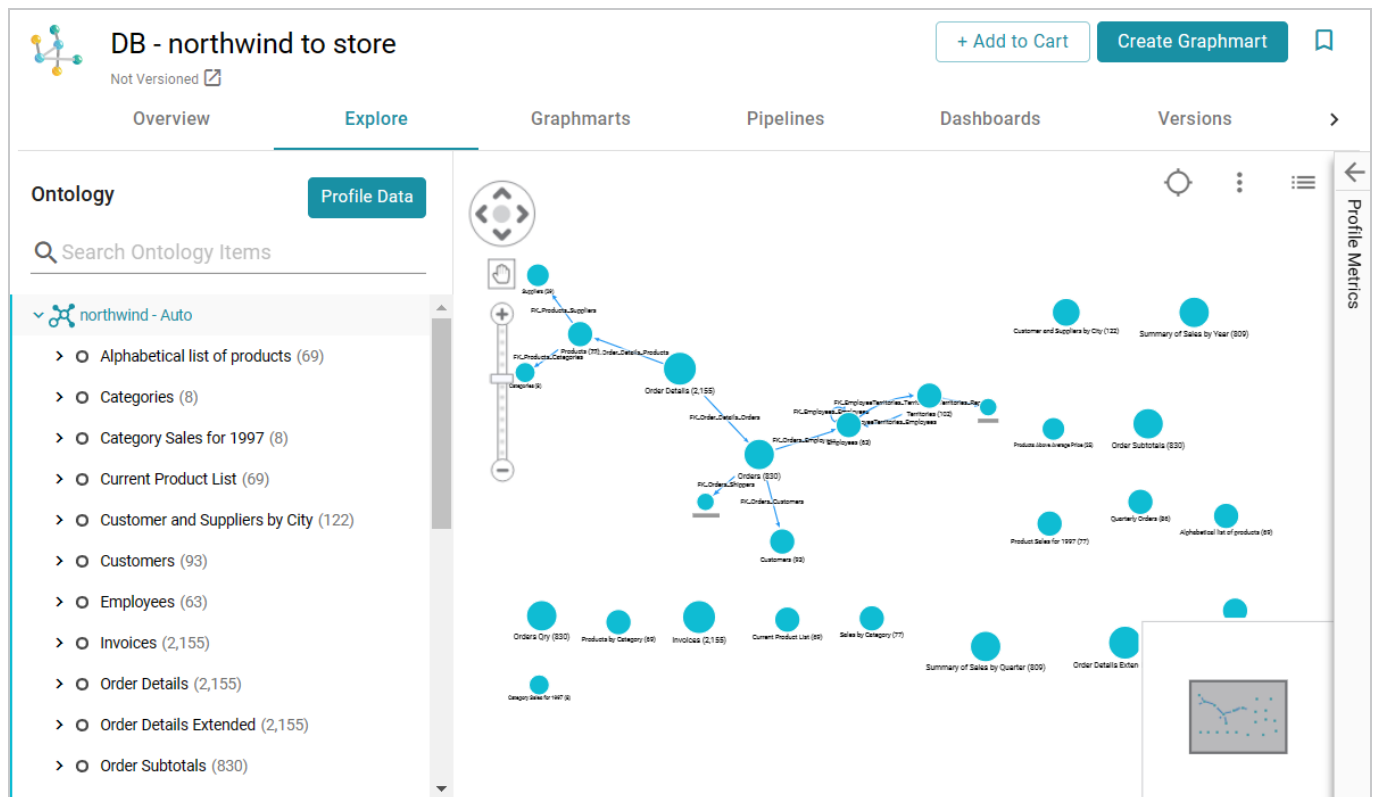
<div><div></div><div></div></div>	Title	↑	Description	Updated Date	Actions	
	<div><div></div>Books</div>		Exported Books dataset		<div><div></div><div></div></div>	<div><div></div><div></div></div>
	<div><div></div>Flights</div>		Exported Flights dataset		<div><div></div><div></div></div>	<div><div></div><div></div></div>
	<div><div></div>Movies</div>		Exported Movies graphmart		<div><div></div><div></div></div>	<div><div></div><div></div></div>
	<div><div></div>Northwind</div>		Exported Northwind graphmart		<div><div></div><div></div></div>	<div><div></div><div></div></div>

Rows per page: 25

1-4 of 4

2. On the Datasets screen, click the name of the dataset that you want to generate a profile for. Anzo displays the Explore tab for the dataset. For example:





- Click the **Profile Data** button on the top left of the screen. The Data Profile dialog box is displayed:

### Profile Data

Generate metrics for data quality and discovery at the class, property, and instance level.  
*Data Profiling can be a time-consuming operation for larger volumes of data.*

Profiling method

☒ **Primary**  
 Executes a subset of metrics focused on data quality.

☐ **Advanced**  
 Executes all available metrics (including Primary ones) for further data discovery and analytics.

☐ Suggest Foreign keys

[LEARN MORE](#)

CANCEL [RUN PROFILING](#)

- On the Data Profile screen, choose the Profiling Method to use. The **Primary** method focuses on data quality type analysis. The **Advanced** method includes the Primary data quality analytics plus several advanced metrics for deeper data discovery and analysis. For details about the Primary and Advanced metrics, see [Data Profiling Metrics](#).


### Note

To run the Advanced metrics, the optional AnzoGraph C++ extensions and dependencies must be installed. If you use dynamic, K8s-based deployments of AnzoGraph, the extensions are included. If you installed a static AnzoGraph instance with the installer, the C++ extensions are optional and are only installed if **yes** was specified for the `Do you want to install C++ UDXs packaged with AnzoGraph DB?` prompt. For information about the C++ dependencies, see [Install the Optional C++ Extension Dependencies](#) in the Deployment Guide.

5. Click **Run Profiling** to start generating the profile.

### Note

If the dataset is used in a graphmart that is active, Anzo displays the Select Graphmart dialog box (shown below), which prompts you to choose whether the online dataset can be used for running the profiling queries or whether to provision another temporary graphmart for the dataset.

A screenshot of a dialog box titled "Select Graphmart". The dialog has a teal header bar with the title. Below the header is a white area containing a text input field with the placeholder text "Select Graphmart" and a downward arrow icon on the right. At the bottom right of the dialog are two buttons: "CANCEL" and "OK".

Select Graphmart

Select Graphmart

CANCEL OK

To use the dataset in the graphmart that is online, click the **Select Graphmart** drop-down list and select the graphmart name. If you want Anzo to provision a temporary graphmart instead, select **Don't reuse the Graphmart**.

The profiling process may take several minutes, especially for large volumes of data. You can check the status of the process in the Activity Log. The Activity Log also presents the option to stop the profiling process by clicking **Cancel** under the progress bar for the task. For example:

**Provisioning Graphmart - Movies graphmart.**  
Provisioning Graphmart - Movies graphmart.  
Successfully activated the graphmart on AnzoGraph  
[View Logs](#)  
Run Time: 2 secs | End time: Nov 2, 2022 - 2:25:26 pm

**Profiling Metrics LinkedDataset - Movies**  
Profiling Metrics LinkedDataset - Movies  
Profiling Metrics  
[View Logs](#)  
Elapsed Time: 7 secs  
Updating Dataset metrics for: Movies  
[Cancel](#)

Rows per page: 25 1-25 of 64

[View All History](#)

- To display the metrics, select a node or edge in the graph view in the middle of the screen or expand the model on the left side of the screen and select a class or property. For example:

**DB - northwind to store**  
Not Versioned [\[icon\]](#)

[+ Add to Cart](#) [Create Graphmart](#) [\[icon\]](#)

Overview **Explore** Graphmarts Pipelines Dashboards Versions Discussion

**Ontology** [Profile Data](#)

Search Ontology Items

- Order Subtotals (830)
- Orders (830)**
  - CustomerID
  - EmployeeID
  - FK\_Orders\_Customers
  - FK\_Orders\_Employees
  - FK\_Orders\_Shippers
  - Freight
  - OrderDate
  - OrderID
  - RequiredDate
  - ShipAddress

**Profile Metrics**  
Last profiled 4 minutes ago  
northwind - Auto / Orders (830)

**Metrics** Sample Data

**CustomerID** [\[icon\]](#) [\[icon\]](#) [\[icon\]](#)

100% of instances have values

CustomerID	Values
SAVEA	30
ERNSH	28
QUICK	25
FOLKO	18
HUNGO	18
RATTC	18
HILAA	18
BERGS	18
BONAP	18
WARTH	15

You can click the **Expand** icon ([\[icon\]](#)) on the right side of the screen to collapse the graph view and expand the metrics view.

For more information about the metrics that are displayed, see [Data Profiling Metrics](#). For information about data profiling for data sources and graphmarts, see [Generating a Source Data Profile](#) and [Generating a Graphmart Data Profile](#).

# Generating a Graphmart Data Profile

Similar to generating a data profile for a dataset, you can profile a graphmart, which may include multiple datasets. When metrics are generated for graphmarts, Anzo profiles the data that results from all of the enabled layers and reports metrics for the classes and properties in the model as well as statistics about the values for the properties. Generating a graphmart profile helps users perform data discovery, assess the quality of the data, and decide on the types of analytics to perform.

Important

To generate a graphmart data profile, AnzoGraph must be online. If you have dynamic AnzoGraph deployments enabled, AnzoGraph will be provisioned automatically when the profile is generated.

- 1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing graphmarts. For example:

Search

Add Graphmart

<div></div>	Title	Description	Status	Statements	Creator	Actions	
	Books		<div>Ready to use</div>	156	System Administrator	<div></div>	<div></div>
	Flights		<div>Ready to use</div>	265,339	System Administrator	<div></div>	<div></div>
	Movies		<div>Ready to use</div>	104,166	System Administrator	<div></div>	<div></div>
	Northwind		<div>Ready to use</div>	131,386	System Administrator	<div></div>	<div></div>
	Tickets		<div>Ready to use</div>	3,696,548	System Administrator	<div></div>	<div></div>

Rows per page:

25

1-5 of 5

- 2. On the Graphmarts screen, click the name of the graphmart for which you want to generate metrics. Anzo displays the Overview for that graphmart. For example:



4. If necessary, activate the graphmart, and then click the **Profile Data** button on the left side of the screen. The Data Profile dialog box is displayed:


### Profile Data

Generate metrics for data quality and discovery at the class, property, and instance level.  
*Data Profiling can be a time-consuming operation for larger volumes of data.*

**Profiling method**

☒ **Primary**  
Executes a subset of metrics focused on data quality.

☐ **Advanced**  
Executes all available metrics (including Primary ones) for further data discovery and analytics.

[LEARN MORE](#) 

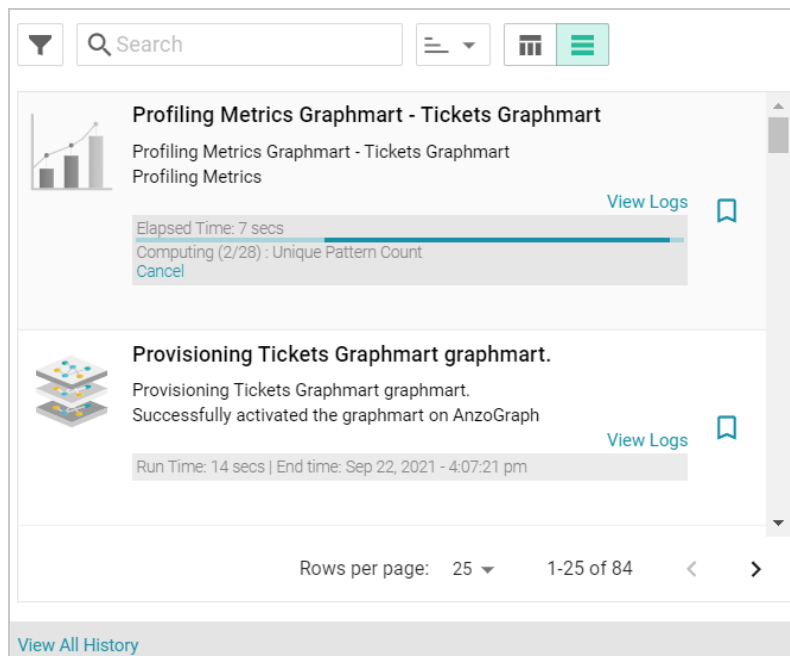
[CANCEL](#) [RUN PROFILING](#)

5. On the Data Profile screen, choose the Profiling Method to use. The **Primary** method focuses on data quality type analysis. The **Advanced** method includes the Primary data quality analytics plus several advanced metrics for deeper data discovery and analysis. For details about the Primary and Advanced metrics, see [Data Profiling Metrics](#).

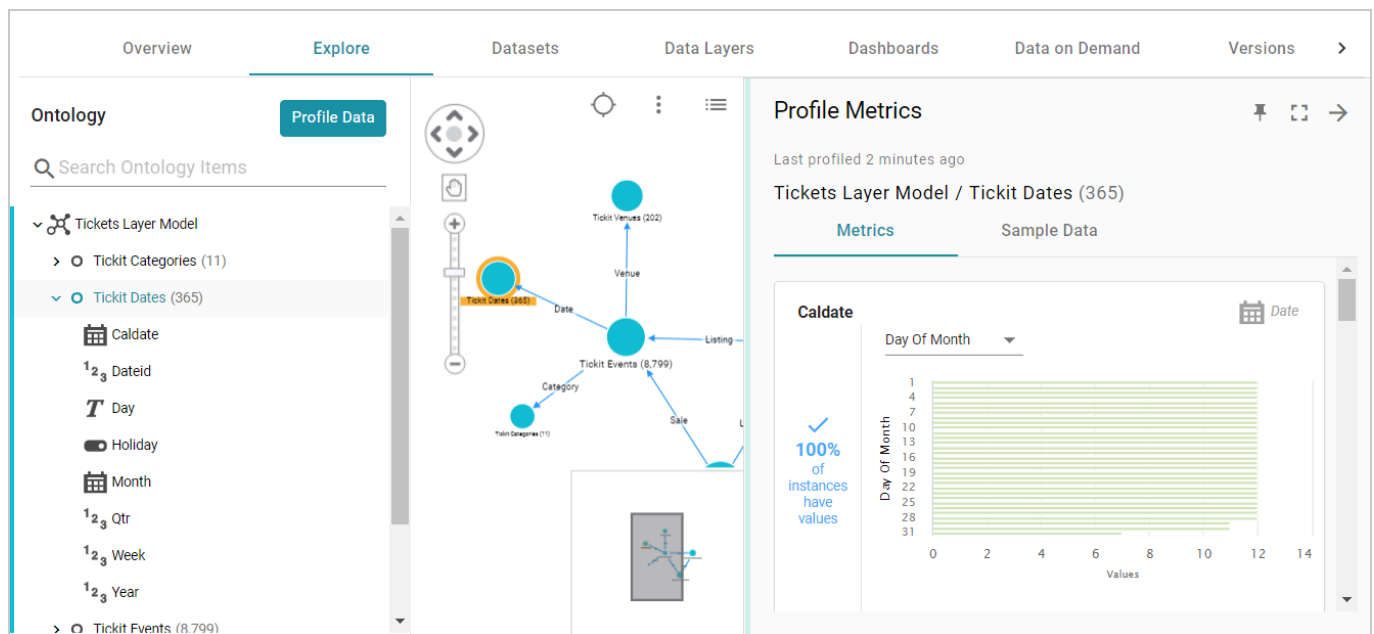
#### Note


To run the Advanced metrics, the optional AnzoGraph C++ extensions and dependencies must be installed. If you use dynamic, K8s-based deployments of AnzoGraph, the extensions are included. If you installed a static AnzoGraph instance with the installer, the C++ extensions are optional and are only installed if **yes** was specified for the `Do you want to install C++ UDXs packaged with AnzoGraph DB?` prompt. For information about the C++ dependencies, see [Install the Optional C++ Extension Dependencies](#) in the Deployment Guide.

6. Click **Run Profiling** to start generating the profile. The process may take several minutes, especially for large volumes of data. You can check the status of the process in the Activity Log. The Activity Log also presents the option to stop the profiling process by clicking **Cancel** under the progress bar for the task. For example:



- Once the profiling is complete, the Profile Metrics panel is expanded on the Explore tab. To populate the panel, click a class or property in the Ontology or a class in the graph view in the middle of the screen. For example:



You can click the Expand icon (  ) on the right side of the screen to collapse the graph view and expand the metrics view.



For more information about the metrics that are displayed, see [Data Profiling Metrics](#). For information about data profiling for datasets and data sources, see [Generating a Dataset Data Profile](#) and [Generating a Source Data Profile](#).

## Data Profiling Metrics

When a Data Profile is generated for a Data Source, Dataset, or Graphmart, Anzo runs several metrics that can help users measure data quality and perform data discovery at the class, property, and instance level. The Data Profile metrics are grouped into two categories: a **Primary** category that focuses on data quality type analysis, and an **Advanced** category that includes the Primary data quality analytics plus several advanced metrics for deeper data discovery and analysis.

The lists below give a summary of the Primary and Advanced metrics. For more information and sample images of the visualizations that are generated, click a metric name.

### Primary

- [Row Count](#): Reports the total row (instance) count per class.
- [Property Count](#): Reports the total number of properties per class.
- [Absent Property](#): For each property, reports the total number of instances that do not have a value.
- [Empty Property](#): For each **string** property, reports the total number of empty strings.
- [Sample Values](#): Returns sample values for each property.
- [Average](#): For each **numeric** property, computes the average of all values.
- [Sum](#): For each **numeric** property, computes the sum of all values.
- [Average String Length](#): For each **string** property, computes the average length of the strings.
- [Value Present](#): For each property, computes the percentage of instances that have at least one value.
- [Unique Pattern Count](#): For each property, counts the total number of unique value patterns.
- [Unique Values Count](#): For each property, counts the total number of unique values.
- [Median](#): For each **numeric** property, computes the median of all values.
- [Standard Deviation](#): For each **numeric** property, computes the standard deviation of all values.
- [Mode](#): For each **numeric** property, computes the mode of all of the values.
- [Presence](#): For each property in a class, reports the percentage of instances that have values vs. do not have values.
- [Top Value Counts](#): For each property, computes the top *N* most occurring values.
- [Bottom Value Counts](#): For each property, computes the *N* least occurring values.
- [Top Pattern Counts](#): For each property, computes the top *N* most common value patterns.
- [Bottom Pattern Counts](#): For each property, computes the *N* least common value patterns.

- **Range**: For **numeric** properties, reports the total range of values.
- **Value Types**: For each property, returns the data types for the instances.
- **DateTime Distribution By Year/Month/Day**: For **dateTime** properties, computes a histogram that shows the distribution of values by year, month, and day.

## Advanced

- **Pearson Skewness**: For each **numeric** property, computes the Pearson coefficient of skewness.
- **Geometric Mean**: For each **numeric** property, computes the geometric mean of all values.
- **Variance**: For each **numeric** property, computes the variance of all values.
- **Discrete Entropy**: For each property, computes the discrete entropy of all values.
- **Discrete Probability**: For each property, computes the discrete probability of all values.
- **String Length Range**: For each **string** property, reports the range of string lengths.
- **Unique Values**: For each property, computes the percentage of unique values.
- **Lower Case Strings**: For each **string** property, computes the percentage of values with all lower case characters.
- **Upper Case Strings**: For each **string** property, computes the percentage of values with all upper case characters.
- **Trivial Values**: For each **string** property, computes the percentage of instances that have one of the following values: **NA**, **N/A**, **NONE**, or **NULL**.

## Row Count

This metric computes the total row count per class. For Source profiles, the Row Count is displayed at the top of the screen under the class name. For example:

Categories					
Creator	System Administrator	Last Modified Date	09/27/2021	Column Count	Row Count 8

For Dataset and Graphmart profiles, the row count is in parenthesis at the top of the screen next to the class name, as shown in the image below.

Profile Metrics	
Last profiled an hour ago	
northwind - Auto / Categories	(8)

## Property Count

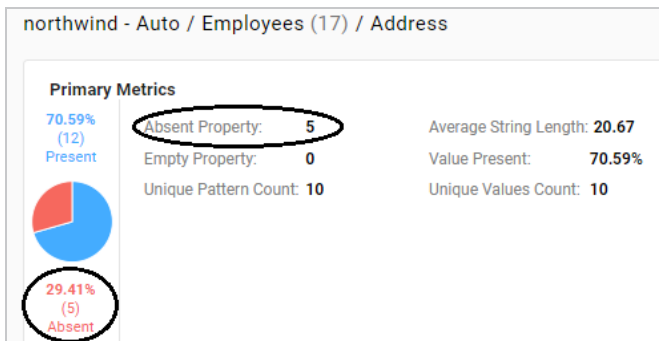
This metric computes the total number of properties per class. For Data Source profiles, the property count is labeled **Column Count**, as shown in the following image.

Categories					
Creator	System Administrator	Last Modified Date	09/27/2023	Column Count	4
				Row Count	8

For Dataset and Graphmart profiles, the property count is not shown.

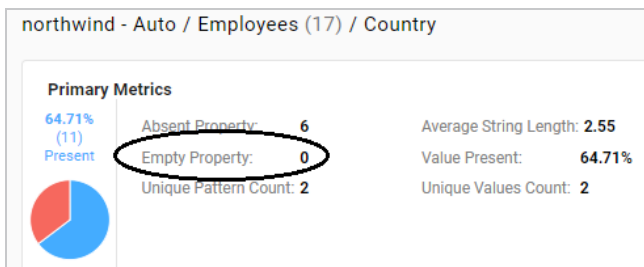
## Absent Property

For each property, this metric reports on the total number of instances that do not have a value for that property.



## Empty Property

For string properties, this metric reports the total number of empty strings.



## Sample Values

This metric returns sample values for each property.

Sample Values	
Beverages	Condiments
Confections	Dairy Products
Grains/Cereals	Meat/Poultry
Produce	Seafood


## Average

For each numeric property, this metric computes the average of all values.

Advanced Metrics	
Mean Absolute Deviation:	14.05
Discrete Entropy:	4.73
Range:	1 - 130
Galton Skewness:	0
Interquartile Range:	20
Average:	23.81
Mode:	20


## Sum

For each numeric property, this metric computes the sum of all values.


northwind - Auto / Products (77) / UnitPrice	
Primary Metrics	
 <b>100%</b> of instances have values	Absent Property: 0
	Range: 3 - 264
	Median: 19.5
	Value Present: 100%
	Sum: 2222.71
	Unique Values Count: 62

## Average String Length

For each string property, this metric computes the average length of the values.


northwind - Auto / Territories (53) / TerritoryDescription	
Primary Metrics	
 <b>100%</b> of instances have values	Absent Property: 0
	Empty Property: 0
	Unique Pattern Count: 17
	Value Present: 100%
	Unique Values Count: 52
	Average String Length: 50



northwind - Auto / Invoices (2,155) / Quantity		
Primary Metrics		
 <b>100%</b> of instances have values	Absent Property:	0
	Range:	1 - 130
	Median:	20
	Value Present:	100%
	Sum:	51317
	Unique Values Count:	55


## Standard Deviation

For each numeric property, this metric computes the standard deviation of all values.

northwind - Auto / Invoices (2,155) / Quantity		
Primary Metrics		
 <b>100%</b> of instances have values	Absent Property:	0
	Range:	1 - 130
	Median:	20
	Value Present:	100%
	Sum:	51317
	Unique Values Count:	55
	Empty Property:	0
	Average:	23.81
	Mode:	20
	Standard Deviation:	19.02
	Unique Pattern Count:	3

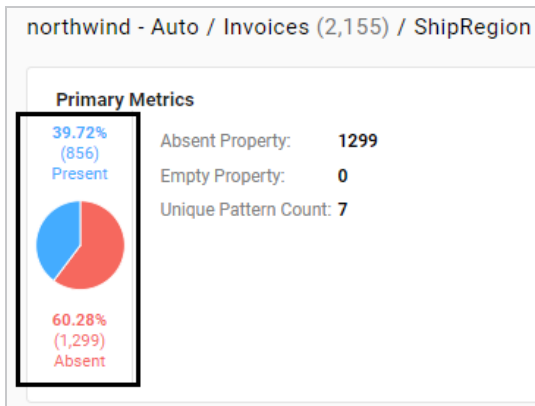
## Mode

For each numeric property, this metric computes the mode of all of the values.

northwind - Auto / Invoices (2,155) / Quantity		
Primary Metrics		
 <b>100%</b> of instances have values	Absent Property:	0
	Range:	1 - 130
	Median:	20
	Value Present:	100%
	Sum:	51317
	Unique Values Count:	55
	Empty Property:	0
	Average:	23.81
	Mode:	20
	Standard Deviation:	19.02
	Unique Pattern Count:	3

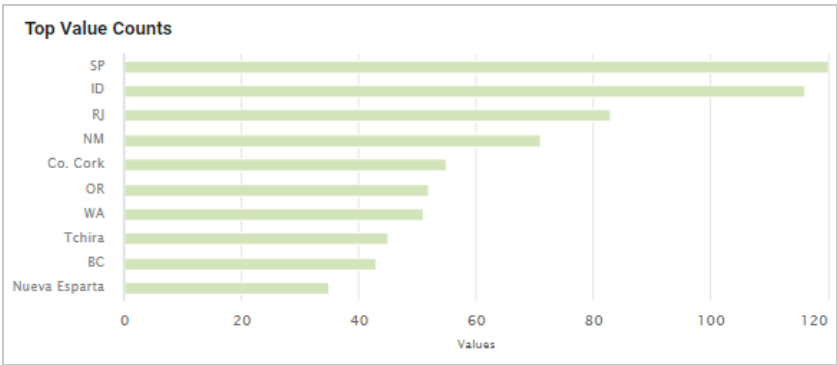
## Presence

The metric is available when viewing a class. For each property in the class, this metric reports on the percentage of instances that have values and the percentage of instances that do not have values.



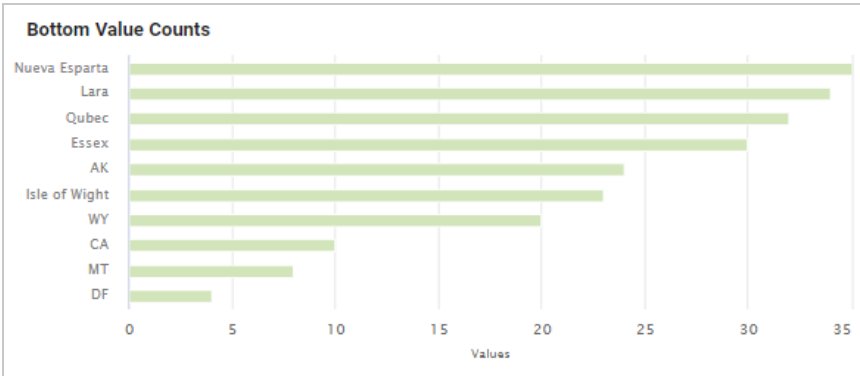
Top Value Counts

For each property, this metric computes the top *N* most occurring values.



Bottom Value Counts

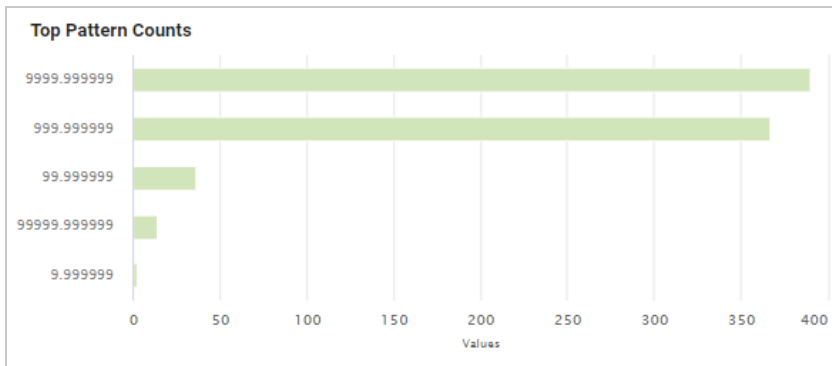
For each property, this metric computes the *N* least occurring values.



Top Pattern Counts

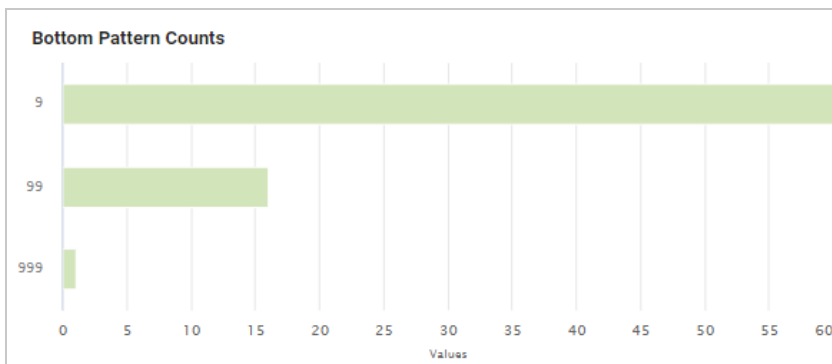
For each property, this metric computes the top *N* most common value patterns.





## Bottom Pattern Counts

For each property, this metric computes the  $N$  least common value patterns.



## Range

For numeric properties, this metric computes the range of all values.

northwind - Auto / Products (77) / UnitsOnOrder

✓

100%

of instances have values

Absent Property: 0

Range: 0 - 100

Median: 0

Value Present: 100%

Sum: 780

Unique Values Count: 10

Empty Property: 0

Average: 10.13

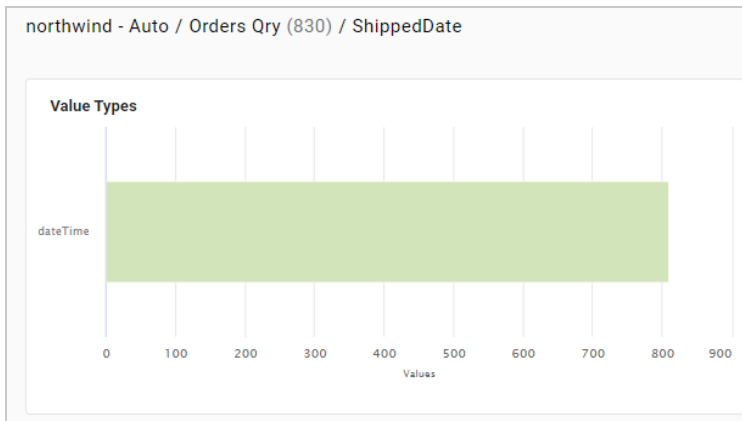
Mode: 0

Standard Deviation: 23.14

Unique Pattern Count: 3

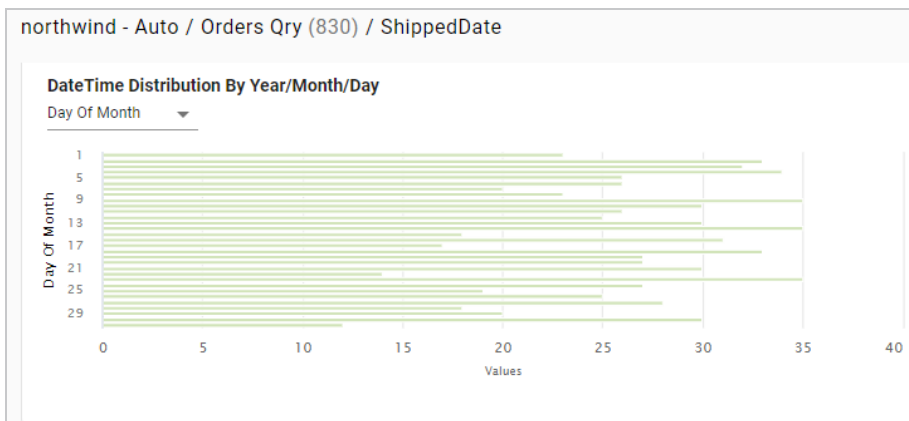
## Value Types

For each property, this metric returns the data types for the instances.



## DateTime Distribution By Year/Month/Day

For each dateTime property, this metric computes a histogram that shows the distribution of values by year, month, and day.



## Pearson Skewness

For each numeric property, this metric computes the Pearson coefficient of skewness to show the distribution of values. A value of 0 indicates no skew, a positive number indicates positive skew, and a negative number indicates negative skew.

Advanced Metrics	
Mean Absolute Deviation:	2.15
Unique Values:	0%
Geometric Mean:	3.59
Interquartile Range:	5
Negative Numbers:	0%
Pearson Skewness:	0.48
Zero Numbers:	0%

## Geometric Mean

For each numeric property, this metric computes the geometric mean of all of the values.

Advanced Metrics	
Mean Absolute Deviation:	2.15
Unique Values:	0%
Geometric Mean:	3.59
Interquartile Range:	5
Negative Numbers:	0%
Pearson Skewness:	0.48
Zero Numbers:	0%

## Variance

For each numeric property, this metric computes the variance of all values.

Advanced Metrics	
Mean Absolute Deviation:	16.15
Unique Values:	0%
Geometric Mean:	0
Interquartile Range:	20
Negative Numbers:	0%
Positive Numbers:	99.67%
Variance:	891.91

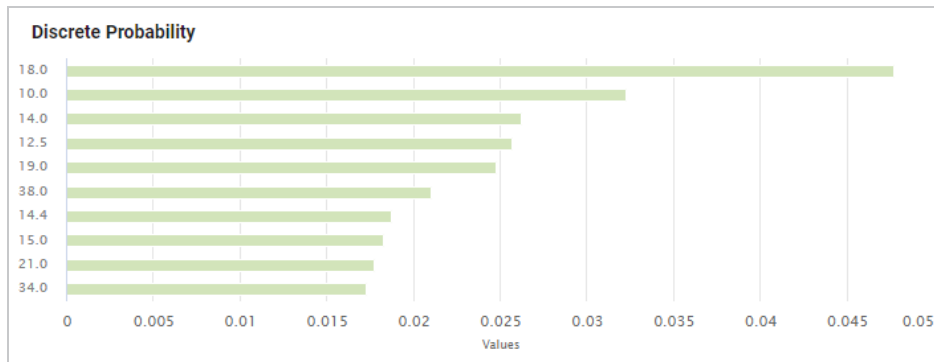
## Discrete Entropy

For each property, this metric computes the discrete entropy of all values.

northwind - Auto / Order Details Extended (2,155) / UnitPrice	
Advanced Metrics	
Mean Absolute Deviation:	16.15
Unique Values:	0%
Geometric Mean:	0
Interquartile Range:	20
Negative Numbers:	0%
Positive Numbers:	99.67%
Variance:	891.91
Discrete Entropy:	6.48
Galton Skewness:	0.36
Interquartile Mean:	19.75
Kurtosis:	33.98
Outliers Count:	97
Pearson Skewness:	0.78
Zero Numbers:	0.33%

## Discrete Probability

For each property, this metric computes the discrete probability of all values.



## String Length Range

For each string property, this metric reports the range of string value lengths.

northwind - Auto / Order Details Extended (2,155) / ProductName			
<b>Advanced Metrics</b>			
Discrete Entropy:	6.1	Unique Values:	0%
HTML Tags:	0%	Lower Case Strings:	0%
Non Printable Characters:	0%	Numeric Strings:	0%
String Length Range:	4 - 32	Spaces in values:	1
Trivial Values:	0%	Upper Case Strings:	0%

## Unique Values

For each property, this metric computes the percentage of unique values.

northwind - Auto / Order Details Extended (2,155) / ProductName			
<b>Advanced Metrics</b>			
Discrete Entropy:	6.1	Unique Values:	0%
HTML Tags:	0%	Lower Case Strings:	0%
Non Printable Characters:	0%	Numeric Strings:	0%
String Length Range:	4 - 32	Spaces in values:	1
Trivial Values:	0%	Upper Case Strings:	0%

## Lower Case Strings

For each string property, this metric computes the percentage of values that have all lower case characters.

northwind - Auto / Orders (830) / ShipAddress			
Advanced Metrics			
Discrete Entropy:	6.23	Unique Values:	0%
HTML Tags:	0%	Lower Case Strings:	2.29%
Non Printable Characters:	0%	Numeric Strings:	0%
String Length Range:	10 - 45	Spaces in values:	2
Trivial Values:	0%	Upper Case Strings:	0%

## Upper Case Strings

For each string property, this metric computes the percentage of values that have all upper case characters.

northwind - Auto / Orders (830) / ShipAddress			
Advanced Metrics			
Discrete Entropy:	6.23	Unique Values:	0%
HTML Tags:	0%	Lower Case Strings:	2.29%
Non Printable Characters:	0%	Numeric Strings:	0%
String Length Range:	10 - 45	Spaces in values:	2
Trivial Values:	0%	Upper Case Strings:	0%

## Trivial Values

For each string property, this metric computes the percentage of instances that have one of the following values: **NA**, **N/A**, **NONE**, or **NULL**.

northwind - Auto / Customers (93) / ContactName			
Advanced Metrics			
Discrete Entropy:	6.54	Unique Values:	100%
HTML Tags:	0%	Lower Case Strings:	0%
Non Printable Characters:	0%	Numeric Strings:	0%
String Length Range:	4 - 23	Spaces in values:	1
Trivial Values:	0%	Upper Case Strings:	0%

For additional metrics based on the type of data quality checks needed, contact Cambridge Semantics.

## Related Topics

[Profiling Data Sources, Datasets, and Graphmarts](#)

# Access

Once data has been onboarded, users have several options for accessing and analyzing the data. Anzo includes the Hi-Res Analytics application where users can create dashboards for exploring and visualizing the data without needing to have specialized query knowledge. The Query Builder enables users to find specific statements or run SPARQL queries. Users can also access data from the SPARQL endpoint or by using the Data on Demand service to generate data feeds for third-party business intelligence tools. The topics in this section provide information about the ways to access data in Anzo.

- [Accessing Data with Hi-Res Analytics Dashboards](#) .....715
- [Accessing Data with the Query Builder](#) .....906
- [Accessing Data on Demand Endpoints](#) .....922
- [Accessing Data from the SPARQL Endpoint](#) .....954
- [Accessing Data from the HTTP Client Interface](#) .....965
- [Sharing Access to Artifacts](#) .....973
- [SPARQL Best Practices and Query Templates](#) .....980
- [Anzo Function and Formula Reference](#) .....994

# Accessing Data with Hi-Res Analytics Dashboards

The Anzo Hi-Res Analytics application enables users to answer both ad-hoc and pre-determined questions using custom dashboards. Automated query generation eliminates the need to have specialized query knowledge, and users can traverse complex, multi-dimensional data by building exploratory charts, filters, tables, and network views.

The topics in this section provide guidance on getting started with Hi-Res Analytics dashboards and include instructions for creating and modifying dashboards and dashboard components.

- Introduction to Hi-Res Analytics ..... 716
- Getting Started: Exploring and Visualizing Data ..... 723
- Creating Dashboards ..... 735
- Creating Lenses ..... 769
- Creating Dashboard Filters ..... 817
- Combining Data from Multiple Classes ..... 871
- Calculating Values in Lenses and Filters ..... 878
- Searching for Text in Unstructured Documents ..... 884
- Sharing Access to a Dashboard or Lens ..... 893
- Cloning a Lens ..... 899
- Exporting a Lens ..... 901
- Deleting a Lens ..... 904

# Introduction to Hi-Res Analytics

Hi-Res Analytics dashboards enable you to create visual representations of your data using the latest in powerful web technologies. This introduction defines the fundamental concepts of working with dashboards and provides an overview of the Hi-Res Analytics user interface.

Tip

To fully leverage the advanced capabilities of Hi-Res Analytics, it helps to have skills working with Excel functions and formulas, SPARQL, and JavaScript and HTML. You can create dashboards without these skills but may not be able to take advantage of all functions.

- [Concepts and Vocabulary](#)
- [User Interface Overview](#)

## Concepts and Vocabulary

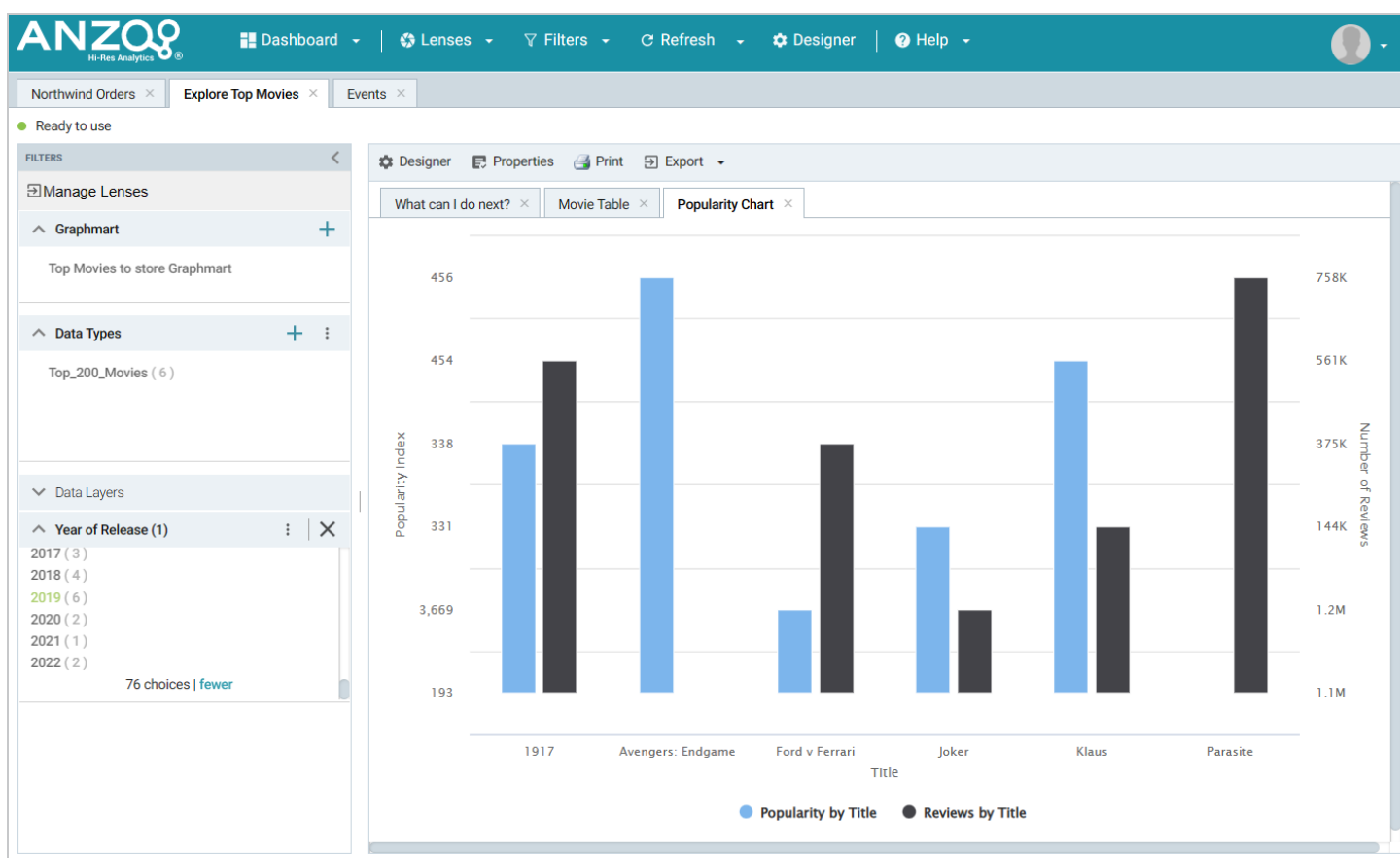
Term	Description
Dashboard	Dashboards enable you to view, analyze, and share your data. Data is viewed through lenses, such as tables and charts, that format the data for display. You can also apply filters to dashboards to refine the results.
Data Layer	Since Graphmarts can include multiple Data Layers, users can include or exclude the data from certain layers when creating or viewing dashboards.
Lens	Lenses are the structures that display your data. You must have at least one lens in a dashboard. You can reuse existing lenses or create new ones. For more information, see <a href="#">Creating Lenses</a> .
Filter	Filters narrow and further define the data to display. Dashboard-level filters apply globally to all lenses in a dashboard. Lens-level filters apply only to a specific lens. For more information, see <a href="#">Creating Dashboard Filters</a> .
Property	A property is a predicate that contains the instance data to display. The data type of a property determines the functional aspects within a dashboard. For example, certain filters act only on dates or numbers.



Term	Description
Path	Paths are relationships. They are transitional elements that allow you to traverse to another class.
Functions and formulas	Functions and formulas can be applied to properties to modify the data that is presented. Available functions depend on the property's data type. For more information, see <a href="#">Calculating Values in Lenses and Filters</a> .

## User Interface Overview

This section gives an overview of the user interface. The images below show an administrator view. Some options are not available to users with lower permission levels.



## Main Toolbar

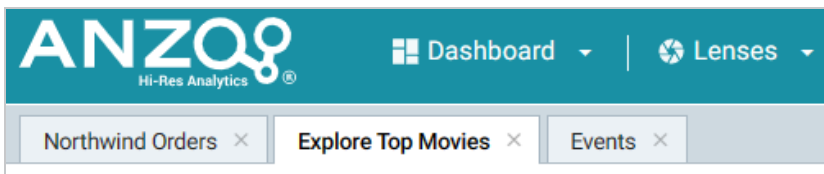
The toolbar at the top of the screen provides the following options:

- **Dashboard:** Accesses dashboard functions, including Save.
- **Lenses:** Creates or opens lenses.

- **Filters:** Creates or manages selected filters.
- **Refresh:** Accesses the automatic refresh check box. Select this box to refresh data automatically. New data will appear and change according to changes elsewhere.
- **Designer:** Controls dashboard layout and design.
- **Help:** Opens help options.
- **User:** Click to access the **Logout** button.

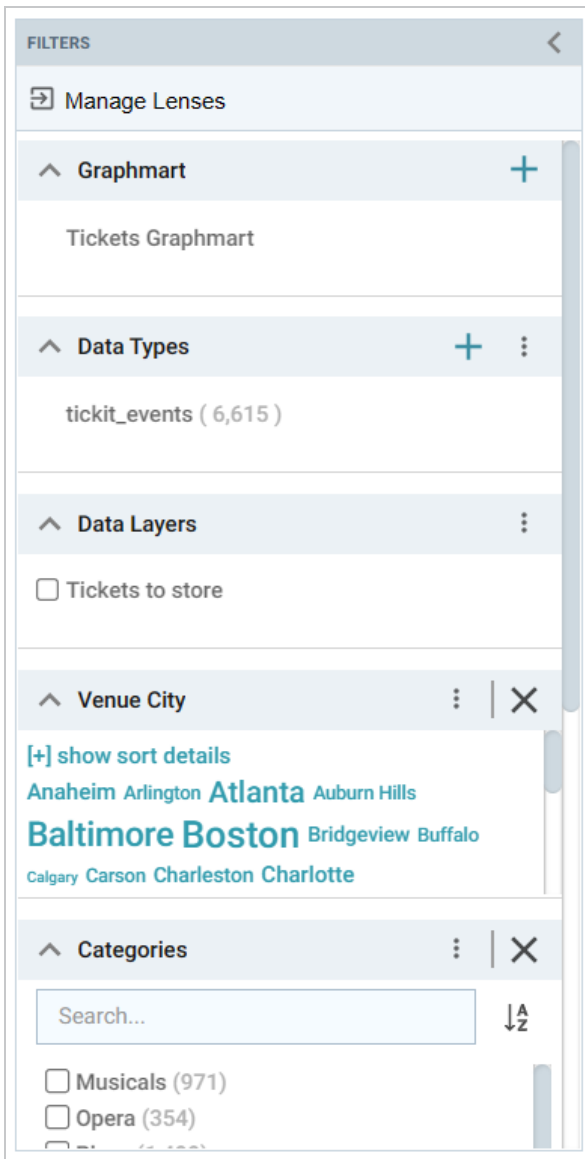
## Dashboard Tabs

The dashboard tabs under the main toolbar display the open dashboards and enable you to navigate between dashboards. When you change a dashboard, an asterisk appears on the dashboard tab. Save the dashboard to preserve the changes.



## Filter Panel

The left panel (shown below) contains the dashboard filters. Each of the panels are described below.



## Manage Lenses

Clicking **Manage Lenses** opens the Manage Lenses dialog box, which enables you to show and hide the items in the panel. For example:

Manage Lenses

×

Filters hidden from view will still impact displayed results

are currently applying filters.

Graphmart	<input checked="" type="checkbox"/> ON
Data Types	<input checked="" type="checkbox"/> ON
Data Layers	<input checked="" type="checkbox"/> ON
Venue City	<input checked="" type="checkbox"/> ON
Categories	<input checked="" type="checkbox"/> ON
Event Date	<input checked="" type="checkbox"/> ON
Event Name	<input checked="" type="checkbox"/> ON
Venue State	<input checked="" type="checkbox"/> ON

Restore Defaults

CANCEL

OK

Clearing an **ON** checkbox hides that item in the panel. You can select a checkbox to show an item or click **Restore Defaults** to show all items.

When the dashboard layout is a **Vertical List Container**, there is also a **Manage Lenses** button available for showing and hiding lenses. For example:

Manage Lenses

^

All Events

:

×

eventid	eventname	starttime
521	West Side Story	1/2/2008 10:00 AM

I<

<

>

>I

1 - 20 of 115

|

Page 1

of 6

⚙

▼

^

Venues

:

×

MODE(

venue/venue

)

MA

Charles Playhouse

Fenway Park

TD Banknorth Garden

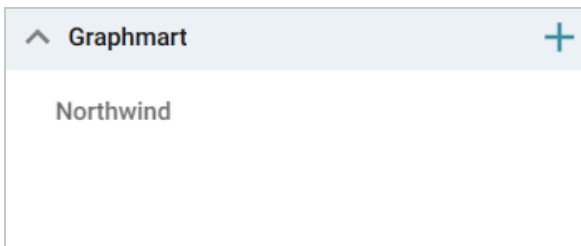
venue/venue

venue/venue

venue/venue

## Graphmart

The Graphmart panel displays the selected graphmart for the dashboard:



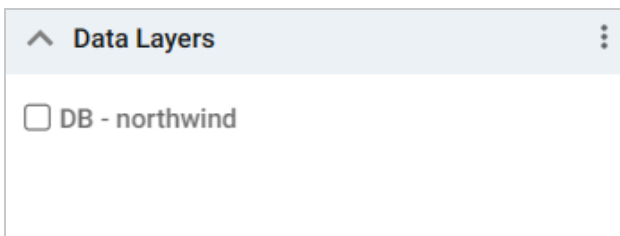
## Data Types

The Data Types panel displays the selected data type from the graphmart.



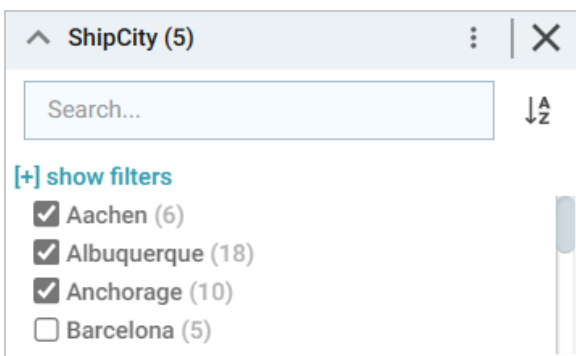
## Data Layers

The Data Layers panel displays the layers in the graphmart.



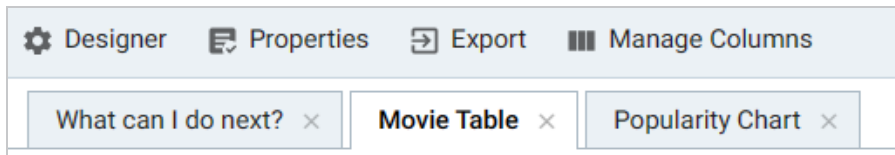
## Filters

By default, filters that you create appear in the left column of the dashboard.



## Object Toolbar and Tabs

The object toolbar and tabs enable you to manage the lenses and filters in the selected dashboard. The tabs display the open lenses, and the toolbar enables you work with the lens properties.



# Getting Started: Exploring and Visualizing Data

When you start to build a new dashboard, you might not know what data exists in the knowledge graph, which values in graph you ultimately want to display, and the most pertinent way to visualize the results. This topic introduces the available lenses and filters and provides guidance on getting started by using the Hi-Res Analytics tools to perform data discovery. By experimenting with simple objects, you can explore the data, determine which questions you want to answer, and start to visualize the end result.

To get started:

- 1. [Create a New Dashboard](#)
- 2. [Explore the Data](#)
- 3. [Create Visualizations of the Data](#)

## Create a New Dashboard

- 1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing graphmarts. For example:

Search

Add Graphmart


<div></div>	Title	Description	Status	Statements	Creator	Actions
<div></div>	Books		<div>Ready to use</div>	156	System Administrator	<div></div>
<div></div>	Flights		<div>Ready to use</div>	265,339	System Administrator	<div></div>
<div></div>	Movies		<div>Ready to use</div>	104,166	System Administrator	<div></div>
<div></div>	Northwind		<div>Ready to use</div>	131,386	System Administrator	<div></div>
<div></div>	Tickets		<div>Ready to use</div>	3,696,548	System Administrator	<div></div>

Rows per page:


25

1-5 of 5

- 2. On the Graphmarts screen, click the name of the graphmart for which you want to create a dashboard. The Overview screen is displayed. For example:




# Northwind

Not Versioned 


INACTIVE

☒

ACTIVE



Ready to use




AZG

Static

Save a Copy

Create Dashboard



Overview

Explore

Datasets

Data Layers

Dashboards

Data on >

Description

None

Load Priority

None

Metrics

Profile data upon activation

☐

Disable Load Counts

Disable populating counts during load operations

☐

Data Loading Settings

Load layers that do not fail

☐

Ignore Source Errors

Ignore any step or view's source layer if they failed to load

☐

Leave Graphmart Online During Refresh

When refreshing layers, leave graphmart and layer online.

☐

Manual Refresh Graphmart

None

General

Type : Graphmart


Creator : System Administrat...

Last Accessed : a few seconds ago

Last Updated : a few seconds ago

Structure Modified : Nov 2, 2022 1:57 PM

Released : Nov 2, 2022 10:06 A...

http://cambridgesemantics.com/Graph 

Dynamic Resource Log

AnzoGraph Server Details

AnzoGraph : AZG

Status : Ready to use

Last Accessed : a few seconds ago

Memory Used : 266.78 MB( 3% )

Memory Total : 11.7 GB

Inactivity Deactivate Timeout

None

- Click the **Create Dashboard** button. The Hi-Res Analytics application opens and displays the Create Dashboard dialog box. Leave **Graphmart Dashboard** selected and click **Next**.



4. Next, type a name for the dashboard in the **Title** field and enter an optional **Description**.

### Create Dashboard

×

Specify details about the new dashboard

#### General Information

Title:\*

Description:

Graphmart:\*

CANCELPREVIOUSFINISH

5. Click **Finish** to create the dashboard. The new dashboard appears as a new tab and contains a sub-tab titled **What can I do next?**. This tab acts as a wizard to guide you through the initial dashboard creation. For example:

ANZOQ  
Hi-Mex Analytics

Dashboard ▾ | Lenses ▾ | Filters ▾ | Refresh ▾ | Designer | Help ▾

\*Northwind Orders ×

Ready to use

FILTERS

Manage Lenses

Graphmart +

Northwind

Data Types + ⋮

No items of the selected type available

Data Layers ⋮

☐ DB - northwind to store

What can I do next? ×

#### Dashboard

This dashboard will allow you to visualize linked data

- Add data you are interested in exploring
- Specify the type of data you would like to see
- Select or create visualizations of your data
- Add facets to filter the data

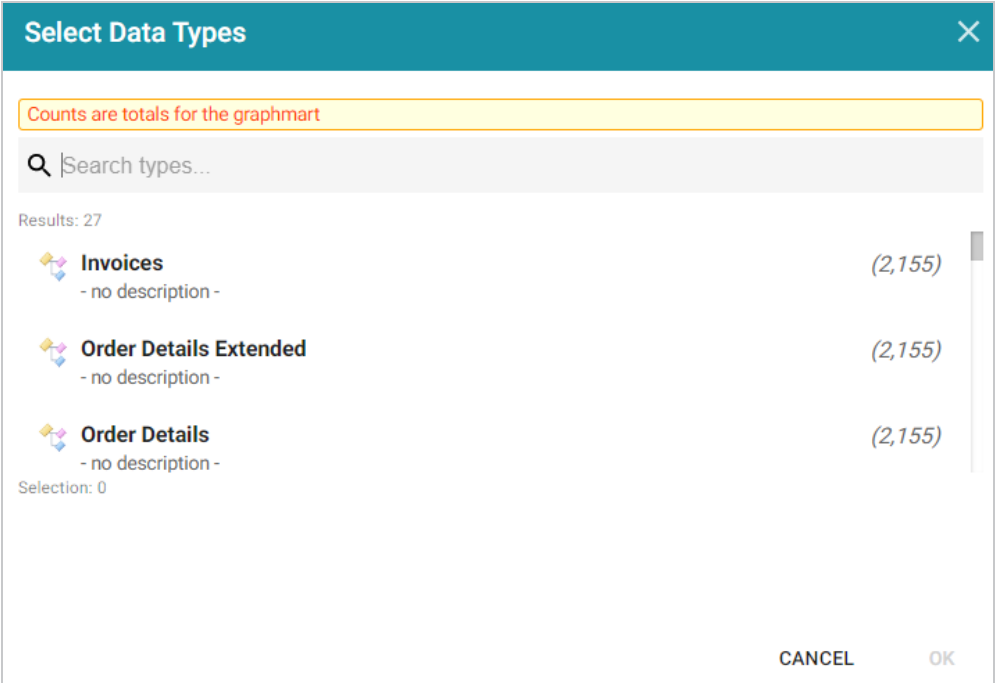
6. In the main toolbar, click the **Dashboard** button and select **Save**. Proceed to [Explore the Data](#) below for guidance on next steps.

### Explore the Data

Once you create a new dashboard, you can experiment with Hi-Res Analytics tools to get to know the data and decide the best way to display it.

### Decide What Type (Class) of Data You Want to See

1. First, review the types of data or classes that exist in the data. On the What can I do next? tab, click **Specify the types of data you would like to see**. The Select Data Types dialog box displays the available data types. The value in parentheses shows the total number of instances of that type exist in the data set:



2. Select one data type. The property that you choose determines the fields that become available to filter on.

#### Tip

Though you must choose one base data type for a dashboard, you can leverage the relationships in the graph to access and integrate data from additional classes. See [Combining Data from Multiple Classes](#) for more information.

Click **OK** to close the Select Data Types dialog box. The data type is added to the Data Types panel on the left side of the screen. Proceed to [Create Filters to See the Values for Properties](#) below for next steps.

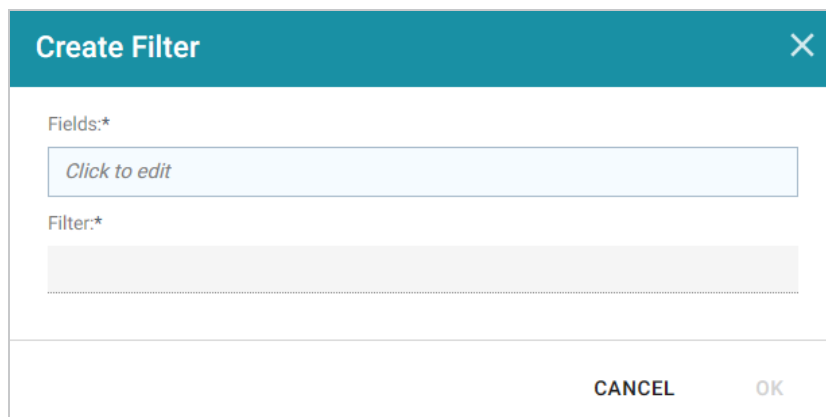
#### Tip

You might want to create multiple dashboards so that you can click between dashboards and view multiple classes of data at the same time.

## Create Filters to See the Values for Properties

To dive deeper into the data and quickly determine what values exist for the class of properties you selected, you can start adding filters to the dashboard. Filters reveal the values associated with fields and help you learn the data set specifics such as whether data exists for certain properties and whether the data includes many duplicate or unique values. Learning more about the details enables you to start making decisions about what properties to group on, what properties have relationships, and what results you want to visualize on the dashboard.






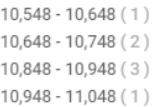
1. To create a filter, click **Add facets to filter the data** on the What can I do next tab. The Create Filter dialog box opens.

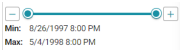


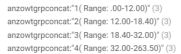
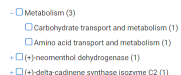



2. In the Create Filter dialog box, click the **Fields** field and select the property or property path to filter on.
3. Then click the **Filter** field to select the filter type. The list of available choices depends on the data type of the property you selected in Fields. The table below describes each filter type.

### Filter Types

Filter	Description
Cloud	Cloud filters display values in term clouds where each term is written in a font size that represents the number of results for that value. Unlike list filters, which enable you to

Filter	Description
	select and filter on multiple values at once, cloud filters allow you to filter on one value at a time. The Cloud filter is available for all data types.
<b>List</b> 	List filters display values in a list and allow you to filter on multiple values at the same time. List filters are available for all data types.
<b>Single Select List</b> 	Single Select List filters are similar to List filters but only allow you to filter one value from the list at a time. This type of filter is available for properties of all data types but is not available for paths.
<b>Limit</b> 	Limit filters are used to limit the results on the dashboard to a specified number of either the largest or smallest values. The Limit filter is available for any data type. For strings, results are ordered alphabetically. "Largest" orders by the last letters in the alphabet and "Smallest" orders by the first letters in the alphabet.
<b>Date Range</b> 	Date Range filters are used to limit the results on a dashboard to data that falls in (or outside of) certain date and time groupings. Date Range filters are available for properties with date, dateTime, and time data types.
<b>Numeric Range</b> 	Numeric Range filters are used to limit the results on a dashboard to data that falls in (or outside of) certain numeric groupings. Numeric Range filters are available for properties with integer and double data types.

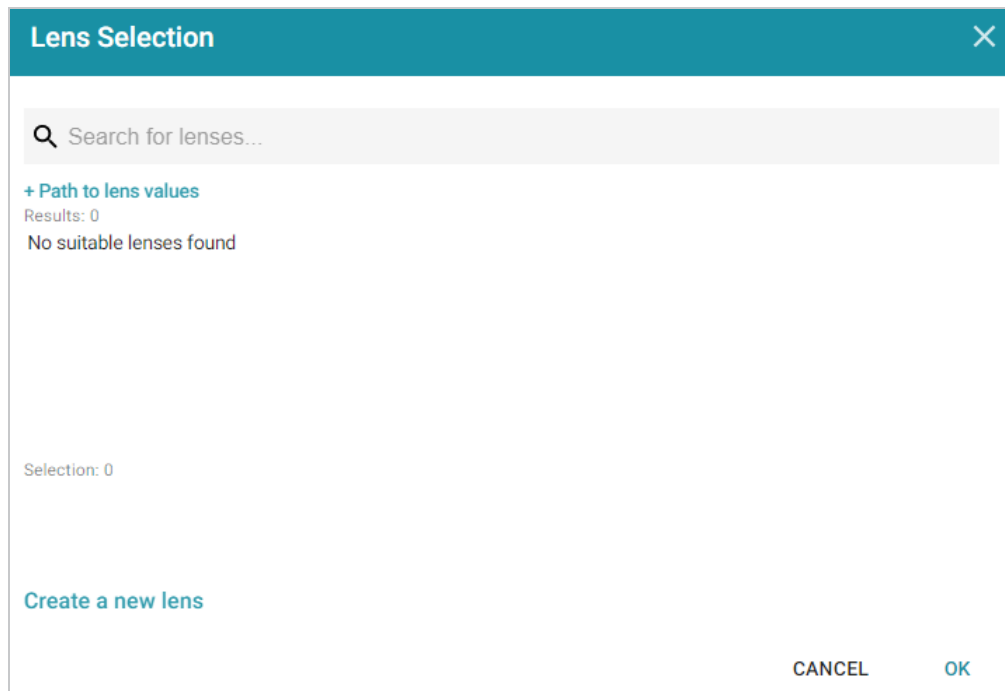
Filter	Description
<b>Range Slider</b> 	Range Slider filters display a slider control that enables you to filter dashboard results by setting one range that you can adjust as needed. This type of filter is available for properties with integer, double, date, time, and dateTime data types. It is not available for paths.
<b>Relative Time</b> 	Relative Time filters are used to filter for records that fall into a specified increment of time relative to the current time. This type of filter is available for date, time, and dateTime data types.
<b>Search</b> 	Search filters are used to search for values of a property that contain a partial match, exact match, or do not equal the text that you specify. The search is case-insensitive. This type of filter is available for all data types. It is not available for use with paths.
<b>Presence</b> <div> Exists ( 9,604 ) Does not exist ( 395 ) </div>	Presence filters group results based on whether the value exists or does not exist. This type of filter is useful for testing whether there are records that are missing a particular value. Presence filters are available for paths and properties of all data types.
<b>Quartile</b> 	Quartile filters group and rank the values for a property into four equal ranges. This filter is available for properties with integer, double, date, time, and dateTime data types. It is not available for paths.
<b>Hierarchy</b> 	<p>If hierarchies exist in your knowledge graph, you can create a Hierarchy filter to explore the parent and child relationships and filter the dashboard based on the relationships.</p> <p>Unlike the majority of dashboard filters, where you select a property to filter on, Hierarchy filters operate on relationships and are only available as a filter type when you select a path to filter on.</p>
<b>Types</b> 	Types filters are used to filter data according to the types of data (classes) that are connected by a specified path. This type of filter is available only for paths and not properties.

For information about configuring each type of filter, see [Creating Dashboard Filters](#).

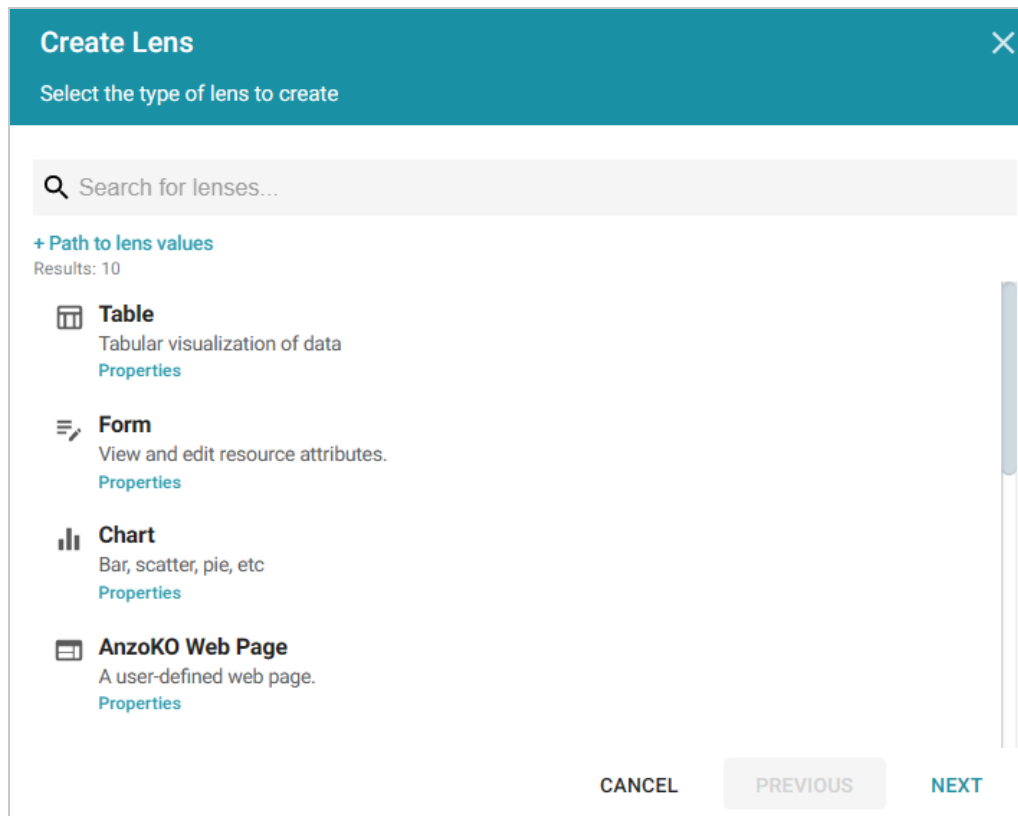
## Create Visualizations of the Data

Once you have a good understanding of the values and relationships that exist in the data , you can experiment with the Hi-Res Analytics lenses and decide on the most appropriate way to display the data. Creating a Table lens is a quick way to view the data that you filtered. This section provides instructions for creating a table lens and describes each of the lenses available in Hi-Res Analytics.

1. To create a Table lens, click **Select or create visualizations of your data** in the What can I do next tab. Anzo displays the Lens Selection dialog box.




2. In the dialog box, click **Create a new lens**. Anzo displays the Create Lens dialog box.



3. Select the **Table** lens and click **Next**.

4. Type a **Title** for the lens, and then click **Finish**. Anzo opens the Table Designer:

5. In the Designer, click the **Auto-generate columns** icon (  ) to add all available columns to the table. Then click **Save**.

The new lens displays as a new sub-tab on the dashboard and displays the data according to the data type and filter or filters that you created. Now that you can view a summary of the data in a table, it can help you determine how to further narrow or expand the results by adding, changing, or removing filters. In addition, you can experiment by adding other lenses to the dashboard to find the ideal way to display the data to answer the questions that you have. The table below describes each type of lens. For more information about each lens, see [Creating Lenses](#).

Lens	Description
<b>AnzoKO Web Page</b>	AnzoKO Web Page lenses include the <a href="#">Knockout JavaScript</a> framework and enable you to display data on a web page that you create using HTML, CSS, and JavaScript.
<b>Chart</b>	Anzo offers several types of Chart lenses. These lenses are useful for displaying large



Column

Bar

Line

Bubble

Funnel

Horizontal Bar

Pie

Panel

Scatter

Area

amounts of complex data and have the widest format range of any lens type. The ability to add an axis enables you to compare data, such as for comparing monthly sales data for multiple stores.

Drill Down

Drill Down lenses create clickable data points that enable you to drill down to view additional details. You can specify mutiple lenses within the Drill Down lens so that clicking a data point presents the data in a different view.

↑

expand details...

▽

1

Upcoming

▽

2

Boris Godunov

▽

3

Salome

Form

Form lenses enable you to create an editable or read-only form on the dashboard. Creating forms can be useful for displaying many details about each record instead of using a table where the large number of columns makes the data hard to read.

aircraft type

Airliner

designed by

Reginald Kirshaw Pier

maximum speed

180.246528

Note

Form lenses are valid in Linked Data Set Dashboards. In Graphmart Dashboards, form lenses do not display instance data.

List

List lenses display results in a list layout, similar to the Microsoft Windows® Explorer interface. The lens enables you to add icons for each data value, and results are grouped onto pages according to the Page Size value that you specify.

Abbas Kiarostami

Abe Vigoda

Adrian Dunbar


Adriana Asti

Aki Kaurismäki

Al Lettieri

Query

Query lenses enable you to retrieve data using a custom SPARQL query and display the results by writing basic HTML and CSS. You can use a Query lens to access data from external sources. Query lenses do not bind directly to the linked data set, data type, or filters defined on the dashboard.

Lens	Description																		
<b>Resource Tree Navigator</b>	<p>Resource Tree Navigator lenses display results in a hierarchical tree view. You can click parent data points to open the successive child data points. This lens is useful for presenting small amounts of data; each discrete group appears on a separate page in the dashboard. You can also click certain objects to view the object's data properties in the left panel.</p> 																		
<b>Table</b>	<p>Table lenses present results in a basic table grid consisting of rows and columns. Table lenses are useful for presenting data aggregates or summaries.</p> <table><tr><th>eventid</th><th></th><th></th></tr><tr><td>1</td><td>Gotterdammerung</td><td>1/25/2008 9:30 AM</td></tr><tr><td>10</td><td>Rigoletto</td><td>9/10/2008 11:00 AM</td></tr><tr><td>100</td><td>Siegfried</td><td>10/30/2008 11:00 AM</td></tr><tr><td>1000</td><td>Gypsy</td><td>4/27/2008 10:00 AM</td></tr><tr><td>1001</td><td>Chicago</td><td>8/20/2008 3:30 PM</td></tr></table>	eventid			1	Gotterdammerung	1/25/2008 9:30 AM	10	Rigoletto	9/10/2008 11:00 AM	100	Siegfried	10/30/2008 11:00 AM	1000	Gypsy	4/27/2008 10:00 AM	1001	Chicago	8/20/2008 3:30 PM
eventid																			
1	Gotterdammerung	1/25/2008 9:30 AM																	
10	Rigoletto	9/10/2008 11:00 AM																	
100	Siegfried	10/30/2008 11:00 AM																	
1000	Gypsy	4/27/2008 10:00 AM																	
1001	Chicago	8/20/2008 3:30 PM																	
<b>Web Page</b>	<p>Web Page lenses enable you to display results on a web page that you create using HTML, CSS, and JavaScript.</p>																		

### Related Topics

- [Creating Dashboards](#)
- [Creating Dashboard Filters](#)
- [Creating Lenses](#)
- [Sharing Access to a Dashboard or Lens](#)

## Creating Dashboards

There are two types of dashboards that you can create to explore and analyze a graphmart's knowledge graphs: a **Graphmart Dashboard** that offers several choices of lens and filter types, and a **Network Navigator Dashboard**, which is an interactive graph visualization tool for exploring the relationships in a graph.

### Note

The Network Navigator Dashboard is available as a **Preview** release. Features are considered "Preview" when the implementation has recently been completed but is not yet thoroughly tested and could be unstable. By default, the only user who has permission to create a Network Navigator Dashboard is the **sysadmin** user. However, the sysadmin user can share created dashboards with other users and groups (see [Sharing Access to a Dashboard or Lens](#)). This feature is available for trial usage, but Cambridge Semantics recommends that you do not rely on Preview features in production environments.

The topics in this section provide instructions for creating both types of dashboards:

























[Creating a Graphmart Dashboard](#)

[Creating a Network Navigator Dashboard \(Preview\)](#)


### Creating a Graphmart Dashboard

Follow the instructions below to create a new Hi-Res Analytics dashboard for a Graphmart.


1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing graphmarts. For example:




	<input type="text" value="Search"/>				<a href="#">Add Graphmart</a>	
<input type="checkbox"/>	Title	Description	Status	Statements	Creator	Actions
	Books		 Ready to use	156	System Administrator	 
	Flights		 Ready to use	265,339	System Administrator	 
	Movies		 Ready to use	104,166	System Administrator	 
	Northwind		 Ready to use	131,386	System Administrator	 
	Tickets		 Ready to use	3,696,548	System Administrator	 
<div>Rows per page: 25 1-5 of 5 &lt; &gt;</div>						

- On the Graphmarts screen, click the name of the graphmart for which you want to create a dashboard. The Overview screen is displayed. For example:



## Northwind

Not Versioned 

INACTIVE
 ACTIVE
 Ready to use
 AZG
Static

[Overview](#)
[Explore](#)
[Datasets](#)
[Data Layers](#)
[Dashboards](#)
[Data on >](#)


Description

None


Load Priority

None


**Metrics**

Profile data upon activation 


**Disable Load Counts**

Disable populating counts during load operations 


**Data Loading Settings**

Load layers that do not fail 

**Ignore Source Errors**

Ignore any step or view's source layer if they failed to load 

**Leave Graphmart Online During Refresh**

When refreshing layers, leave graphmart and layer online. 

**Manual Refresh Graphmart**

**General**

Type: Graphmart


Creator: System Administrat...

Last Accessed: a few seconds ago

Last Updated: a few seconds ago

Structure Modified: Nov 2, 2022 1:57 PM

Released: Nov 2, 2022 10:06 A...

<http://cambridgesemantics.com/Graph> 

[Dynamic Resource Log](#)

**AnzoGraph Server Details**

AnzoGraph: AZG

Status: Ready to use

Last Accessed: a few seconds ago

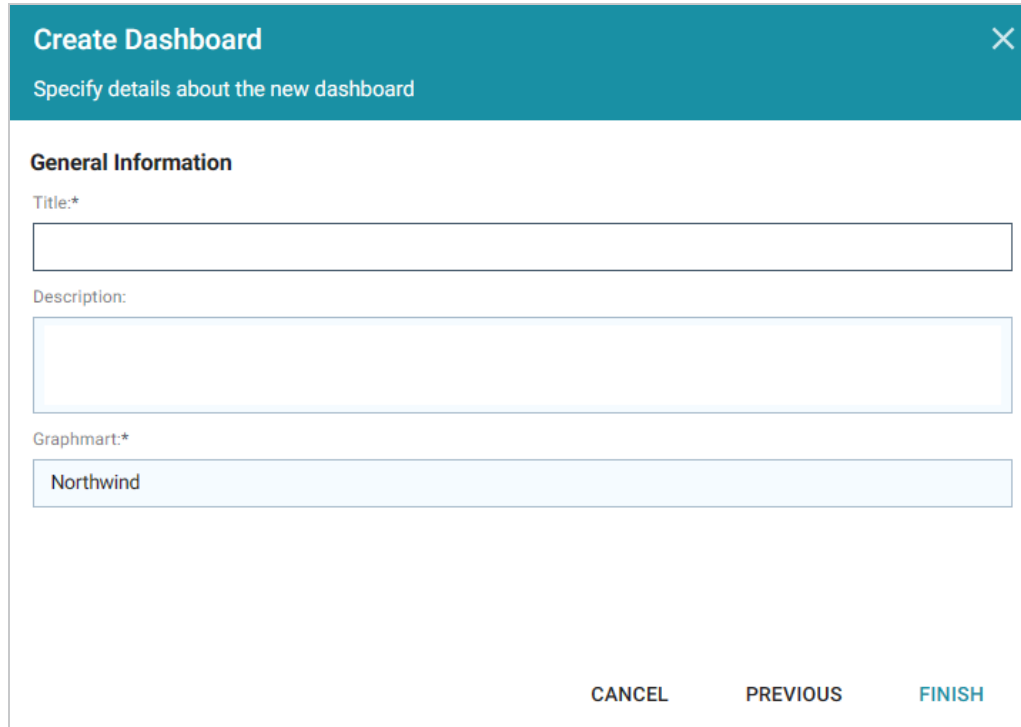
Memory Used: 266.78 MB( 3% )

Memory Total: 11.7 GB

Inactivity Deactivate Timeout

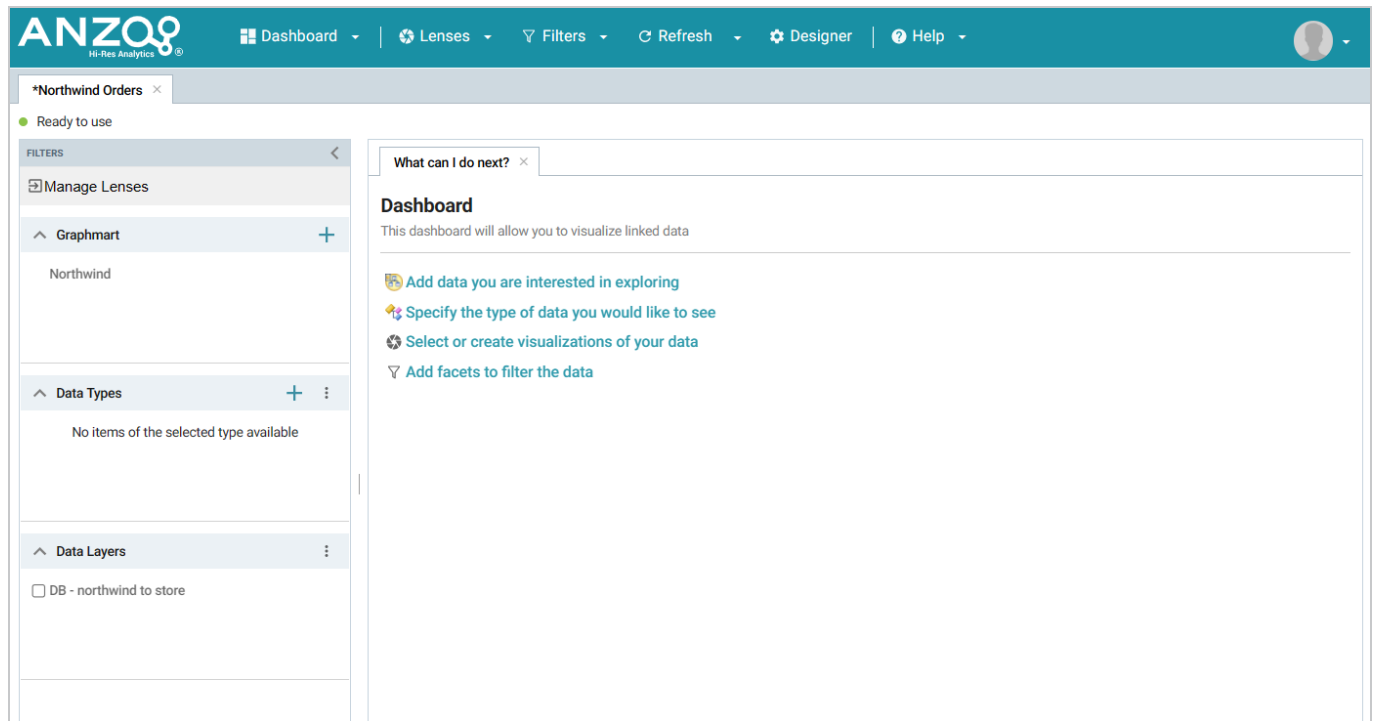
None

3. Click the **Create Dashboard** button. The Hi-Res Analytics application opens and displays the Create Dashboard dialog box. Leave **Graphmart Dashboard** selected and click **Next**.
4. Next, type a name for the dashboard in the **Title** field and enter an optional **Description**.

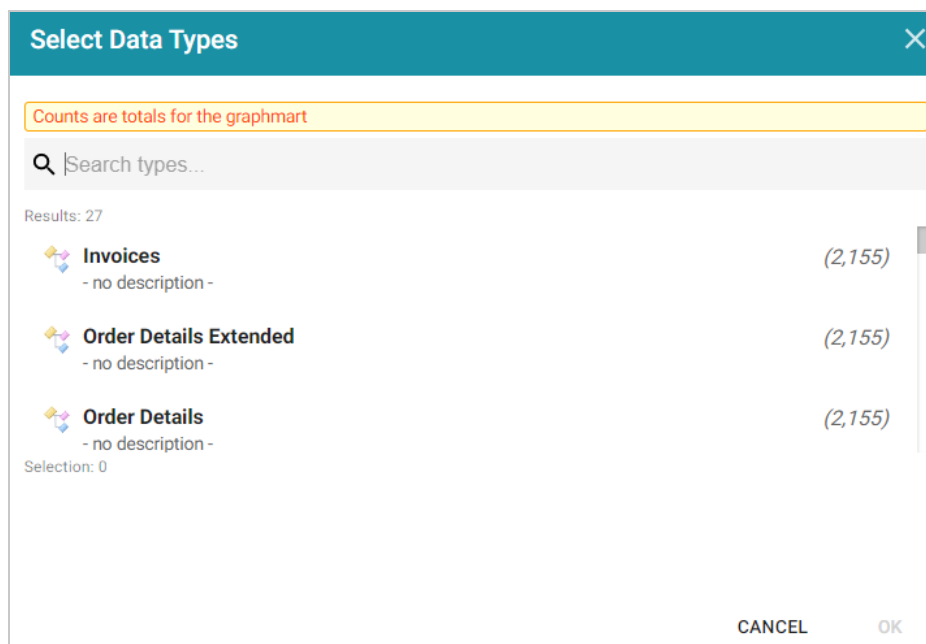


The image shows a 'Create Dashboard' dialog box with a teal header. The header contains the title 'Create Dashboard' and a close button (X). Below the header, the subtitle 'Specify details about the new dashboard' is displayed. The main content area is titled 'General Information' and contains three input fields: 'Title:\*' (a single-line text box), 'Description:' (a multi-line text box), and 'Graphmart:\*' (a dropdown menu with 'Northwind' selected). At the bottom of the dialog, there are three buttons: 'CANCEL', 'PREVIOUS', and 'FINISH'.

5. Click **Finish** to create the dashboard. The new dashboard appears as a new tab and contains a sub-tab titled **What can I do next?**. This tab acts as a wizard to guide you through the initial dashboard creation. For example:



6. On the What can I do next? tab, click **Specify the types of data you would like to see**. The Select Data Types dialog box displays the available data types. The value in parentheses shows the total number of instances of that type exist in the graphmart:



7. In the Select Data Types dialog box, select the data type or class of data that you want to display on the dashboard. Anzo uses the type, along with any filters, to populate the visualizations (lenses) that you add to the dashboard.

### Tip

Though you must choose one base data type for a dashboard, you can leverage the relationships in the graph to access and integrate data from additional classes. See [Combining Data from Multiple Classes](#) for more information.

8. Click **OK** to close the Select Data Types dialog box. The data type is added to the Data Types panel on the left side of the dashboard.
9. In the main Hi-Res Analytics toolbar, click the **Dashboard** button and select **Save** to save your progress.

Now that the dashboard basics are defined, see [Creating Lenses](#) and [Creating Dashboard Filters](#) for instructions on adding lenses and filters to the dashboard.



























## Creating a Network Navigator Dashboard (Preview)

The Network Navigator Dashboard is an interactive graph visualization tool that enables you to find relationships in the knowledge graph and explore the paths to build out a Network View. This topic helps you get started building a Network Navigator Dashboard by covering the basic steps and functionality that is presented by default. Additional topics describe the functionality in further detail and cover more advanced dashboard configuration options.


### Note

The Network Navigator Dashboard is available as a **Preview** release. Features are considered "Preview" when the implementation has recently been completed but is not yet thoroughly tested and could be unstable. By default, the only user who has permission to create a Network Navigator Dashboard is the **sysadmin** user. However, the sysadmin user can share created dashboards with other users and groups (see [Sharing Access to a Dashboard or Lens](#)). This feature is available for trial usage, but Cambridge Semantics recommends that you do not rely on Preview features in production environments.


1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing graphmarts. For example:





	<input type="text" value="Search"/>				<a href="#">Add Graphmart</a>	
<input type="checkbox"/>	Title	Description	Status	Statements	Creator	Actions
	Books		 Ready to use	156	System Administrator	 
	Flights		 Ready to use	265,339	System Administrator	 
	Movies		 Ready to use	104,166	System Administrator	 
	Northwind		 Ready to use	131,386	System Administrator	 
	Tickets		 Ready to use	3,696,548	System Administrator	 
<div>Rows per page: 25</div> <div>1-5 of 5</div> <div></div> <div></div>						


- On the Graphmarts screen, click the name of the graphmart for which you want to create a dashboard. The Overview screen is displayed. For example:



## Supply Chain Graphmart

Not Versioned 

INACTIVE
 ACTIVE
 Ready to use
 AZG
 Static

[Save a Copy](#)
[Create Dashboard](#)


[Overview](#)
[Explore](#)
[Datasets](#)
[Data Layers](#)
[Dashboards](#)
[Data on](#)


Description

None


Load Priority

100


Metrics

Profile data upon activation 


Disable Load Counts

Disable populating counts during load operations 


Data Loading Settings

Load layers that do not fail 

Ignore Source Errors

Ignore any step or view's source layer if they failed to load 

Leave Graphmart Online During Refresh

When refreshing layers, leave graphmart and layer online. 

Manual Refresh Graphmart

**General**

Type: Graphmart


Creator: System Administrat...

Last Accessed: 9 minutes ago

Last Updated: 9 minutes ago

Structure Modified: 9 minutes ago

Released: 9 minutes ago

<http://cambridgesemantics.com/Graph> 

[Dynamic Resource Log](#)

**AnzoGraph Server Details**

AnzoGraph: AZG

Status: Ready to use

Last Accessed: 8 minutes ago

Memory Used: 232.18 MB( 2%)

Memory Total: 11.7 GB

Inactivity Deactivate Timeout

None





3. Click the **Create Dashboard** button. The Hi-Res Analytics application opens and displays the Create Dashboard dialog box. Select **Network Navigator Dashboard** and click **Next**. Remember that only the **sysadmin** user has the option to create this type of dashboard. If you do not see the option and think you should, check with your system administrator.

Create Dashboard

Select the type of dashboard to create

☐ Show advanced dashboards

**Graphmart Dashboard**  
Set up a new Graphmart dashboard. Graphmart dashboards display data from graphmarts.  
[Properties](#)

**Network Navigator**  
Interactively explore Graphmart data.  
[Properties](#)

CANCEL

PREVIOUS

NEXT

Creating a Network Navigator Dashboard (Preview)

741

4. Next, type a name for the dashboard in the **Title** field and enter an optional **Description**. For example:

Create Dashboard

×

Specify details about the new dashboard

General Information

Title:\*

Explore Supply Chain

Description:

Explore suppliers for raw materials

Graphmart:\*

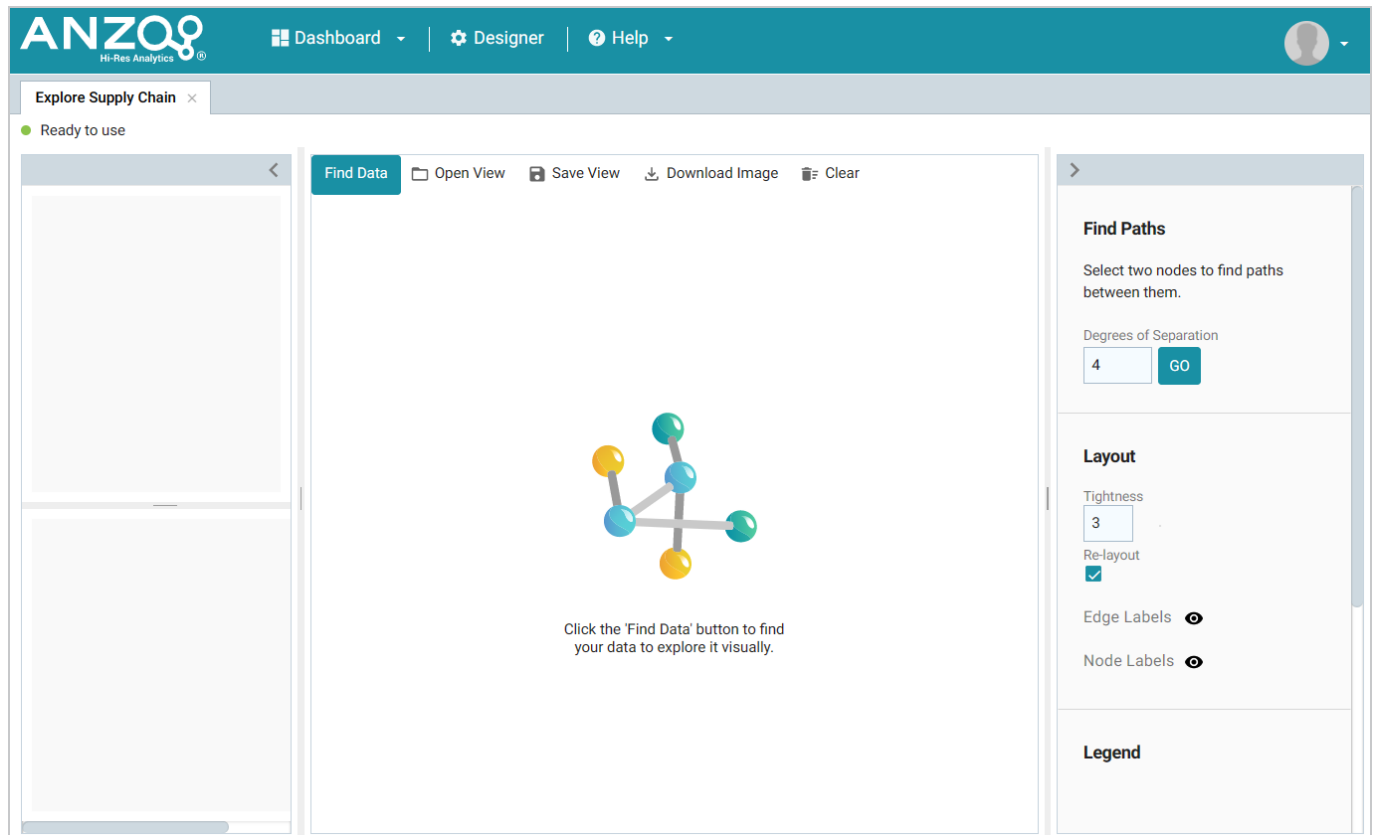
[VW] Supply Chain Graphmart

CANCEL

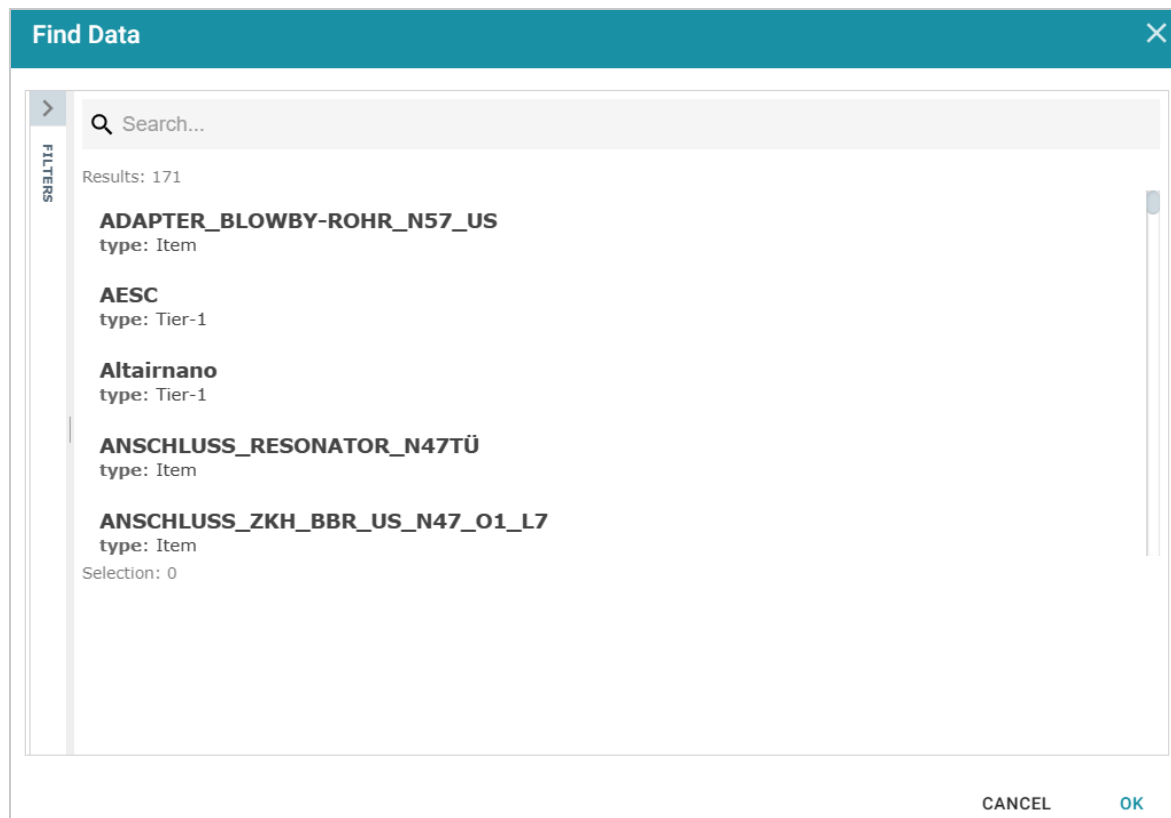
PREVIOUS

FINISH

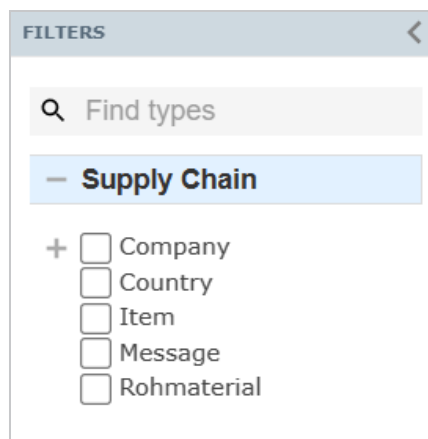
5. Click **Finish** to create the dashboard. The dashboard appears as a new tab in the application. For example:



- The first step in rendering the Network View is to choose a node (or group of nodes) as the starting point. A node is an object in the data. In order to be able to traverse the network, start with an object for a property that has one or more relationships with other properties. To select a value, click the **Find Data** button at the top of the dashboard. The Find Data dialog box is displayed and lists all of the values for each of the properties in the data model. Select one or more objects in the list.

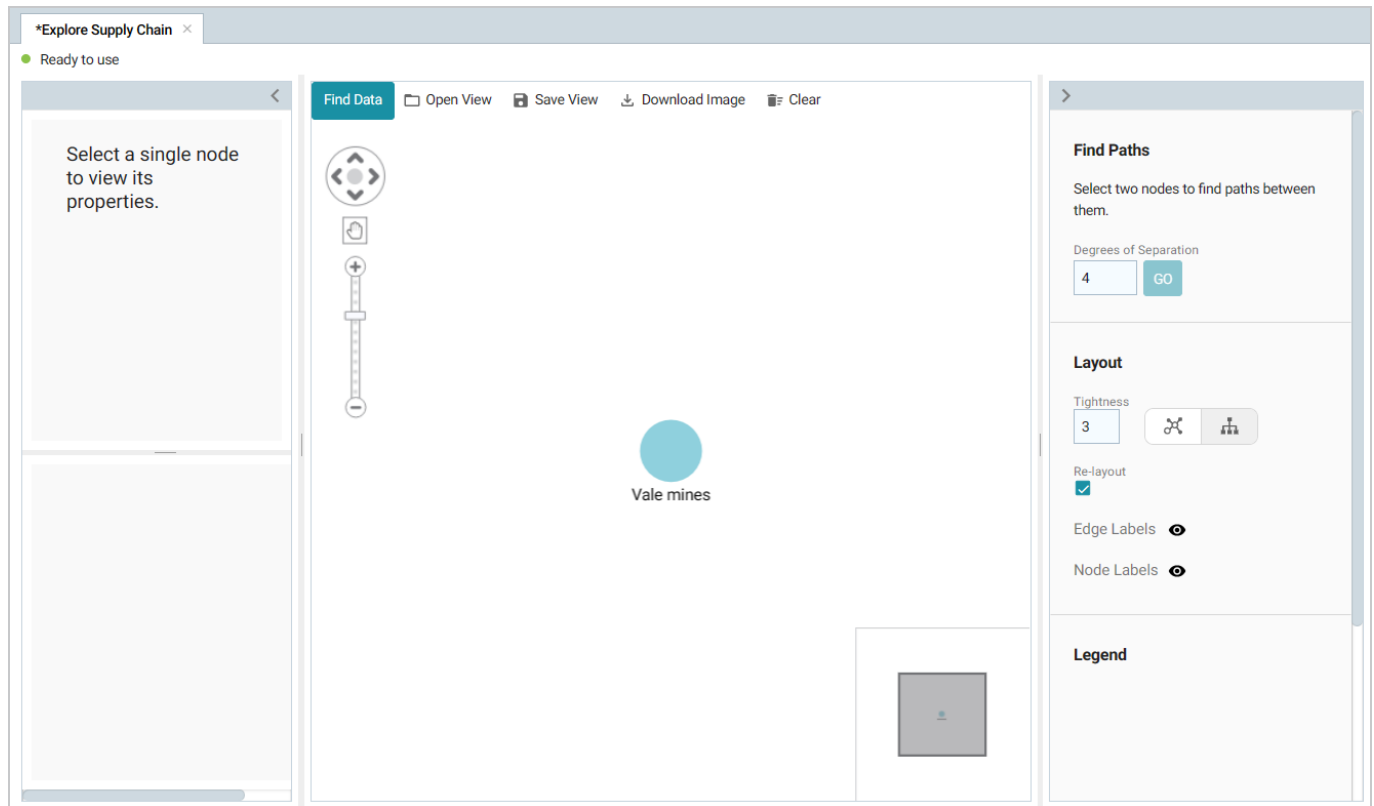


To narrow down the list of values by class or type of data, you can click **Filter** to open the Filter panel on the left side of the screen. The Filter panel lists the classes and subclasses from the model. For example, the image below shows the classes for a knowledge graph that contains supply chain data for an automobile manufacturer:



You can expand or collapse the classes to display or hide subclasses. And you can start typing text in the **Find Types** field to search for specific classes. Select the checkbox next to any of the classes from which you want to display data. The list in the main part of the screen is refreshed to display only the values for the properties in the selected classes.

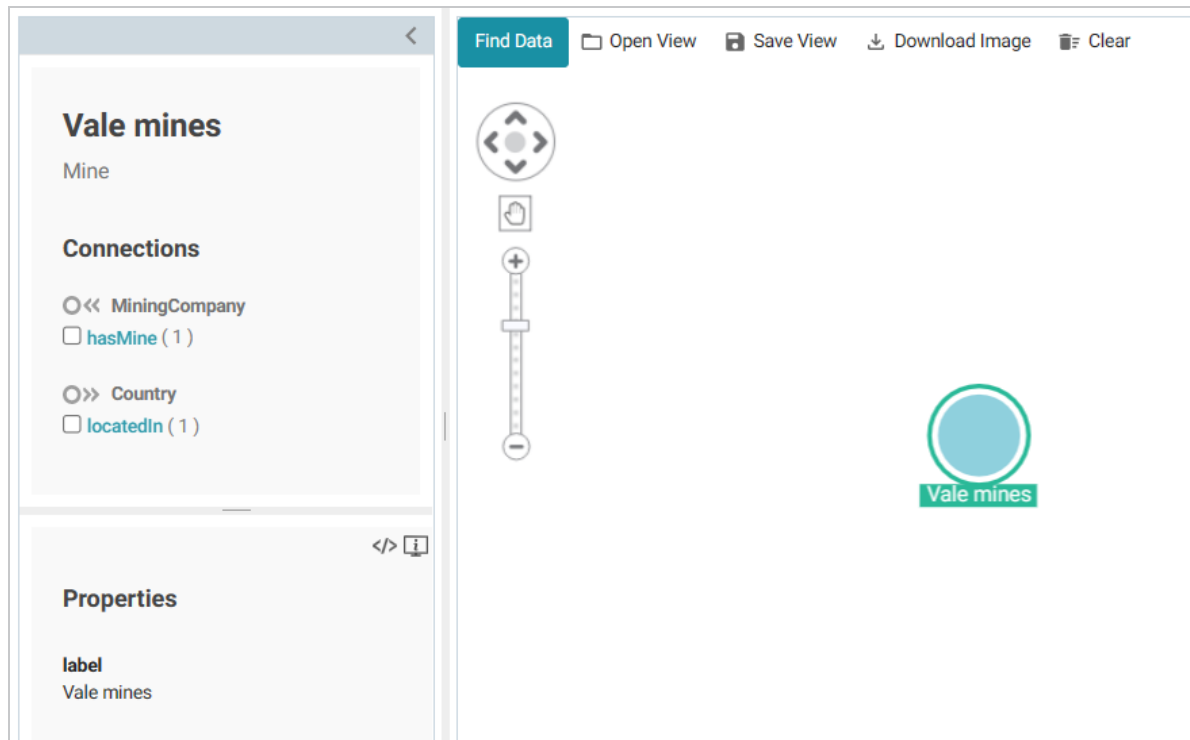
- Once you have selected the starting node in the Find Data dialog box, click **OK** to add the node to the Network View. For example, in the image below the value **Vale mines** was selected as the starting point, and the Vale mines node is added to the View:



### Tip

You can configure the dashboard to **Auto-Expand** by a certain number of degrees or hops so that adding a node to a View automatically adds the specified number of related nodes and paths. For information, see [Auto-Expanding a Network View](#).

- To start building out the network, select a node to view its connections and properties on the left side of the screen. For example, selecting the **Vale mines** node shows its incoming and outgoing connections as well as properties like the label.

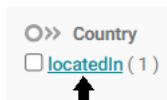


The image shows that Vale mines has an incoming connection (or backward path ◯◀) from **Mining Company** via **hasMine** and an outgoing connection (or forward path ◯▶) to **Country** via **locatedIn**.

9. There are two options for exploring the connections:

### See a list of connected nodes without adding them to the View

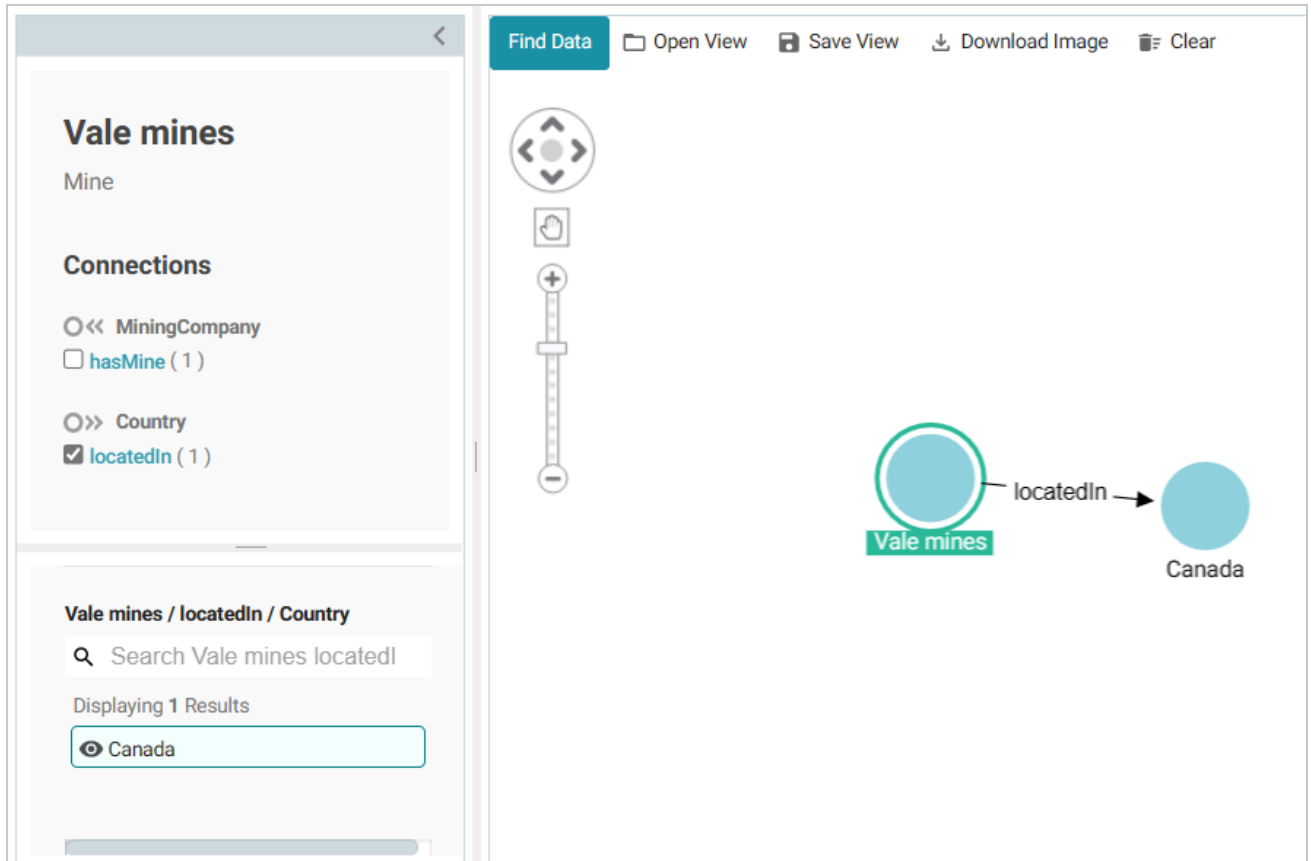
To see a list of nodes that are connected by a path without adding the nodes and path to the View, you can click the path name link.



Details are shown in the lower panel, but the nodes are hidden from the network View by default. For example, clicking the **locatedIn** path from the image above, shows that Vale mines is located in Canada.

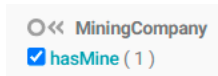


If you decide you want a node and path to be added to the View, you can click the hidden node to make it visible. For example, clicking **Canada** adds the path and node to the View:

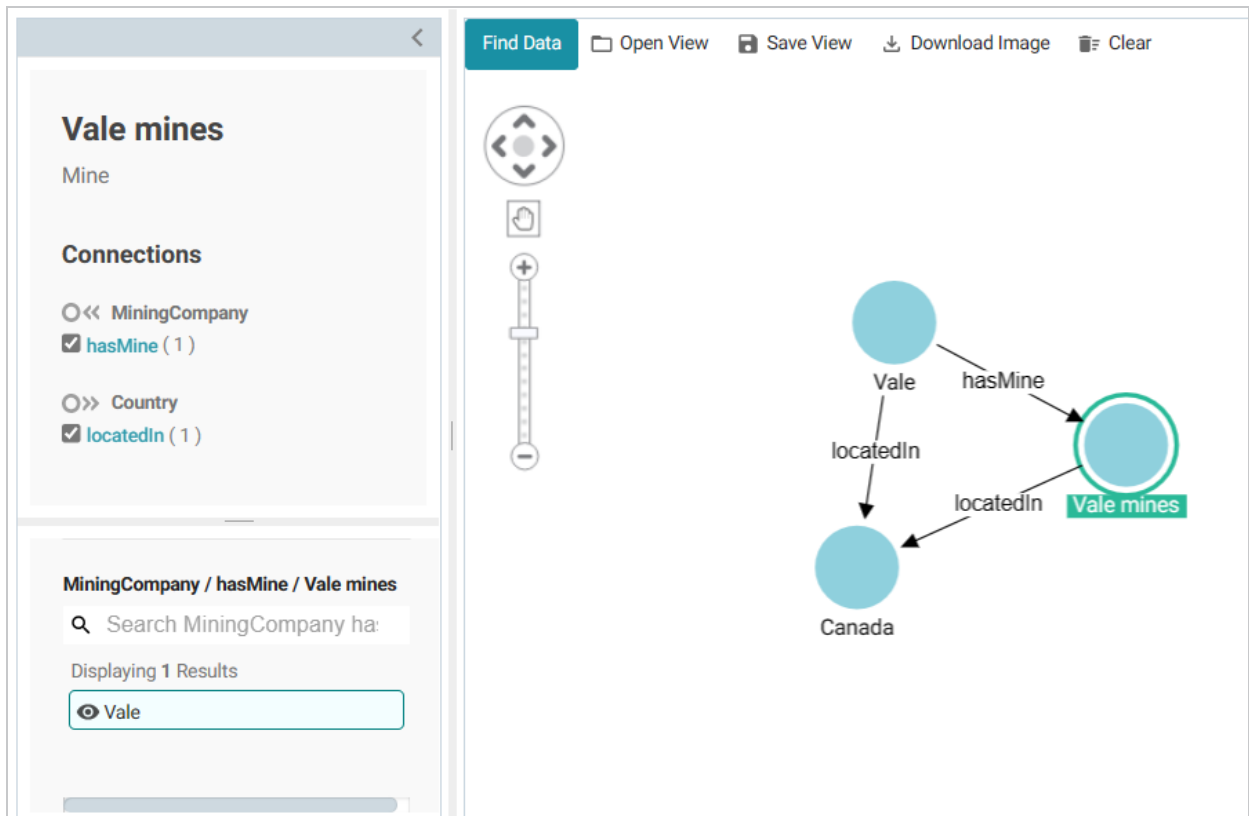


## Add a path and node directly to the View

To add a path and node directly to the network View, you can select the checkbox next to the path.



For example, selecting the **hasMine** checkbox adds the path and the MiningCompany value (**Vale**) to the View:



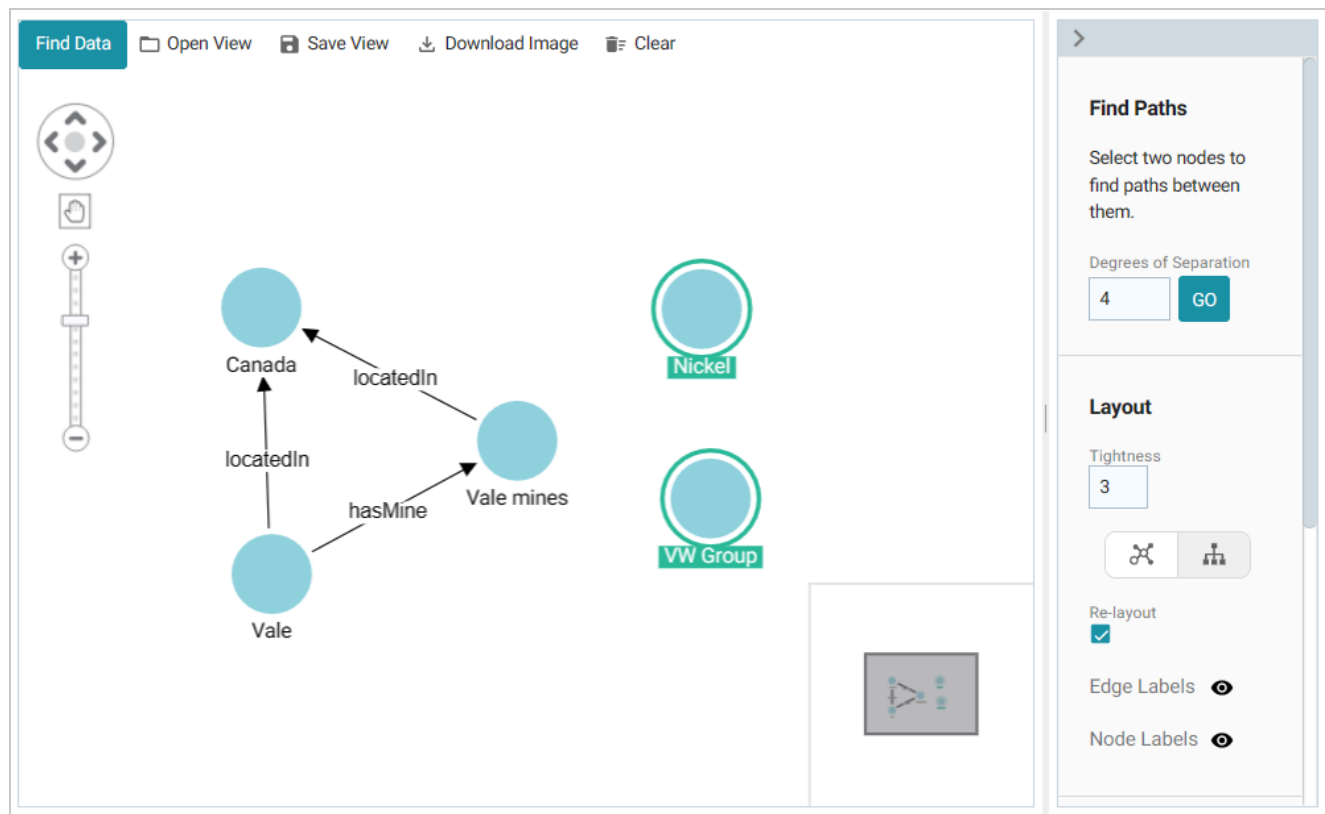
### Tip

Any time you want to clear the View and start again, you can click **Clear** at the top of the screen. To toggle between showing and hiding a path or node, you can click the path or node in the panel on the left side of the screen to select or deselect the item.

You can save the View by clicking **Save View** and specifying a Title and optional Description in the dialog box. When you save the dashboard, the View that is visible at the time it is saved is presented when the dashboard is reopened.

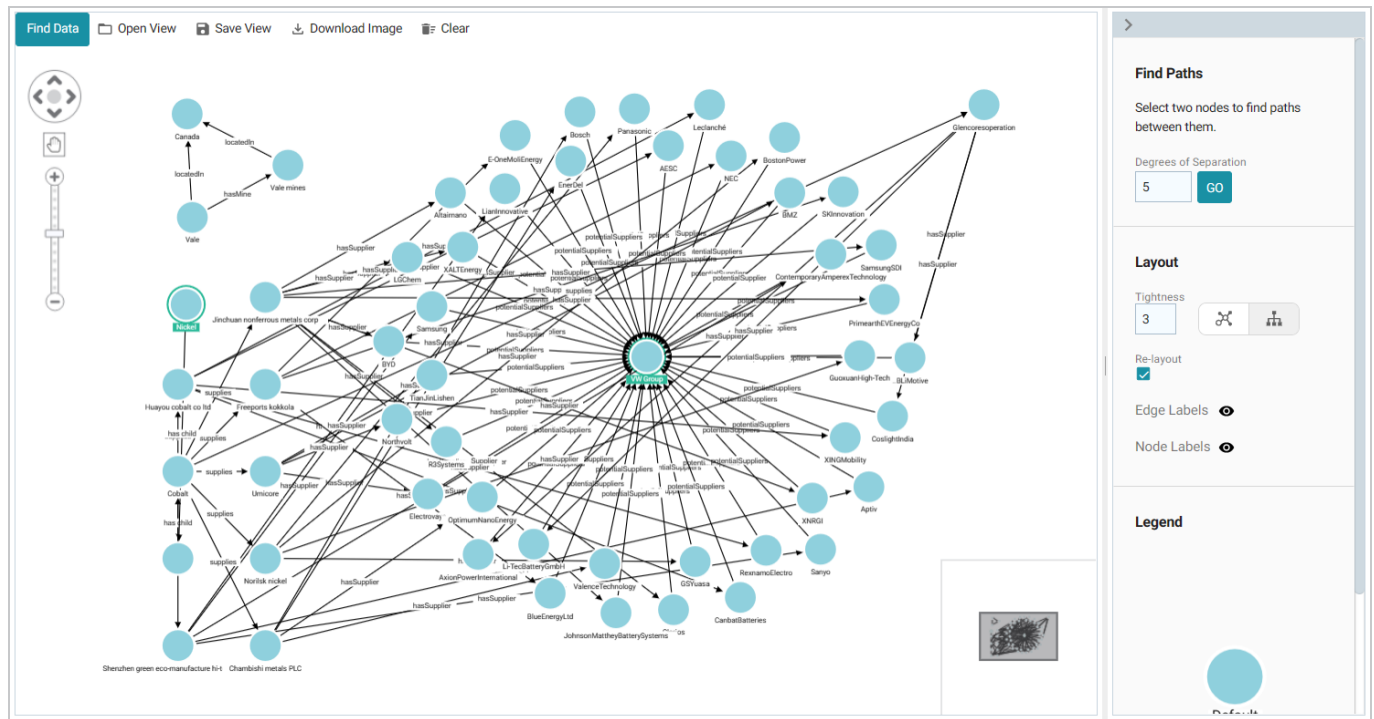
10. To continue to explore and expand the Network View, you can select another node and repeat the previous step to identify additional paths to follow. Or you also have the option to select two nodes and do a search for paths between them. To find paths, find and add to the view, if necessary, the two nodes whose relationships you want to explore. Then hold the **Ctrl** button and select the two nodes. For example, in the image below, two nodes, an OEM called VW Group, and a raw material, Nickel, were added to the View, and both nodes are selected:





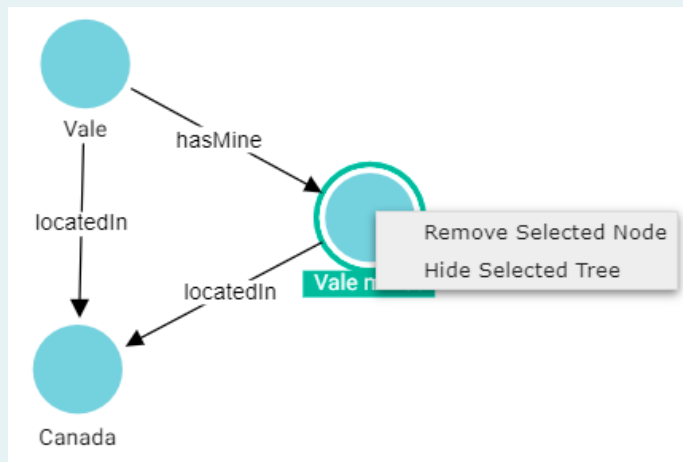
11. On the right side of the screen under **Find Paths**, adjust the **Degrees of Separation** value as needed (the default is 4) and then click **Go**. Anzo runs an All Paths graph algorithm to find all of the paths that exist between the two nodes—where the maximum number of hops is the value in the **Degrees of Separation** field. If a "No paths found" message is returned, that means there is no path between the two nodes that is  $N$  or fewer hops away, where  $N$  is the value in Degrees of Separation. You can change the value and try again.

In the example for VW Group and Nickel, finding paths with the default value of 4 returned "No paths found." However, increasing Degrees of Separation to 5 renders several nodes and relationships.



### Tip

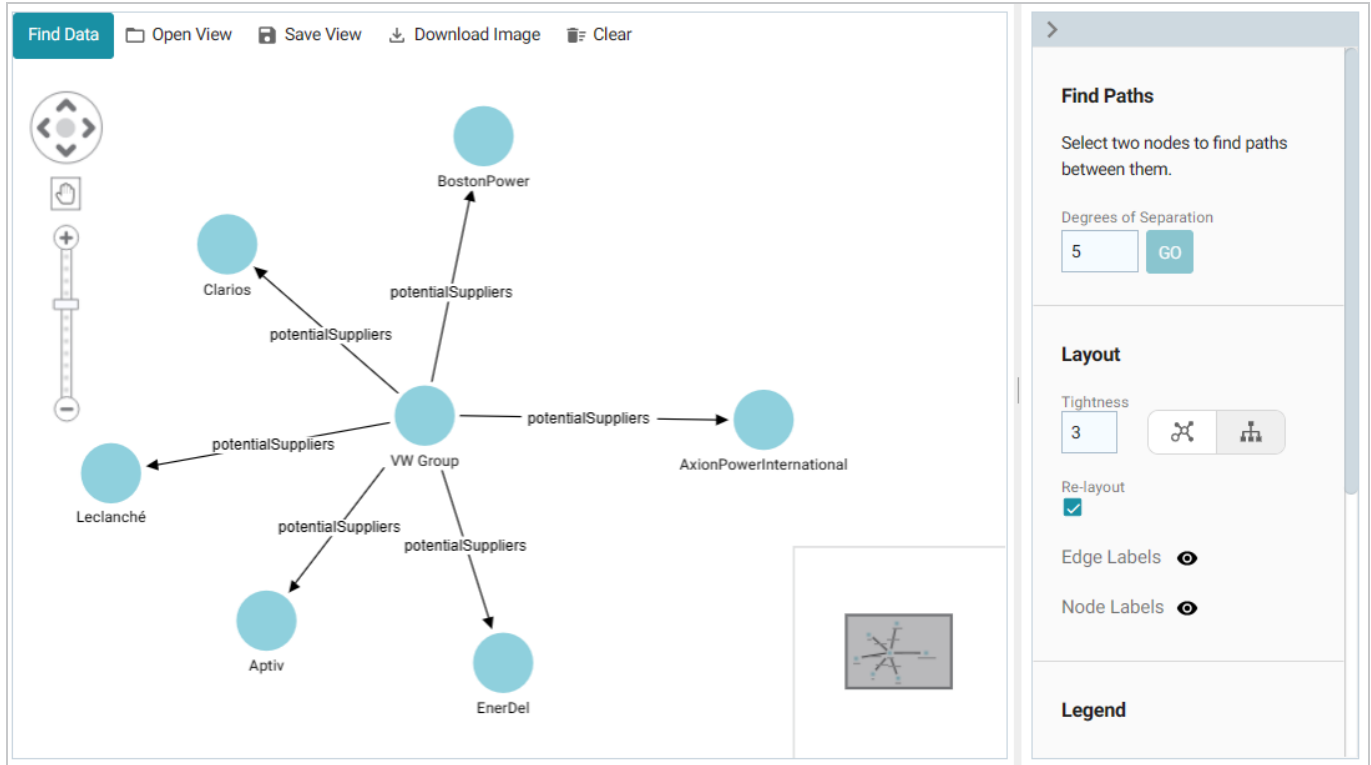
You can remove a single node or a hide a group of nodes by right-clicking a node and selecting an option from the pop-up menu (shown below). **Remove Selected Node** deletes the node and its incoming and outgoing connections. **Hide Selected Tree** hides the selected node and all of its connected nodes and paths.



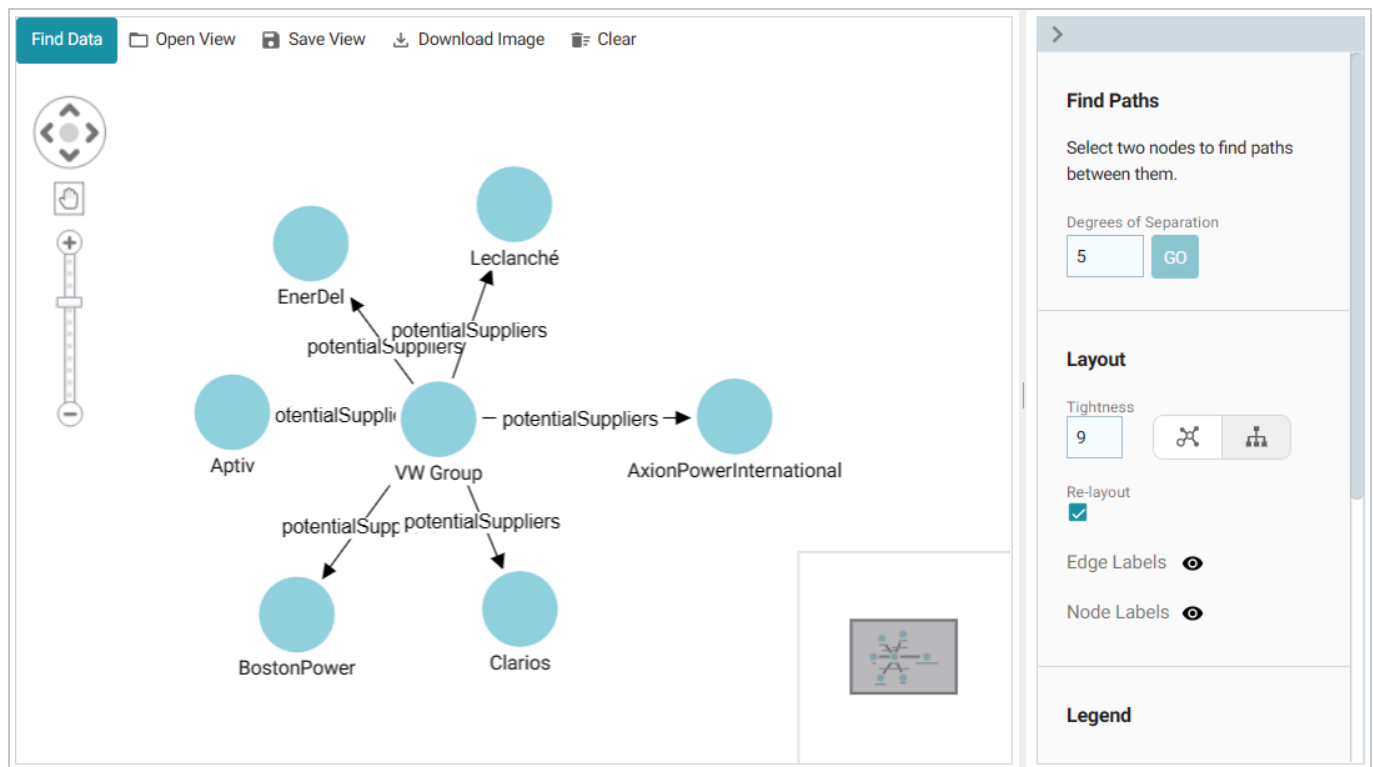
- You can continue to explore the network using the methods described above. In addition, you can adjust the layout of the View using the options under **Layout** on the right side of the screen:

## Tightness

The **Tightness** setting controls how close or far apart the nodes are. Valid values are **1 – 9**. The lower the Tightness value, the looser the nodes are. Increasing Tightness brings the nodes closer together. Type a number in the field and then press **Enter** to apply it. In the simple example below, the default Tightness value of **3** shows nodes that are far apart.

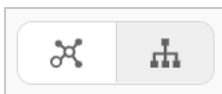


Increasing Tightness to **9** attracts the nodes closer together:

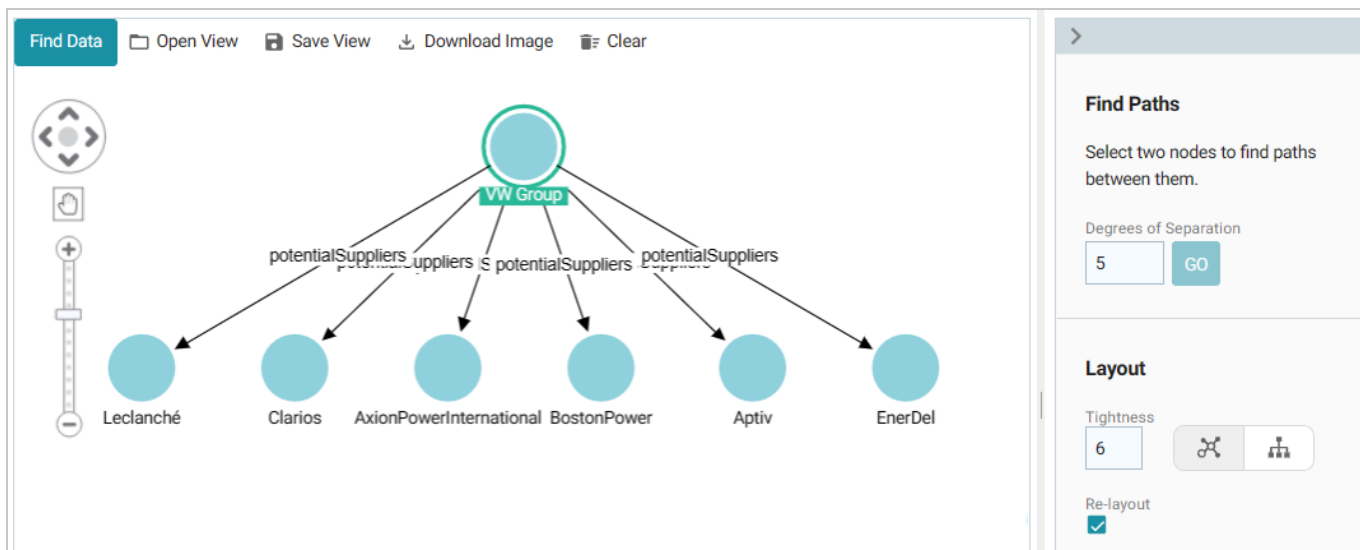


## Standard vs. Hierarchical Layout

The layout buttons (shown below) enable you to switch between a **Standard** (🔗) layout (the default setting) and a **Hierarchical** (📊) layout.



To change to a Hierarchical layout from a Standard layout, select the node in the View that is the root node and then click the **Hierarchy** (📊) button. For example, in the image below, **VW Group** was selected as the root node. Changing to the Hierarchical layout changes the View to a hierarchy:



## Re-Layout

The **Re-Layout** setting controls whether the entire View is refreshed and rearranged when a node is added. When Re-Layout is enabled, the View is rearranged when a node is added. If Re-Layout is disabled, new nodes are added to the View and existing nodes remain in place.

## Show or Hide Edge and Node Labels

The Eye (👁) icons next to **Edge Labels** and **Node Labels** enable you to toggle between showing or hiding the edge and node labels in the View.

You can save the View any time by clicking **Save View**. You are prompted to specify a **Title** and optional **Description**. To avoid creating multiple Views with the same name, make sure that you specify a unique Title. When you save the dashboard, the View that is visible at the time it is saved is presented when the dashboard is reopened. Other Views can be opened by clicking **Open View** and selecting a View. Only one View is displayed at time per dashboard. You also have the option to export a PNG version of the Network View by clicking **Download Image**.

## Related Topics

[Adding Icons to a Network Navigator Dashboard](#)

[Auto-Expanding a Network View](#)

[Swapping the Graphmart in a Network Navigator Dashboard](#)

## Adding Icons to a Network Navigator Dashboard

In Network Navigator Dashboards, you can use custom icons to represent different types (classes) of data in the knowledge graph. The steps below guide you through identifying the types in the Network View, if necessary, so that you can choose icons, uploading the icons the library, and applying the icons to specific classes.

- [Identifying Types and Preparing Icons](#)
- [Uploading and Applying Icons to a Dashboard](#)

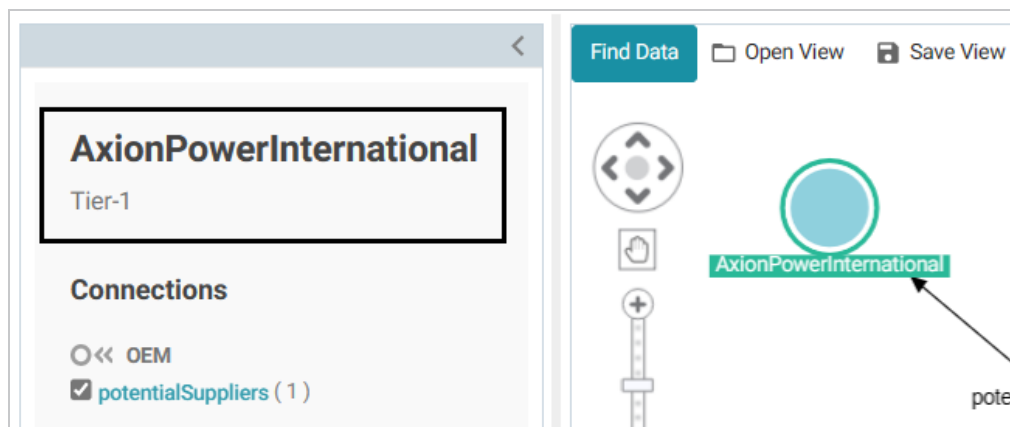
### Note

To be able to upload and apply icons, a Network Navigator Dashboard must exist and have a Network View that displays at least one node. For instructions on creating a dashboard, see [Creating a Network Navigator Dashboard \(Preview\)](#).

## Identifying Types and Preparing Icons

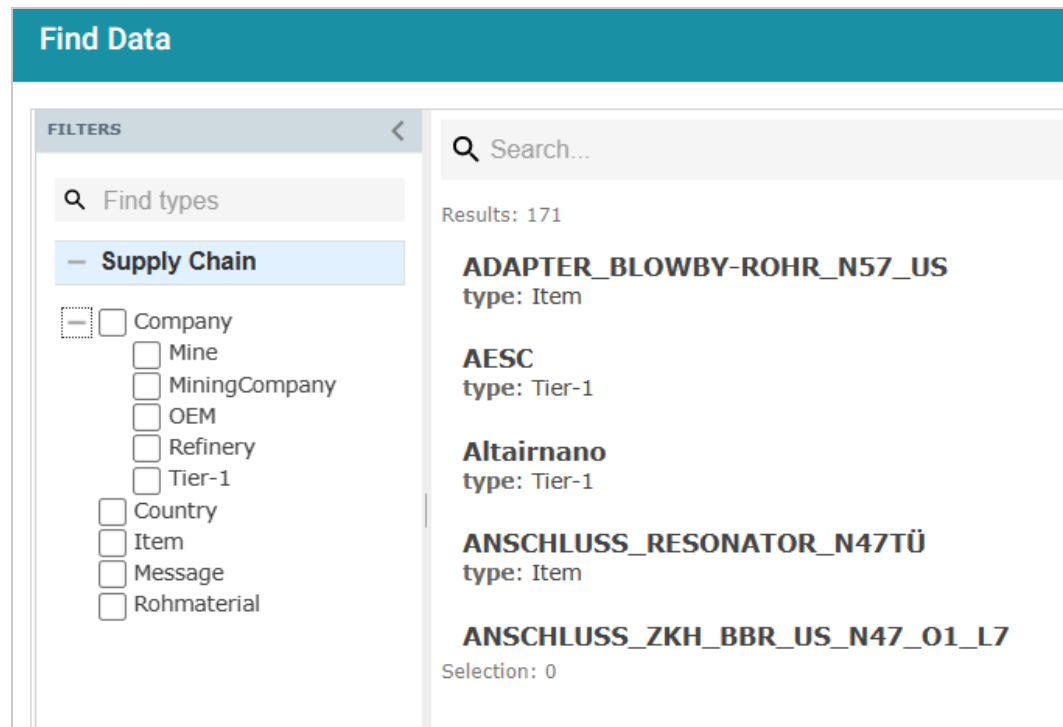
When configuring a dashboard to display icons, the icons are applied at the class (type) level. That way each property of the same type is represented by the same icon. If you are unfamiliar with the data that is displayed in a View, there are a couple of ways you can identify the classes:

- You can select a node in the View and see the type below the node label on the left side of the screen. For example, the image below shows that the **AxionPowerInternational** node is in the **Tier-1** class.



- You can click **Find Data** and open the **Filter** panel. Expand the contents to see the classes and

subclasses. For example:

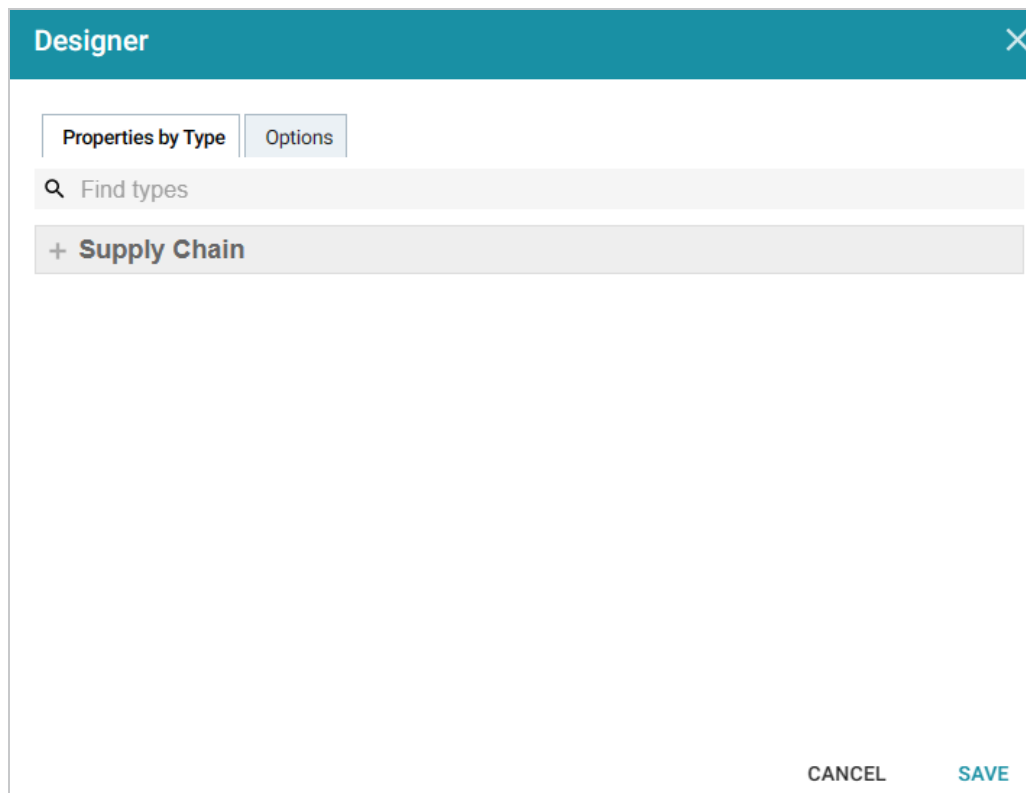


Once you know the types of the nodes in the View, you can determine the appropriate icon to use for each type. Download the icons to your computer, if necessary. When uploading icons to the library, you will browse your computer to select them. The application supports standard image formats like PNG, JPEG, and SVG.

The examples from the list above show a portion of a View and a list of the classes and subclasses for an automobile supply chain knowledge graph. The entire View (shown below) shows companies that are potential suppliers of Cobalt to the VW Group. In the example, adding icons would help users distinguish between the first tier companies, mining companies, refineries, mines, etc.



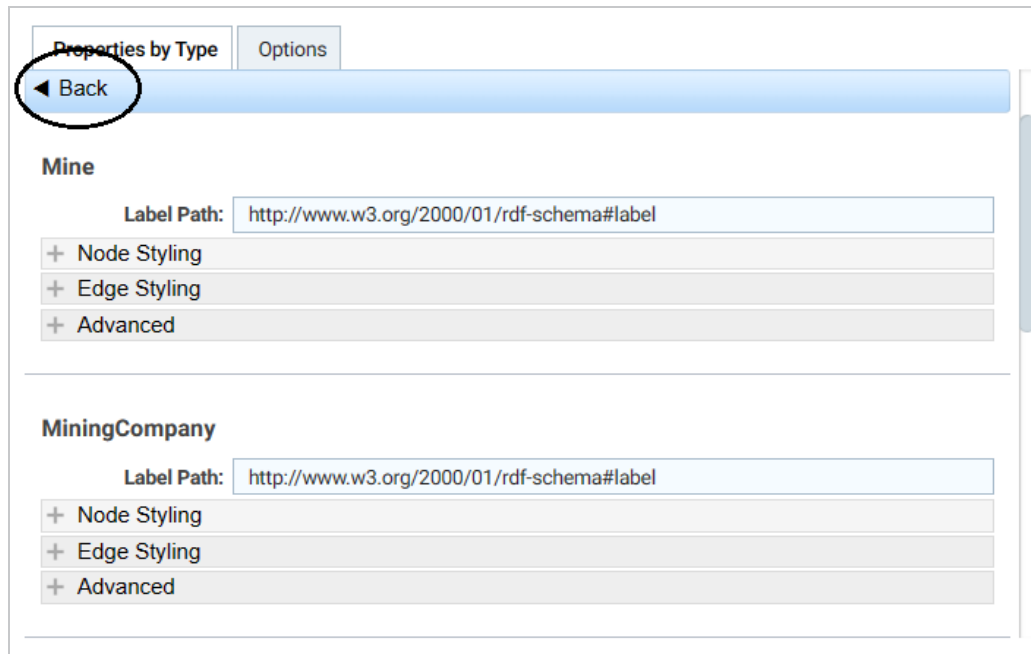




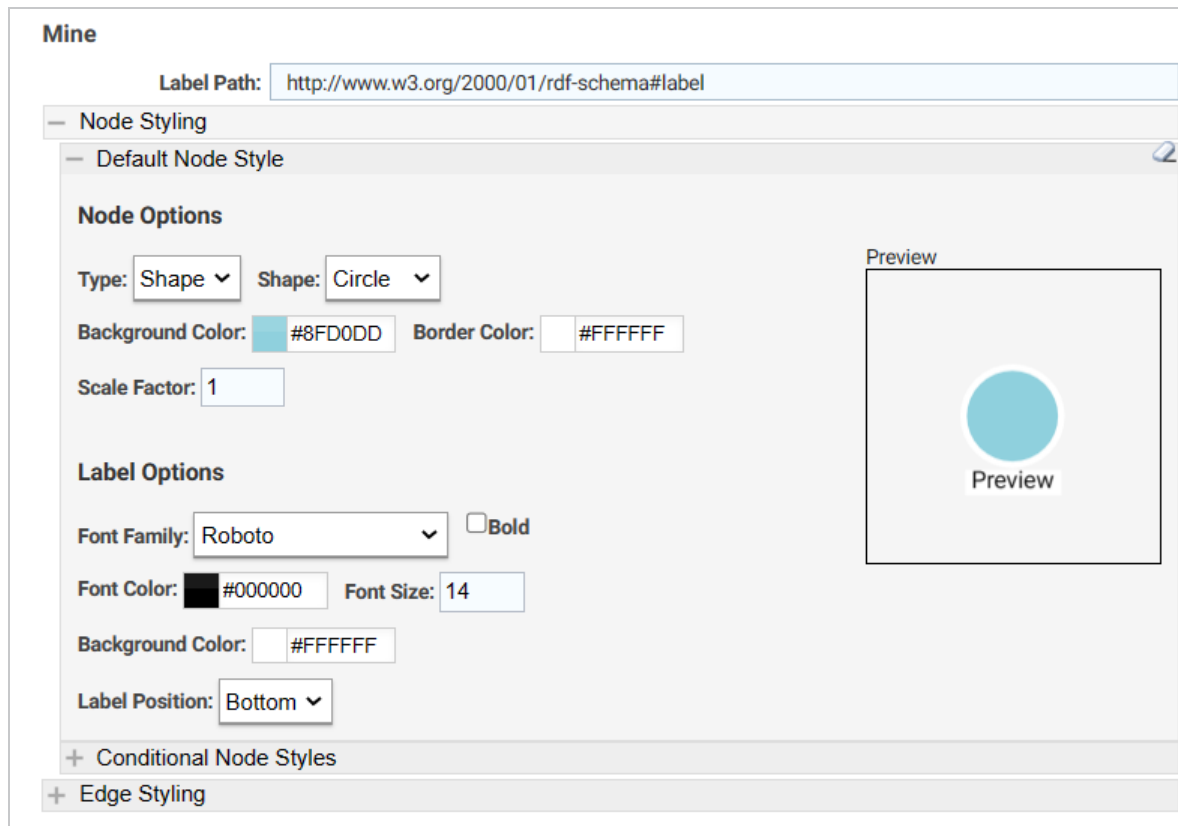
3. Expand the view of the model to display each of the types (classes). If the model contains subclasses, you can view them by clicking the **View Sub Types** link next to the parent class (shown below).



When viewing the subtypes, you can click **Back** to return to the parent level. For example:



- Under the class that you want to apply an icon to, expand **Node Styling**. Then expand **Default Node Style**. For example, the image below shows the default node style configuration for the **Mine** type.

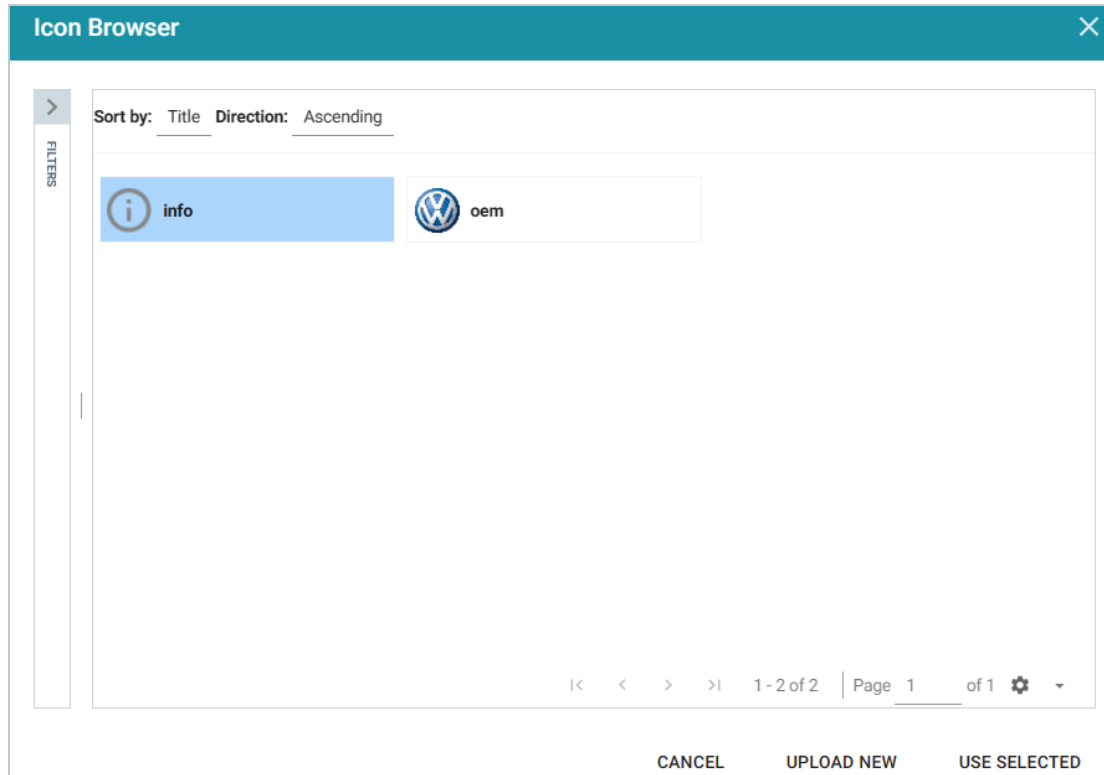


- Under Node Options, click the **Type** drop-down list and select **Icon**. An Icon field with a **Browse** button is displayed below Type:

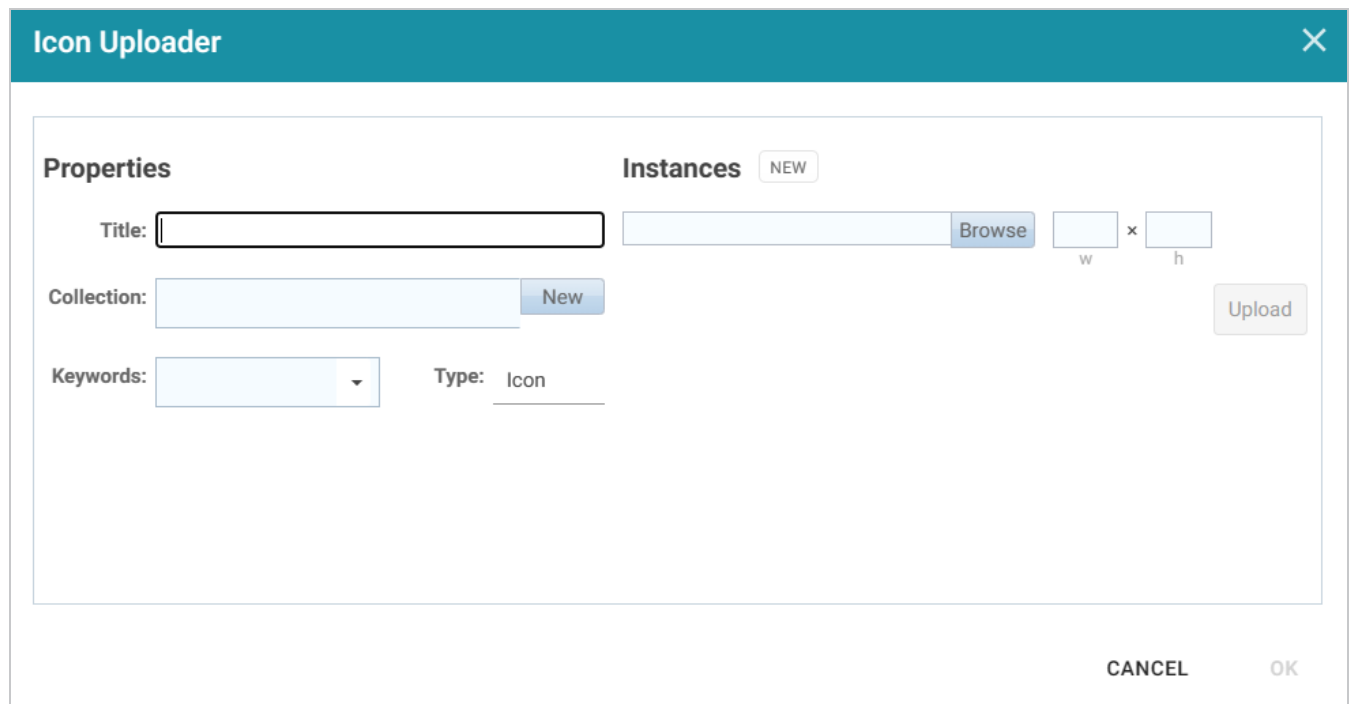
Type: **Icon** Shape: **Circle**

Icon:  **Browse**

- Click **Browse** to open the Icon Browser. The Browser lists the icons that have been uploaded to the application and gives you the option to upload new icons. For example, the image below shows that two icons have been uploaded previously and are in the library.



- To apply an existing icon, select the icon and click **Use Selected**. To add new icons, click **Upload New**. The Icon Uploader dialog box is displayed:



The image shows a dialog box titled "Icon Uploader" with a close button (X) in the top right corner. The dialog is divided into two main sections: "Properties" and "Instances".

**Properties Section:**

- Title:** A text input field.
- Collection:** A text input field with a "New" button to its right.
- Keywords:** A dropdown menu.
- Type:** A text input field with the value "Icon" entered.

**Instances Section:**

- A "NEW" button is located at the top right of this section.
- Below the "NEW" button is a "Browse" button.
- To the right of the "Browse" button are two input fields for width (labeled "w") and height (labeled "h"), separated by a multiplication sign "x".
- An "Upload" button is located at the bottom right of the "Instances" section.

At the bottom right of the entire dialog box are "CANCEL" and "OK" buttons.

8. In the Icon Uploader dialog box, complete the following fields as needed:

- **Title:** Required field that specifies the name of the icon.
- **Instances:** Required field that specifies the icon to upload. Only one icon can be selected at a time. Click **Browse** to navigate to the icon on your computer and select it. When an icon is selected, the width and height fields are populated to show the size of the selected image. You can adjust the size but it is not required. The images are automatically downsized when displayed in the View.
- **Collection:** Optional field that specifies the collection to add the icon to. If you want to be able to filter icons by collection, you can select an existing collection from the drop-down list or click **New** and type a new name to add to the list.
- **Keywords:** Optional field that specifies any keywords to tag the icon with. Keywords also enable users to filter icons.
- **Type:** Required field that specifies the type of image you are adding to the library. **Icon** is the default value and should not be changed.

The image below shows an example of the properties set for an icon that will be used to represent the nodes for mines.

**Properties**

**Instances**
NEW

Title: mine
mine.jpeg
Browse
1490 x 1609  
w h

Collection:
New

Keywords:
Type: Icon

Upload

- When you have finished configuring the icon properties, click **Upload** to upload the icon. The dialog box shows a preview of the new icon. Click **OK** to close the dialog box and add the icon to the Icon Browser.

### Tip

It is helpful to repeat Steps 7 – 9 until you have uploaded all of the icons that you want to use for the dashboard. That way you can more efficiently apply the icons to the types in the Designer.

- When you have finished uploading icons to the Icon Browser, the next step is to apply them to the classes. First, select the icon that you want to apply to the class you selected in Step 4. For example, in the image below, the **mine** icon is selected for the **Mine** class.

**Icon Browser**

>

FILTERS

Sort by: Title Direction: Ascending

info

mine

mining company

oem

refinery

tier1

1 - 6 of 6 | Page 1 of 1

CANCEL UPLOAD NEW USE SELECTED

11. Click **Use Selected** to apply the icon and return to the Designer where you can preview the icon and configure additional node options.

**Node Options**

Type: Icon Shape: Circle

Icon: *mine*  32 Browse

Background Color: #8FD0DD Border Color: #FFFFFF

☐ Auto-trim Images

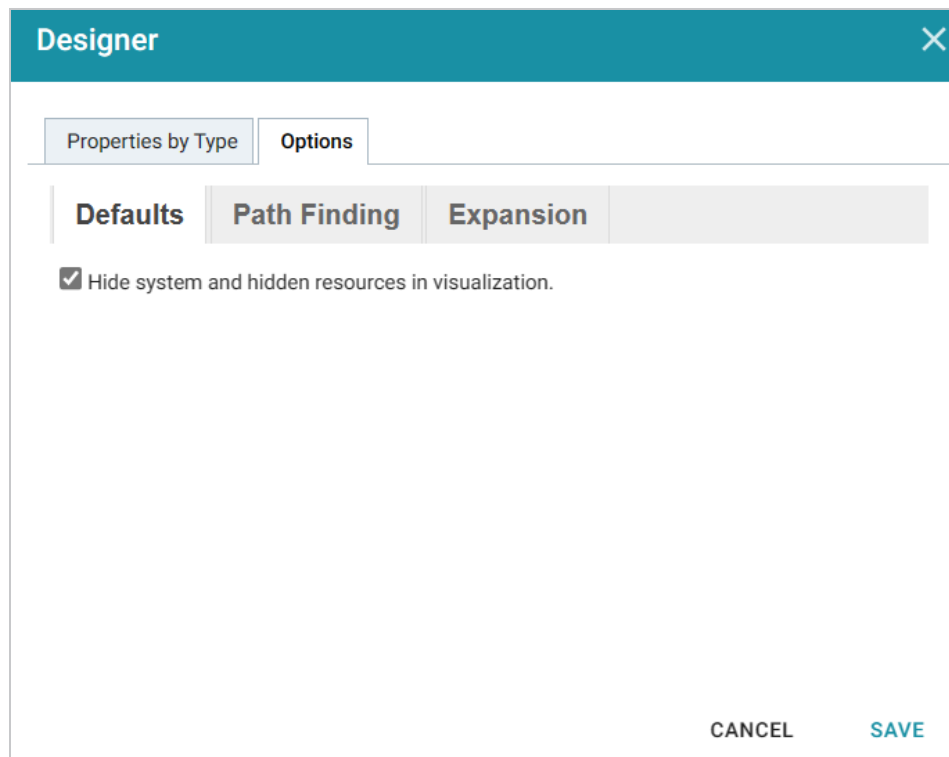
Scale Factor: 1

Preview



12. If you want to change the shape, size, and background or border color for the icon, you can adjust the following properties:
- **Shape:** Enables you to set the icon shape to Circle or Square.
  - **Background Color:** Controls the color of the background if the icon is transparent or is smaller than the chosen shape. You can click the colored square to use the color picker or type the hex value for the desired color in the text box.
  - **Border Color:** Controls the color of the border around the icon. You can click the colored square to use the color picker or type the hex value for the desired color in the text box.
  - **Auto-Trim Images:** This setting trims the image to fit in the chosen shape if it is larger than the shape.
  - **Scale Factor:** Enables you to scale the icon size up or down.
13. When you have finished configuring the icon for the selected class, navigate to the other classes and repeat the steps above to apply and configure additional icons. When you have finished applying icons, click **Save** to apply the changes and close the Designer. The new icons will be applied to the nodes in each Network View for the dashboard. For example, the image below shows the revised View of the sample automobile supply chain knowledge graph.





3. Click the **Expansion** tab to view the expansion options. For example, the image below shows the default configuration.



Properties by TypeOptions

DefaultsPath FindingExpansion

Auto expand 3 degrees by default.

Limit number of outgoing adjacent nodes to show to 4 nodes per node.

Limit number of incoming adjacent nodes to show to 4 nodes per node.

Show all connections: True

When expanding multiple levels, don't show edges for inverse properties of edges that are already visible.

**Property Inclusion List**

☐ Use Inclusion List

Select the properties to auto expand. You will still be able to manually expand properties that are not selected. Selecting none is the same as selecting all.

CANCELSAVE

4. On the Expansion tab, configure the options as needed:

- **Number of Degrees to Expand:** At the top of the screen, select the default number of degrees (hops) to auto-expand. The options are 0 – 6. When this option is set to a value greater than 0 and a node is added to a View in the dashboard, Anzo automatically populates the View with the nodes and paths that are *N* or fewer hops from the added node, where *N* is the specified value. If the View does not auto-expand when a node is added, that means there are no paths to or from that node that are *N* or fewer hops.
- **Show All Connections:** This option controls whether or not all connections are shown for all nodes.
- **Show or Hide Inverse Edges:** When auto-expanding the network multiple levels, this option controls whether to show or hide the inverse edges, i.e. whether or not to show both the incoming and outgoing connections to the same nodes. When this option is set to **don't show**, only one edge (path) will be shown between nodes. When this option is set to **show**, both the incoming and outgoing paths will be displayed.

- **Limit the Number of Adjacent Nodes:** This option controls the number of adjacent nodes to show per node.
- **Property Inclusion List:** If you would like auto-expansion to apply only to particular properties, you can select the **Use Inclusion List** checkbox. When Use Inclusion List is enabled, the dashboard's properties are listed at the bottom of the screen. Select each of the properties that you want to include.

5. When you have finished configuring the expansion options, click **Save** to apply the changes and close the Designer.

Once Auto-Expand is enabled, adding a node or path to a Network View in this dashboard will also automatically add to the View the related nodes and paths that meet the specified criteria.

## Related Topics

[Creating a Network Navigator Dashboard \(Preview\)](#)

[Adding Icons to a Network Navigator Dashboard](#)

## Swapping the Graphmart in a Network Navigator Dashboard

Follow the instructions below if you want to change a Network Navigator dashboard to access a different graphmart.

1. Open in the Hi-Res Analytics application the Network Navigator Dashboard that you want to configure.
2. In the main toolbar, click the **Dashboard** menu and select **Properties**. The Properties dialog box for the dashboard is displayed. For example:

General
Security
Advanced

### General Information

Title:\*

Explore Supply Chain

Description:

Explore suppliers for raw materials.

Graphmart:\*

Automobile Supply Chain Risk

CANCEL
SAVE

- On the General tab, click the Graphmart drop-down list and select the graphmart that you want to use in place of the current selection.

General
Security
Advanced

### General Information

Title:\*

Explore Supply Chain

Description:

Explore suppliers for raw materials.

Graphmart:\*

Automobile Supply Chain Risk

Q Search...

Automobile Supply Chain Risk
- no description -

Movies
- no description -

Northwind
- no description -

Tickets Graphmart
- no description -

4. Click **Save** to save the selection and close the Properties dialog box.

## Creating Lenses

Lenses define the data's visual presentation. Each type of lens represents a unique method for displaying data. For instance, in a column chart, you can present multiple data series for comparison. You can also apply custom formats such as fonts and colors to any lens. The topics in this section provide instructions for creating lenses.

[Creating an AnzoKO Web Page Lens](#)

[Creating Chart Lenses](#)

[Creating a Drill Down Lens](#)

[Creating a Form Lens](#)

[Creating a List Lens](#)

[Creating a Query Lens](#)

[Creating a Resource Tree Navigator Lens](#)

[Creating a Table Lens](#)

[Creating a Web Page Lens](#)

### Related Topics

[Cloning a Lens](#)

### Creating an AnzoKO Web Page Lens

The AnzoKO Web Page lens includes the [Knockout JavaScript](#) framework and enables you to create visualizations of RDF resources and metadata using knockout.js-like syntax without needing to write additional JavaScript to declare which parts of the data to render in which sections of the HTML. Follow the steps below to create an AnzoKO Web Page Lens.

1. Open the dashboard that you want to add a lens to.
2. In the Hi-Res Analytics main toolbar, click **Lenses** and select **New**. The Create Lens dialog box is displayed:

The 'Create Lens' dialog box has a teal header with the title 'Create Lens' and a close button. Below the header, it says 'Select the type of lens to create'. There is a search bar with the placeholder text 'Search for lenses...'. Below the search bar, there is a link '+ Path to lens values' and a note 'Results: 10'. A list of lens types is shown, each with an icon, a title, a description, and a 'Properties' link:

- Table**: Tabular visualization of data. Properties
- Form**: View and edit resource attributes. Properties
- Chart**: Bar, scatter, pie, etc. Properties
- AnzoKO Web Page**: A user-defined web page. Properties

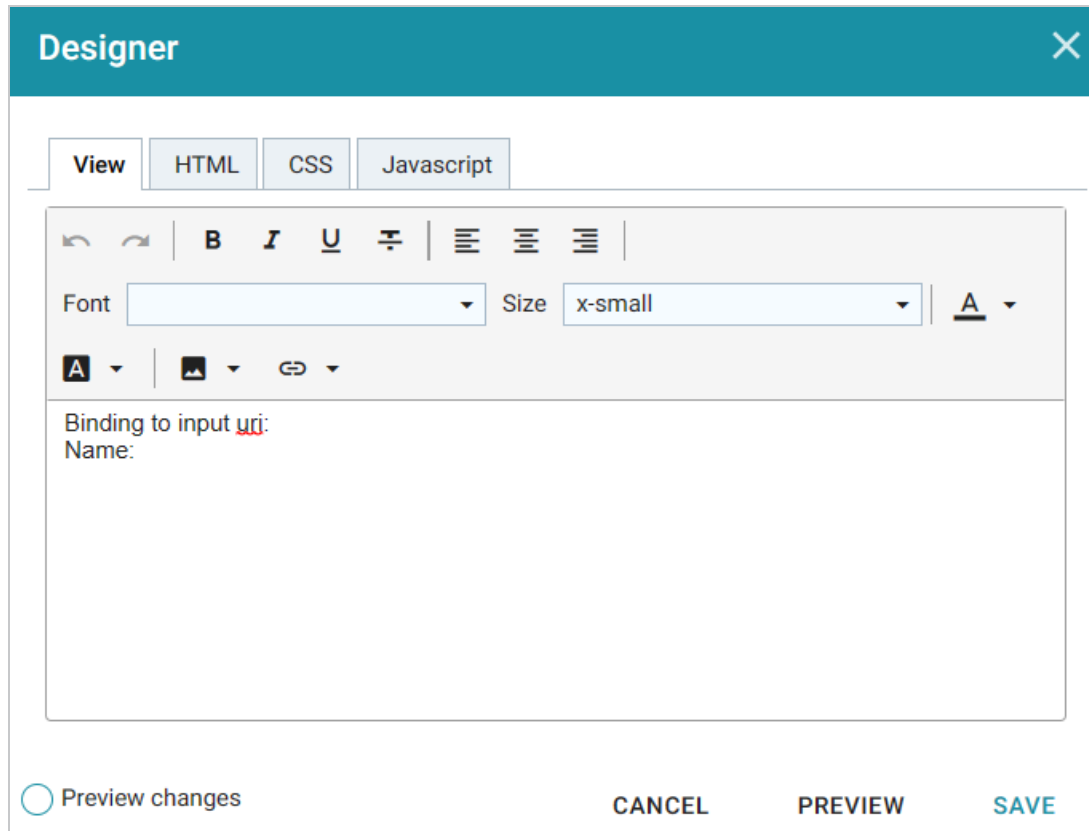
At the bottom right, there are three buttons: 'CANCEL', 'PREVIOUS' (disabled), and 'NEXT'.

- On the Create Lens dialog box, select **AnzoKO Web Page**, and then click **Next**. Anzo displays the General Information dialog box.

The 'General Information' dialog box has a teal header with the title 'Create Lens' and a close button. Below the header, it says 'Specify details about the new lens'. The section is titled 'General Information'. It contains two text input fields: 'Title:\*' and 'Description:'. Below the 'Description' field, there is a link '[+] more...'. At the bottom right, there are three buttons: 'CANCEL', 'PREVIOUS', and 'FINISH'.

- Type a **Title** and optional **Description** for the lens.

5. Click **Finish**. The lens Designer dialog box is displayed so that you can configure the lens.



The image shows a 'Designer' dialog box with a teal header and a close button (X). Below the header are three tabs: 'View' (selected), 'HTML', and 'JavaScript'. The main area contains a rich text editor toolbar with icons for undo, redo, bold, italic, underline, strikethrough, bulleted list, numbered list, and indent. Below the toolbar are two dropdown menus: 'Font' and 'Size' (set to 'x-small'), followed by a text color dropdown (set to 'A'). Below these are three more icons: a background color dropdown, an image icon, and a link icon. The main text area contains the text 'Binding to input uri:' and 'Name:'. At the bottom, there is a radio button labeled 'Preview changes', and three buttons: 'CANCEL', 'PREVIEW', and 'SAVE'.

6. Configure the lens and then click **Save** to save the configuration and add the lens to the dashboard.

## Creating Chart Lenses

Anzo Hi-Res Analytics employs the [Highcharts](#) API to provide interactive chart lenses. Follow the steps below to create a Chart lens.

1. Open the dashboard that you want to add a lens to.
2. In the Hi-Res Analytics main toolbar, click **Lenses** and select **New**. The Create Lens dialog box is displayed:

**Create Lens** [X]

Select the type of lens to create

Search for lenses...

+ Path to lens values  
Results: 10

- Table**  
Tabular visualization of data  
[Properties](#)
- Form**  
View and edit resource attributes.  
[Properties](#)
- Chart**  
Bar, scatter, pie, etc  
[Properties](#)
- AnzoKO Web Page**  
A user-defined web page.  
[Properties](#)

CANCEL PREVIOUS NEXT

- On the Create Lens dialog box, select **Chart**, and then click **Next**. Anzo displays the General Information dialog box.

**Create Lens** [X]

Specify details about the new lens

**General Information**

Title:\*

Description:

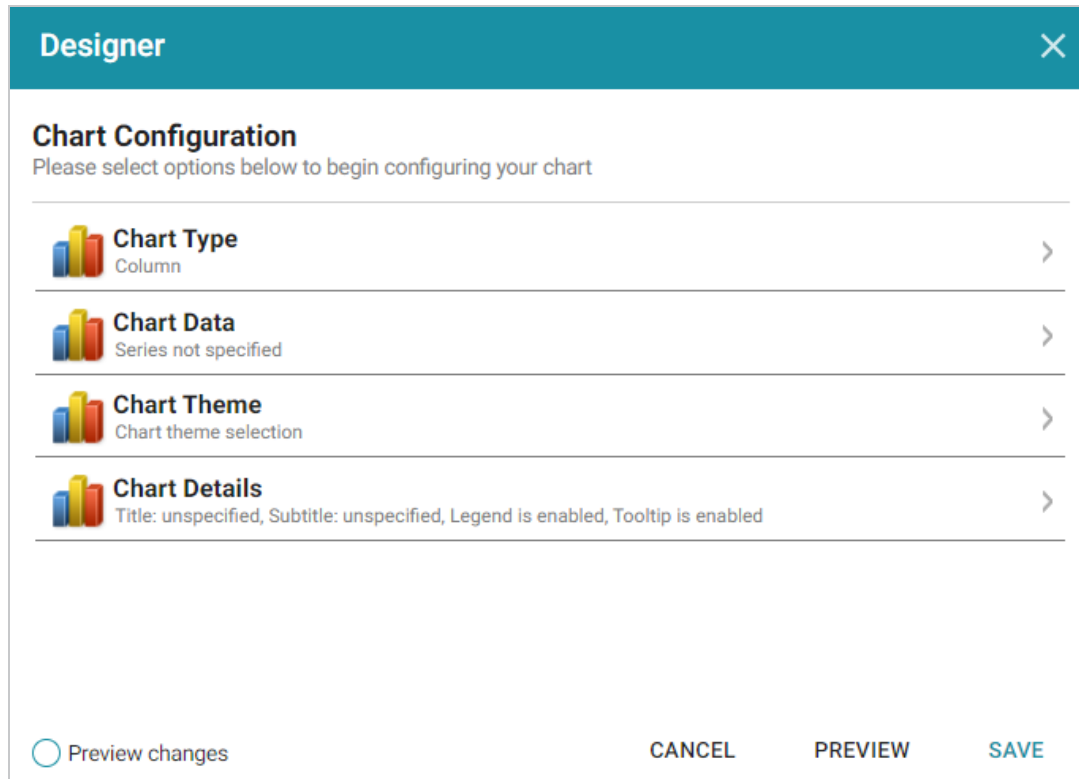
[+] more...

CANCEL PREVIOUS FINISH

- Type a **Title** and optional **Description** for the lens.



5. Click **Finish**. The lens Designer dialog box is displayed so that you can configure the lens.



The image shows a 'Designer' dialog box with a teal header and a close button (X) in the top right corner. Below the header, the title 'Chart Configuration' is displayed, followed by the instruction 'Please select options below to begin configuring your chart'. The main area contains four configuration sections, each with a bar chart icon, a title, a subtitle, and a right-pointing arrow:

- Chart Type**: Column
- Chart Data**: Series not specified
- Chart Theme**: Chart theme selection
- Chart Details**: Title: unspecified, Subtitle: unspecified, Legend is enabled, Tooltip is enabled

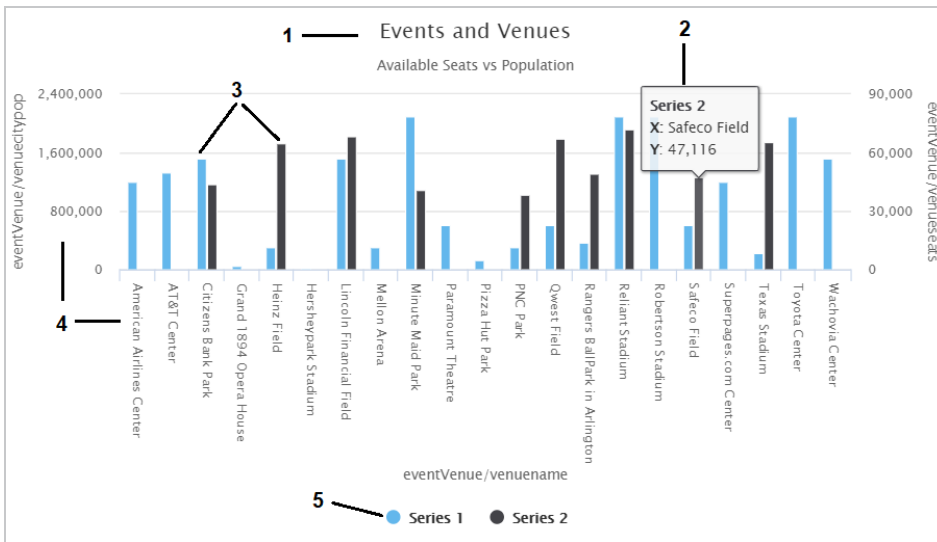
At the bottom of the dialog, there is a radio button labeled 'Preview changes' (which is currently unselected), and three buttons: 'CANCEL', 'PREVIEW', and 'SAVE' (which is highlighted in teal).

6. Configure the lens and then click **Save** to save the configuration and add the lens to the dashboard. See the sections below for configuration information.

- [Chart Concepts](#)
- [General Chart Configuration](#)
- [Area Chart](#)
- [Bar Chart](#)
- [Bubble Chart](#)
- [Column Chart](#)
- [Funnel Chart](#)
- [Heat Map](#)
- [Line Chart](#)

## Chart Concepts

This section provides information about chart concepts and describes the general, shared chart settings.



1. **Title and Subtitle:** You can configure titles and subtitles for all charts.
2. **Tooltips:** Configurable tooltips display details when users place their cursor over an item in the chart.
3. **Series:** A series is a set of properties and formulas used to display data on the chart. You can include multiple data series and configure each series individually.
4. **Axes:** The X and Y axes define the horizontal and vertical coordinates for displaying the data.
5. **Legend:** The legend differentiates each series in the chart. You can also click a series in the legend to show or hide that series in the chart.

## General Chart Configuration


When creating a chart, select the chart type that best suits your intended data presentation. All charts allow you to add multiple data series and configure each series individually.


The Chart Configuration screen is the initial screen in the Designer window that appears after you name and create the chart lens.


Designer


Chart Configuration

Please select options below to begin configuring your chart

 **Chart Type**  
Column

 **Chart Data**  
Series not specified

 **Chart Theme**  
Chart theme selection

 **Chart Details**  
Title: unspecified, Subtitle: unspecified, Legend is enabled, Tooltip is enabled

☐ Preview changes

CANCEL




PREVIEW

SAVE

Click an option to configure the chart:

- **Chart Type:** Enables you to select the type of chart that you want to display, such as column, pie, or line.
- **Chart Data:** Enables you to specify the data that will populate the chart.
- **Chart Theme:** Enables you to select a theme or color scheme for the chart.
- **Chart Details:** Enables finer-grained customization than the Chart Theme settings. You can further customize the chart design by adding details such as a chart title and subtitle and modifying styles, fonts, legend, and tooltip formats.

## Chart Designer Interface Functions

Icon	Description
	Click the eraser icon to erase all series. This action cannot be undone.
	Click the back icon to return to the previous screen.
	Click the home icon to return to the Chart Configuration screen.

Icon	Description
+	Click the plus icon to add a series.

## Chart Data

The Chart Data screen allows you to configure the Properties and formulas that populate your chart. Most charts share the same settings.

**Designer**

**Chart Data**  
Please specify what you would like to see in the chart

**Series Configurations**  
Select a configuration to edit

! Series 1  
Clone | Rename | Disable | Remove

+ Add a series

☐ Preview changes

**Series Details**  
Set properties of the selected series configuration

\*Group:  
Click to set the formula used to group by !

Group label:  
Click to set formula for the Group label value !

\*Value:  
Click to set formula for the Value value !

+ Show formulas used for creating calculated series

Data | Plot | X Axis | Y Axis | Filters

CANCEL PREVIEW SAVE

## Series Configurations

The Series Configurations section contains settings to manage the data series. Click a series to select it. The Series Details appear in the section to the right. The Series Configuration options are:

- **Clone:** Creates a new series with the same settings.
- **Rename:** Renames the series.
- **Disable:** Removes series data from the chart without deleting it.
- **Remove:** Deletes the series. You must have at least one active series for a functioning chart. Removing a series cannot be undone.

Series Details

Depending on the chart type, five tabs appear at the bottom of the Series Details screen:

- **Data**: Defines properties and formulas used to populate the chart.
- **Plot**: Defines chart formatting, including data labels, legends, and other display options.
- **X Axis**: Defines formats and labels for the X axis values.
- **Y Axis**: Defines formats and labels for the Y axis values.
- **Filters**: Defines any filters to apply to this lens.

Data

\*Group:

Click to set the formula used to group by

!

Group label:

Click to set formula for the Group label value

!

\*Value:

Click to set formula for the Value value

!

+ Show formulas used for creating calculated series

Data

Plot

X Axis

Y Axis

Filters

Field	Description
Group	Defines property and optional formulas for grouping data.
Group label	Typically the same as Group. Defines the properties and formulas to serve as the group label.
Value	Defines the property and optional formulas to populate the values in the chart.
Show formulas used for...	<div>Creates a calculated series using the following fields:</div> <div><b>Series Group</b>: Selects property and functions or formulas used to group data in addition to the Group setting.</div> <div><b>Series Label</b>: Typically the same as Series Group. The property to use as the series label.</div>

Plot

Series Chart Type

Column

>

Series Chart Style

Plot style information

>

Series Chart Data Labels

Data labels are set to automatic enablement

>

Show:

Largest

70

-Automatic-

Data

Plot

X Axis

Y Axis

Filters

Field	Description
Series Chart Type	Enables you to select a chart type for the series.
Series Chart Style	Enables you to change chart formats such as fill colors and border lines.
Series Chart Data Labels	Enables you to change chart data label formats.
Show	Enables you to define a portion of the data to display based on the largest or smallest Group Labels or Values.
Show in legend	Indicates whether to show the series' name in the legend.

X Axis

Create a new axis | Delete current axis

Axis:

X Axis 1

Title:

Sort by:

Group label ValuesAscending

☐ Display axis on the opposite side

Axis Title Details

Axis title is unspecified

Axis Labels

Axis labels are enabled

Axis Style

Axis style information

Data

Plot

X Axis

Y Axis

Filters

Field	Description
Axis	Enables you to select the X axis to use for the series if multiple axes exist.
Title	Defines the title for the X axis.
Sort by	Enables you to select the sort value (either Group label or Value) for string data types.
Display axis on the opposite side	Moves the X axis to the opposite side of the chart.
Axis Title Details	Enables you to change the format for the axis title.
Axis Labels	Enables you to change the format for axis labels.

Field	Description
Axis Style	Enables you to make axis style changes.

### Y Axis

[Create a new axis](#) | [Delete current axis](#)

Axis:

Y Axis 1

Title:

☐ Display axis on the opposite side

**Axis Title Details**
Axis title is unspecified
>

**Axis Labels**
Axis labels are enabled
>

**Axis Style**
Axis style information
>

Data
Plot
X Axis
**Y Axis**
Filters

Field	Description
Axis	Enables you to select the Y axis to use for the series if multiple axes exist.
Title	Defines the title for the Y axis.
Display axis on the opposite side	Moves the Y axis to the opposite side of the chart.
Axis Title Details	Enables you to change the format for the axis title.
Axis Labels	Enables you to change the format for axis labels.



Field	Description
Axis Style	Enables you to make axis style changes.

Filters

Series Filters

No filters specified

>

Group Filters

No filters specified

>

Value Filters

No filters specified

>

Data

Plot

X Axis

Y Axis

Filters

Field	Description
Series Filters	Enables you to define filters that apply to the entire series.
Group Filters	Enables you to define filters that apply only to the Group values.
Value Filters	Enables you to define filters that apply only to the Value values.

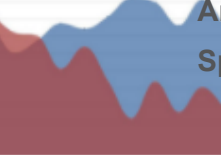
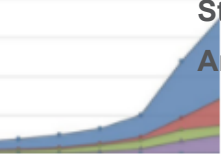
For more information about Series Details settings, see the documentation for specific lens types:

- [Area Chart](#)
- [Bar Chart](#)
- [Bubble Chart](#)
- [Column Chart](#)
- [Funnel Chart](#)
- [Heat Map](#)
- [Line Chart](#)

# Area Chart

Area charts are useful for emphasizing trends. Area charts are similar to line charts but have the added ability to display stacked data series.

There are five types of area charts:

Example	Type	Description
	Area	Connects value points on the chart with straight lines and shades the area below the line.
	Step Area	Connects value points on the chart with short horizontal steps and shades below the line. This chart emphasizes the extent of value change by expanding the data points across the x axis.
	Area Spline	Connects value points on the chart with curved lines and shades the area below the line.
	Stacked Area	Connects value points on the chart with straight lines and shades the area below the line. Add a Series Group to define the groups within the totals. Hover the mouse pointer over a colored area to view the value.
	100% Stacked Area	Compares each value as a percentage of the total and shades the area below each series. Add a Series Group to define the groups presented as a percentage within the total.

In addition to the [General Chart Configuration](#) options, the Area Chart Designer includes the following area-chart-specific settings on the Plot tab:

Series Chart Type
Area
>

Series Chart Style
Plot style information
>

Series Chart Data Labels
Data labels are set to automatic enablement
>

Series Chart Markers
Markers are enabled
>

Show:

Largest
70
-Automatic-

Show in legend:
☒

Connect missing points:
☐

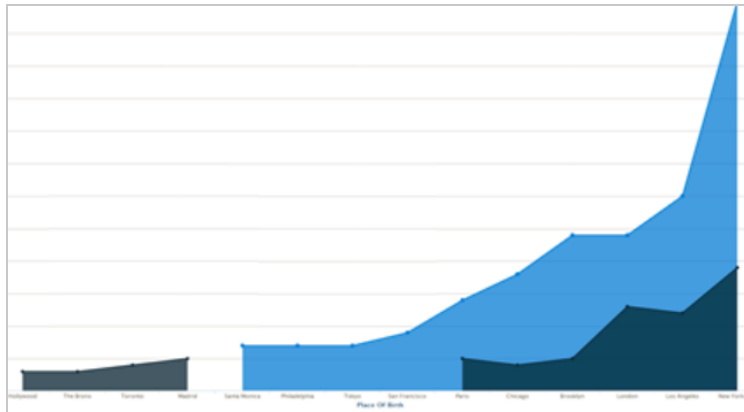
Threshold:

0

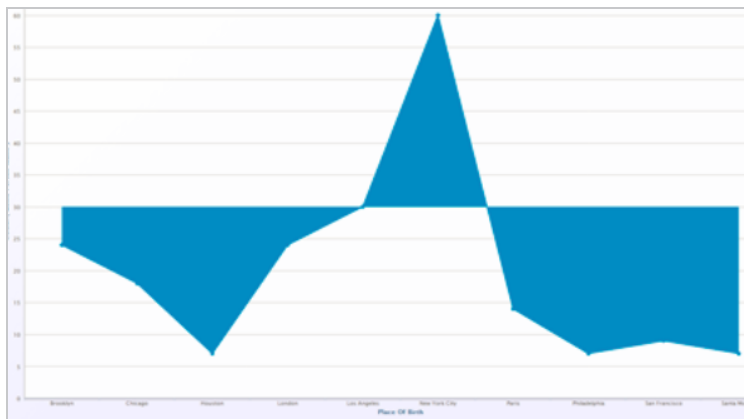
Data
Plot
X Axis
Y Axis
Filters

- Series Chart Markers:** The options in this category enable you to customize the data points that appear on the series line:
  - Enabled:** Enables or disables series chart markers.
  - Symbol:** Selects a symbol to mark data points.
  - Marker Radius:** Defines the marker size in pixels.
  - Fill Color:** Defines the marker color.
  - Outline Thickness:** Defines the thickness of the marker outline.
  - Outline Color:** Defines the color of the marker outline.
- Connect missing points:** Selecting this option connects the graph line across missing points.
 

For example, selecting **Connect missing points** for the example below would connect the two dark blue areas by filling in the space between them.





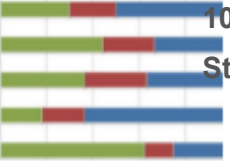
- **Threshold:** Defines the Y axis value to use as a base (starting point) for the shaded area. For example, a Threshold of 0 begins all shading at the value 0. A Threshold of 10 begins the shading at 10 and draws the area chart above or below the threshold as required. For example, the image below shows an area chart with a threshold of 30.



## Bar Chart

There are three types of bar charts:

Example	Type	Description
	<b>Clustered Bar</b>	Compares values across categories.

Example	Type	Description
	<b>Stacked Bar</b>	Compares the contribution of each value to a total across categories. Add a Series Group to define the groups within the totals. Hover the mouse pointer over a colored area to view the value.
	<b>100% Stacked Bar</b>	Compares each value as a percentage of the total. Add a Series Group to define the groups within the total.

The Bar Chart Designer uses the [General Chart Configuration](#) options.

## Bubble Chart

Bubble charts are useful for displaying data that has a third dimension. Bubble charts plot points for the X axis, Y axis, and represent relative size.

In addition to the [General Chart Configuration](#) options, the Bubble Chart Designer includes the following bubble-chart-specific settings on the Data tab:

\*Group:

Click to set the formula used to group by

!

\*Y:

Click to set formula for the Y value

!

\*X:

Click to set formula for the X value

!

\*Size:

Click to set formula for the Size value

!

+ Show formulas used for creating calculated series

Data

Plot

X Axis

Y Axis

Filters

- **Y:** Selects the Y axis values.
- **X:** Selects the X axis values.

- **Size:** Selects the property to use to determine the proportionate bubble size.

The Bubble Chart Designer also includes the following bubble-chart-specific settings on the Plot tab:

**Series Chart Type**  
Bubble

**Series Chart Style**  
Plot style information

**Series Chart Data Labels**  
Data labels are set to automatic enablement

**Series Chart Markers**  
Markers are enabled

Show:

Largest 70 -Automatic-

Show in legend:

Data Plot X Axis Y Axis Filters

**Series Chart Markers:** The options in this category enable you to customize the data points, such as outlines, that appear on the bubbles:

- **Enabled:** Enables or disables series chart markers.
- **Symbol:** Selects a symbol to mark data points.
- **Marker Radius:** Defines the marker size in pixels.
- **Fill Color:** Defines the marker color.
- **Outline Thickness:** Defines the thickness of the marker outline.
- **Outline Color:** Defines the color of the marker outline.

The Bubble Chart Designer enables you to create filters for the X, Y, and Z (Size) axes from the Filters tab.

Series Filters

No filters specified

Y Filters

No filters specified

X Filters

No filters specified

Size Filters

No filters specified

Data

Plot

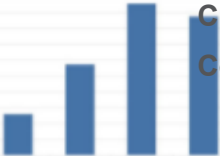
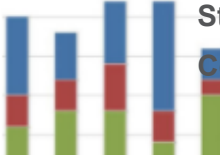
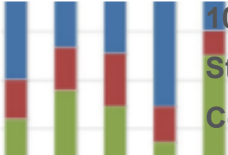
X Axis

Y Axis

Filters

Column Chart

There are three types of column charts:

Example	Type	Description
	Clustered Column	A basic column chart that compares values across categories.
	Stacked Column	Compares the contribution of each value to a total across categories. Add a Series Group to define the groups within the totals. Hover the mouse pointer over a colored area to view the value.
	100% Stacked Column	Compares each value as a percentage of the total. Add a Series Group to define the groups within the total.

The Column Chart Designer uses the [General Chart Configuration](#) options.

Funnel Chart

Funnel charts display a wide area at the top, with other data area proportionally smaller below it.



The Funnel Chart Designer uses the [General Chart Configuration](#) options.

Heat Map

Heat maps display data in tabular format within defined value ranges, such as low, medium, or high. Data points are rendered as a block of color depending on where they fall in the range.

47	46	52	16	115	120
24	117	64	19	117	6
8	76	123	114	8	12

In addition to the [General Chart Configuration](#) options, the Heat Map Chart Designer includes the following heat-map-chart-specific settings on the Data tab:

\*Group:

Click to set the formula used to group by

!

\*X:

Click to set formula for the X value

!

\*Y:

Click to set formula for the Y value

!

\*Value:

Click to set formula for the Value value

!

+ Show formulas used for creating calculated series

Data

Plot

X Axis

Y Axis

Color Axis

Filters



- **X**: Selects the X axis values.
- **Y**: Selects the Y axis values.
- **Value**: Selects the property to use for the value range.

The Heat Map Designer also includes a Color Axis tab that enables you to customize the value range block colors and axis labels and styles.

Minimum color:

Maximum color:

Axis Labels

Axis labels are enabled

Axis Style

Axis style information

Data

Plot

X Axis

Y Axis

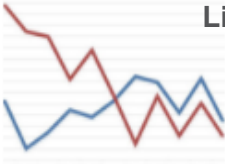
Color Axis

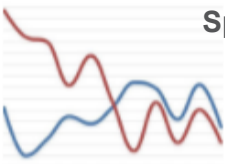
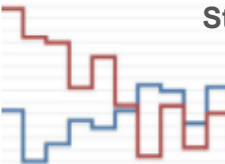
Filters

- **Minimum color**: The color to use for values in the minimum range.
- **Maximum color**: The color to use for values in the maximum range.
- **Axis Labels**: Enables you to customize the styles of the labels on the X and Y axes.
- **Axis Style**: Enables you to customize axis styles such as grid and tick lines.

Line Chart

There are three types of line charts:

Example	Type	Description
	Line	Connects value points with straight lines.

Example	Type	Description
	<b>Spline</b>	Connects value points with curved lines.
	<b>Step Line</b>	Connects value points with short horizontal steps. This chart emphasizes the extent of value change by expanding the data points across the X axis.

In addition to the [General Chart Configuration](#) options, the Line Chart Designer includes the following line-chart-specific settings on the Plot tab:

**Series Chart Type**  
Line

**Series Chart Style**  
Plot style information

**Series Chart Data Labels**  
Data labels are set to automatic enablement

**Series Chart Markers**  
Markers are enabled

Show:

Largest
70
-Automatic-

Show in legend:
☒

Connect missing points:
☐

Data
Plot
X Axis
Y Axis
Filters

**Series Chart Markers:** The options in this category enable you to customize the data points on the lines:

- **Enabled:** Enables or disables series chart markers.
- **Symbol:** Selects a symbol to mark data points.
- **Marker Radius:** Defines the marker size in pixels.
- **Fill Color:** Defines the marker color.

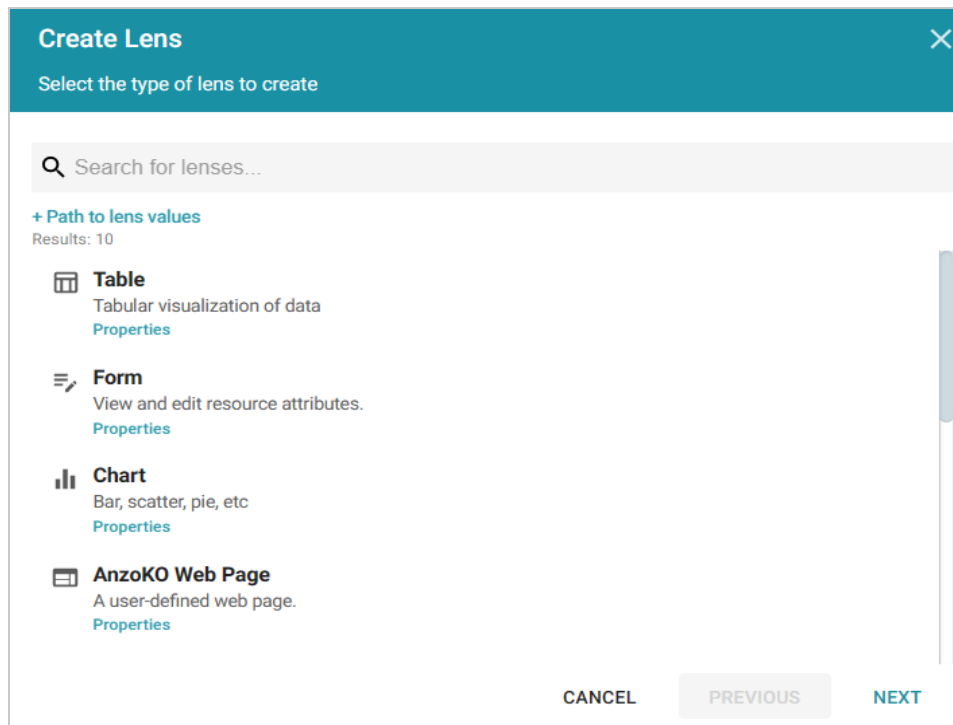
- **Outline Thickness:** Defines the thickness of the marker outline.
- **Outline Color:** Defines the color of the marker outline.

**Connect Missing Points:** Selecting this option connects the lines across missing points.

## Creating a Drill Down Lens

Drill Down lenses combine other lenses into a hierarchical interface. Clicking on an object in one lens opens the next lens in successive order. Follow the steps below to create a Drill Down lens.

1. Open the dashboard that you want to add a lens to.
2. In the Hi-Res Analytics main toolbar, click **Lenses** and select **New**. The Create Lens dialog box is displayed:



3. On the Create Lens dialog box, select **Drill Down**, and then click **Next**. Anzo displays the General Information dialog box.

### Create Lens

×

Specify details about the new lens

#### General Information

Title:\*

Description:

[+] more...

CANCELPREVIOUSFINISH

4. Type a **Title** and optional **Description** for the lens.
5. Click **Finish**. The lens Designer dialog box is displayed so that you can configure the lens.

### Designer

×

#### Lenses

+🗑️

Drag to reorder views.

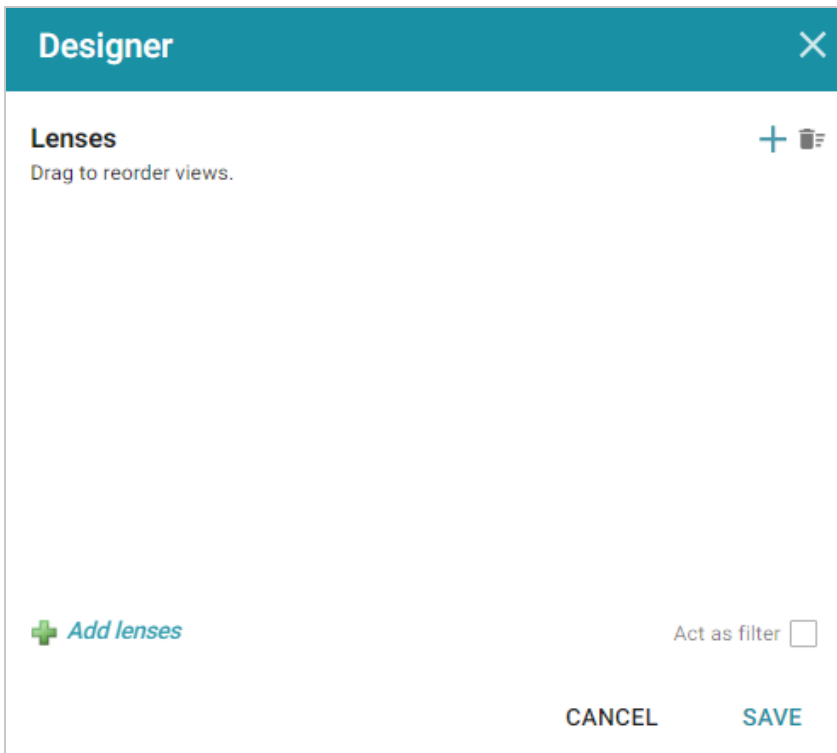
+ *Add lenses*Act as filter ☐

CANCELSAVE

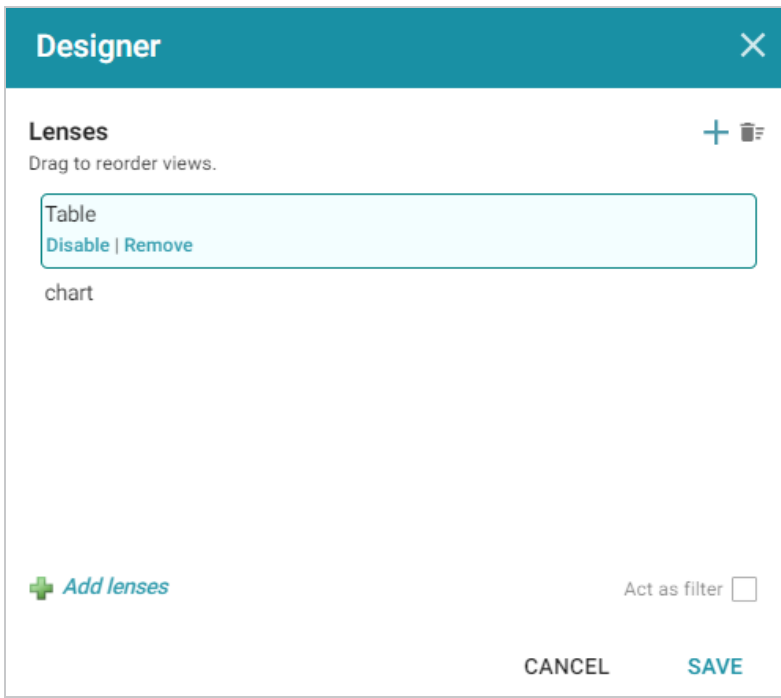
6. Configure the lens and then click **Save** to save the configuration and add the lens to the dashboard.  
See [Drill Down Lens Configuration](#) below for information about the configuration.

## Drill Down Lens Configuration

The drill down lens Designer does not require any property selections or format configurations. Instead, you configure each of the lenses that you choose.



Click the plus icon (✚) at the top or bottom of the Designer to add lenses. When you finish adding lenses, click **Save**. The lens listed first becomes the lens with the drill down functionality. Clicking a drill down icon takes you to the next lens. You can drag the lenses in the Designer to change the display order.



Anzo adds the drill down lens to the dashboard, and you can configure each lens using the Designer for that lens.

In the example below, clicking the drill down icon (▼) next to an Event ID displays the Venues lens, which shows the venue for that event:

events		
	Eventid	Eventname
▽	1	Gotterdammerung
▽	2	Boris Godunov
▽	3	Salome
▽	4	La Cenerentola (Cinderella)
▽	5	Il Trovatore
▽	6	L'Elisir d'Amore
▽	7	Doctor Atomic
▽	8	The Magic Flute
▽	9	The Fly
▽	10	Rigoletto
▽	11	Doctor Atomic
▽	12	Ring Cycle
▽	13	Lucia di Lammermoor
▽	14	La Rondine
▽	15	Die Walkure
▽	16	La Gioconda
▽	17	La Gioconda
▽	18	Gotterdammerung

## Creating a Form Lens

Form lenses enable you to create an editable or read-only form on the dashboard. Creating forms can be useful for displaying many details about each record instead of using a table where the large number of columns makes the data hard to read. Follow the steps below to create a Form lens.

### Important

By default, only the sysadmin user has access to create Form lenses. In addition, Form lenses are valid in Linked Dataset dashboards only. They do not display data for Graphmart dashboards. To create a Linked Dataset dashboard, select the **Show advanced dashboards** checkbox when creating a dashboard, and then select **Linked Dataset Dashboard**.

1. Open the dashboard that you want to add a lens to.
2. In the Hi-Res Analytics main toolbar, click **Lenses** and select **New**. The Create Lens dialog box is displayed:

**Create Lens** [Close]

Select the type of lens to create

Search for lenses...

+ Path to lens values  
Results: 10

- Table**  
Tabular visualization of data  
[Properties](#)
- Form**  
View and edit resource attributes.  
[Properties](#)
- Chart**  
Bar, scatter, pie, etc  
[Properties](#)
- AnzoKO Web Page**  
A user-defined web page.  
[Properties](#)

CANCEL PREVIOUS NEXT

- On the Create Lens dialog box, select **Form**, and then click **Next**. Anzo displays the General Information dialog box.

**Create Lens** [Close]

Specify details about the new lens

**General Information**

Title:\*

Description:

[\[+\] more...](#)

CANCEL PREVIOUS FINISH

- Type a **Title** and optional **Description** for the lens.



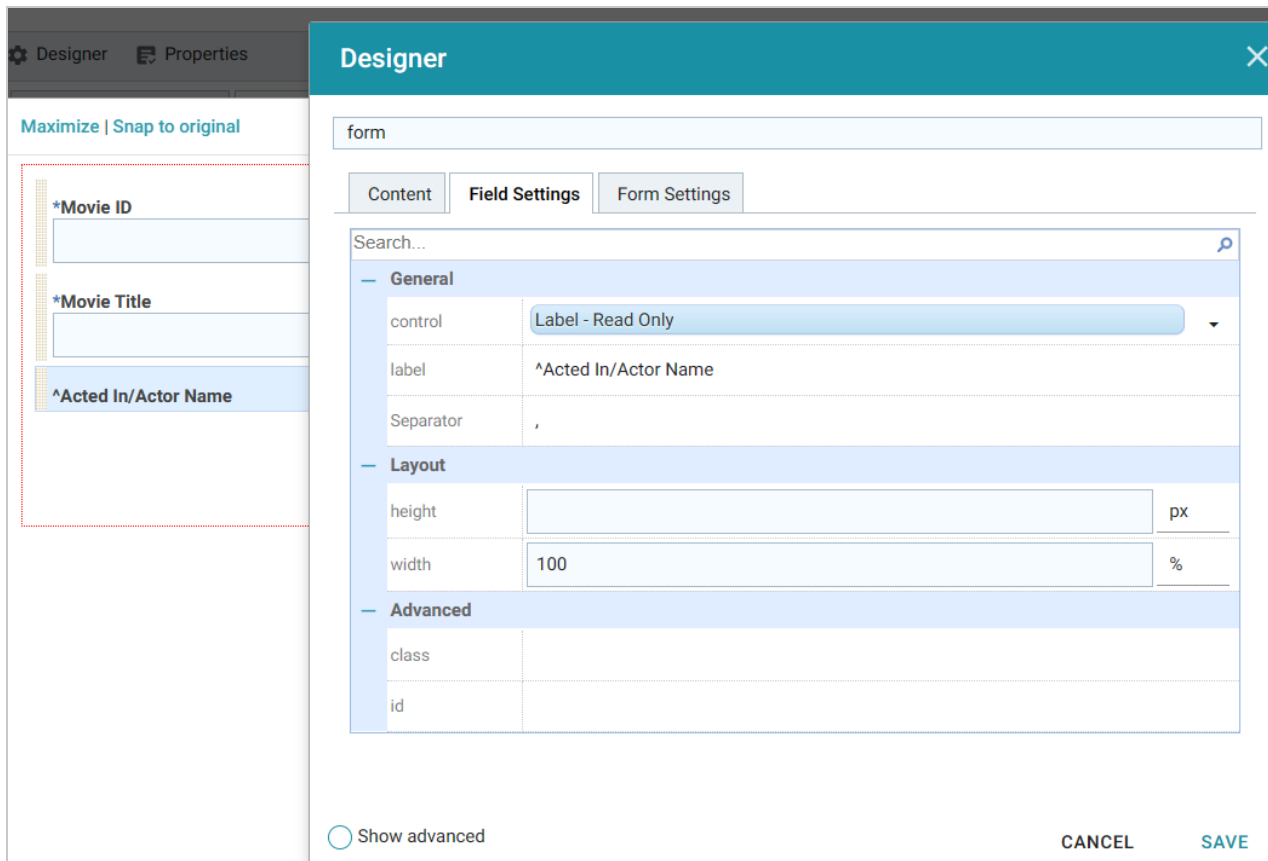
5. Click **Finish**. The lens Designer dialog box is displayed so that you can configure the lens.

The image shows a 'Designer' dialog box with a teal header and a close button (X) in the top right corner. Below the header is a search bar containing the text 'form'. Underneath the search bar are three tabs: 'Content' (selected), 'Field Settings', and 'Form Settings'. The 'Content' tab is active, showing a 'Data' section with an upward arrow and the text 'Please select a property...'. Below this is a search bar with a magnifying glass icon and the text 'Search for properties...'. Underneath the search bar, it says 'Results: 7'. There are three expandable sections: 'O>> Movies', '1 2 3 Movie ID' (with a subtext '- no description -'), and 'T Movie Title' (with a subtext '- no description -'). Below these is a section for 'O<< Movie Actors' with a subtext 'The subject is an instance of a class.' At the bottom of the 'Content' tab is a 'Layout' section with a downward arrow. At the very bottom of the dialog box, there is a 'Show advanced' link, a 'CANCEL' button, and a 'SAVE' button.

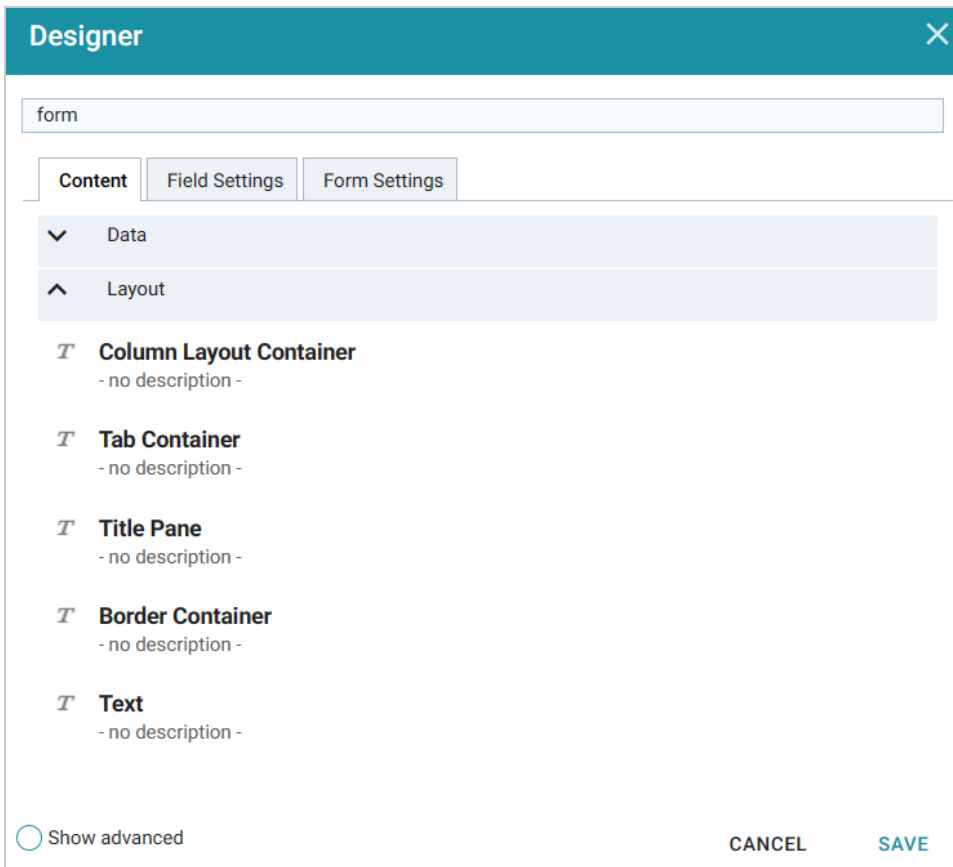
6. Configure the lens and then click **Save** to save the configuration and add the lens to the dashboard.  
See [Form Lens Configuration](#) below for information about the configuration.

## Form Lens Configuration

On the Content tab in the Designer, drag onto the dashboard each property or relative path that you want to appear as a field on the form. After adding objects, you can rearrange the form layout and use the Field Settings tab to further configure each field.



To arrange the fields in a different layout, such as a two-column layout, click **Layout** below the list of properties on the Content tab. The Designer displays the available layout containers.



Drag a container onto the form to create the layout template. You can then drag properties into the template.

## Creating a List Lens

List lenses display the values for the selected property in a list layout with icons, similar to a directory explorer view. Follow the steps below to create a List lens.

1. Open the dashboard that you want to add a lens to.
2. In the Hi-Res Analytics main toolbar, click **Lenses** and select **New**. The Create Lens dialog box is displayed:

The screenshot shows the 'Create Lens' dialog box with a teal header. The title is 'Create Lens' and the subtitle is 'Select the type of lens to create'. Below the header is a search bar with the placeholder text 'Search for lenses...'. Under the search bar, there is a link '+ Path to lens values' and a note 'Results: 10'. A list of lens types is displayed, each with an icon, a title, a description, and a link to 'Properties':

- Table**: Tabular visualization of data. Properties
- Form**: View and edit resource attributes. Properties
- Chart**: Bar, scatter, pie, etc. Properties
- AnzoKO Web Page**: A user-defined web page. Properties

At the bottom of the dialog are three buttons: 'CANCEL', 'PREVIOUS' (disabled), and 'NEXT'.

3. On the Create Lens dialog box, select **List**, and then click **Next**. Anzo displays the General Information dialog box.

The screenshot shows the 'Create Lens' dialog box with a teal header. The title is 'Create Lens' and the subtitle is 'Specify details about the new lens'. Below the header is a section titled 'General Information'. It contains two text input fields: 'Title:\*' and 'Description:'. Below the 'Description' field is a link '[+] more...'. At the bottom of the dialog are three buttons: 'CANCEL', 'PREVIOUS', and 'FINISH'.

4. Type a **Title** and optional **Description** for the lens.

- Click **Finish**. The lens Designer dialog box is displayed so that you can configure the lens.

**Designer** [Close]

**Title**

Path\* *Click to select a path* [Error Icon]

Default

**Subtitle**

Path *Click to select a path*

Default

**Icon**

Path *Click to select a path*

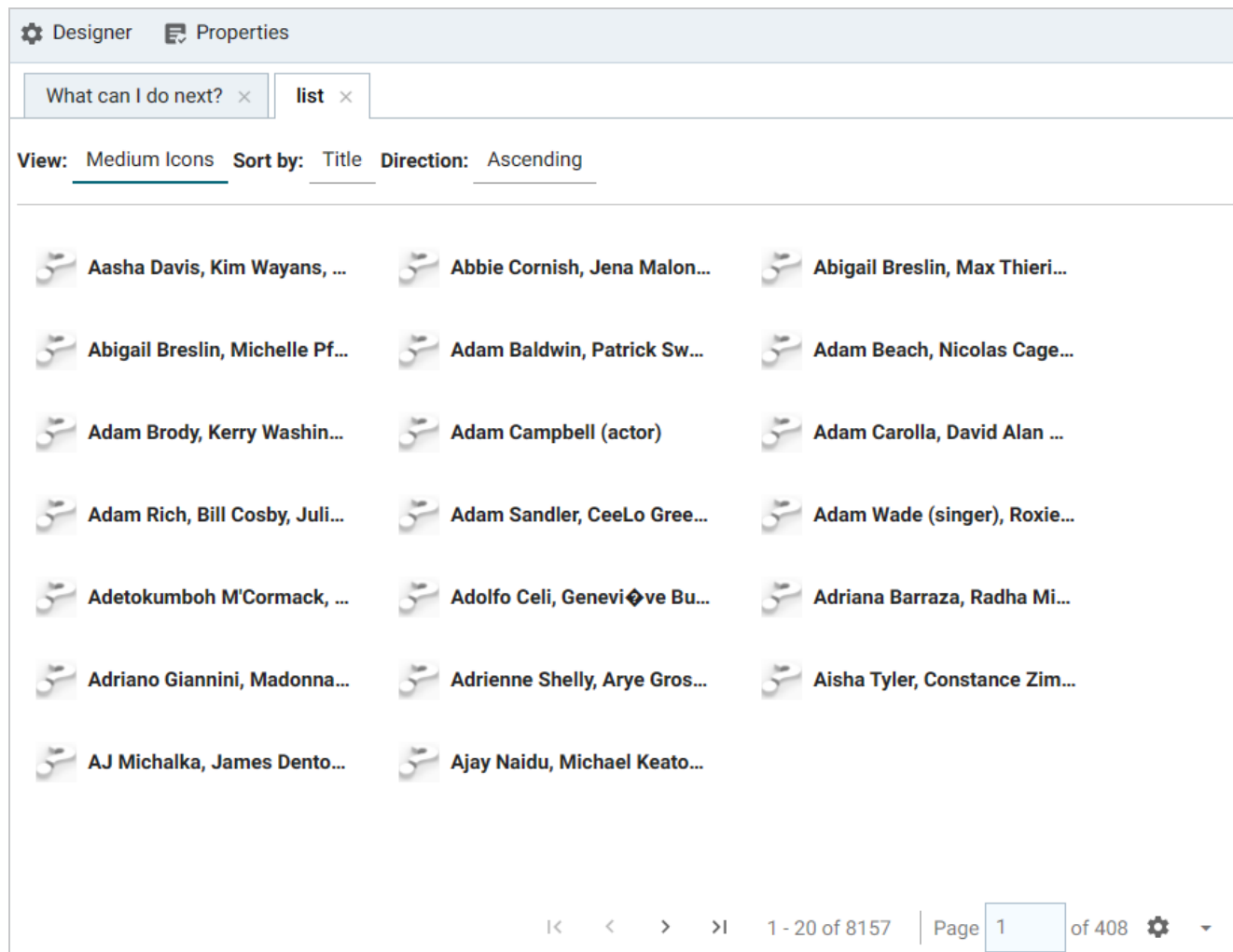
Default

Hyperlink

☒ Preview changes

CANCEL SAVE

- In the **Path** field under Title, select the property that contains the values you want to display in the list.
- If you want to include a subtitle for each value in the list, click the **Path** field under Subtitle and select the property to supply the subtitle values. For example, in the image below, Actor Name is chosen as the Title with Movie Name as the subtitle.



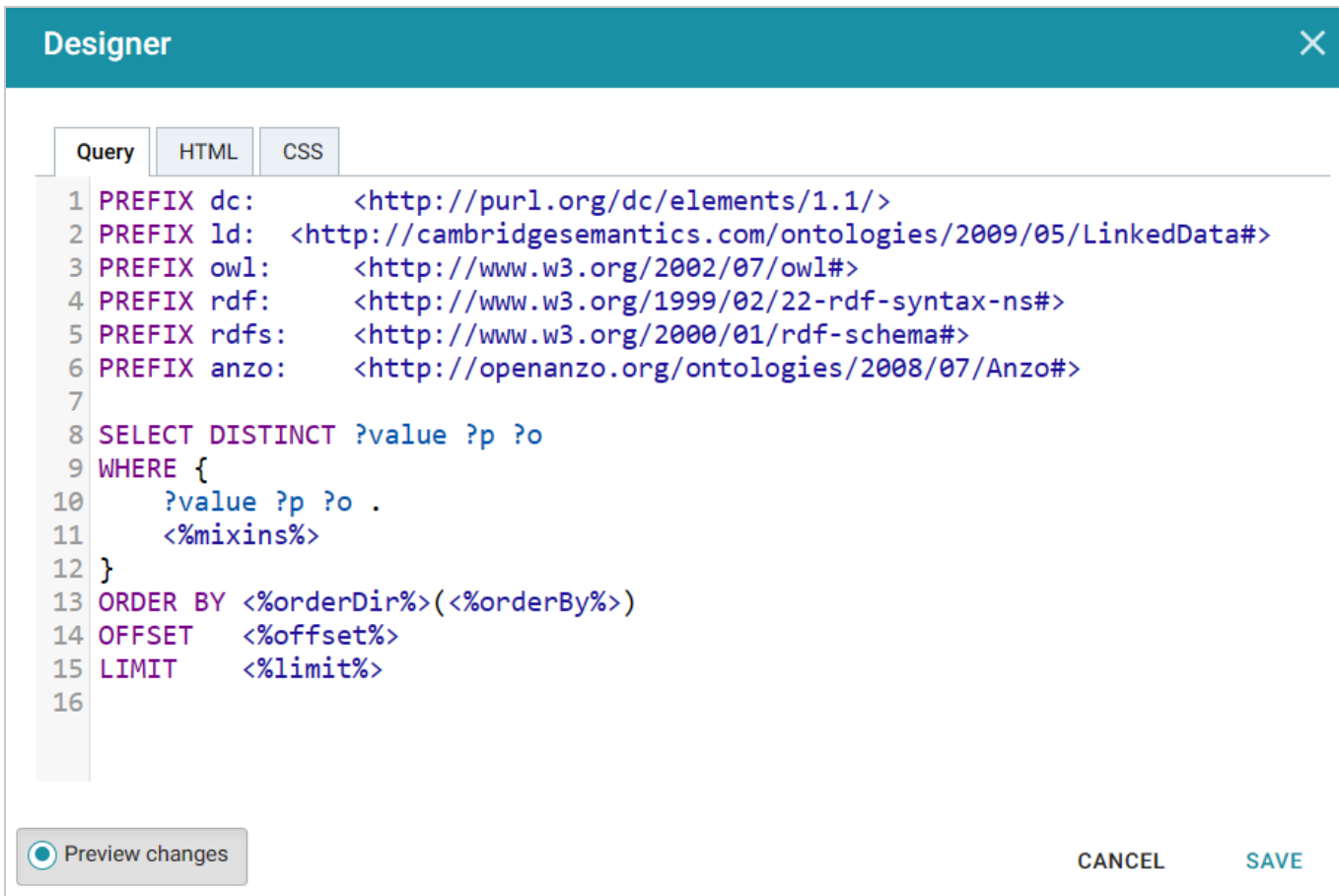
- If you have a property that contains an icon to use in place of the default question mark icons, click the Path field under Icon and select the property.
- If you want the Title to be formatted as a hyperlink to another lens, you can click **Hyperlink** and select the lens to link to.
- When you have finished configuring the lens, click **Save** to save the configuration and add the lens to the dashboard.

## Creating a Query Lens

The query lens allows you to retrieve and display data using custom a SPARQL query. You format the query results using HTML and CSS. This lens can access external SPARQL-compatible data sources. See [SPARQL Best Practices and Query Templates](#) for guidance on writing SPARQL queries.

## Query Lens Configuration

The Query lens Designer has three tabs:



- **Query:** This tab displays a SPARQL query template that you can use to write the query. Note the default code that reflects inherent functionality:
  - **<%mixins%>:** Incorporates a filter function.
  - **ORDER BY:** Incorporates a sort function.
- **HTML:** This tab includes default HTML and basic JavaScript code with sample values. You can edit the content to design the results that the query returns. The default HTML code automatically adds returned query data to a table and organizes it so that new rows are created for each record. Make sure that the `<option>` elements correspond to the elements in your query.
- **CSS:** This tab enables you to create a cascading style sheet to format the HTML and define the look and feel of the lens. Cambridge Semantics recommends that you define all CSS classes as namespaces to avoid global format changes.

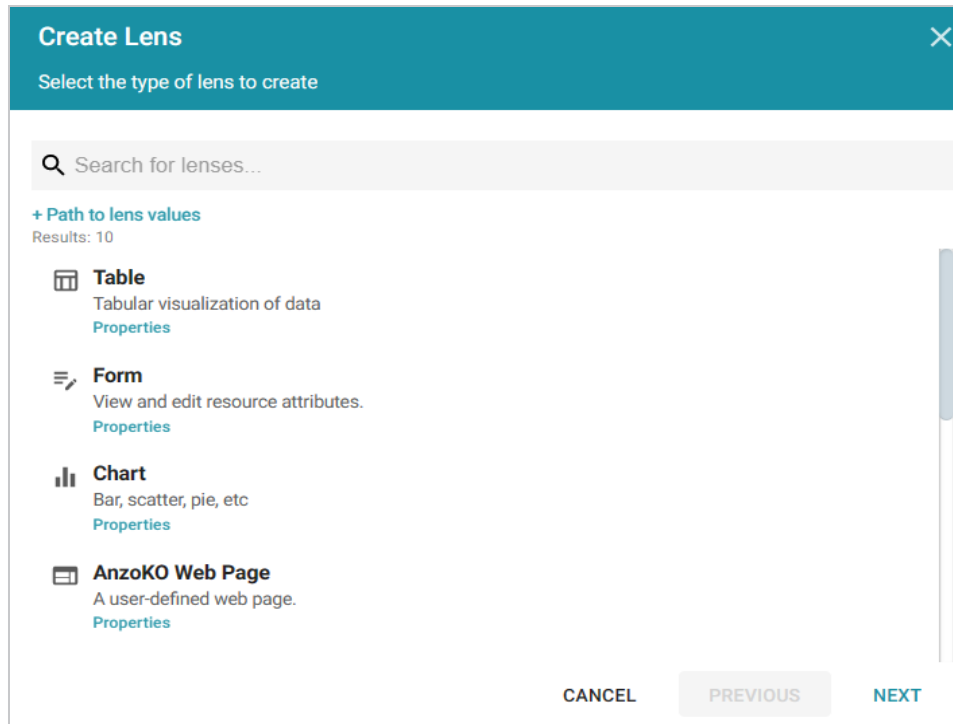
## Creating a Resource Tree Navigator Lens

The Resource Tree Navigator lens displays data in a tree format with points that you can click to open successive child data points. Follow the steps below to create a lens.

## Note

By default, the only user who has permission to create a Resource Tree Navigator lens is the **sysadmin** user. However, the sysadmin user can share created lenses with other users and groups (see [Sharing Access to a Dashboard or Lens](#)).

1. Open the dashboard that you want to add a lens to.
2. In the Hi-Res Analytics main toolbar, click **Lenses** and select **New**. The Create Lens dialog box is displayed:



3. On the Create Lens dialog box, select **Resource Tree Navigator**, and then click **Next**. Anzo displays the General Information dialog box.



Create Lens

Specify details about the new lens

General Information

Title:\*

Description:

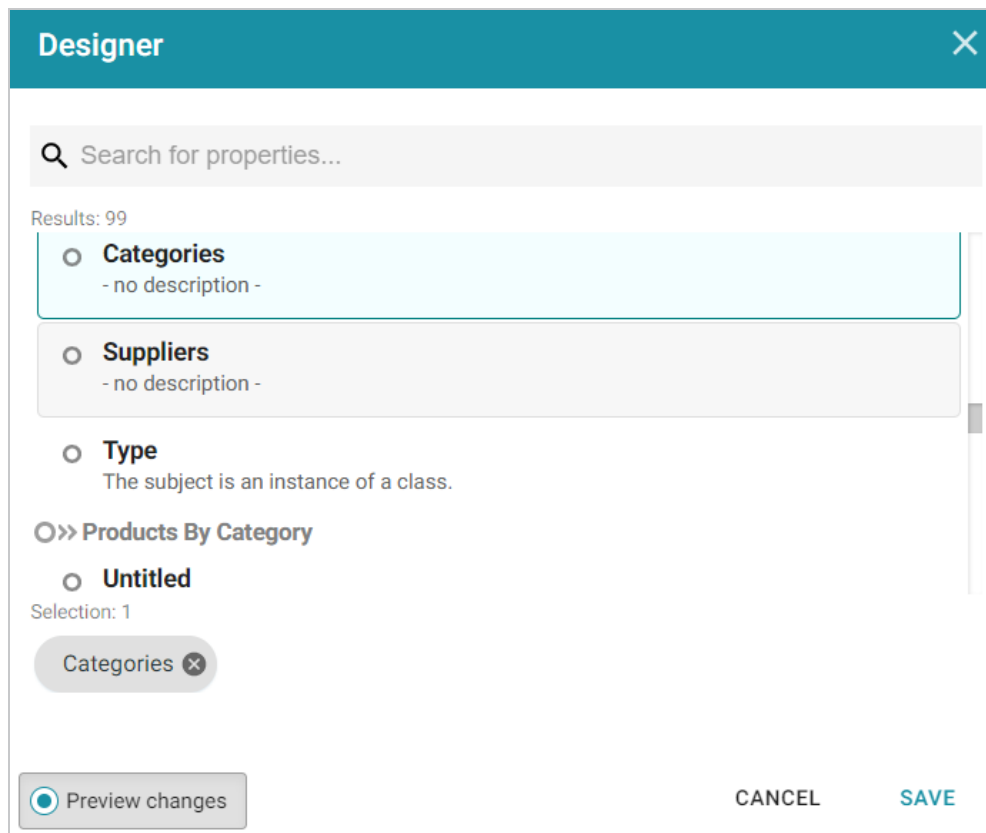
[+] more...

CANCEL

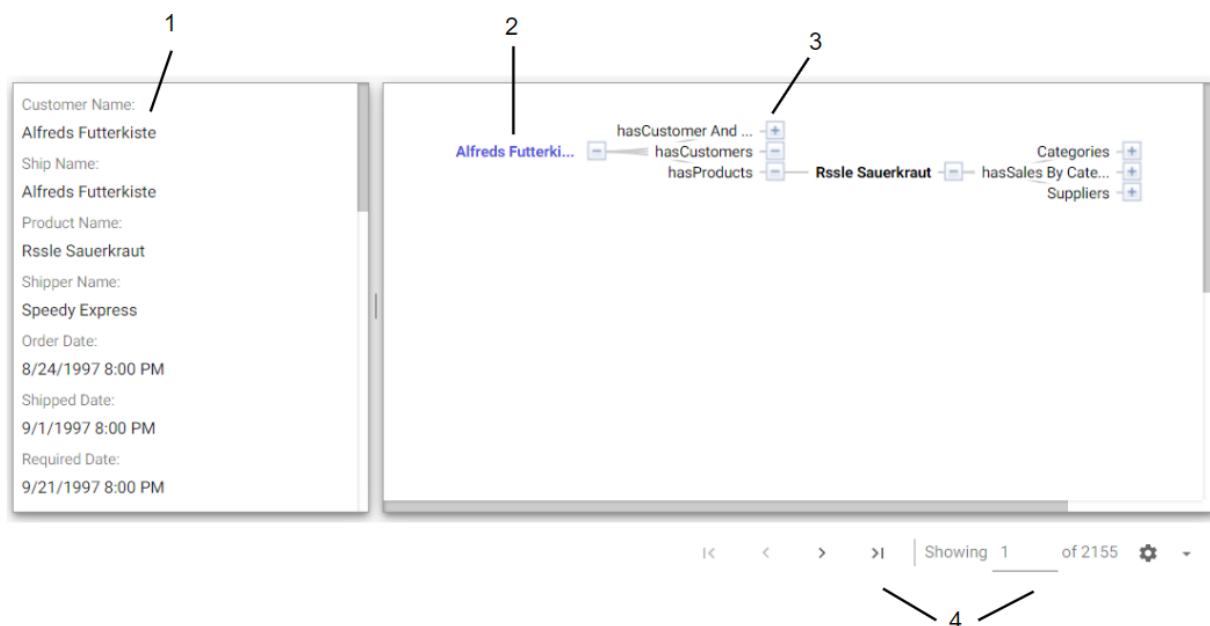
PREVIOUS

FINISH

4. Type a **Title** and optional **Description** for the lens.
5. Click **Finish**. The lens Designer dialog box is displayed so that you can configure the lens. The Designer displays all of the properties for the linked classes. Select each property that you want the resource tree to include.



- Click **Save** to save the configuration and add the lens to the dashboard. The image below shows the information that is displayed.

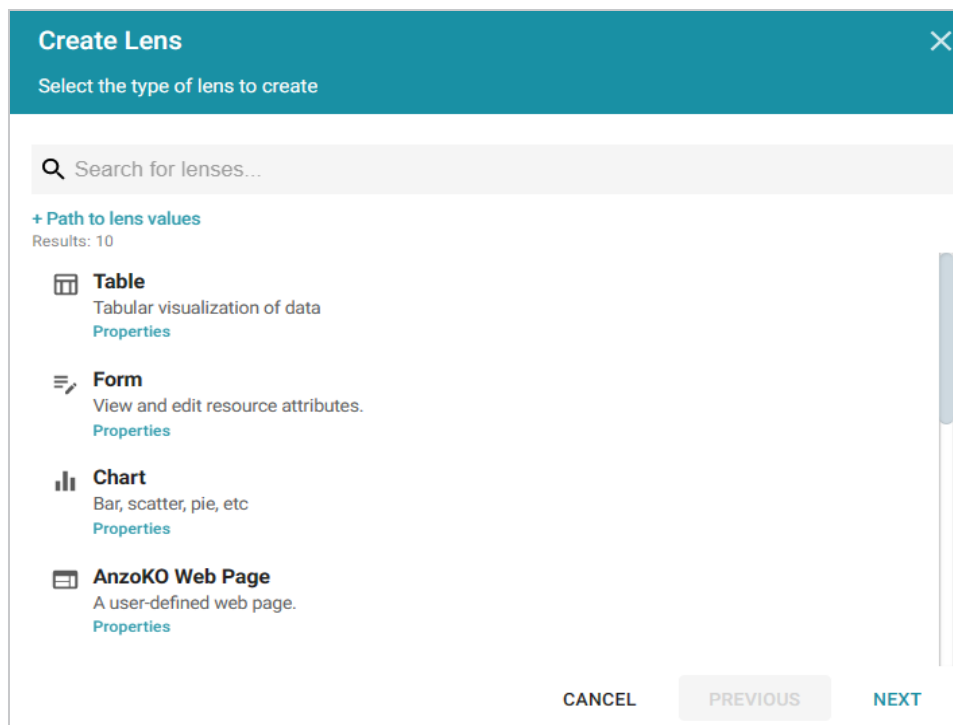


1. **Related data:** Displays the class data related to the selected data property. Data changes when another class is selected.
2. **Class property:** Displays the label property of the target class as the initial (start) point of the resource tree. Expand the tree to view child properties by clicking the plus icon for a data point.
3. **Selected linked property:** Displays the initial selected property that links to other classes.
4. **Navigation tools:** Use the arrows to navigate to other pages. The Showing text box displays the current page number and total number of pages.

## Creating a Table Lens

Table lenses display data in a standard row-and-column grid layout. Follow the steps below to create a Table lens.

1. Open the dashboard that you want to add a lens to.
2. In the Hi-Res Analytics main toolbar, click **Lenses** and select **New**. The Create Lens dialog box is displayed:



3. On the Create Lens dialog box, select **Table**, and then click **Next**. Anzo displays the General Information dialog box.

# Create Lens

×

Specify details about the new lens

## General Information

Title:\*

Description:

[+] more...

CANCEL

PREVIOUS

FINISH

4. Type a **Title** and optional **Description** for the lens.

5. Click **Finish**. The lens Designer dialog box is displayed so that you can configure the lens.

Designer

Columns

List of column configurations.

Untitled

Disable | View | Clone

Remove

Column Details

Set properties of the selected column.

Column Header Label:

Column Value Expression:

Click to edit

Format | Hyperlink | Editor

Column Footer Expression:

Click to edit

Format | Hyperlink

Column Width:

Content

Filters

Group rows by:

Click to edit

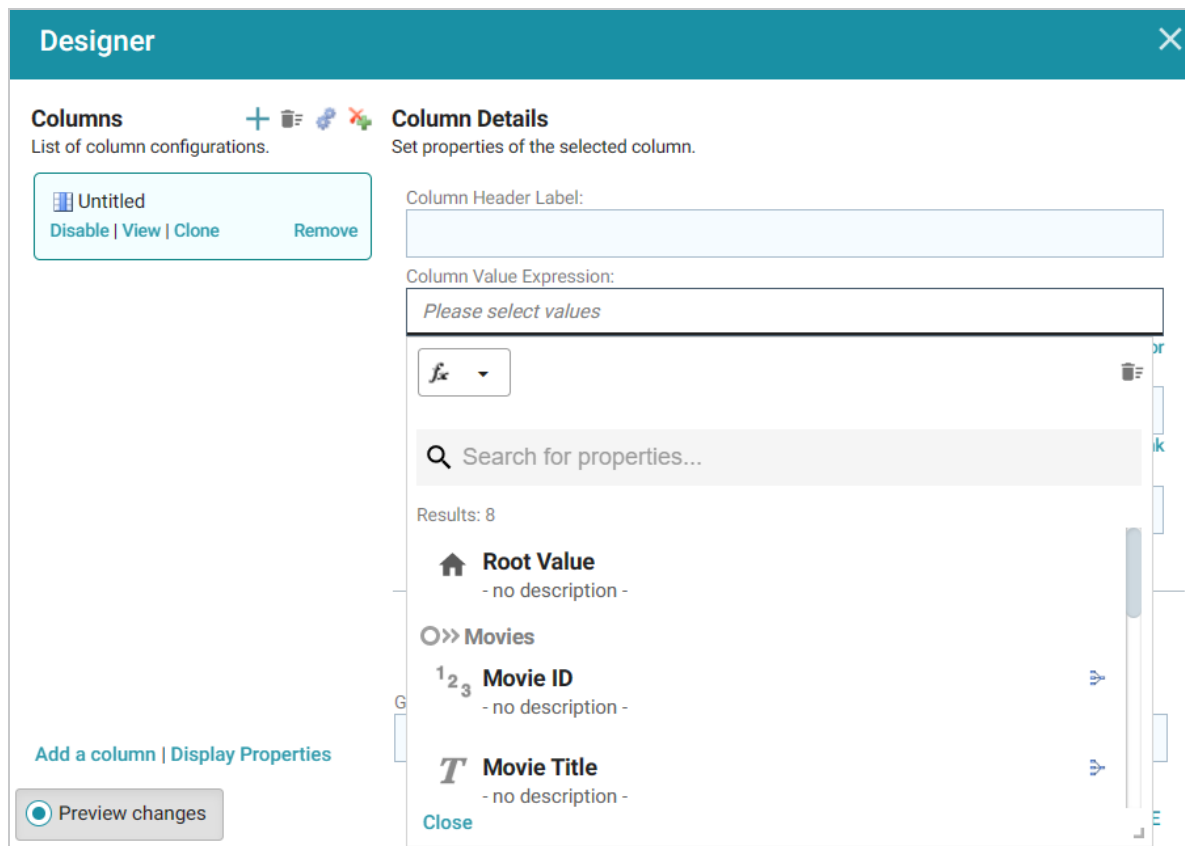
Add a column | Display Properties

Preview changes


CANCEL

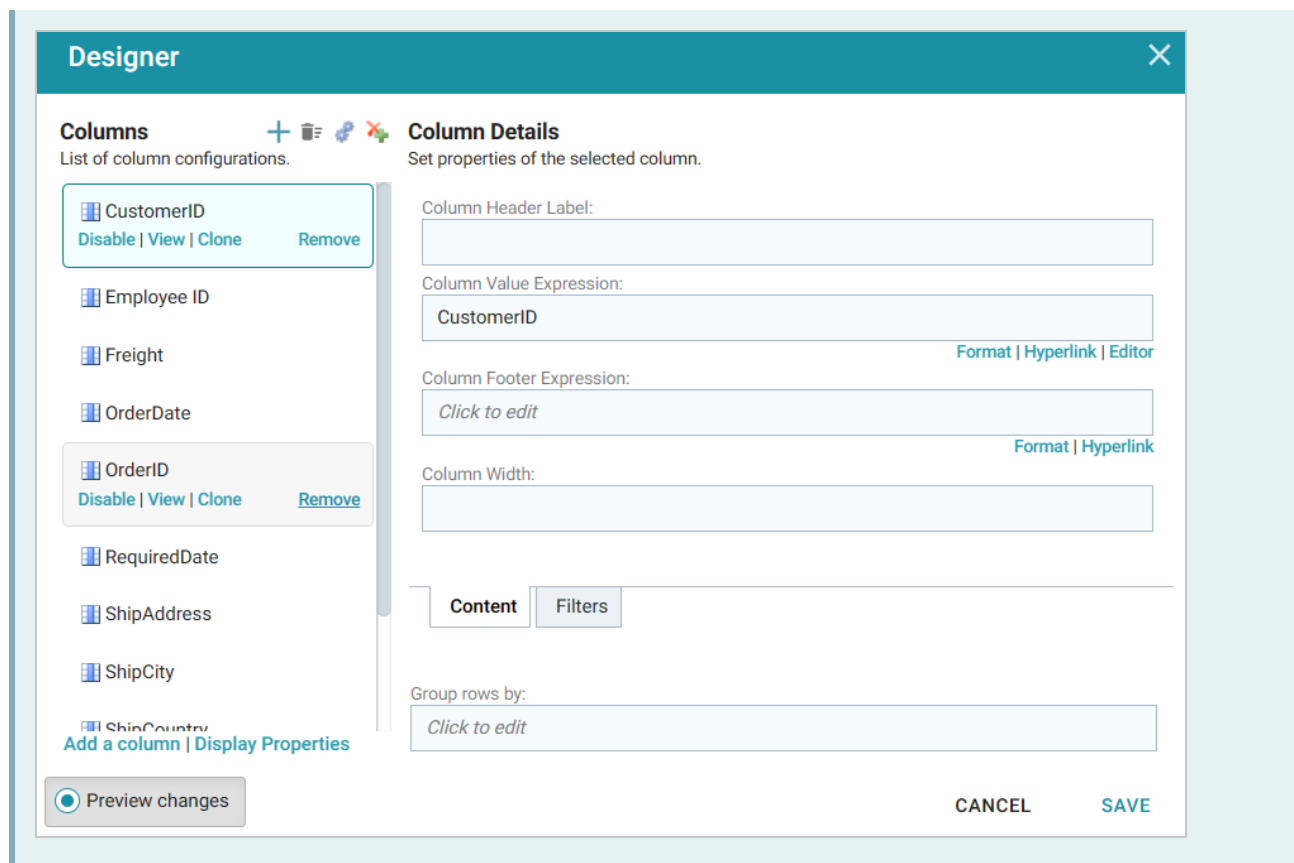
SAVE

6. To add a column to the table, click the **Column Value Expression** field and select the property name or calculation to use to populate the values in the column. For information about calculating values, see [Calculating Values in Lenses and Filters](#).



### Tip

As a shortcut to browse the available columns, you can click the Auto-generate columns icon (  ) to add all properties (for the data type selected on the dashboard) as columns. Then you can hover the pointer over columns to remove the ones you do not want to keep. For example:



7. Configure the lens and then click **Save** to save the configuration and add the lens to the dashboard. See [Table Lens Configuration](#) below for information about the configuration.

## Table Lens Configuration

The Table lens Designer enables you to choose the properties to become columns as well as apply functions and filters to the data that the table displays.

### Column Details

This section describes the fields that are available on the Content and Filters tabs in the Designer. Click a property or column on the left side of the screen to configure the options for that column.

- **Column Header Label:** (Optional) The column name to display. Overrides the Column Value Expression property name.
- **Column Value Expression:** The property name or calculation to use to populate the values in the column.
- **Column Footer Expression:** (Optional) The property to use for the table footer.
- **Column Width:** (Optional) The width of the column in pixels.
- **Group rows by:** (Optional) The property to use to group data on.

- **Filters Tab: Create filter:** Enables you to create a filter on the column. For more information, see [Creating Dashboard Filters](#).

## Default Data Display Formats

This section describes the default display formats for date and numeric values in tables.

- **Date:** By default Anzo displays date values in "short" date format. The order of the month, day, and year depends on the location of your browser. For example, in the United States the default date format is MM/DD/YYYY. In Australia, the default date format is DD/MM/YYYY. Note that this is not dependent on the Anzo server location but on the location auto-detected by the browser.
- **Numeric:** Anzo displays the complete value without a limit on precision. Numeric formats are also dependent on the location of the browser. For example, in the United States the default format for a large number is 4,294,967,295.00 and in Canada the default format is 4 294 967 295,000. Note that this is not dependent on the Anzo server location but on the location detected by the browser.

## Creating a Web Page Lens

Web Page lenses enable you to display data by creating a web page using HTML, CSS, and JavaScript. This lens is for advanced users with coding skills in these areas. A powerful feature of the this lens is the ability to bind data to Anzo graphs so that updates are reflected in real time. Follow the steps below to create a Web Page lens.

1. Open the dashboard that you want to add a lens to.
2. In the Hi-Res Analytics main toolbar, click **Lenses** and select **New**. The Create Lens dialog box is displayed:



The 'Create Lens' dialog box has a teal header with the title 'Create Lens' and a close button. Below the header, it says 'Select the type of lens to create'. There is a search bar with the placeholder text 'Search for lenses...'. Below the search bar, there is a link '+ Path to lens values' and a note 'Results: 10'. A list of lens types is shown, each with an icon, a title, a description, and a link to 'Properties':

- Table**: Tabular visualization of data. Properties
- Form**: View and edit resource attributes. Properties
- Chart**: Bar, scatter, pie, etc. Properties
- AnzoKO Web Page**: A user-defined web page. Properties

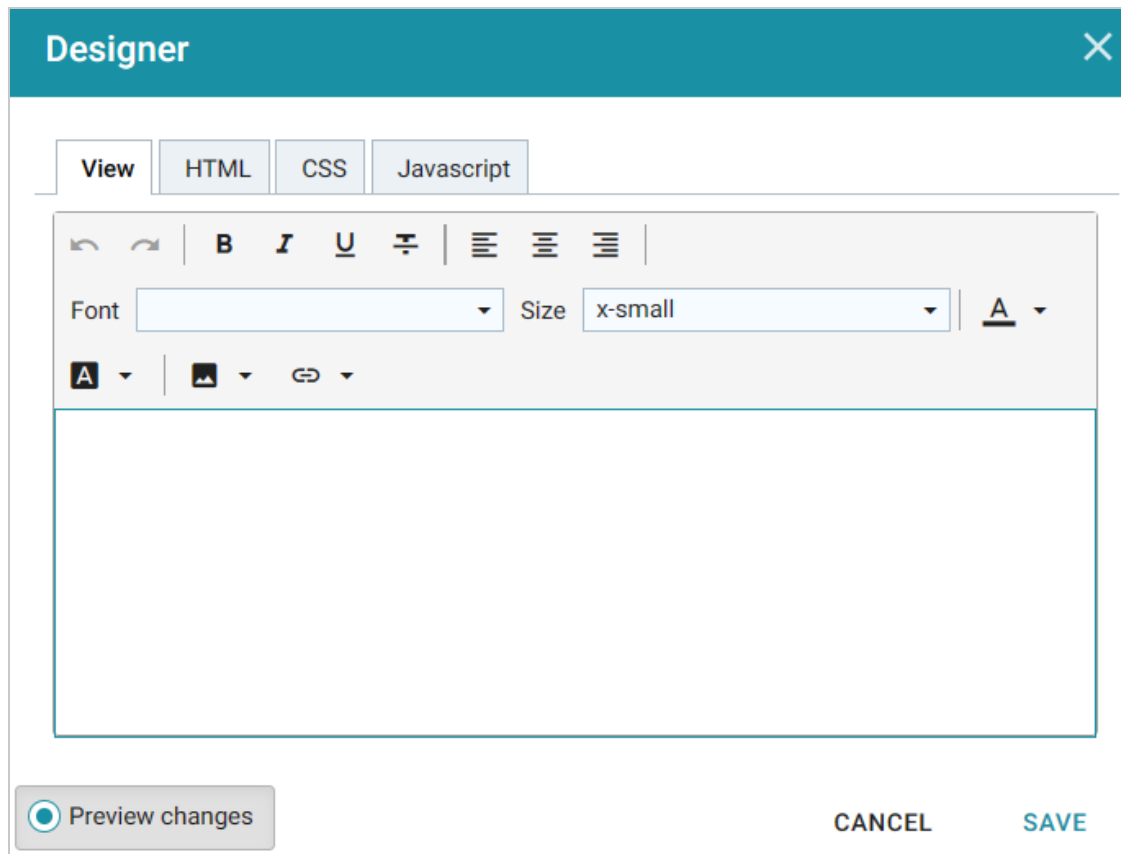
At the bottom right, there are three buttons: 'CANCEL', 'PREVIOUS' (disabled), and 'NEXT'.

- On the Create Lens dialog box, select **Web Page**, and then click **Next**. Anzo displays the General Information dialog box.

The 'General Information' dialog box has a teal header with the title 'Create Lens' and a close button. Below the header, it says 'Specify details about the new lens'. The section is titled 'General Information'. It contains two text input fields: 'Title:\*' and 'Description:'. Below the 'Description' field, there is a link '[+] more...'. At the bottom right, there are three buttons: 'CANCEL', 'PREVIOUS', and 'FINISH'.

- Type a **Title** and optional **Description** for the lens.

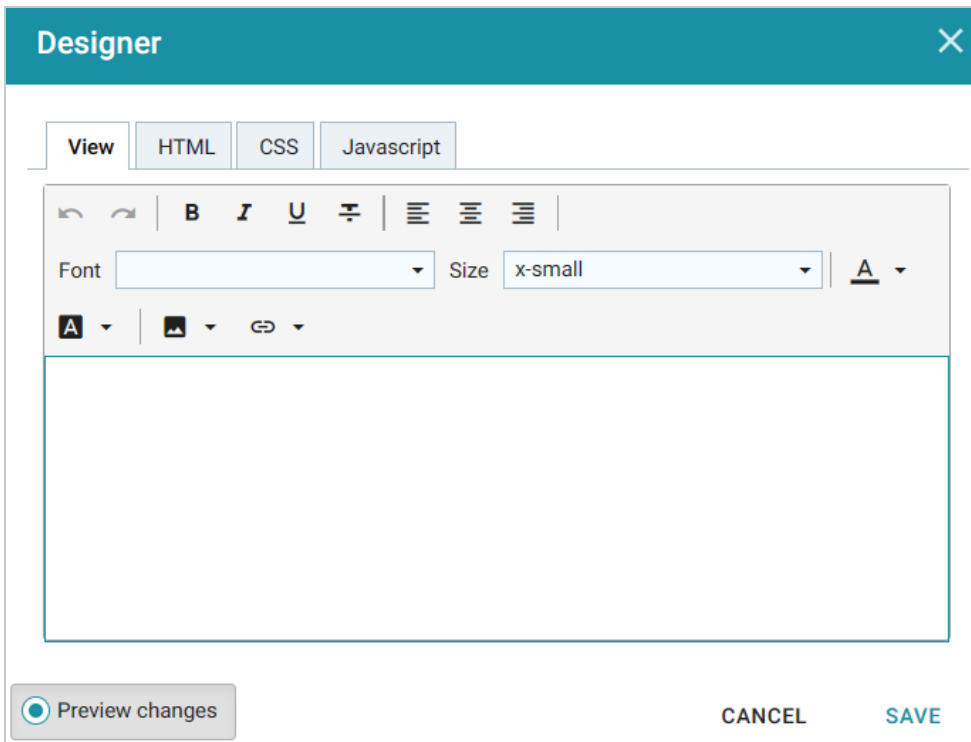
5. Click **Finish**. The lens Designer dialog box is displayed so that you can configure the lens.



6. Configure the lens and then click **Save** to save the configuration and add the lens to the dashboard.  
See [Web Page Lens Configuration](#) below for information about the configuration.

## Web Page Lens Configuration

The Web Page Designer has four tabs:



- **View:** Provides a rich text interface for viewing the page (WYSIWYG). Changes made to this page are reflected in the HTML code.
- **HTML:** This tab enables HTML coding and data binding. The example HTML image below shows code that defines text format as well as data binding using the `anzowbind:innerhtml` command.



For more information about data binding, see the [Data Binding Example](#) section below.

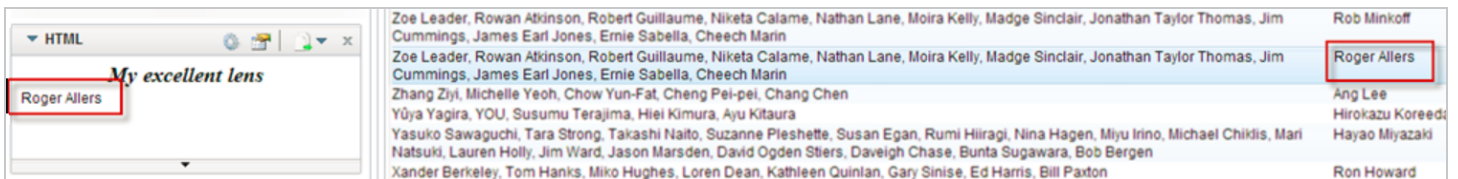
- **CSS:** This tab enables you to create a cascading style sheet to format the HTML and define the look and feel of the web page. Cambridge Semantics recommends that you define all CSS classes as namespaces to avoid global format changes.
- **Javascript:** This tab enables you to write JavaScript code to implement functions such as if statements, animations, or event notifications.

## Data Binding Example

When data is bound to a web page lens using HTML code, the web page lens behaves as follows:

- The lens will reflect data changes in real time.
- If the lens is oriented to the left-hand column (using the Orientation drop-down), selecting data in an active lens prompts the web page lens to display the related data.

In the example below, the active table lens row is selected, prompting the web page lens on the left (“My excellent lens”) to display the corresponding data.



## Creating Dashboard Filters

Filters narrow the data presented in a dashboard. You can define filter criteria using Microsoft Excel-like functions such as AVG, SUM, or UPPER, or groupings such as a date range or aggregation. When you add a filter to a dashboard, all lenses on the dashboard update simultaneously based on your filter selection. Though you can also filter data in some lens objects, such as a column, a filter applies across the entire dashboard. Unlike lenses, filters cannot be shared by other users or dashboards and must be created for each dashboard.

The topics in this section provide instructions for creating and configuring each type of filter.

- [Creating a Cloud Filter](#)
- [Creating a Date Range Filter](#)
- [Creating a Hierarchy Filter](#)
- [Creating a Limit Filter](#)
- [Creating a List Filter](#)
- [Creating a Numeric Range Filter](#)
- [Creating a Presence Filter](#)
- [Creating a Quartile Filter](#)
- [Creating a Range Slider Filter](#)
- [Creating a Relative Time Filter](#)
- [Creating a Search Filter](#)
- [Creating a Single Select List Filter](#)
- [Creating a Types Filter](#)

### Related Topics

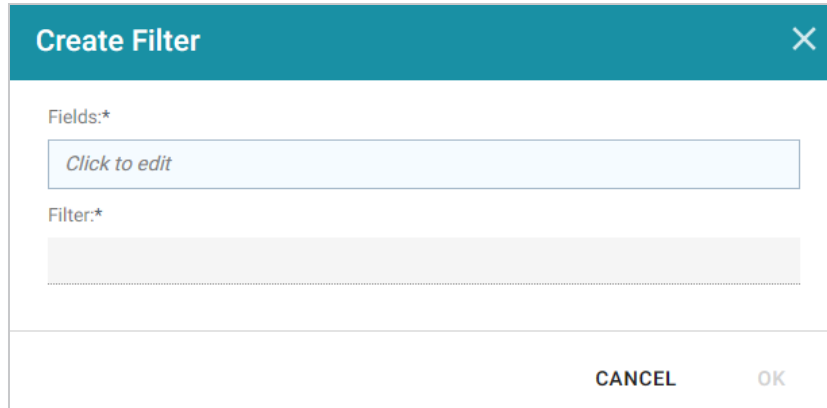
- [Creating Lenses](#)
- [Anzo Function and Formula Reference](#)

### Creating a Cloud Filter

Cloud filters display values in term clouds where each term is written in a font size that represents the number of results for that value. Unlike list filters, which enable you to select and filter on multiple values at once, cloud filters allow you to filter on one value at a time. The Cloud filter is available for all data types.

Follow the instructions below to create a Cloud filter.

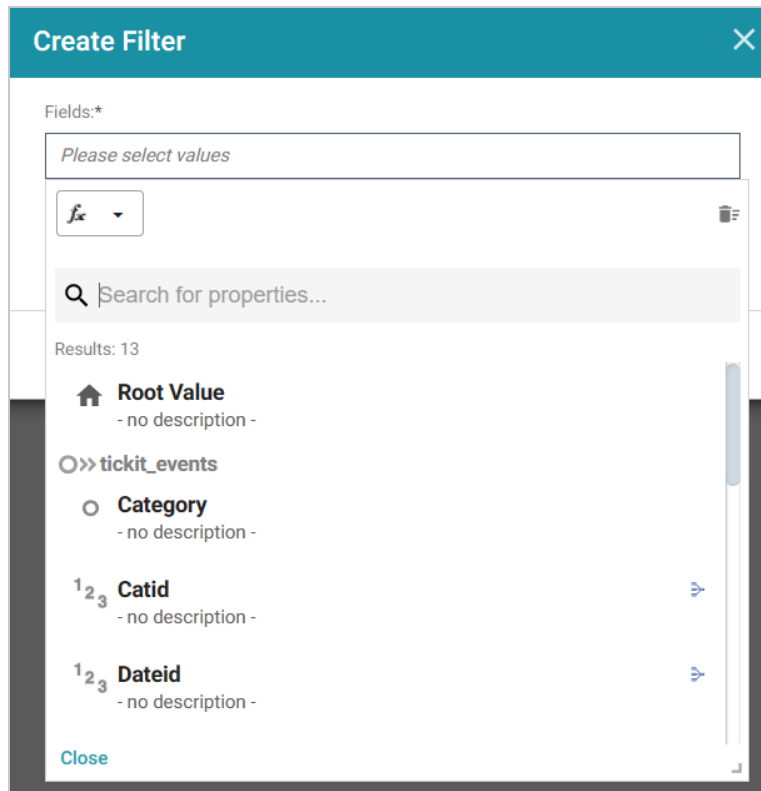
1. Open the dashboard that you want to add the filter to.
2. In the Hi-Res Analytics main toolbar, click **Filters** and select **Create a Filter**. The Create Filter dialog box is displayed.



The 'Create Filter' dialog box has a teal header with the title 'Create Filter' and a close button (X). Below the header, there are two sections: 'Fields:\*' and 'Filter:\*'. The 'Fields:\*' section contains a light blue button labeled 'Click to edit'. The 'Filter:\*' section contains a light gray rectangular input field. At the bottom right of the dialog, there are two buttons: 'CANCEL' and 'OK'.

3. Click in the **Fields** field to open the Property drop-down list and determine the property to filter on. The values for this property will be the terms that are displayed in the cloud. The list of available properties depends on the selected data type for the dashboard.

For example, the following image shows the list of properties that are available for a dashboard whose source is a graphmart that contains data about tickets sold for various types of events. The type for the dashboard is tickit\_events:



This image shows the 'Create Filter' dialog box with the 'Fields:\*' section expanded. The 'Fields:\*' section now displays a list of properties. At the top of this list is a search bar with the placeholder text 'Search for properties...'. Below the search bar, the text 'Results: 13' is shown. The list of properties includes: 'Root Value' (with a house icon and '- no description -'), 'tickit\_events' (with a double arrow icon), 'Category' (with a circle icon and '- no description -'), 'Catid' (with a '1 2 3' icon and '- no description -'), and 'Dateid' (with a '1 2 3' icon and '- no description -'). A 'Close' button is located at the bottom left of the property list. The 'Filter:\*' section remains empty.

The list below describes the icons and options that are available when choosing a property:

- The **Root Value** (📍) is the instance URI for the root resource—the URI for the instances of the class that was chosen as the data type for the dashboard.

**Tip**

To view the Root Value values, you can use the STR function to show a string representation of the URIs.

- Linked classes are represented by incoming (◀) and outgoing (▶) connection icons. The properties in those classes with a path to another class are denoted with a circle icon (○). Selecting a linked property navigates to that class and displays its properties.
  - When a property or path is selected, the breadcrumbs at the top of the dialog box show you the property path. You can click the Clear icon (🗑️) to clear the path and start again.
  - After you have selected a property, you can apply a function or formula to that property to calculate the values that are displayed in the filter. To add a function, click the function button (fx) at the top of the drop-down list. The functions that become available depend on the data type of the selected property. To choose a more advanced function or type a formula, click **Advanced**. The Calculated Value dialog box opens and enables you to choose additional properties and functions. For more information, see [Calculating Values in Lenses and Filters](#).
4. After you have selected the property to filter on, click **Close** to close the Fields drop-down list.
  5. Next, click the **Filter** field and select **Cloud** from the drop-down list. The dialog box is refreshed to show the Filter Properties and other options that are available for the filter type:

**Create Filter** [X]

Fields:\*  
venue/venuecity

Filter:\*  
Cloud

**Filter Properties**

Title:  
[Empty Input]

Label field:  
Click to edit

Exclude: ☐

Show counts: ☒ Respond to other filters: ☒

**Subfilters**  
+ Create Filter

CANCEL OK

6. Configure any of the following properties. All of the fields are optional:

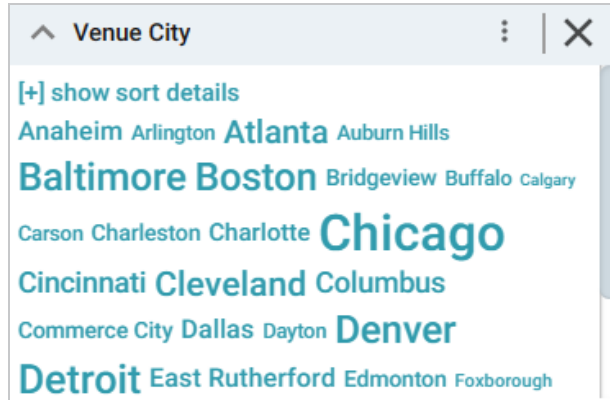
- **Title:** Defines the title that appears at the top of the filter when it is added to the dashboard. If Title is blank, the value from the Fields field is used as the title.
- **Label Field:** If you want to populate the cloud with values from a property other than the one specified in Fields, you can select an alternate property in this field.
- **Exclude:** This setting controls whether selecting a term in the filter narrows the results to show only the records that *include* that term or whether selecting a term *excludes* the records that include that term. When Exclude is disabled, selecting a term narrows the dashboard results to show only the records that include that term. When Exclude is enabled, selecting a term filters out all of the records that include that term.
- **Show Counts:** This setting controls whether the number of results for each term are displayed when you hover the pointer over a term.
- **Respond to Other Filters:** This setting controls whether the results of this filter change based on selections in other filters on the dashboard.

7. If you would like to be able to further constrain the data that appears in the filter, you can add one or more subfilters. To add a subfilter, click **Create Filter** under Subfilters. The process of creating a subfilter is the same as the process for the parent filter. However, the subfilter is not displayed on the dashboard. It is visible only when editing the parent filter, and the subfilter's configuration affects only the parent filter and any sibling subfilters.



8. When you have finished configuring the filter, click **OK** to add it to the dashboard. The new filter appears in the left pane of the dashboard and displays the values that are available for filtering the displayed data.

For example, the filter in the image below shows cities with event venues. The size of the terms represent the number of events that were held in venues in that city.



Depending on whether the Exclude option is enabled or disabled, clicking a term in the Cloud refreshes the dashboard to show only the data that either contains or excludes the selected term.

When working with the filter on the dashboard, the following options are available for sorting and configuration:

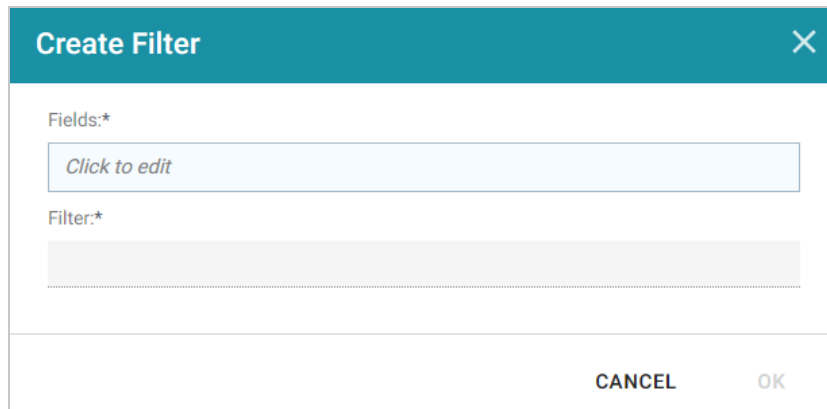
- **show/hide sort details:** Shows or hides the following options for sorting the results in the filter:
  - **Sort by:** Select **Count** to order results according to the total number of results for each value, or select **Value** to sort string values alphabetically.
  - **Direction:** Select **Ascending** to order results in alphabetical order. Or select **Descending** to order results in reverse order.
- **show/hide filters:** This option is displayed when a term is selected in the Cloud. It shows or hides the selection.
- **Menu (⋮):** The menu contains the following options:
  - **Select All Visible:** This option does not work for Cloud filters.
  - **Clear:** This option becomes available when a term is selected. Clicking **Clear** removes the selection.
  - **Designer:** Selecting this option opens the filter Designer so that you can view or change the filter configuration.
- **Close (X):** Clicking **X** removes the filter from the dashboard. This action cannot be undone.

## Creating a Date Range Filter

Date Range filters are used to limit the results on a dashboard to data that falls in (or outside of) certain date and time groupings. Date Range filters are available for properties with date, dateTime, and time data types.

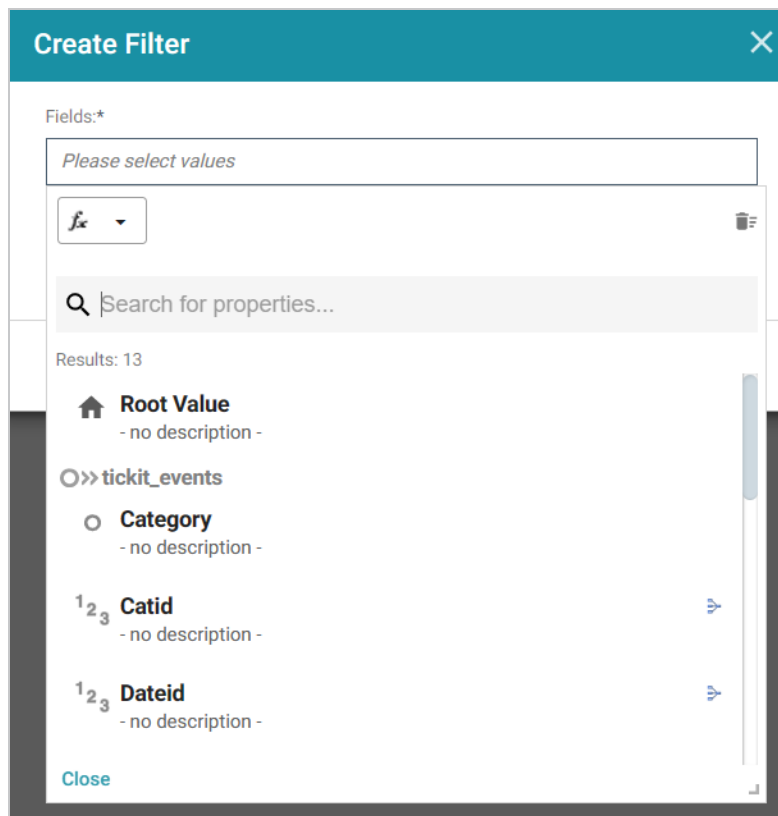
Follow the instructions below to create a Date Range filter.

1. Open the dashboard that you want to add the filter to.
2. In the Hi-Res Analytics main toolbar, click **Filters** and select **Create a Filter**. The Create Filter dialog box is displayed.



3. Click in the **Fields** field to open the Property drop-down list and determine the date, dateTime, or time type property to filter on. The values for this property will be used to determine the date ranges for the filter. The list of available properties depends on the selected data type for the dashboard.

For example, the following image shows the list of properties that are available for a dashboard whose source is a graphmart that contains data about tickets sold for various types of events. The type for the dashboard is tickit\_events:



The list below describes the icons and options that are available when choosing a property:

- The **Root Value** (🏠) is the instance URI for the root resource—the URI for the instances of the class that was chosen as the data type for the dashboard.

#### Tip

To view the Root Value values, you can use the STR function to show a string representation of the URIs.

- Linked classes are represented by incoming (◁) and outgoing (▷) connection icons. The properties in those classes with a path to another class are denoted with a circle icon (○). Selecting a linked property navigates to that class and displays its properties. For this example filter, selecting **EventDate**, navigates to the tickit\_dates class where a date type property can be selected.
- When a property or path is selected, the breadcrumbs at the top of the dialog box show you the property path. You can click the Clear icon (🗑️) to clear the path and start again.
- After you have selected a property, you can apply a function or formula to that property to calculate the values that are displayed in the filter. To add a function, click the function button (fx) at the top of the drop-down list. The functions that become available depend on the data type

of the selected property. To choose a more advanced function or type a formula, click **Advanced**. The Calculated Value dialog box opens and enables you to choose additional properties and functions. For more information, see [Calculating Values in Lenses and Filters](#).

- After you have selected the property to filter on, click **Close** to close the Fields drop-down list.
- Next, click the **Filter** field and select **Date Range** from the drop-down list. The dialog box is refreshed to show the Filter Properties, Format, and other options that are available for the filter type:

**Create Filter** ✕

Fields:\*  
eventDate/caldate

Filter:\*  
Date Range

**Filter Properties**

Title:

Interval Unit:\*  
MILLENNIUM

Interval:\*

Exclude: ☐ Show Bars: ☐ Always Show Checkboxes: ☒  
Show counts: ☒ Respond to other filters: ☒

**Format**

Format Type:  
Automatic

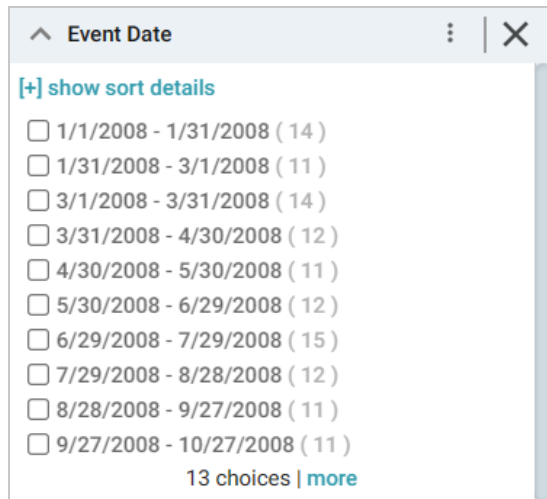
**Subfilters**  
[+ Create Filter](#)

CANCEL OK

- Configure any of the following properties. **Interval Unit** and **Interval** are required fields, and the rest of the fields are optional:
  - Title:** Defines the title that appears at the top of the filter when it is added to the dashboard. If Title is blank, the value from the Fields field is used as the title.
  - Label Field:** If you want to populate the ranges with a label other than the one specified in Fields, you can select an alternate property in this field.

- **Interval Unit:** Click this field to choose the unit of time for the **Interval** value. Depending on the data type of the selected property, a subset of the following values are available to choose from: Millennium, Century, Decade, Year, Month, Week, Day, Hour, Minute, or Second.
  - **Interval:** This setting specifies a number that defines the length of time in each grouping. For example, if the Interval Unit is "Decade," an Interval value of 2 creates groups of two-decade increments.
  - **Exclude:** This setting controls whether selecting a range in the filter narrows the results to show only the records that are *included* in that range or whether selecting a range *excludes* the records that fall in that range. When Exclude is disabled, selecting a range narrows the dashboard results to show only the records that fall in that range. When Exclude is enabled, selecting a range filters out all of the records that fall in that range.
  - **Show Bars:** This setting controls whether the total values for the selected property appear as a bar graphic in the background of the filter.
  - **Always Show Checkboxes:** This setting controls whether checkboxes are shown next to the items in the filter.
  - **Show Counts:** This setting controls whether the number of results for each range are displayed in parentheses next to the range.
  - **Respond to Other Filters:** This setting controls whether the results of this filter change based on selections in other filters on the dashboard.
7. If you would like to specify the format to for date values that are displayed in the filter, click the **Format Time** field and select a format from the drop-down list.
  8. If you would like to be able to further constrain the data that appears in the filter, you can add one or more subfilters. To add a subfilter, click **Create Filter** under Subfilters. The process of creating a subfilter is the same as the process for the parent filter. However, the subfilter is not displayed on the dashboard. It is visible only when editing the parent filter, and the subfilter's configuration affects only the parent filter and any sibling subfilters.
  9. When you have finished configuring the filter, click **OK** to add it to the Dashboard. The new filter appears in the left pane of the Dashboard and displays the values that are available for filtering the displayed data.

For example, the filter in the image below shows date ranges for events. The Interval Unit is Month, and the Interval is 3 months.



Depending on whether the Exclude option is enabled or disabled, clicking a range in the filter refreshes the dashboard to show only the data that is in the selected range or only the data that is outside of the range.

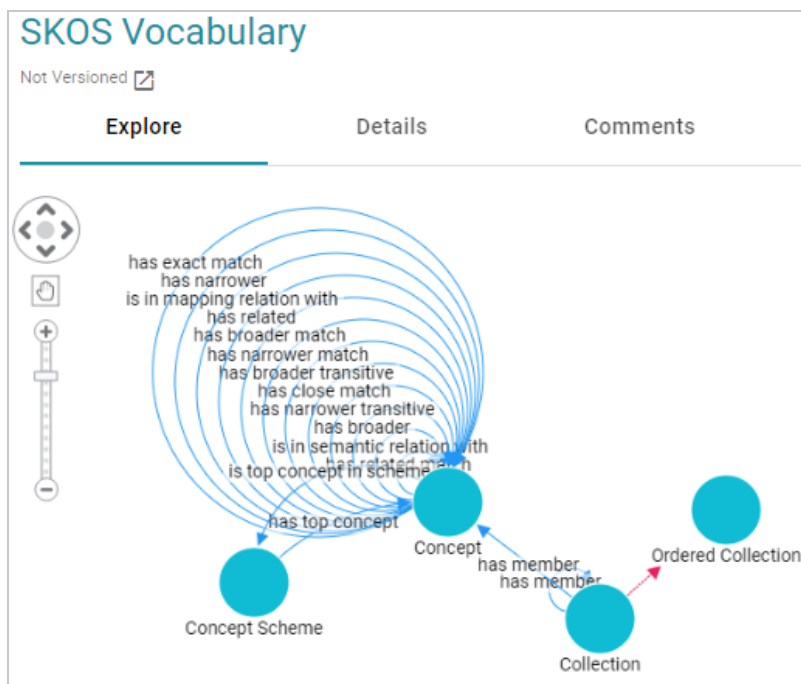
When working with the filter on the dashboard, the following options are available for sorting and configuration:

- **show/hide sort details:** Shows or hides the following option for sorting the results in the filter:
  - **Direction:** This option controls how you want to order the date ranges in the filter, depending on the Format specified for the values in the filter. For number values, **Count Ascending** orders results from the earliest to latest date and **Count Descending** orders results from the latest to earliest date. For character values, **Name Ascending** orders results in alphabetical order and **Name Descending** orders results in reverse alphabetical order.
- **show/hide filters:** This option is displayed when a range is selected. It shows or hides the selection.
- **Menu (⋮):** The menu contains the following options:
  - **Select All Visible:** This option selects all of the ranges that are listed in the filter.
  - **Clear:** This option is available when one or more ranges are selected. Clicking **Clear** removes the selection..
  - **Designer:** Selecting this option opens the filter Designer so that you can view or change the filter configuration.
- **Close (X):** Clicking **X** removes the filter from the dashboard. This action cannot be undone.

## Creating a Hierarchy Filter

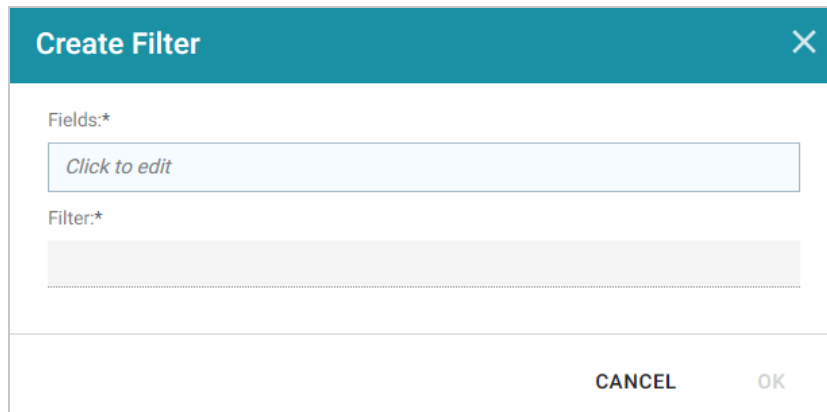
If hierarchies exist in your knowledge graph, you can create a Hierarchy filter to explore the parent and child relationships and filter the dashboard based on the relationships. Unlike the majority of dashboard filters, where you select a property to filter on, Hierarchy filters operate on relationships and are only available as a filter type when you select a path to filter on.

In order to produce hierarchies in the data, you typically need a self-referential data model, where properties have relationships to themselves. The SKOS ontology is a good example of a self-referential model. As shown in the image below, many of the properties in the Concept class refer to themselves. These paths define a hierarchy.



Follow the instructions below to create a Hierarchy Filter.

1. Open the dashboard that you want to add the filter to.
2. In the Hi-Res Analytics main toolbar, click **Filters** and select **Create a Filter**. The Create Filter dialog box is displayed.



**Create Filter** [X]

Fields:\*

*Click to edit*

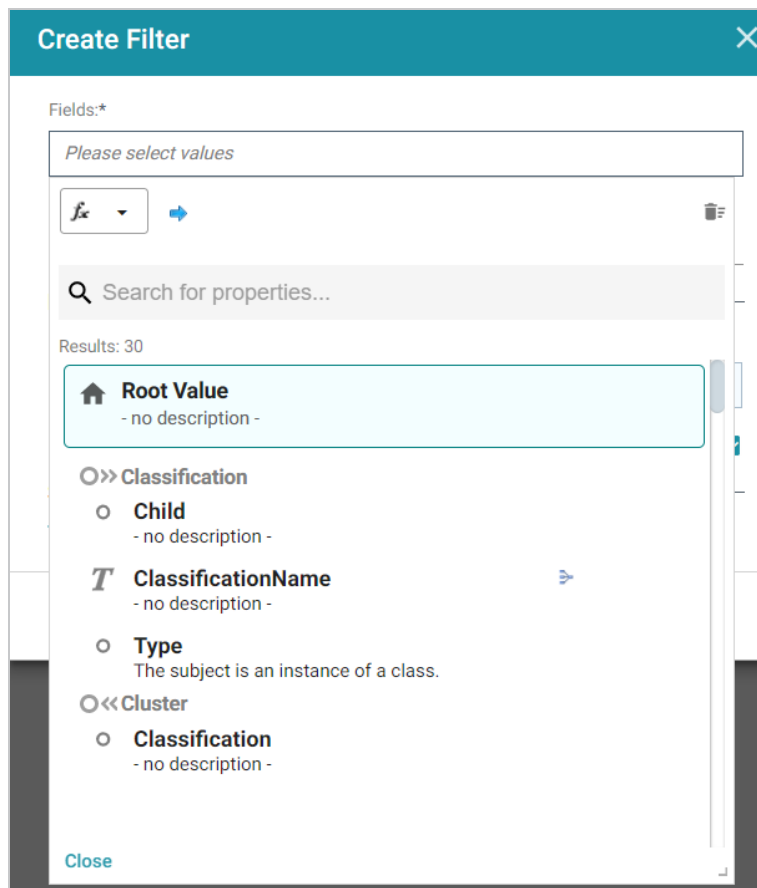
Filter:\*

[Empty text box]

CANCEL OK

3. Click in the **Fields** field to open the Property drop-down list and determine the path to filter on. The list of available classes, paths, and properties depends on the selected data type for the dashboard.

For example, the following image shows the list of properties that are available for a dashboard whose source is a graphmart that contains data about protein classification. The type for the dashboard is Classification:



**Create Filter** [X]

Fields:\*

*Please select values*

*f<sub>x</sub>* [Dropdown] [Add] [Remove]

Search for properties...

Results: 30

- Root Value**  
- no description -
- Classification**
  - Child**  
- no description -
  - ClassificationName**  
- no description -
  - Type**  
The subject is an instance of a class.
- Cluster**
  - Classification**  
- no description -

Close

The list below describes the icons and options that are available when choosing a path:



- The **Root Value** (📍) is the instance URI for the root resource—the URI for the instances of the class that was chosen as the data type for the dashboard. Typically **Root Value** is the chosen Field for Hierarchy filters.

#### Tip

To view the Root Value values, you can use the STR function to show a string representation of the URIs.

- Linked classes are represented by incoming (◀) and outgoing (▶) connection icons. The properties in those classes with a path to another class are denoted with a circle icon (○). Selecting a linked property navigates to that class and displays its properties.
  - When a property or path is selected, the breadcrumbs at the top of the dialog box show you the property path. You can click the Clear icon (🗑️) to clear the path and start again.
  - After you have selected a path, you can apply a function or formula to calculate the values that are displayed in the filter. To add a function, click the function button (fx) at the top of the drop-down list. The functions that become available depend on the data type of the selected path. To choose a more advanced function or type a formula, click **Advanced**. The Calculated Value dialog box opens and enables you to choose additional properties and functions. For more information, see [Calculating Values in Lenses and Filters](#).
4. After you have selected the path to filter on, click **Close** to close the Fields drop-down list.
  5. Next, click the **Filter** field and select **Hierarchy** from the drop-down list. The dialog box is refreshed to show the Filter Properties and other options that are available for the filter type:

Create Filter

×

Fields:\*

ROOT

Filter:\*

Hierarchy

Filter Properties

Title:

Show counts:

☒

Respond to other filters:

☒

Label Field:\*

Click to select a path

Children Field:\*

Click to select a path

Subfilters

+ Create Filter

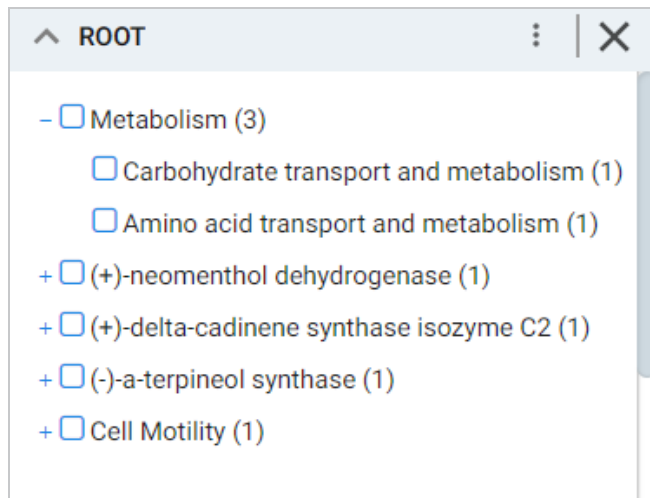
CANCEL

OK

6. Configure any of the following properties. **Label Field** and **Children Field** are required fields, and the rest of the fields are optional:
  - **Title:** Defines the title that appears at the top of the filter when it is added to the dashboard. If Title is blank, the value from the Fields field is used as the title.
  - **Show Counts:** This setting controls whether the number of results for each filter value are displayed in parentheses next to the value.
  - **Respond to Other Filters:** This setting controls whether the results of this filter change based on selections in other filters on the dashboard.
  - **Label Field:** Species the property that supplies the label for the parent values in the hierarchy.
  - **Children Field:** Specifies the child value in the relationship.
7. If you would like to be able to further constrain the data that appears in the filter, you can add one or more subfilters. To add a subfilter, click **Create Filter** under Subfilters. The process of creating a subfilter is the same as the process for the parent filter. However, the subfilter is not displayed on the dashboard. It is visible only when editing the parent filter, and the subfilter's configuration affects only the parent filter and any sibling subfilters.

- When you have finished configuring the filter, click **OK** to add it to the dashboard. The new filter appears in the left pane of the dashboard and displays the values that are available for filtering the displayed data.

For example, the filter in the image below shows a hierarchy of classifications.



Selecting checkboxes in the filter refreshes the dashboard to show only the data that includes the selected values.

When working with the filter on the dashboard, the following options are available:

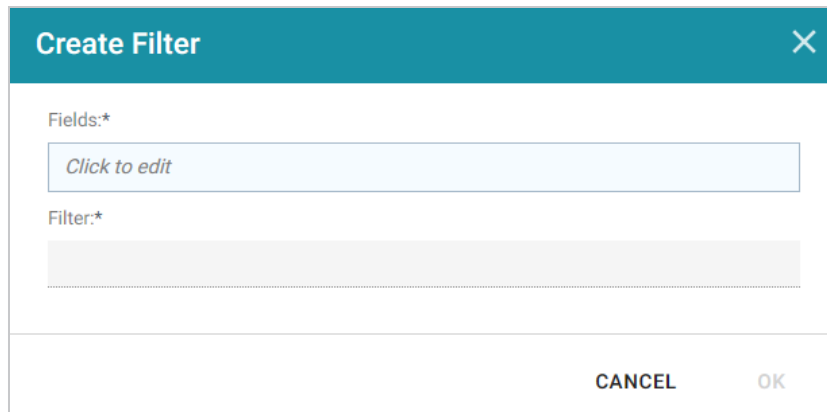
- **Menu** (⋮): The menu contains the following options:
  - **Clear**: This option becomes available when an item is selected. Clicking **Clear** removes the selection.
  - **Designer**: Selecting this option opens the filter Designer so that you can view or change the filter configuration.
- **Close** (X): Clicking **X** removes the filter from the dashboard. This action cannot be undone.

## Creating a Limit Filter

Limit filters are used to limit the results on the dashboard to a specified number of either the largest or smallest values. The Limit filter is available for any data type. For strings, results are ordered alphabetically. "Largest" orders by the last letters in the alphabet and "Smallest" orders by the first letters in the alphabet.

Follow the instructions below to create a Limit Filter.

1. Open the dashboard that you want to add the filter to.
2. In the Hi-Res Analytics main toolbar, click **Filters** and select **Create a Filter**. The Create Filter dialog box is displayed.



**Create Filter** [X]

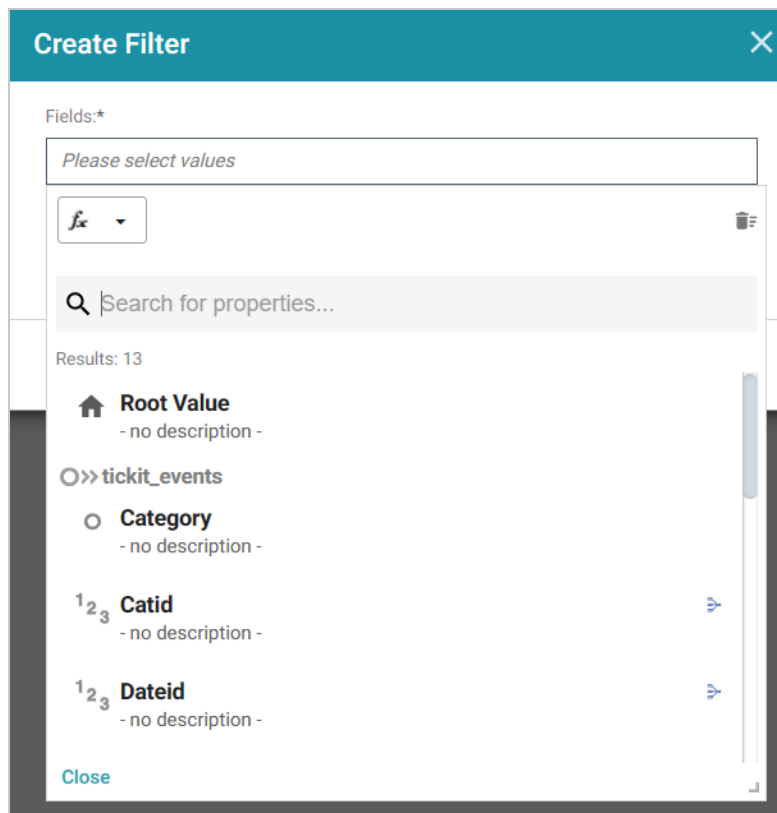
Fields:\*

Filter:\*

CANCEL OK

3. Click in the **Fields** field to open the Property drop-down list and determine the property to filter on. The list of available properties depends on the selected data type for the dashboard.

For example, the following image shows the list of properties that are available for a dashboard whose source is a graphmart that contains data about tickets sold for various types of events. The type for the dashboard is tickit\_events:



**Create Filter** [X]

Fields:\*

[X] [≡]

Results: 13

- ☒ **Root Value**  
- no description -
- ☐ **tickit\_events**
  - ☐ **Category**  
- no description -
  - ☐ **Catid**  
- no description -
  - ☐ **Dateid**  
- no description -

Close

The list below describes the icons and options that are available when choosing a property:

- The **Root Value** (🏠) is the instance URI for the root resource—the URI for the instances of the class that was chosen as the data type for the dashboard.

### Tip

To view the Root Value values, you can use the STR function to show a string representation of the URIs.

- Linked classes are represented by incoming (◀) and outgoing (▶) connection icons. The properties in those classes with a path to another class are denoted with a circle icon (○). Selecting a linked property navigates to that class and displays its properties. For this example filter, selecting the **Venue** path navigates to the tickit\_venues class where an integer type property, Venueseats, is selected.
  - When a property or path is selected, the breadcrumbs at the top of the dialog box show you the property path. You can click the Clear icon (🗑️) to clear the path and start again.
  - After you have selected a property, you can apply a function or formula to that property to calculate the values that are displayed in the filter. To add a function, click the function button (fx) at the top of the drop-down list. The functions that become available depend on the data type of the selected property. To choose a more advanced function or type a formula, click **Advanced**. The Calculated Value dialog box opens and enables you to choose additional properties and functions. For more information, see [Calculating Values in Lenses and Filters](#).
4. After you have selected the property to filter on, click **Close** to close the Fields drop-down list.
  5. Next, click the **Filter** field and select **Limit** from the drop-down list. The dialog box is refreshed to show the Filter Properties and other options that are available for the filter type:

**Create Filter** [X]

Fields:\*  
venue/venueseats

Filter:\*  
Limit

**Filter Properties**

Title:  
[ ]

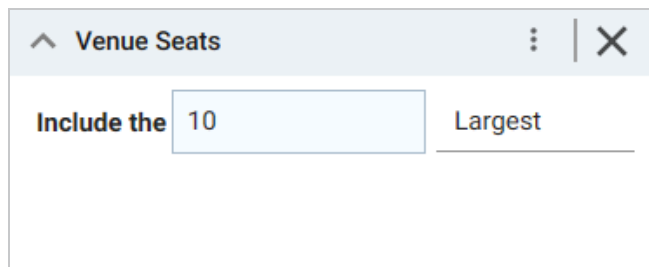
Limit by resource: ☒

**Subfilters**  
+ Create Filter

CANCEL OK

6. Configure any of the following properties. All of the fields are optional:
  - **Title**: Defines the title that appears at the top of the filter when it is added to the dashboard. If Title is blank, the value from the Fields field is used as the title.
  - **Limit by Resource**: This setting controls whether the limit is also applied to the resource that is the data type of the dashboard. When **Limit by Resource** is enabled, the specified limit applies to the resource as well as the property. When **Limit by Resource** is disabled, the limit applies only to the specified property. Using the example above for the property of Venueseats and the resource (data type) tickit\_events, when Limit by Resource is enabled, filtering for the 5 largest values of Venueseats returns the 5 events with the largest venues. When Limit by Resource is disabled, filtering for the 5 largest venues returns all of the events that were held in one of the 5 largest venues.
7. If you would like to be able to further constrain the data that appears in the filter, you can add one or more subfilters. To add a subfilter, click **Create Filter** under Subfilters. The process of creating a subfilter is the same as the process for the parent filter. However, the subfilter is not displayed on the dashboard. It is visible only when editing the parent filter, and the subfilter's configuration affects only the parent filter and any sibling subfilters.
8. When you have finished configuring the filter, click **OK** to add it to the dashboard. The new filter appears in the left pane of the dashboard and displays the options for configuring the limit.
9. To configure the limit for filtering data, specify a number in the **Include the** field. And click the drop-down list on the right to choose **Largest** or **Smallest**.

For example, the filter in the image below excludes all but the 10 venues with the most number of seats.



The image shows a filter configuration window titled "Venue Seats". At the top, there is a header bar with a chevron icon, the title "Venue Seats", and a close button (X). Below the header, the main configuration area contains the text "Include the" followed by a text input field containing the number "10". To the right of the input field is a dropdown menu currently displaying the word "Largest".

Changing the values in the filter refreshes the dashboard according to the new limit.

When working with the filter on the dashboard, the following options are available:

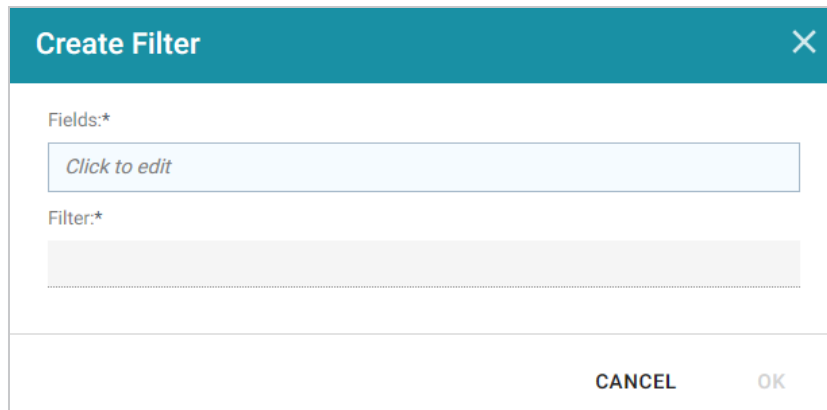
- **Menu (⋮):** The menu contains the following options:
  - **Clear:** Clicking **Clear** removes the value in the **Include the** field.
  - **Designer:** Selecting this option opens the filter Designer so that you can view or change the filter configuration.
- **Close (X):** Clicking **X** removes the filter from the dashboard. This action cannot be undone.

## Creating a List Filter

List filters display values in a list and allow you to filter on multiple values at the same time. List filters are available for all data types.

Follow the instructions below to create a List filter.

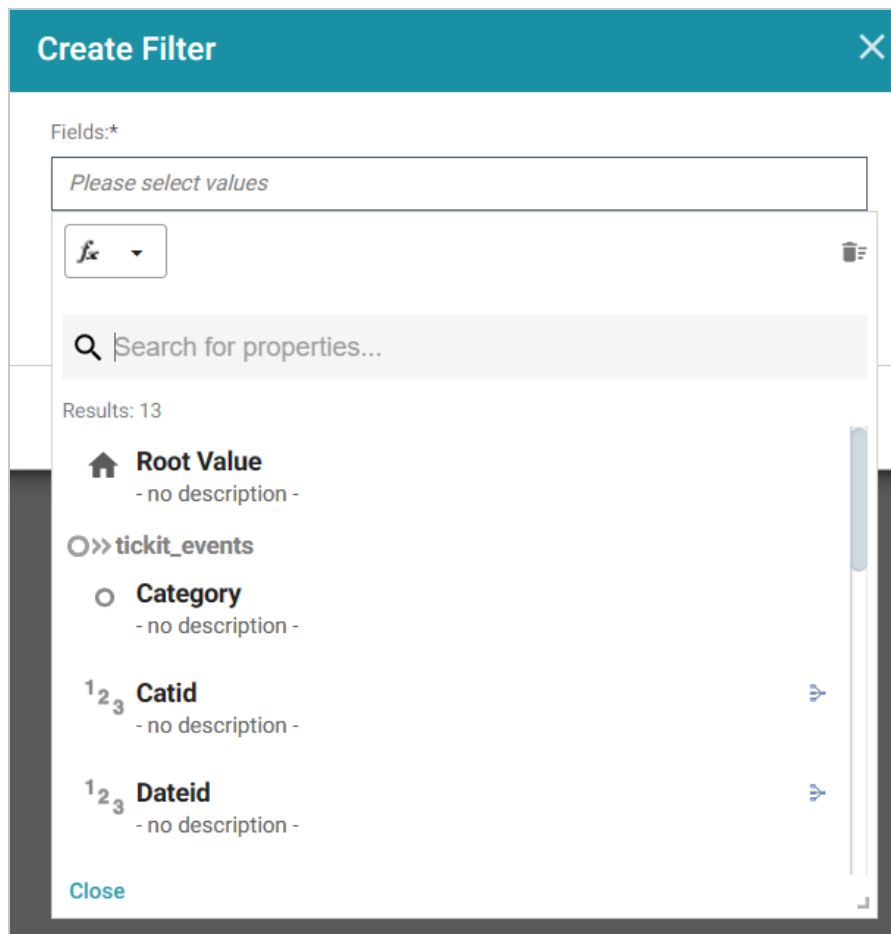
1. Open the dashboard that you want to add the filter to.
2. In the Hi-Res Analytics main toolbar, click **Filters** and select **Create a Filter**. The Create Filter dialog box is displayed.



The screenshot shows a 'Create Filter' dialog box. It has a teal header bar with the text 'Create Filter' and a close button (X) on the right. Below the header, there are two sections: 'Fields:\*' with a light blue button labeled 'Click to edit', and 'Filter:\*' with a light gray input field. At the bottom right, there are 'CANCEL' and 'OK' buttons.

3. Click in the **Fields** field to open the Property drop-down list and determine the property to filter on. The list of available properties depends on the selected data type for the dashboard.

For example, the following image shows the list of properties that are available for a dashboard whose source is a graphmart that contains data about tickets sold for various types of events. The type for the dashboard is tickit\_events:



The list below describes the icons and options that are available when choosing a property:

- The **Root Value** (🏠) is the instance URI for the root resource—the URI for the instances of the class that was chosen as the data type for the dashboard.

#### Tip

To view the Root Value values, you can use the STR function to show a string representation of the URIs.

- Linked classes are represented by incoming (◀) and outgoing (▶) connection icons. The properties in those classes with a path to another class are denoted with a circle icon (○). Selecting a linked property navigates to that class and displays its properties.
- When a property or path is selected, the breadcrumbs at the top of the dialog box show you the property path. You can click the Clear icon (🗑️) to clear the path and start again.
- After you have selected a property, you can apply a function or formula to that property to calculate the values that are displayed in the filter. To add a function, click the function button (fx) at the top of the drop-down list. The functions that become available depend on the data type



of the selected property. To choose a more advanced function or type a formula, click **Advanced**. The Calculated Value dialog box opens and enables you to choose additional properties and functions. For more information, see [Calculating Values in Lenses and Filters](#).

4. After you have selected the property to filter on, click **Close** to close the Fields drop-down list.
5. Next, click the **Filter** field and select **List** from the drop-down list. The dialog box is refreshed to show the Filter Properties and other options that are available for the filter type:

**Create Filter** [X]

Fields:\*  
category/catname

Filter:\*  
List

**Filter Properties**

Title:  
[ ]

Label field:  
Click to edit

Exclude: ☐ Show Bars: ☐ Show Blanks: ☐ Always Show Checkboxes: ☒

Show counts: ☒ Respond to other filters: ☒

**Subfilters**  
[+ Create Filter](#)

CANCEL OK

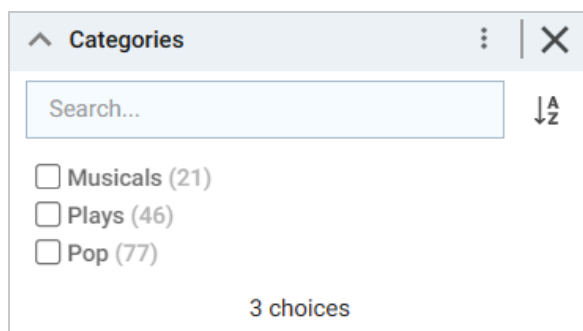
6. Configure any of the following properties. All of the fields are optional:
  - **Title:** Defines the title that appears at the top of the filter when it is added to the dashboard. If Title is blank, the value from the Fields field is used as the title.
  - **Label Field:** If you want to populate the list with values from a property other than the one specified in Fields, you can select an alternate property in this field.
  - **Exclude:** This setting controls whether selecting a value in the List Filter narrows the results to show only the records that *include* that value or whether selecting a value *excludes* the records that include that value. When Exclude is disabled, selecting a value in the list narrows the

Dashboard results to show only the records that include that value. When Exclude is enabled, selecting a value filters out all of the records that include that value.

- **Show Bars:** This setting controls whether the total values for the selected property appear as a bar graphic in the background of the filter.
- **Show Blanks:** This setting controls whether to include null values for the selected property by listing them as **Blank** in the filter.
- **Always Show Checkboxes:** This setting controls whether checkboxes are shown next to the items in the filter.
- **Show Counts:** This setting controls whether the number of results for each item in the filter is displayed in parentheses next to the item.
- **Respond to Other Filters:** This setting controls whether the results of this filter change based on selections in other filters on the dashboard.


7. If you would like to be able to further constrain the data that appears in the filter, you can add one or more subfilters. To add a subfilter, click **Create Filter** under Subfilters. The process of creating a subfilter is the same as the process for the parent filter. However, the subfilter is not displayed on the dashboard. It is visible only when editing the parent filter, and the subfilter's configuration affects only the parent filter and any sibling subfilters.
8. When you have finished configuring the filter, click **OK** to add it to the dashboard. The new filter appears in the left pane of the dashboard and displays the values that are available for filtering the displayed data.

For example, the filter in the image below shows a filter that lists the categories of events. One or more of the categories can be selected to filter the data on the dashboard.



Depending on whether the Exclude option is enabled or disabled, selecting an item in the filter refreshes the dashboard to show only the data includes the selected value or only the data that does not include the selected value.

When working with the filter on the dashboard, the following options are available for sorting and configuration:

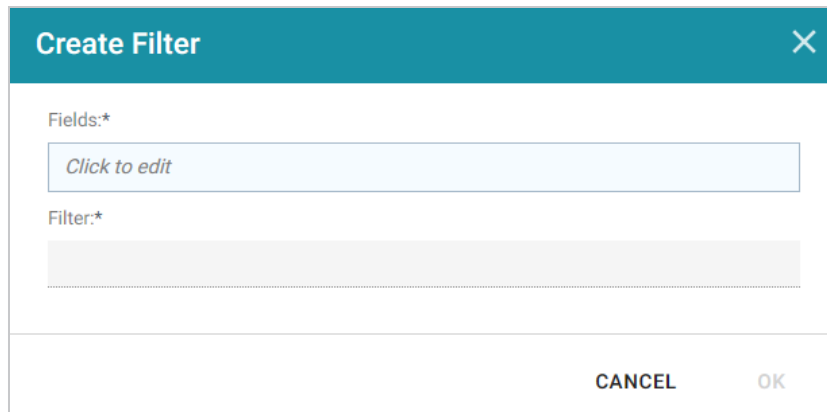
- **Search:** Enables you to search for a value in the list. The search is case-insensitive.
- **Sort** (

## Creating a Numeric Range Filter

Numeric Range filters are used to limit the results on a dashboard to data that falls in (or outside of) certain numeric groupings. Numeric Range filters are available for properties with integer and double data types.

Follow the instructions below to create a Numeric Range filter.

1. Open the dashboard that you want to add the filter to.
2. In the Hi-Res Analytics main toolbar, click **Filters** and select **Create a Filter**. The Create Filter dialog box is displayed.



**Create Filter** [X]

Fields:\*

*Click to edit*

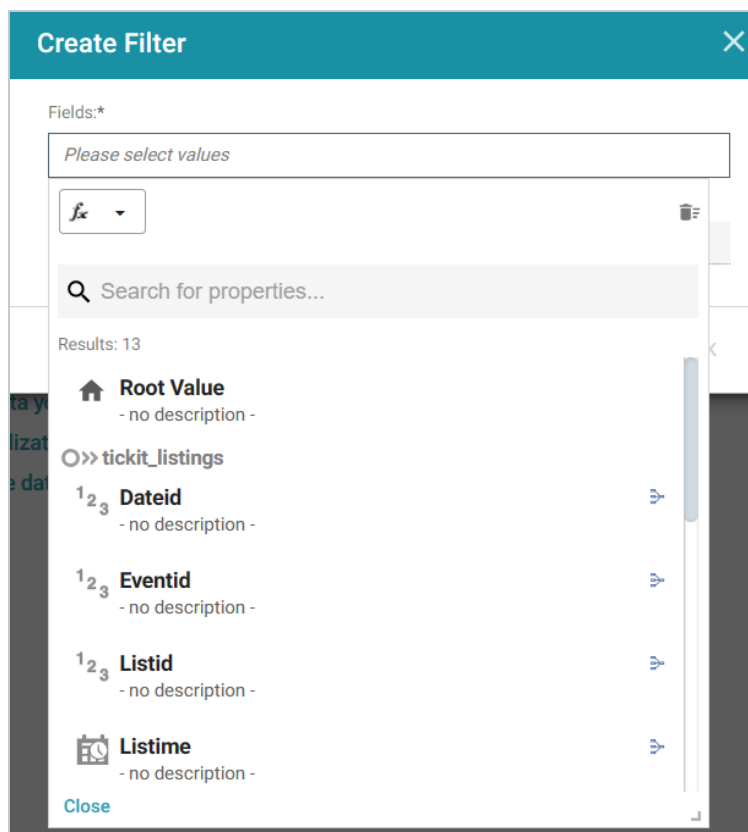
Filter:\*

[Empty text box]

CANCEL OK

- Click in the **Fields** field to open the Property drop-down list and determine the integer or double type property to filter on. The values for this property will be used to determine the numeric ranges for the filter. The list of available properties depends on the selected data type for the dashboard.

For example, the following image shows the list of properties that are available for a dashboard whose source is a graphmart that contains data about tickets sold for various types of events. The type for the dashboard is tickit\_listings:



**Create Filter** [X]

Fields:\*

*Please select values*

*fx* [Dropdown icon] [Trash icon]

Search for properties...

Results: 13

- Root Value**  
- no description -
- tickit\_listings**
- 1 2 3 Dateid**  
- no description - [Psi icon]
- 1 2 3 Eventid**  
- no description - [Psi icon]
- 1 2 3 Listid**  
- no description - [Psi icon]
- Listime**  
- no description - [Psi icon]

Close

The list below describes the icons and options that are available when choosing a property:

- The **Root Value** (📍) is the instance URI for the root resource—the URI for the instances of the class that was chosen as the data type for the dashboard.

#### Tip

To view the Root Value values, you can use the STR function to show a string representation of the URIs.

- Linked classes are represented by incoming (◁) and outgoing (▷) connection icons. The properties in those classes with a path to another class are denoted with a circle icon (○). Selecting a linked property navigates to that class and displays its properties.
  - When a property or path is selected, the breadcrumbs at the top of the dialog box show you the property path. You can click the Clear icon (🗑️) to clear the path and start again.
  - After you have selected a property, you can apply a function or formula to that property to calculate the values that are displayed in the filter. To add a function, click the function button (fx) at the top of the drop-down list. The functions that become available depend on the data type of the selected property. To choose a more advanced function or type a formula, click **Advanced**. The Calculated Value dialog box opens and enables you to choose additional properties and functions. For more information, see [Calculating Values in Lenses and Filters](#).
4. After you have selected the property to filter on, click **Close** to close the Fields drop-down list.
  5. Next, click the **Filter** field and select **Numeric Range** from the drop-down list. The dialog box is refreshed to show the Filter Properties, Format, and other options that are available for the filter type:

Create Filter

×

Fields:\*

totalprice

Filter:\*

Numeric Range

Filter Properties

Title:

Interval:\*

Exclude: ☐

Show Bars: ☐

Always Show Checkboxes: ☒

Show counts: ☒

Respond to other filters: ☒

Format

Type:

Automatic

Subfilters

+ Create Filter

CANCEL

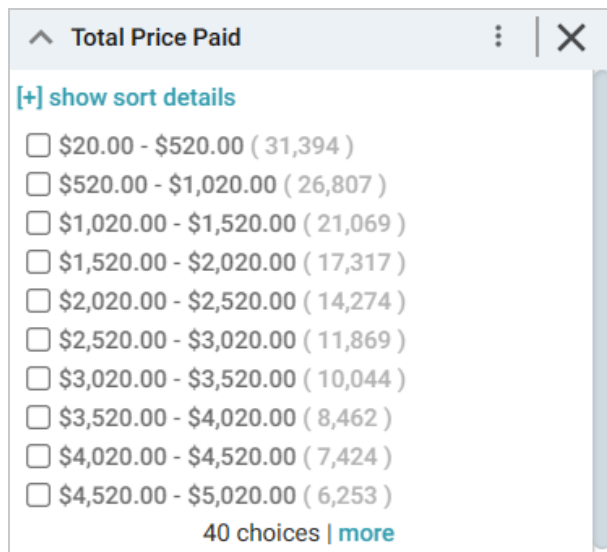
OK

6. Configure any of the following properties. **Interval** is a required field, and the rest of the fields are optional:
- **Title:** Defines the title that appears at the top of the filter when it is added to the dashboard. If Title is blank, the value from the Fields field is used as the title.
  - **Interval:** This setting specifies a number that defines the size of the groupings.
  - **Exclude:** This setting controls whether selecting a range in the filter narrows the results to show only the records that are *included* in that range or whether selecting a range *excludes* the records that fall in that range. When Exclude is disabled, selecting a range narrows the dashboard results to show only the records that fall in that range. When Exclude is enabled, selecting a range filters out all of the records that fall in that range.
  - **Show Bars:** This setting controls whether the total values for the selected property appear as a bar graphic in the background of the filter.

- **Always Show Checkboxes:** This setting controls whether checkboxes are shown next to the items in the filter.
- **Show Counts:** This setting controls whether the number of results for each range are displayed in parentheses next to the range.
- **Respond to Other Filters:** This setting controls whether the results of this filter change based on selections in other filters on the dashboard.

7. If you would like to specify the format for the numeric values that are displayed in the filter, click the **Type** field and select a format from the drop-down list.
8. If you would like to be able to further constrain the data that appears in the filter, you can add one or more subfilters. To add a subfilter, click **Create Filter** under Subfilters. The process of creating a subfilter is the same as the process for the parent filter. However, the subfilter is not displayed on the dashboard. It is visible only when editing the parent filter, and the subfilter's configuration affects only the parent filter and any sibling subfilters.
9. When you have finished configuring the filter, click **OK** to add it to the dashboard. The new filter appears in the left pane of the dashboard and displays the values that are available for filtering the displayed data.

For example, the filter in the image below shows numeric ranges for the total price paid for tickets to events. The Interval is 500, and format is Money, resulting in ranges for each group of \$500.



Depending on whether the Exclude option is enabled or disabled, clicking a range in the filter refreshes the dashboard to show only the data that is in the selected range or only the data that is outside of the range.

When working with the filter on the dashboard, the following options are available for sorting and configuration:

- **show/hide sort details:** Shows or hides the following option for sorting the results in the filter:
  - **Direction:** This option controls how you want to order the ranges in the filter, depending on the Format specified for the values in the filter. **Count Ascending** and **Name Ascending** order results from the smallest range to the largest and **Count Descending** and **Name Descending** order results from the largest range to the smallest.
- **show/hide filters:** This option is displayed when a range is selected. It shows or hides the selection.
- **Menu (⋮):** The menu contains the following options:
  - **Select All Visible:** This option selects all of the ranges that are listed in the filter.
  - **Clear:** This option becomes available when a range is selected. Clicking **Clear** removes the selection.
  - **Designer:** Selecting this option opens the filter Designer so that you can view or change the filter configuration.
- **Close (X):** Clicking **X** removes the filter from the dashboard. This action cannot be undone.

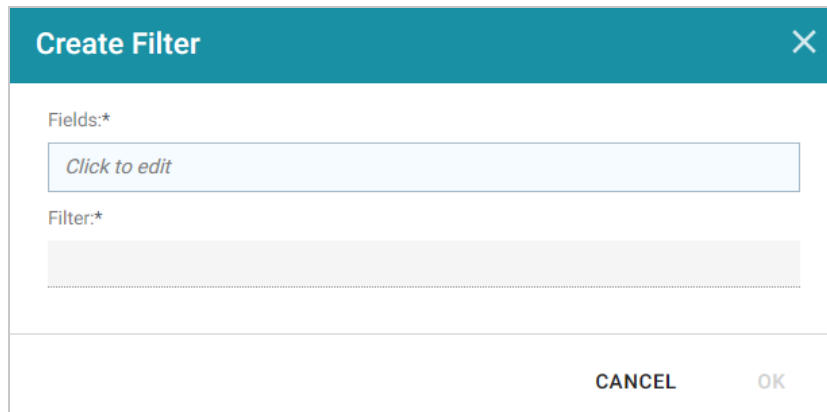
## Creating a Presence Filter

Presence filters group results based on whether the value exists or does not exist. This type of filter is useful for testing whether there are records that are missing a particular value. Presence filters are available for paths and properties of all data types.

Follow the instructions below to create a Presence filter.

1. Open the dashboard that you want to add the filter to.
2. In the Hi-Res Analytics main toolbar, click **Filters** and select **Create a Filter**. The Create Filter dialog box is displayed.





**Create Filter** [X]

Fields:\*

*Click to edit*

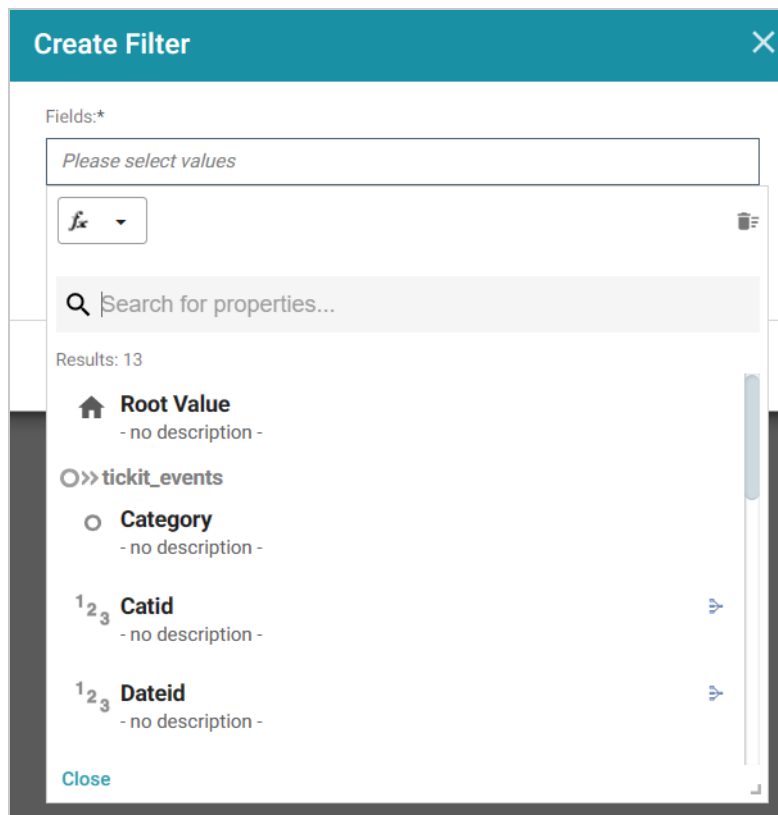
Filter:\*

[Empty text box]

CANCEL OK

3. Click in the **Fields** field to open the Property drop-down list and determine the path or property to filter on. The list of available properties depends on the selected data type for the dashboard.

For example, the following image shows the list of properties that are available for a dashboard whose source is a graphmart that contains data about tickets sold for various types of events. The type for the dashboard is tickit\_events:



**Create Filter** [X]

Fields:\*

*Please select values*

[Icon] [Dropdown arrow]

[Icon]

Search for properties...

Results: 13

- Root Value** (Home icon) - no description -
- tickit\_events** (Expand icon)
- Category** (Radio icon) - no description -
- Catid** (List icon) - no description -
- Dateid** (List icon) - no description -

Close

The list below describes the icons and options that are available when choosing a property:

- The **Root Value** (🏠) is the instance URI for the root resource—the URI for the instances of the class that was chosen as the data type for the dashboard.

### Tip

To view the Root Value values, you can use the STR function to show a string representation of the URIs.

- Linked classes are represented by incoming (◁) and outgoing (▷) connection icons. The properties in those classes with a path to another class are denoted with a circle icon (○). Selecting a linked property navigates to that class and displays its properties.
  - When a property or path is selected, the breadcrumbs at the top of the dialog box show you the property path. You can click the Clear icon (✕) to clear the path and start again.
  - After you have selected a property, you can apply a function or formula to that property to calculate the values that are displayed in the filter. To add a function, click the function button (fx) at the top of the drop-down list. The functions that become available depend on the data type of the selected property. To choose a more advanced function or type a formula, click **Advanced**. The Calculated Value dialog box opens and enables you to choose additional properties and functions. For more information, see [Calculating Values in Lenses and Filters](#).
4. After you have selected the property to filter on, click **Close** to close the Fields drop-down list.
  5. Next, click the **Filter** field and select **Presence** from the drop-down list. The dialog box is refreshed to show the Filter Properties and other options that are available for the filter type:

Create Filter

×

Fields:\*

eventid

Filter:\*

Presence

Filter Properties

Title:

Show counts: ☒ Respond to other filters: ☒

Subfilters

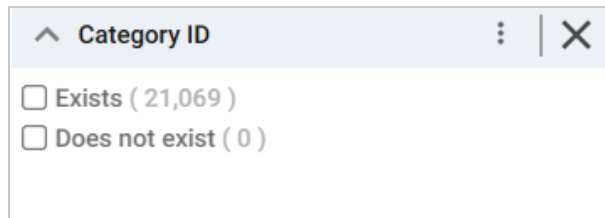
+ Create Filter

CANCEL

OK

6. Configure any of the following properties. All of the fields are optional:
  - **Title:** Defines the title that appears at the top of the filter when it is added to the dashboard. If Title is blank, the value from the Fields field is used as the title.
  - **Show Counts:** This setting controls whether the number of results for each item in the filter is displayed in parentheses next to the item.
  - **Respond to Other Filters:** This setting controls whether the results of this filter change based on selections in other filters on the dashboard.
7. If you would like to be able to further constrain the data that appears in the filter, you can add one or more subfilters. To add a subfilter, click **Create Filter** under Subfilters. The process of creating a subfilter is the same as the process for the parent filter. However, the subfilter is not displayed on the dashboard. It is visible only when editing the parent filter, and the subfilter's configuration affects only the parent filter and any sibling subfilters.
8. When you have finished configuring the filter, click **OK** to add it to the Dashboard. The new filter appears in the left pane of the dashboard and displays the values that are available for filtering the displayed data.

For example, the filter in the image below shows a Presence Filter that tests whether there are records that are missing the Category ID value. Selecting **Exists** or **Does not exist** filters the dashboard data to show only the records that fall into that category.



When working with the filter on the dashboard, the following options are available for sorting and configuration:

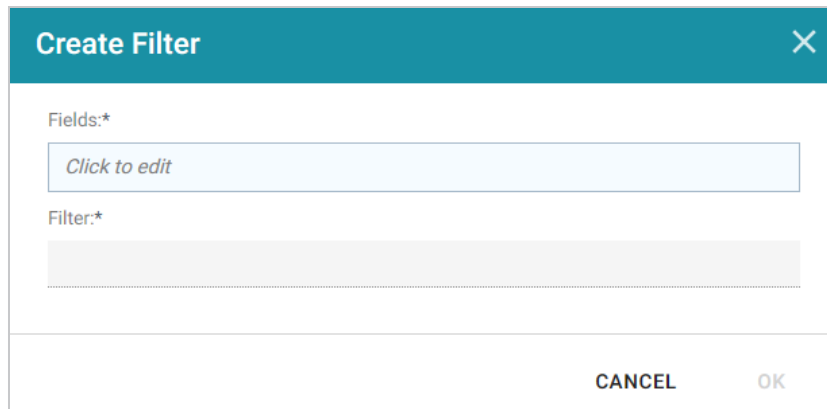
- **Menu (⋮):** The menu contains the following options:
  - **Clear:** This option becomes available when an item is selected. Clicking **Clear** removes the selection.
  - **Designer:** Selecting this option opens the filter Designer so that you can view or change the filter configuration.
- **Close (X):** Clicking **X** removes the filter from the dashboard. This action cannot be undone.

## Creating a Quartile Filter

Quartile filters group and rank the values for a property into four equal ranges. This filter is available for properties with integer, double, date, time, and dateTime data types. It is not available for paths.

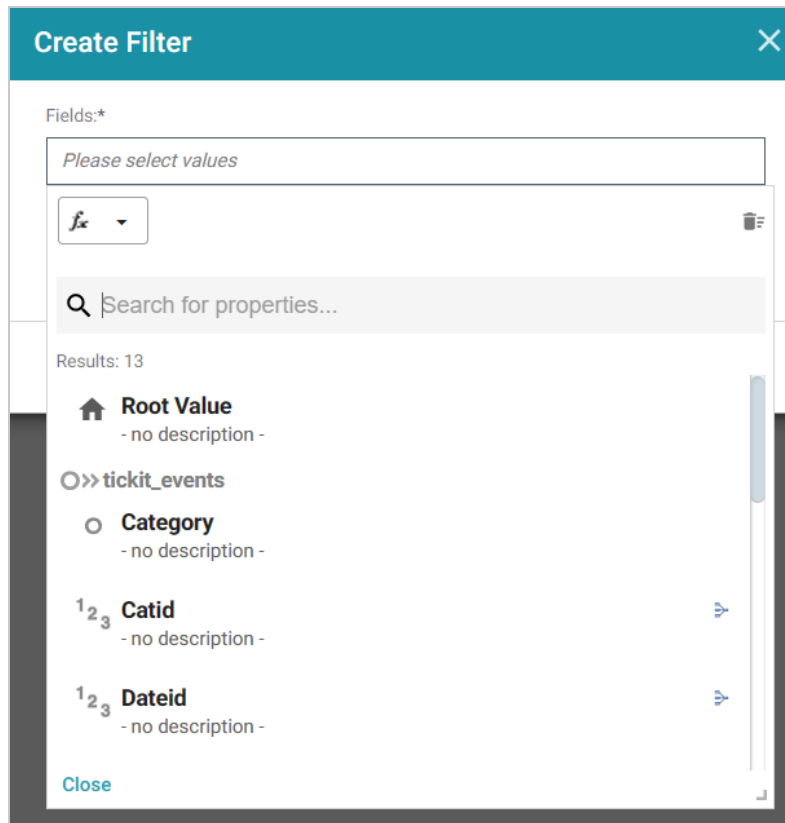
Follow the instructions below to create a Quartile filter.

1. Open the dashboard that you want to add the filter to.
2. In the Hi-Res Analytics main toolbar, click **Filters** and select **Create a Filter**. The Create Filter dialog box is displayed.

The image shows a 'Create Filter' dialog box. It has a teal header bar with the title 'Create Filter' and a close button (X). Below the header, there are two main sections. The first section is labeled 'Fields:\*' and contains a light blue rectangular button with the text 'Click to edit'. The second section is labeled 'Filter:\*' and contains a light gray rectangular input field. At the bottom of the dialog box, there are two buttons: 'CANCEL' and 'OK'.

3. Click in the **Fields** field to open the Property drop-down list and determine the property to filter on. The list of available properties depends on the selected data type for the dashboard.

For example, the following image shows the list of properties that are available for a dashboard whose source is a graphmart that contains data about tickets sold for various types of events. The type for the dashboard is tickit\_events:



The list below describes the icons and options that are available when choosing a property:

- The **Root Value** (🏠) is the instance URI for the root resource—the URI for the instances of the class that was chosen as the data type for the dashboard.

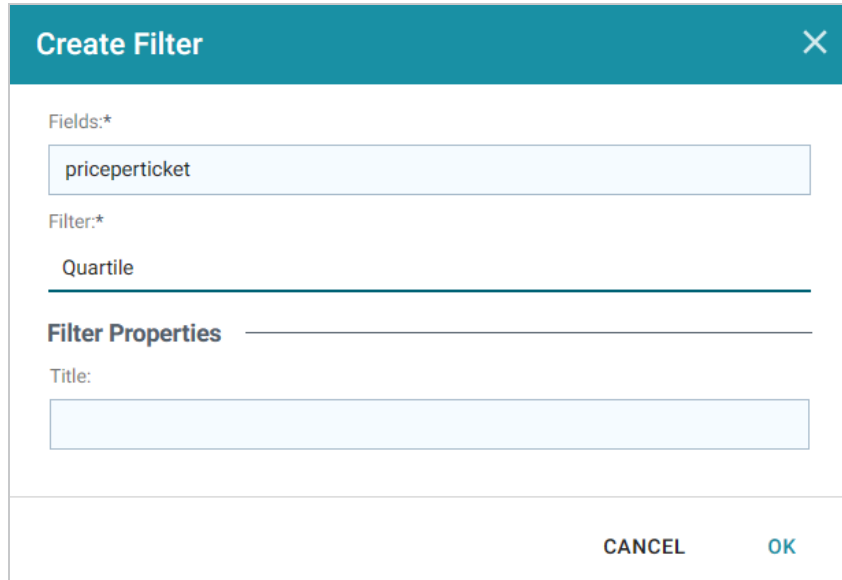
#### Tip

To view the Root Value values, you can use the STR function to show a string representation of the URIs.

- Linked classes are represented by incoming (◁◁) and outgoing (◁◁) connection icons. The properties in those classes with a path to another class are denoted with a circle icon (◯). Selecting a linked property navigates to that class and displays its properties.
- When a property or path is selected, the breadcrumbs at the top of the dialog box show you the property path. You can click the Clear icon (🗑️) to clear the path and start again.
- After you have selected a property, you can apply a function or formula to that property to calculate the values that are displayed in the filter. To add a function, click the function button (fx) at the top of the drop-down list. The functions that become available depend on the data type of the selected property. To choose a more advanced function or type a formula, click

**Advanced.** The Calculated Value dialog box opens and enables you to choose additional properties and functions. For more information, see [Calculating Values in Lenses and Filters](#).

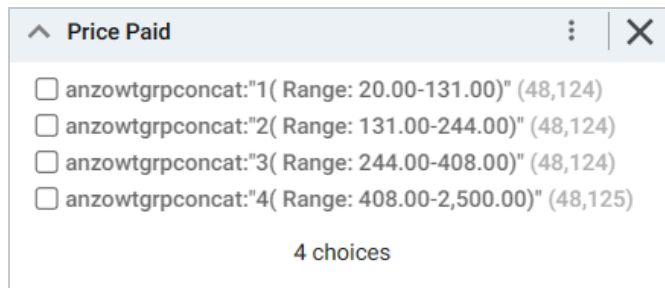
- After you have selected the property to filter on, click **Close** to close the Fields drop-down list.
- Next, click the **Filter** field and select **Quartile** from the drop-down list. The dialog box is refreshed to show the Filter Properties that are available for the filter type:



The 'Create Filter' dialog box has a teal header with the title 'Create Filter' and a close button (X). Below the header, there are three sections: 'Fields:\*' with a text input containing 'priceperticket'; 'Filter:\*' with a dropdown menu showing 'Quartile'; and 'Filter Properties' with a 'Title:' label and an empty text input. At the bottom right, there are 'CANCEL' and 'OK' buttons.

- The only property that is configurable for Quartile filters is the **Title**. This field is optional and defines the title that appears at the top of the filter when it is added to the dashboard. If Title is blank, the value from the Fields field is used as the title.
- When you have finished configuring the filter, click **OK** to add it to the dashboard. The new filter appears in the left pane of the dashboard and displays the values that are available for filtering the displayed data.

For example, the filter in the image below shows a filter that groups the price paid for tickets to events into four equal ranges.



The 'Price Paid' filter widget has a header with an expand/collapse arrow, the title 'Price Paid', a settings icon (three dots), and a close button (X). The main area contains four checkboxes, each followed by a label indicating the range and the number of records: 'anzowtgrpconcat:"1( Range: 20.00-131.00)" (48,124)', 'anzowtgrpconcat:"2( Range: 131.00-244.00)" (48,124)', 'anzowtgrpconcat:"3( Range: 244.00-408.00)" (48,124)', and 'anzowtgrpconcat:"4( Range: 408.00-2,500.00)" (48,125)'. At the bottom, it says '4 choices'.

One or more ranges can be selected to filter the dashboard data to show only the records from the selected ranges.

When working with the filter on the dashboard, the following options are available for sorting and configuration:

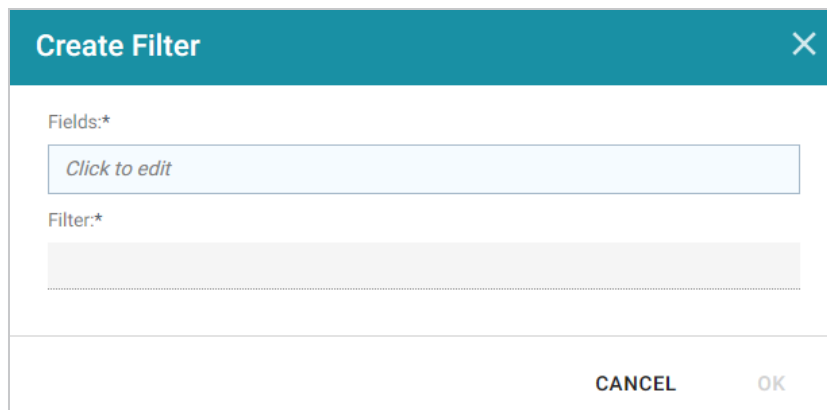
- **show/hide filters:** This option is displayed when a value is selected. It shows or hides the selection.
- **Menu (⋮):** The menu contains the following options:
  - **Select All Visible:** This option selects all of the items that are listed in the filter.
  - **Clear:** This option becomes available when an item is selected. Clicking **Clear** removes the selection.
  - **Designer:** Selecting this option opens the filter Designer so that you can view or change the filter configuration.
- **Close (X):** Clicking **X** removes the filter from the dashboard. This action cannot be undone.

### Creating a Range Slider Filter

Range Slider filters display a slider control that enables you to filter dashboard results by setting one range that you can adjust as needed. This type of filter is available for properties with integer, double, date, time, and dateTime data types. It is not available for paths.

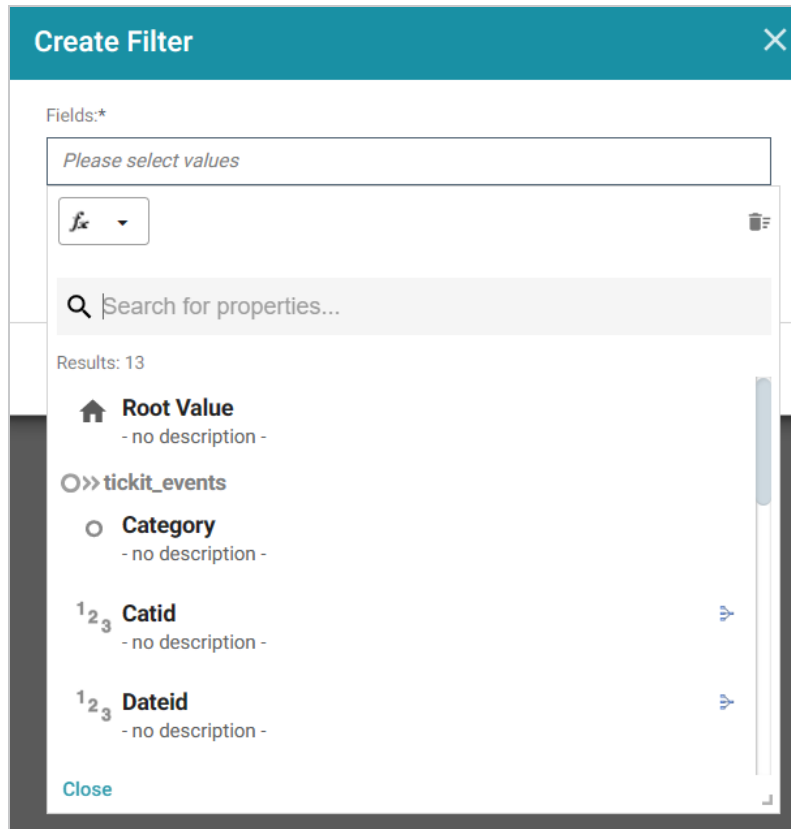
Follow the instructions below to create a Range Slider filter.

1. Open the dashboard that you want to add the filter to.
2. In the Hi-Res Analytics main toolbar, click **Filters** and select **Create a Filter**. The Create Filter dialog box is displayed.

The image shows a 'Create Filter' dialog box. It has a teal header bar with the title 'Create Filter' and a close button (X) on the right. Below the header, there are two main sections. The first section is labeled 'Fields:\*' and contains a light blue rectangular button with the text 'Click to edit'. The second section is labeled 'Filter:\*' and contains a light gray rectangular area with a dotted line at the bottom. At the bottom of the dialog box, there are two buttons: 'CANCEL' and 'OK'.

3. Click in the **Fields** field to open the Property drop-down list and determine the property to filter on. The list of available properties depends on the selected data type for the dashboard.

For example, the following image shows the list of properties that are available for a dashboard whose source is a graphmart that contains data about tickets sold for various types of events. The type for the dashboard is `ticket_events`:



The list below describes the icons and options that are available when choosing a property:

- The **Root Value** (🏠) is the instance URI for the root resource—the URI for the instances of the class that was chosen as the data type for the dashboard.

#### Tip

To view the Root Value values, you can use the STR function to show a string representation of the URIs.

- Linked classes are represented by incoming (◀) and outgoing (▶) connection icons. The properties in those classes with a path to another class are denoted with a circle icon (○). Selecting a linked property navigates to that class and displays its properties.
- When a property or path is selected, the breadcrumbs at the top of the dialog box show you the property path. You can click the Clear icon (🗑️) to clear the path and start again.



- After you have selected a property, you can apply a function or formula to that property to calculate the values that are displayed in the filter. To add a function, click the function button (**fx**) at the top of the drop-down list. The functions that become available depend on the data type of the selected property. To choose a more advanced function or type a formula, click **Advanced**. The Calculated Value dialog box opens and enables you to choose additional properties and functions. For more information, see [Calculating Values in Lenses and Filters](#).
4. After you have selected the property to filter on, click **Close** to close the Fields drop-down list.
  5. Next, click the **Filter** field and select **Range Slider** from the drop-down list. The dialog box is refreshed to show the Filter Properties and other options that are available for the filter type:

The screenshot shows the 'Create Filter' dialog box. It has a teal header bar with the text 'Create Filter' and a close button (X). Below the header, there are several sections:

- Fields:\*** A text input field containing 'eventDate/caldate'.
- Filter:\*** A dropdown menu showing 'Range Slider'.
- Filter Properties** A section with two input fields:
  - Title:** An empty text input field.
  - Label field:** A text input field containing the text 'Click to edit'.
- Subfilters** A section with a link that says '+ Create Filter'.

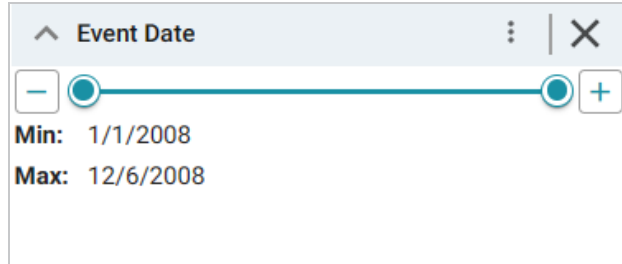
At the bottom of the dialog, there are two buttons: 'CANCEL' and 'OK'.

6. Configure any of the following properties. All of the fields are optional:
  - **Title:** Defines the title that appears at the top of the filter when it is added to the dashboard. If Title is blank, the value from the Fields field is used as the title.
  - **Label Field:** If you want to populate the list with values from a property other than the one specified in Fields, you can select an alternate property in this field.
7. If you would like to be able to further constrain the data that appears in the filter, you can add one or more subfilters. To add a subfilter, click **Create Filter** under Subfilters. The process of creating a subfilter is the same as the process for the parent filter. However, the subfilter is not displayed on the

dashboard. It is visible only when editing the parent filter, and the subfilter's configuration affects only the parent filter and any sibling subfilters.

8. When you have finished configuring the filter, click **OK** to add it to the dashboard. The new filter appears in the left pane of the dashboard and displays the values that are available for filtering the displayed data.

For example, the filter in the image below shows a Range Slider filter that uses a date property to define the range. The dashboard can be filtered by adjusting the Min and Max values to decrease or increase the range.



9. To refresh the results on the dashboard you can click and drag the Min or Max end of the slider or click the plus and minus buttons to adjust the range in small increments.

When working with the filter on the dashboard, the following options are available for sorting and configuration:

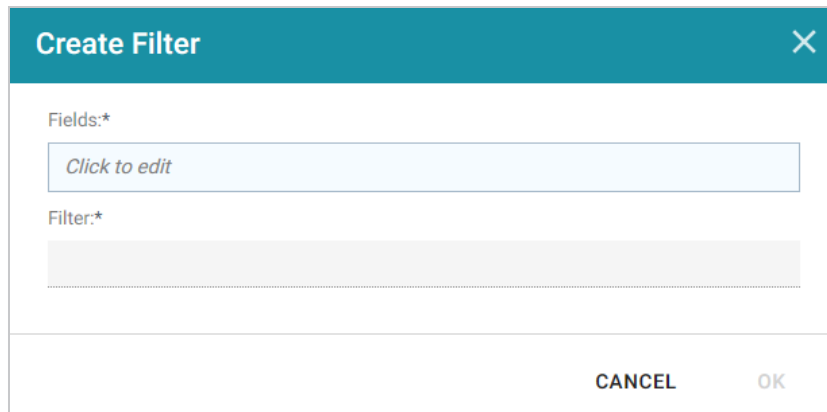
- **Menu (⋮):** The menu contains the following options:
  - **Clear:** This option is available once the slider has been adjusted. Clicking **Clear** resets the range to the default configuration.
  - **Designer:** Selecting this option opens the filter Designer so that you can view or change the filter configuration.
- **Close (X):** Clicking **X** removes the filter from the dashboard. This action cannot be undone.

## Creating a Relative Time Filter

Relative Time filters are used to filter for records that fall into a specified increment of time relative to the current time. This type of filter is available for date, time, and dateTime data types.

Follow the instructions below to create a Relative Time filter.

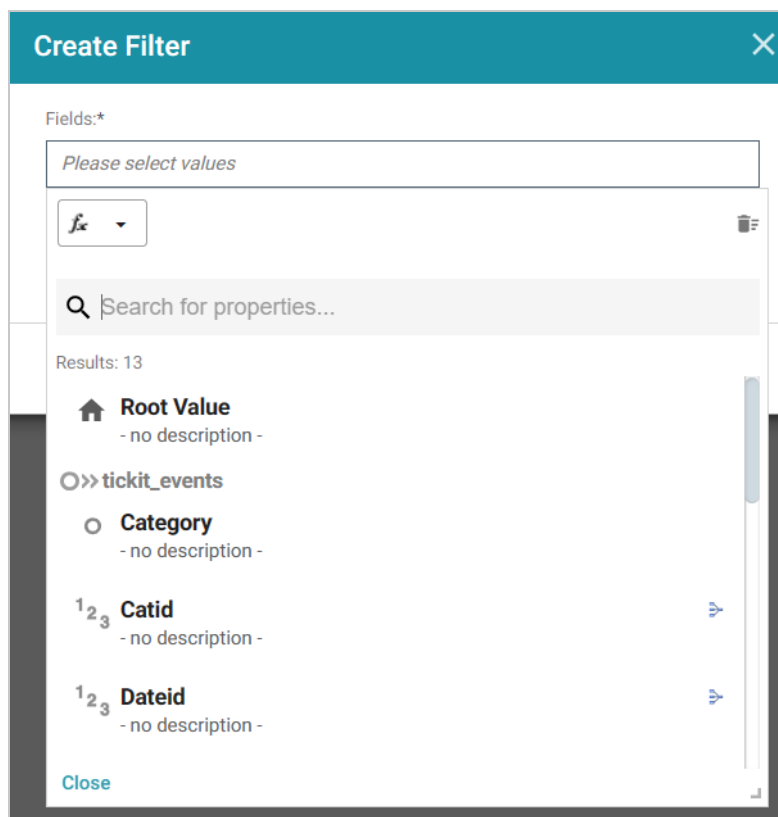
1. Open the dashboard that you want to add the filter to.
2. In the Hi-Res Analytics main toolbar, click **Filters** and select **Create a Filter**. The Create Filter dialog box is displayed.



The 'Create Filter' dialog box has a teal header with a close button (X). Below the header, there are two sections: 'Fields:\*' and 'Filter:\*'. The 'Fields:\*' section contains a light blue button labeled 'Click to edit'. The 'Filter:\*' section contains a light gray rectangular input field. At the bottom right, there are two buttons: 'CANCEL' and 'OK'.

3. Click in the **Fields** field to open the Property drop-down list and determine the date, dateTime, or time type property to filter on. The list of available properties depends on the selected data type for the dashboard.

For example, the following image shows the list of properties that are available for a dashboard whose source is a graphmart that contains data about tickets sold for various types of events. The type for the dashboard is tickit\_events:



This image shows the 'Create Filter' dialog box with the 'Fields:\*' section expanded. The 'Fields:\*' section has a text input field with the placeholder 'Please select values'. Below this is a dropdown menu with a magnifying glass icon and a trash icon. A search bar with the placeholder 'Search for properties...' is also present. Below the search bar, it says 'Results: 13'. The list of results includes:
 

- Root Value** - no description -
- tickit\_events** (expanded)
  - Category** - no description -
  - Catid** - no description -
  - Dateid** - no description -

 At the bottom left, there is a 'Close' button.

The list below describes the icons and options that are available when choosing a property:

- The **Root Value** (📍) is the instance URI for the root resource—the URI for the instances of the class that was chosen as the data type for the dashboard.

#### Tip

To view the Root Value values, you can use the STR function to show a string representation of the URIs.

- Linked classes are represented by incoming (◁) and outgoing (▷) connection icons. The properties in those classes with a path to another class are denoted with a circle icon (○). Selecting a linked property navigates to that class and displays its properties.
  - When a property or path is selected, the breadcrumbs at the top of the dialog box show you the property path. You can click the Clear icon (🗑️) to clear the path and start again.
  - After you have selected a property, you can apply a function or formula to that property to calculate the values that are displayed in the filter. To add a function, click the function button (fx) at the top of the drop-down list. The functions that become available depend on the data type of the selected property. To choose a more advanced function or type a formula, click **Advanced**. The Calculated Value dialog box opens and enables you to choose additional properties and functions. For more information, see [Calculating Values in Lenses and Filters](#).
4. After you have selected the property to filter on, click **Close** to close the Fields drop-down list.
  5. Next, click the **Filter** field and select **Relative Time** from the drop-down list. The dialog box is refreshed to show the Filter Properties and other options that are available for the filter type:

Create Filter

×

Fields:\*

starttime

Filter:\*

Relative Time

Filter Properties

Title:

Label field:

Click to edit

Respond to other filters:

☒

Subfilters

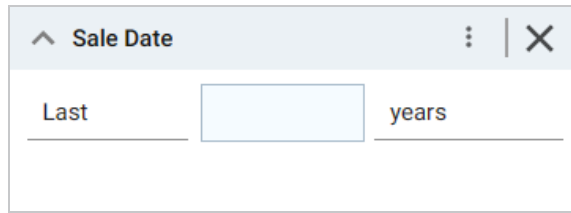
+ Create Filter

CANCEL

OK

6. Configure any of the following properties. All of the fields are optional:
  - **Title:** Defines the title that appears at the top of the filter when it is added to the dashboard. If Title is blank, the value from the Fields field is used as the title.
  - **Label Field:** If you want to populate the filter with a label other than the one specified in Fields, you can select an alternate property in this field.
  - **Respond to Other Filters:** This setting controls whether the results of this filter change based on selections in other filters on the dashboard.
7. If you would like to be able to further constrain the data that appears in the filter, you can add one or more subfilters. To add a subfilter, click **Create Filter** under Subfilters. The process of creating a subfilter is the same as the process for the parent filter. However, the subfilter is not displayed on the dashboard. It is visible only when editing the parent filter, and the subfilter's configuration affects only the parent filter and any sibling subfilters.
8. When you have finished configuring the filter, click **OK** to add it to the dashboard. The new filter appears in the left pane of the dashboard and displays the values that are available for filtering the displayed data.

For example, the filter in the image below filters data by the Sale Date for tickets.



9. To configure the requirements for the data to display on the Dashboard, specify the following options:
  - On the left, select **Last** or **Next** to configure the relative time direction.
  - In the middle, specify a number to represent the amount of time.
  - On the right, select the time increment to use.

When working with the filter on the dashboard, the following options are also available:

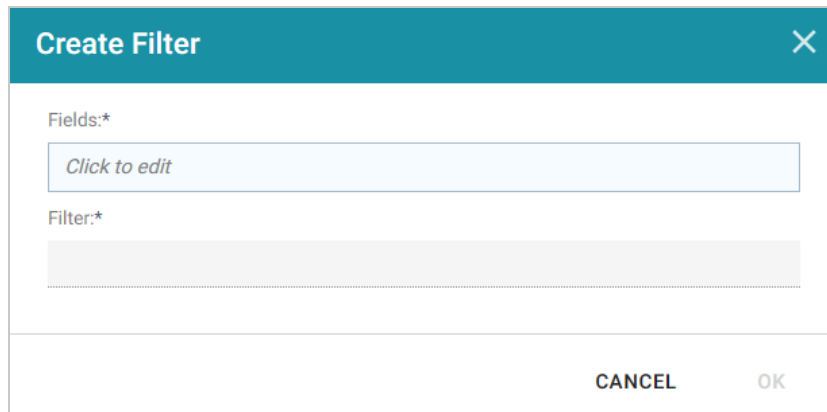
- **Menu** (⋮): The menu contains the following options:
  - **Clear**: This option is available when once the filter options are selected. Clicking **Clear** resets the filter to the default values.
  - **Designer**: Selecting this option opens the filter Designer so that you can view or change the filter configuration.
- **Close** (X): Clicking **X** removes the filter from the dashboard. This action cannot be undone.

## Creating a Search Filter

Search filters are used to search for values of a property that contain a partial match, exact match, or do not equal the text that you specify. The search is case-insensitive. This type of filter is available for all data types. It is not available for use with paths.

Follow the instructions below to create a Search Filter.

1. Open the dashboard that you want to add the filter to.
2. In the Hi-Res Analytics main toolbar, click **Filters** and select **Create a Filter**. The Create Filter dialog box is displayed.



**Create Filter** [X]

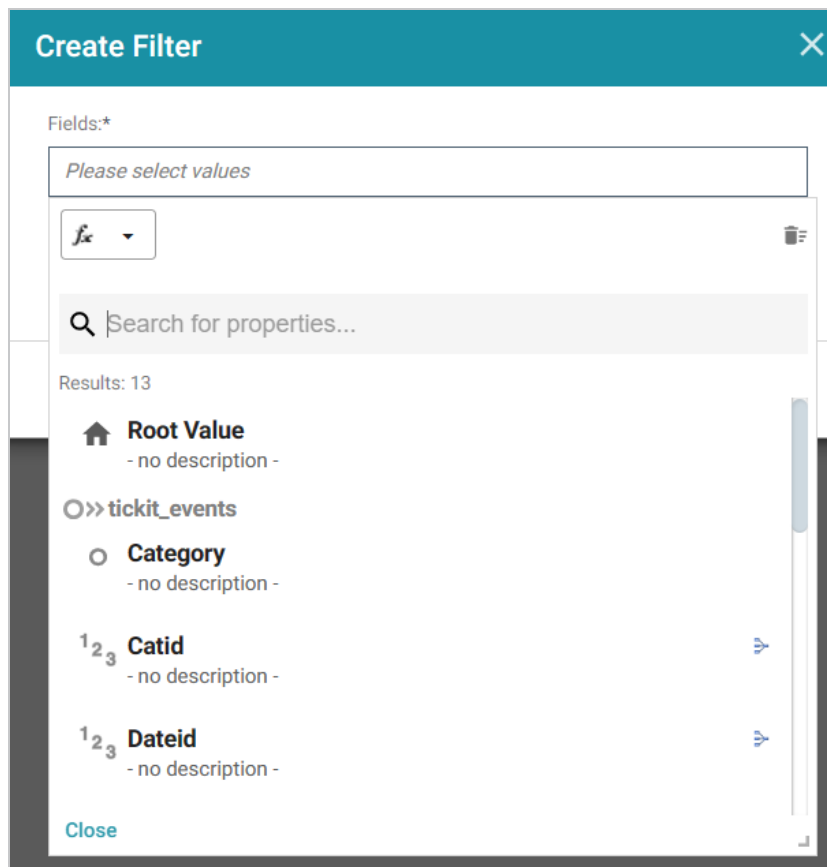
Fields:\*

Filter:\*

CANCEL OK

3. Click in the **Fields** field to open the Property drop-down list and determine the property to filter on. The list of available properties depends on the selected data type for the dashboard.

For example, the following image shows the list of properties that are available for a dashboard whose source is a graphmart that contains data about tickets sold for various types of events. The type for the dashboard is tickit\_events:



**Create Filter** [X]

Fields:\*

[v] [icon]

Results: 13

- Root Value**  
- no description -
- tickit\_events**
  - Category**  
- no description -
  - Catid**  
- no description -
  - Dateid**  
- no description -

Close

The list below describes the icons and options that are available when choosing a property:

- The **Root Value** (📍) is the instance URI for the root resource—the URI for the instances of the class that was chosen as the data type for the dashboard.

#### Tip

To view the Root Value values, you can use the STR function to show a string representation of the URIs.

- Linked classes are represented by incoming (◀) and outgoing (▶) connection icons. The properties in those classes with a path to another class are denoted with a circle icon (○). Selecting a linked property navigates to that class and displays its properties.
  - When a property or path is selected, the breadcrumbs at the top of the dialog box show you the property path. You can click the Clear icon (🧼) to clear the path and start again.
  - After you have selected a property, you can apply a function or formula to that property to calculate the values that are displayed in the filter. To add a function, click the function button (fx) at the top of the drop-down list. The functions that become available depend on the data type of the selected property. To choose a more advanced function or type a formula, click **Advanced**. The Calculated Value dialog box opens and enables you to choose additional properties and functions. For more information, see [Calculating Values in Lenses and Filters](#).
4. After you have selected the property to filter on, click **Close** to close the Fields drop-down list.
  5. Next, click the **Filter** field and select **Search Filter** from the drop-down list. The dialog box is refreshed to show the Filter Properties and other options that are available for the filter type:

Create Filter

Fields:\*

eventname

Filter:\*

Search Filter

Filter Properties

Title:

Subfilters

+ Create Filter

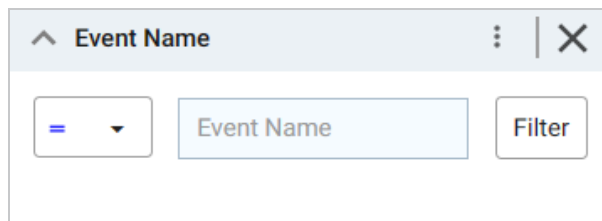
CANCEL

OK

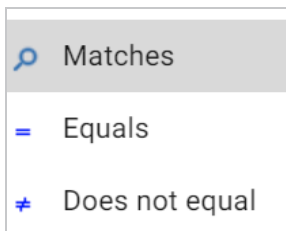


6. The only property that is configurable for Search Filters is the **Title**. This field is optional and defines the title that appears at the top of the filter when it is added to the dashboard. If Title is blank, the value from the Fields field is used as the title.
7. If you would like to be able to further constrain the data that appears in the filter, you can add one or more subfilters. To add a subfilter, click **Create Filter** under Subfilters. The process of creating a subfilter is the same as the process for the parent filter. However, the subfilter is not displayed on the dashboard. It is visible only when editing the parent filter, and the subfilter's configuration affects only the parent filter and any sibling subfilters.
8. When you have finished configuring the filter, click **OK** to add it to the dashboard. The new filter appears in the left pane of the dashboard and displays the values that are available for filtering the displayed data.

For example, the filter in the image below defines Event Name as the property to search on.



9. To perform a search, configure the following options:
  - On the left, select the type of match to perform. **Matches** includes partial matches and **Equals** is an exact match. Note that matches are case-insensitive.



- In the middle, specify the value to search for. The search is case-insensitive.

10. Press **Enter** or click **Filter** to perform the search and refresh the data on the dashboard.

When working with the filter on the dashboard, the following options are also available:

- **Menu** (⋮): The menu contains the following options:
  - **Clear**: This option becomes available when a search term is entered. Clicking **Clear** removes the term.

- **Designer:** Selecting this option opens the filter Designer so that you can view or change the filter configuration.

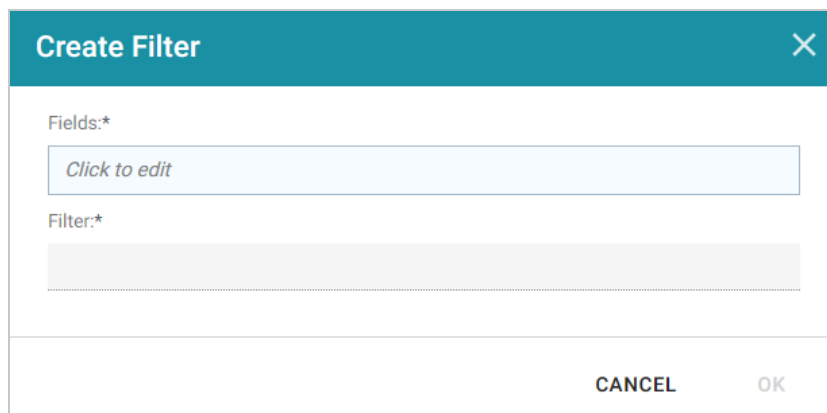
- **Close (X):** Clicking **X** removes the filter from the dashboard. This action cannot be undone.

## Creating a Single Select List Filter

Single Select List filters are similar to List filters but only allow you to filter one value from the list at a time. This type of filter is available for properties of all data types but is not available for paths.

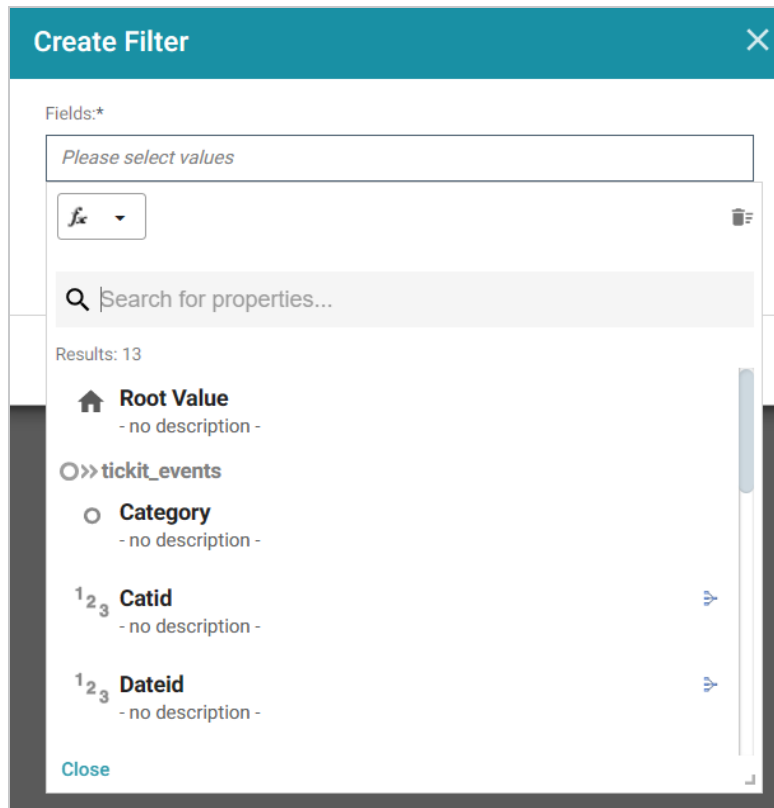
Follow the instructions below to create a Single Select List filter.

1. Open the dashboard that you want to add the filter to.
2. In the Hi-Res Analytics main toolbar, click **Filters** and select **Create a Filter**. The Create Filter dialog box is displayed.

The image shows a 'Create Filter' dialog box. It has a teal header bar with the title 'Create Filter' and a close button (X) on the right. Below the header, there are two main sections. The first section is labeled 'Fields:\*' and contains a light blue rectangular button with the text 'Click to edit'. The second section is labeled 'Filter:\*' and contains a light gray rectangular area. At the bottom of the dialog box, there are two buttons: 'CANCEL' and 'OK'.

3. Click in the **Fields** field to open the Property drop-down list and determine the property to filter on. The list of available properties depends on the selected data type for the dashboard.

For example, the following image shows the list of properties that are available for a dashboard whose source is a graphmart that contains data about tickets sold for various types of events. The type for the dashboard is ticket\_events:



The list below describes the icons and options that are available when choosing a property:

- The **Root Value** (🏠) is the instance URI for the root resource—the URI for the instances of the class that was chosen as the data type for the dashboard.

#### Tip

To view the Root Value values, you can use the STR function to show a string representation of the URIs.

- Linked classes are represented by incoming (🔍) and outgoing (🔍) connection icons. The properties in those classes with a path to another class are denoted with a circle icon (○). Selecting a linked property navigates to that class and displays its properties.
- When a property or path is selected, the breadcrumbs at the top of the dialog box show you the property path. You can click the Clear icon (🗑️) to clear the path and start again.
- After you have selected a property, you can apply a function or formula to that property to calculate the values that are displayed in the filter. To add a function, click the function button (fx) at the top of the drop-down list. The functions that become available depend on the data type of the selected property. To choose a more advanced function or type a formula, click

**Advanced.** The Calculated Value dialog box opens and enables you to choose additional properties and functions. For more information, see [Calculating Values in Lenses and Filters](#).

- After you have selected the property to filter on, click **Close** to close the Fields drop-down list.
- Next, click the **Filter** field and select **Single Select List** from the drop-down list. The dialog box is refreshed to show the Filter Properties and other options that are available for the filter type:

The screenshot shows the 'Create Filter' dialog box. It has a teal header bar with the text 'Create Filter' and a close button (X). Below the header, there are several sections:

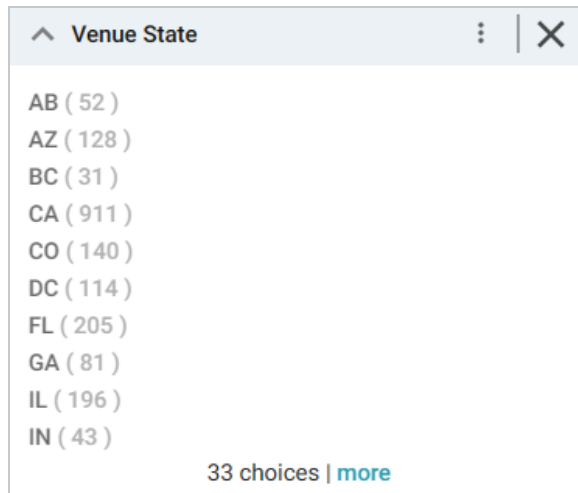
- Fields:\*** A text input field containing 'category/catname'.
- Filter:\*** A text input field containing 'Single Select List'.
- Filter Properties** A section with a horizontal line above it. It contains:
  - Title:** A text input field.
  - Label field:** A text input field containing 'Click to edit'.
  - Exclude:** A checkbox.
  - Show Bars:** A checkbox.
  - Show Blanks:** A checkbox.
  - Show counts:** A checked checkbox.
  - Respond to other filters:** A checked checkbox.
- Subfilters** A section with a horizontal line above it. It contains a link '+ Create Filter'.

At the bottom of the dialog, there are two buttons: 'CANCEL' and 'OK'.

- Configure any of the following properties. All of the fields are optional:
  - Title:** Defines the title that appears at the top of the filter when it is added to the dashboard. If Title is blank, the value from the Fields field is used as the title.
  - Label Field:** If you want to populate the list with values from a property other than the one specified in Fields, you can select an alternate property in this field.
  - Exclude:** This setting controls whether selecting a value in the filter narrows the results to show only the records that *include* that value or whether selecting a value *excludes* the records that include that value. When Exclude is disabled, selecting a value in the list narrows the dashboard results to show only the records that include that value. When Exclude is enabled, selecting a value filters out all of the records that include that value.
  - Show Bars:** This setting controls whether the total values for the selected property appear as a bar graphic in the background of the filter.

- **Show Blanks:** This setting controls whether to include null values for the selected property by listing them as **Blank** in the filter.
  - **Show Counts:** This setting controls whether the number of results for each item in the filter is displayed in parentheses next to the item.
  - **Respond to Other Filters:** This setting controls whether the results of this filter change based on selections in other filters on the dashboard.
7. If you would like to be able to further constrain the data that appears in the filter, you can add one or more subfilters. To add a subfilter, click **Create Filter** under Subfilters. The process of creating a subfilter is the same as the process for the parent filter. However, the subfilter is not displayed on the dashboard. It is visible only when editing the parent filter, and the subfilter's configuration affects only the parent filter and any sibling subfilters.
  8. When you have finished configuring the filter, click **OK** to add it to the dashboard. The new filter appears in the left pane of the dashboard and displays the values that are available for filtering the displayed data.

For example, the filter in the image below shows a filter that lists each of the states that have event venues. One state at a time can be selected to filter the data on the dashboard.



Depending on whether the Exclude option is enabled or disabled, selecting an item in the filter refreshes the Dashboard to show only the data includes the selected value or only the data that does not include the selected value.

When working with the filter on the dashboard, the following options are available for sorting and configuration:

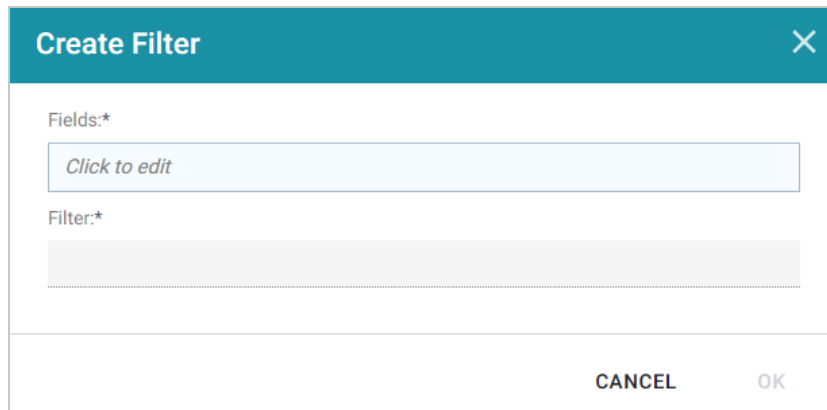
- **Menu (⋮):** The menu contains the following options:
  - **Clear:** This option is available when an item is selected. Clicking **Clear** removes the selection.
  - **Designer:** Selecting this option opens the filter Designer so that you can view or change the filter configuration.
- **Close (X):** Clicking **X** removes the filter from the dashboard. This action cannot be undone.

## Creating a Types Filter

Types filters are used to filter data according to the types of data (classes) that are connected by a specified path. This type of filter is available only for paths and not properties.

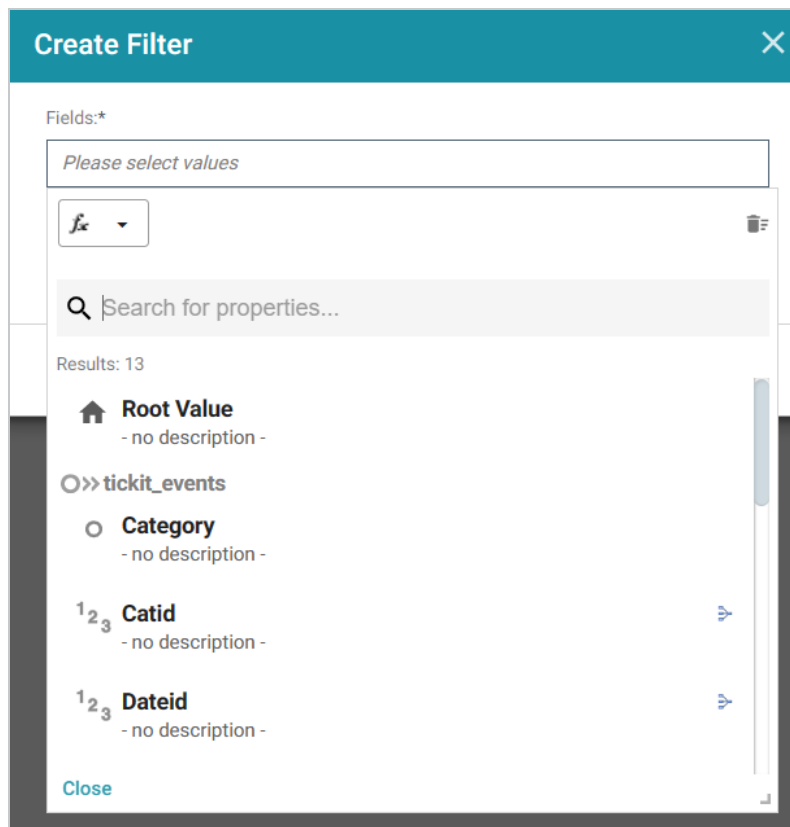
Follow the instructions below to create a Types filter.

1. Open the dashboard that you want to add the filter to.
2. In the Hi-Res Analytics main toolbar, click **Filters** and select **Create a Filter**. The Create Filter dialog box is displayed.

The image shows a 'Create Filter' dialog box with a teal header bar containing the title 'Create Filter' and a close button (X). The main area is white and contains two input fields. The first field is labeled 'Fields:\*' and contains the text 'Click to edit'. The second field is labeled 'Filter:\*' and is currently empty. At the bottom of the dialog, there are two buttons: 'CANCEL' and 'OK'.

3. Click in the **Fields** field to open the Property drop-down list and determine the path to filter on. The list of available classes, paths, and properties depends on the selected data type for the dashboard.

For example, the following image shows the list of properties that are available for a dashboard whose source is a graphmart that contains data about tickets sold for various types of events. The type for the dashboard is tickit\_events:



The list below describes the icons and options that are available when choosing a path:

- The **Root Value** (🏠) is the instance URI for the root resource—the URI for the instances of the class that was chosen as the data type for the dashboard.

#### Tip

To view the Root Value values, you can use the STR function to show a string representation of the URIs.

- Linked classes are represented by incoming (🔍) and outgoing (🔍) connection icons. The properties in those classes with a path to another class are denoted with a circle icon (○). Selecting a linked property navigates to that class and displays its properties.
- When a property or path is selected, the breadcrumbs at the top of the dialog box show you the property path. You can click the Clear icon (🗑️) to clear the path and start again.
- After you have selected a path, you can apply a function or formula to calculate the values that are displayed in the filter. To add a function, click the function button (fx) at the top of the drop-down list. The functions that become available depend on the data type of the selected path. To choose a more advanced function or type a formula, click **Advanced**. The Calculated Value

dialog box opens and enables you to choose additional properties and functions. For more information, see [Calculating Values in Lenses and Filters](#).

- After you have selected the path to filter on, click **Close** to close the Fields drop-down list.
- Next, click the **Filter** field and select **Types** from the drop-down list. The dialog box is refreshed to show the Filter Properties and other options that are available for the filter type:

**Create Filter** [X]

Fields:\*  
ROOT

Filter:\*  
Types

**Filter Properties**

Title:  
[ ]

Exclude: ☐ Show Bars: ☐ Show Blanks: ☐ Always Show Checkboxes: ☒  
Show counts: ☒ Respond to other filters: ☒

**Subfilters**  
[+ Create Filter](#)

CANCEL OK

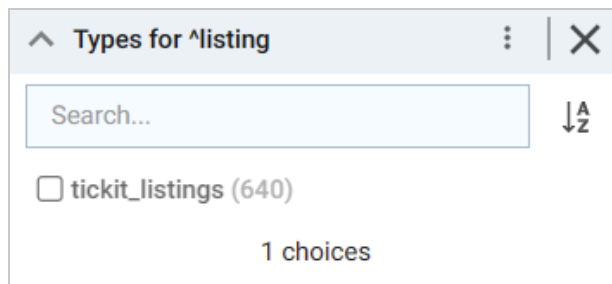
- Configure any of the following properties. All of the fields are optional:
  - Title:** Defines the title that appears at the top of the filter when it is added to the dashboard. If Title is blank, the value from the Fields field is used as the title.
  - Exclude:** This setting controls whether selecting a value in the filter narrows the results to show only the records that *include* that value or whether selecting a value *excludes* the records that include that value. When Exclude is disabled, selecting a value in the filter narrows the dashboard results to show only the records that include that value. When Exclude is enabled, selecting a value filters out all of the records that include that value.
  - Show Bars:** This setting controls whether the total values for the selected property appear as a bar graphic in the background of the filter.
  - Show Blanks:** This setting controls whether to include null values for the selected property by listing them as **Blank** in the filter.



- **Always Show Checkboxes:** This setting controls whether checkboxes are shown next to the items in the filter.
- **Show Counts:** This setting controls whether the number of results for each filter value are displayed in parentheses next to the value.
- **Respond to Other Filters:** This setting controls whether the results of this filter change based on selections in other filters on the dashboard.

7. If you would like to be able to further constrain the data that appears in the filter, you can add one or more subfilters. To add a subfilter, click **Create Filter** under Subfilters. The process of creating a subfilter is the same as the process for the parent filter. However, the subfilter is not displayed on the dashboard. It is visible only when editing the parent filter, and the subfilter's configuration affects only the parent filter and any sibling subfilters.
8. When you have finished configuring the filter, click **OK** to add it to the dashboard. The new filter appears in the left pane of the dashboard and displays the values that are available for filtering the displayed data.

For example, the filter in the image below shows that there is one type of class of data that is connected by the chosen path.



Depending on whether the Exclude option is enabled or disabled, selecting an item in the filter refreshes the dashboard to show only the data includes the selected value or only the data that does not include the selected value.

When working with the filter on the dashboard, the following options are available for sorting and configuration:

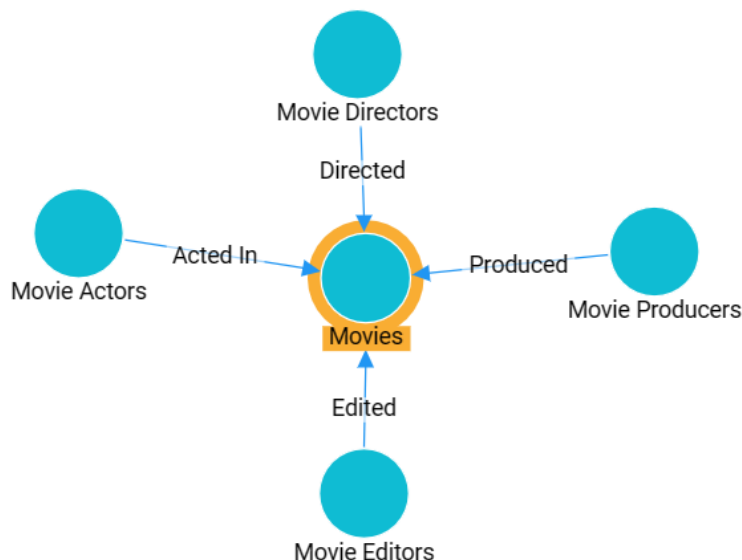
- **Search:** Enables you to search for a value in the filter. The search is case-insensitive.
- **Sort (↕):** Shows the following options for sorting the results in the filter:
  - **Count Ascending:** Orders results from the smallest count to the largest.
  - **Count Descending:** Orders results from the largest count to the smallest.

- **Name Ascending:** Orders results in alphabetical order.
- **Name Descending:** Orders results in reverse alphabetical order.
- **show/hide filters:** This option is displayed when a value is selected. It shows or hides the selection.
- **Menu (⋮):** The menu contains the following options:
  - **Select All Visible:** This option selects all of the items that are listed in the filter.
  - **Clear:** This option is available when one or more items are selected. Clicking **Clear** removes the selections.
  - **Designer:** Selecting this option opens the filter Designer so that you can view or change the filter configuration.
- **Close (X):** Clicking **X** removes the filter from the dashboard. This action cannot be undone.

## Combining Data from Multiple Classes

Though you must choose one base data type (or class) for each Hi-Res Analytics dashboard, selecting a data type with connections to other classes enables you to configure lenses and filters that combine the data from those classes. This powerful capability can help surface the semantic relationships in your data and enable you to leverage those relationships to access and integrate all of the data in the knowledge graph. When choosing the base data type for a dashboard, it helps to consider all of the desired filters.

For example, consider the following data model for a movie data set:



By creating a dashboard that specifies **Movies** as the base data type, the lenses and filters in the dashboard can navigate the paths to properties in the other classes, Movie Actors, Movie Directors, Movie Producers, and so on. This topic provides guidance on accessing data from multiple classes in filters and lenses.

- [Combining Classes in a Lens](#)
- [Filtering on Multiple Classes](#)

### Combining Classes in a Lens

The image below shows a dashboard that accesses the graph for the above model. The specified data type is **Movies**, and a table lens displays all of the columns/properties in the Movies class:

\*Movies

Ready to use

FILTERS

Manage Lenses

Graphmart

Movies Graphmart

Data Types

Movies ( 8,157 )

Data Layers

Movies

Designer Properties Export Manage Columns

What can I do next?

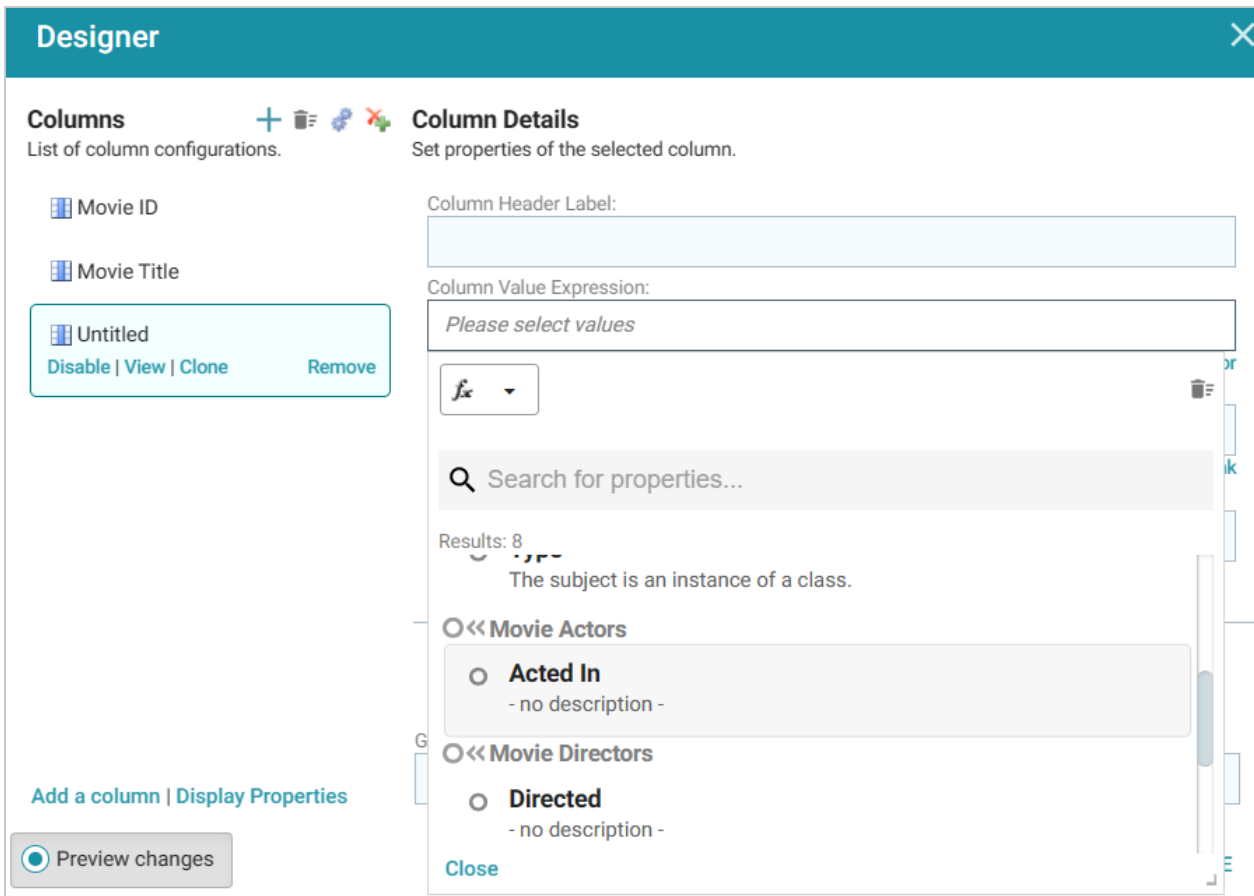
All Movies

Movie ID	Movie Title
3217	Army of Darkness
3746	Blade Runner
3837	Blazing Saddles
3947	Blue Velvet (film)
4227	Barry Lyndon
4231	Buffy the Vampire Slayer (film)
4560	Braveheart
4726	Batman (1989 film)
4727	Batman (1966 film)
4728	Batman Returns
4729	Batman & Robin (film)
4730	Batman Forever
5313	Crouching Tiger, Hidden Dragon
7906	Destry Rides Again
8481	Dressed to Kill (1980 film)

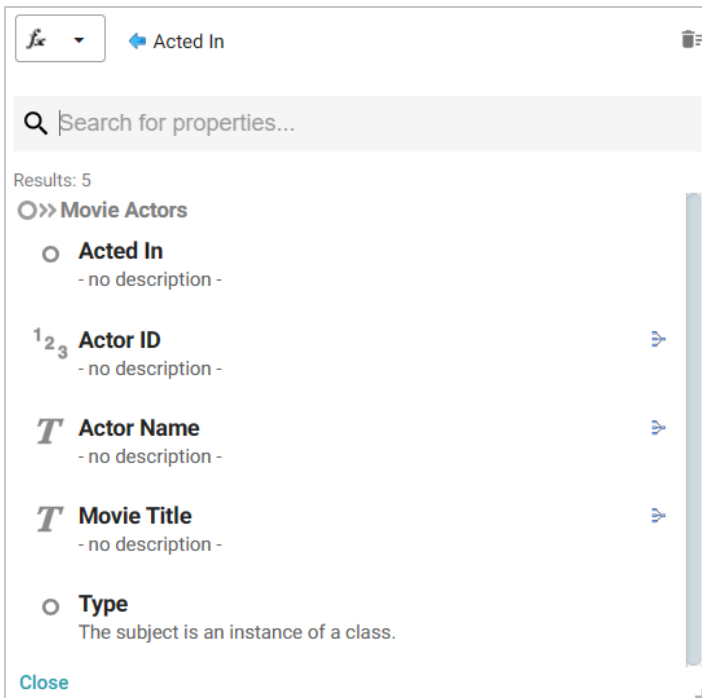
1 - 20 of 8157

Page 1 of 408

Lenses and filters can be configured to leverage the relationships from the base class to the connected classes. For example, adding a column that navigates the **ActedIn** path to access the **Movie Actors** class could be used to display values such as the names of the actors who starred in the movies. To navigate the relationship in the lens Designer, the **ActedIn** path from the **MovieActors** class is selected for the new column:



Once the path is chosen, all of the properties from the **Movie Actor** class are displayed:



Selecting **ActorName** adds the column to the dashboard. The actors from each movie are now integrated into the lens even though the actor name values are not in the base class.

Designer

Properties

Export

Manage Columns

All Movies

Movie ID	Movie Title	Actor Name
4727	Batman (1966 film)	Lee Meriwether, Frank Gorshin, Cesar Romero, Burt Ward, Burgess Meredith, Adam West
4728	Batman Returns	Pat Hingle, Michelle Pfeiffer, Michael Murphy (actor), Michael Keaton, Michael Gough, Danny DeVito, Christopher Walken
4729	Batman & Robin (film)	Uma Thurman, Pat Hingle, Michael Gough, John Glover (actor), George Clooney, Elle Macpherson, Chris O'Donnell, Arnold Schwarzenegger, Alicia Silverstone
4730	Batman Forever	Val Kilmer, Tommy Lee Jones, Pat Hingle, Nicole Kidman, Michael Gough, Jim Carrey, Chris O'Donnell
5313	Crouching Tiger, Hidden Dragon	Zhang Ziyi, Michelle Yeoh, Chow Yun-fat, Chang Chen
7906	Destry Rides Again	Mischa Auer, Marlene Dietrich, James Stewart, Brian Donlevy
8481	Dressed to Kill (1980 film)	Nancy Allen (actress), Michael Caine, Keith Gordon, Angie Dickinson

<<<>>>

1 - 20 of 8157

Page 1 of 408

Settings

## Filtering on Multiple Classes

In addition to combining classes in lenses, you can also apply filters across classes. Like the example above, the image below shows a dashboard where the specified data type is **Movies**, and a Table lens displays all of the columns/properties in the Movies class:

\*Movies

Ready to use

FILTERS

Manage Lenses

Graphmart

Movies Graphmart

Data Types

Movies ( 8,157 )

Data Layers

Movies

Designer

Properties

Export

Manage Columns

What can I do next?

All Movies

Movie ID	Movie Title
3217	Army of Darkness
3746	Blade Runner
3837	Blazing Saddles
3947	Blue Velvet (film)
4227	Barry Lyndon
4231	Buffy the Vampire Slayer (film)
4560	Braveheart
4726	Batman (1989 film)
4727	Batman (1966 film)
4728	Batman Returns
4729	Batman & Robin (film)
4730	Batman Forever
5313	Crouching Tiger, Hidden Dragon
7906	Destry Rides Again
8481	Dressed to Kill (1980 film)

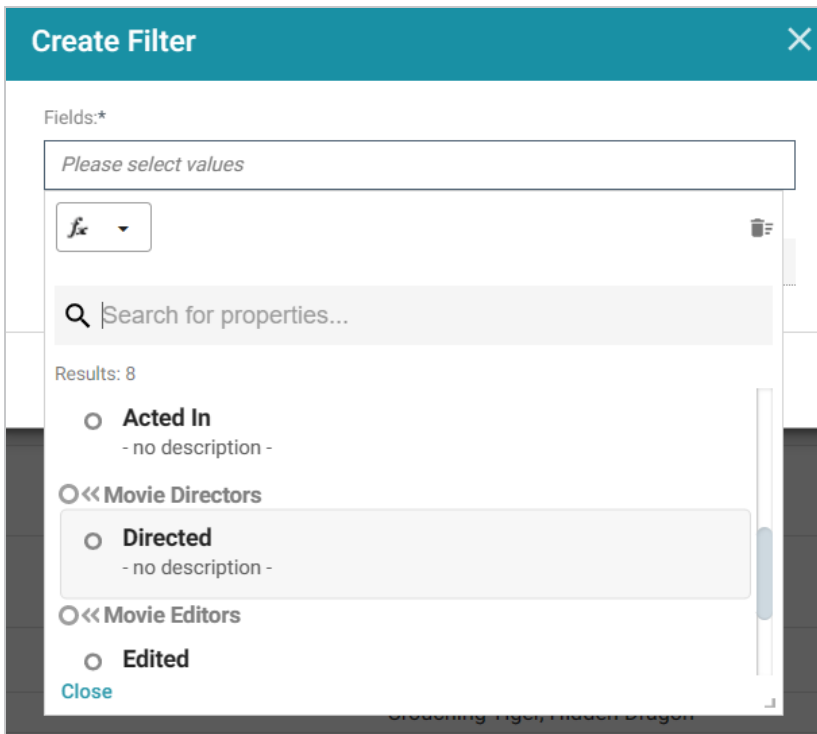
1 - 20 of 8157

Page 1 of 408

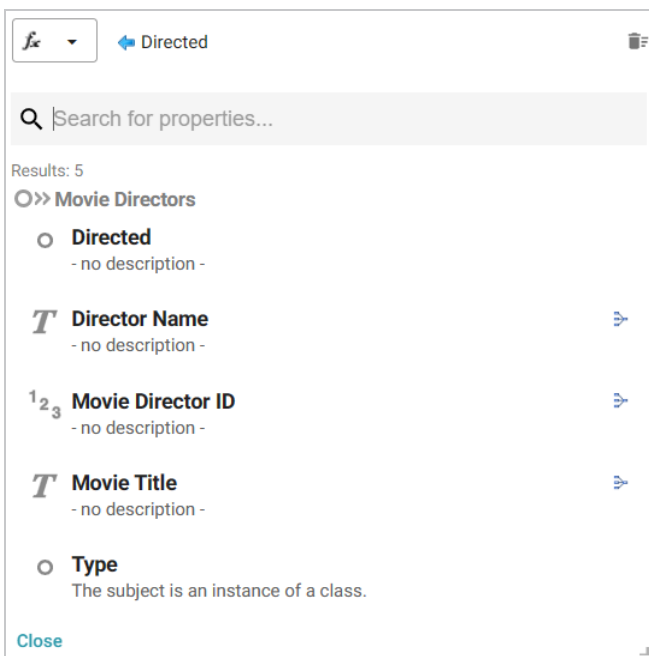
A filter can be configured to leverage the relationships from the base class to the connected classes. For example, adding a filter that navigates the **Directed** path to access the **Movie Directors** class could be used to display, and filter on, the name of the director for each movie. To navigate the relationship in the Create Filter dialog box, the **Directed** path from the **MovieDirectors** class is selected for the Field to filter on:

Combining Data from Multiple Classes

875



Once the path is chosen, the properties from the **Movie Directors** class are displayed:



Selecting the **Director Name** property and choosing **List** as the type of filter adds a filter to the dashboard that lists all of the directors in the graph. Users can select particular director names to filter the lens so that it only shows the movies that include one or more of the selected directors.



\*Movies x

Ready to use

FILTERS

Manage Lenses

Graphmart +

Data Types

Data Layers

^Directed/Director Name (2) | X

Search...

[+] show filters

☒ A. Edward Sutherland (4)
 ☐ Aaron Katz (filmmaker) (1)
 ☒ Aaron Norris (4)
 ☐ Aaron Schneider (1)
 ☐ Aaron Scott Moorhead (1)
 ☐ Abel Ferrara (9)

Designer Properties Export Manage Columns

All Movies x

Movie ID	Movie Title	Actor Name
1308304	Sidekicks (1992 film)	
2683330	The Invisible Woman (1940 film)	
5638447	The Hitman	Saved by the Bell: The New Class, Michael Parks, Chuck Norris, Al Waxman
6653265	Delta Force 2: The Colombian Connection	Richard Jaeckel, Paul Perri, Mark Margolis, John P. Ryan (actor), Chuck Norris, Billy Drago
8001525	Follow the Boys	Vera Zorina, George Raft
9058143	Top Dog (1995 film)	Chuck Norris
9095434	Dixie (film)	
32129077	The Navy Comes Through	

Using the same path traversal, filters could also be created to narrow the data to certain actors or producers by following the relationships to the Movie Actors or Movie Producers classes. For more information about creating or editing lenses and filters, see [Creating Lenses](#) and [Creating Dashboard Filters](#).

## Related Topics

[Creating Lenses](#)

[Creating Dashboard Filters](#)

[Sharing Access to a Dashboard or Lens](#)

## Calculating Values in Lenses and Filters

Anzo provides many standard and advanced functions that you can use to compute the values that are displayed on a dashboard. When selecting properties and paths for lenses and filters, you can add calculations by selecting functions from a list or by writing your own formula. The Hi-Res Analytics application enables you to save your formulas as computed properties that can be reused on other dashboards, lenses, and filters.

This topic provides instructions for using functions and formulas to calculate displayed values, saving formulas as computed properties, and reusing computed properties.

- [Computations in Filters and Lenses](#)
- [Applying Functions and Formulas to Properties](#)
- [Saving Formulas for Reuse](#)
- [Reusing Computed Properties](#)

### Computations in Filters and Lenses

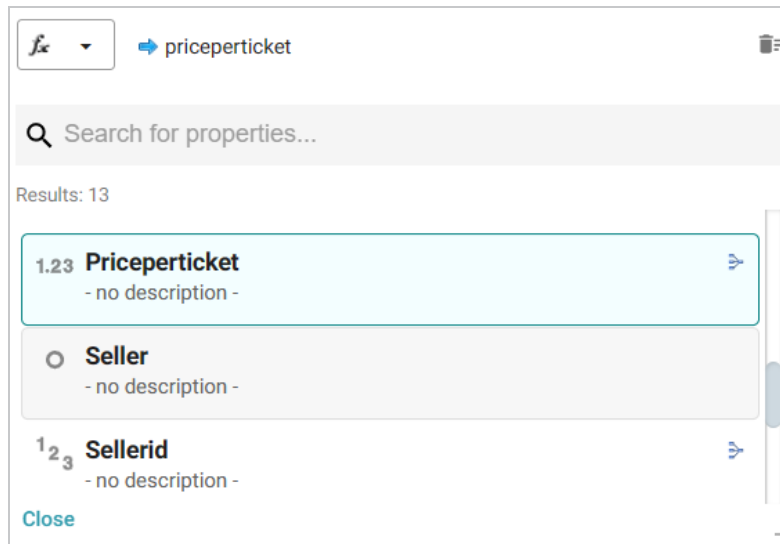
When you apply formulas to properties in filters, Anzo performs the calculation across all of the values that exist for the selected property and then groups the results into the list of values that the calculations return. For multiple value properties, all values for that property are included in the calculations.


When you apply formulas to properties in lenses, the calculation results depend on the data type of the dashboard or lens. If the property belongs to a class that allows multiple values, Anzo performs the calculation on each set of multiple values and returns the results as one record in the lens. If the class includes single value properties, the calculation is performed separately for each single value.

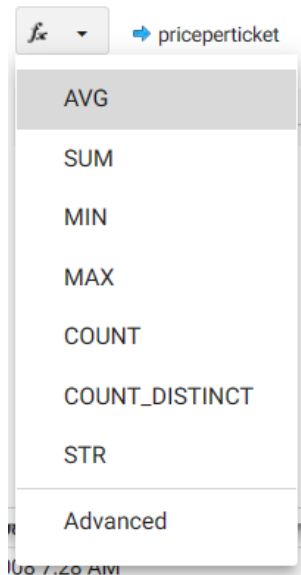
### Applying Functions and Formulas to Properties

Follow these instructions to use a function or formula to compute the values in a lens or filter.

1. Create a new lens or filter or open the Designer for an existing lens or filter.
2. In the drop-down list for selecting properties or fields, select the property or path for which you want to compute the values. For example:



- Click the function button (  ) to display the list of standard functions. The list varies depending on the data type of the selected property.



- Click a function to apply it to the property that you chose. For information about each of the available functions, see [Anzo Function and Formula Reference](#).
- To choose a more advanced function or type a formula, click **Advanced**. The Calculated Value dialog box opens and enables you to choose additional properties and functions as well as type your own calculation.

Calculated Value

Please select values

Formula

[{tickit\_listings}priceperticket]

Save as computed property | Clear

Fields

Search for properties...

Results: 13

Root Value

- no description -

O>> tickit\_listings

123 Dateid

- no description -

Functions

Search for functions...

Results: 182

ABS( number )

Calculate absolute value of term.

ADD( value1, value2 )

Calculate addition between 2 terms.

CLOSE

### Tip

To create an advanced formula, it might help to get started by viewing the functions listed in the Functions column on the right side of the screen. Each function includes the syntax to use for creating a formula that uses that function.

- In the Functions column, double-click a function to add it to the Formula field at the top of the dialog box. For information about each of the available functions, see [Anzo Function and Formula Reference](#).
- Place your cursor in the Formula where you want to insert the property to perform the calculation on (for example, inside the parentheses) and then double-click the property in the Fields column. If the syntax for the function includes characters such as commas, type the characters in the appropriate location in the formula. You can click the **Clear** link on the bottom right of the Formula field any time to clear that field and start over.

For example, the formula below calculates the total price paid by multiplying the values in the price per ticket column with the value in the number of tickets column:

Calculated Value

Please select values

Formula

MULTIPLY([tickit\_listings]numtickets],[tickit\_listings]priceperticket))

Save as computed property | Clear

Fields

Search for properties...

Results: 13

123 Numtickets

- no description -

1.23 Priceperticket

- no description -

Insert Field | Reset

Functions

Search multi

Results: 1

fx MULTIPLY(v1, v2)

Calculate numeric multiply

CLOSE

8. When you are finished writing a formula, you have two options:

- If you want to use the formula now without saving it for later use, click **Close** to close the Calculated Value dialog box. Then complete the lens or filter configuration.
- If you want save the formula for reuse, click the **Save as computed property** link and follow the instructions below in [Saving Formulas for Reuse](#).

## Saving Formulas for Reuse

Follow these instructions to save a formula as a computed property that you can use in other lenses and filters that target the same class of data.

1. When you have finished writing a formula in the Calculated Value dialog box, click the **Save as computed property** link below the Formula field. The Save formula as computed property dialog box opens.

2. In the **Title** field, type a name for the new computed property.

Save formula as a computed property

Title:\*

Description:

Ontology:

Tickets - Auto

CANCEL

SAVE

3. Type a description of the new property in the **Description** field.
4. If necessary, click in the **Ontology** field to choose another ontology to save the property in. If you want to save this property in multiple ontologies, you can click the **Save as computed property** link again after saving the property in the current ontology.
5. Click **Save**. Anzo saves the new property and labels it with an asterisk (\*). The property becomes available in the Fields column in the Calculated Value dialog box. For example, the image below shows the Computed Total Price property that saves the formula from the [Applying Functions and Formulas to Properties](#) section above.

Fields

Search for properties...

Results: 13

Root Value

- no description -

O>> ticket\_listings

T

\*Computed Total Price

- no description -

Delete

1 2 3 Dateid

- no description -

Insert Field | Reset

Functions

Search for functions...

Results: 182

fx ABS( number )

Calculate absolute value of term.

fx ADD( value1, value2 )

Calculate addition between 2 terms.

fx AND( logical1, logical2 )

Calculate logical AND of params.

fx AVERAGEIF( values\_to\_test, criteria, valu...

Averages only those elements that match the

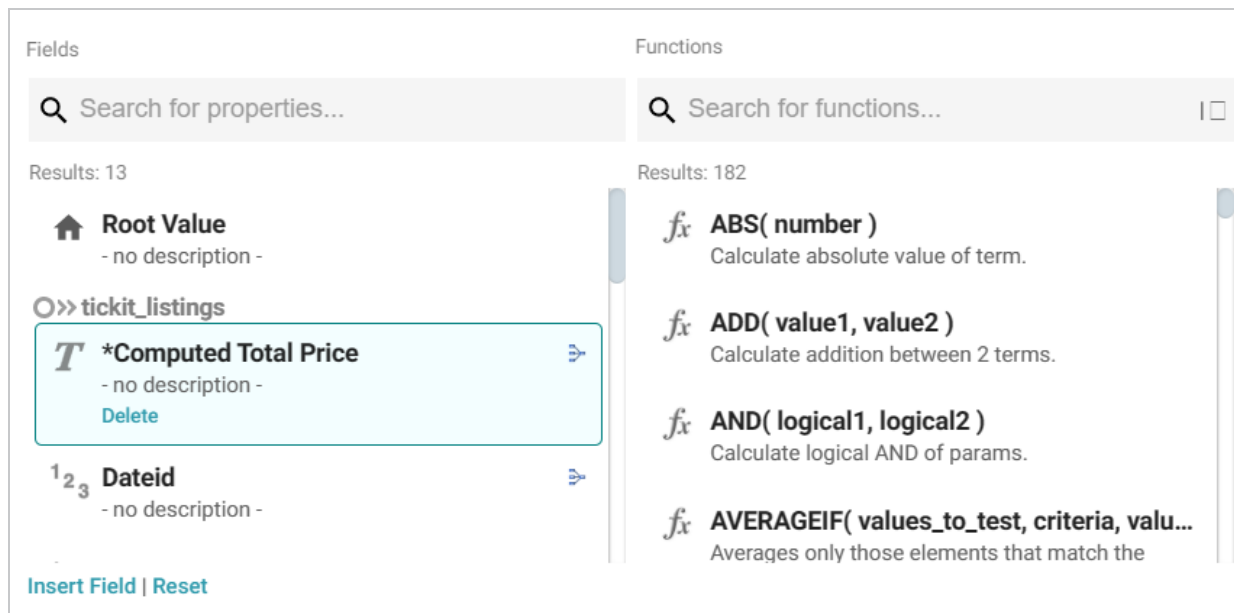
6. Click **Close** to close the dialog box.

## Reusing Computed Properties

When an ontology contains computed properties, any other dashboards, lenses, and filters that use that ontology can also use the computed properties as long as they also use the same data type or class of data that the computed property is saved in.

To use a computed property:

1. Open the Designer for the filter or lens where you want to apply the computed property.
2. Click in the **Fields** or **Column Value Expression** field to open the property selection drop-down list. The drop-down list includes any computed properties that are available for use with the selected data type. Computed properties are labeled with an asterisk (\*).



3. To use the property as-is, select the property and then close the drop-down list.
4. If you want to make changes to the formula and save it as a different computed property, select the property and then click the function button (fx) to open the Calculated Value dialog box. Follow the instructions in [Saving Formulas for Reuse](#) above to edit the formula and then save a new computed property.

## Related Topics

[Creating Dashboard Filters](#)

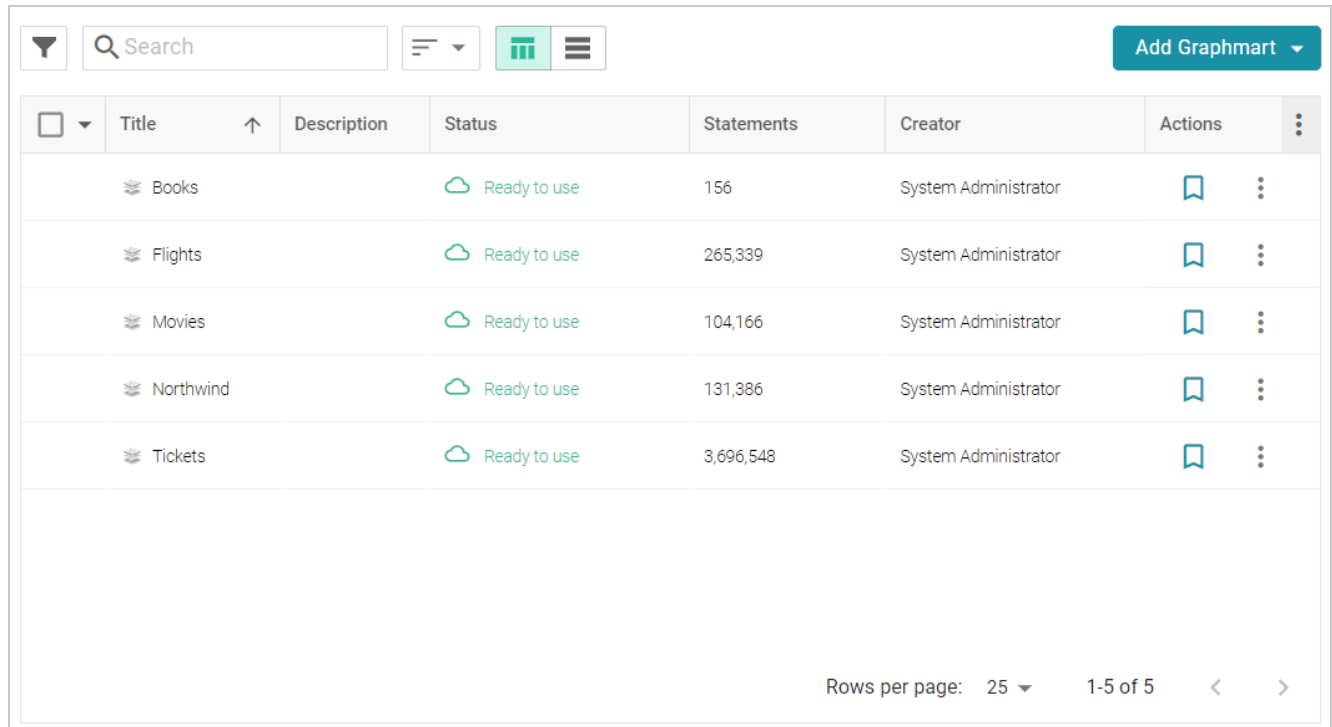
[Creating Lenses](#)

[Anzo Function and Formula Reference](#)

## Searching for Text in Unstructured Documents

Anzo Hi-Res Analytics incorporates the [Elasticsearch](#) search engine to enable you to perform full text searches on unstructured documents. This topic provides instructions for creating a dashboard with text search capability and running a search across unstructured documents.

1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing graphmarts. For example:

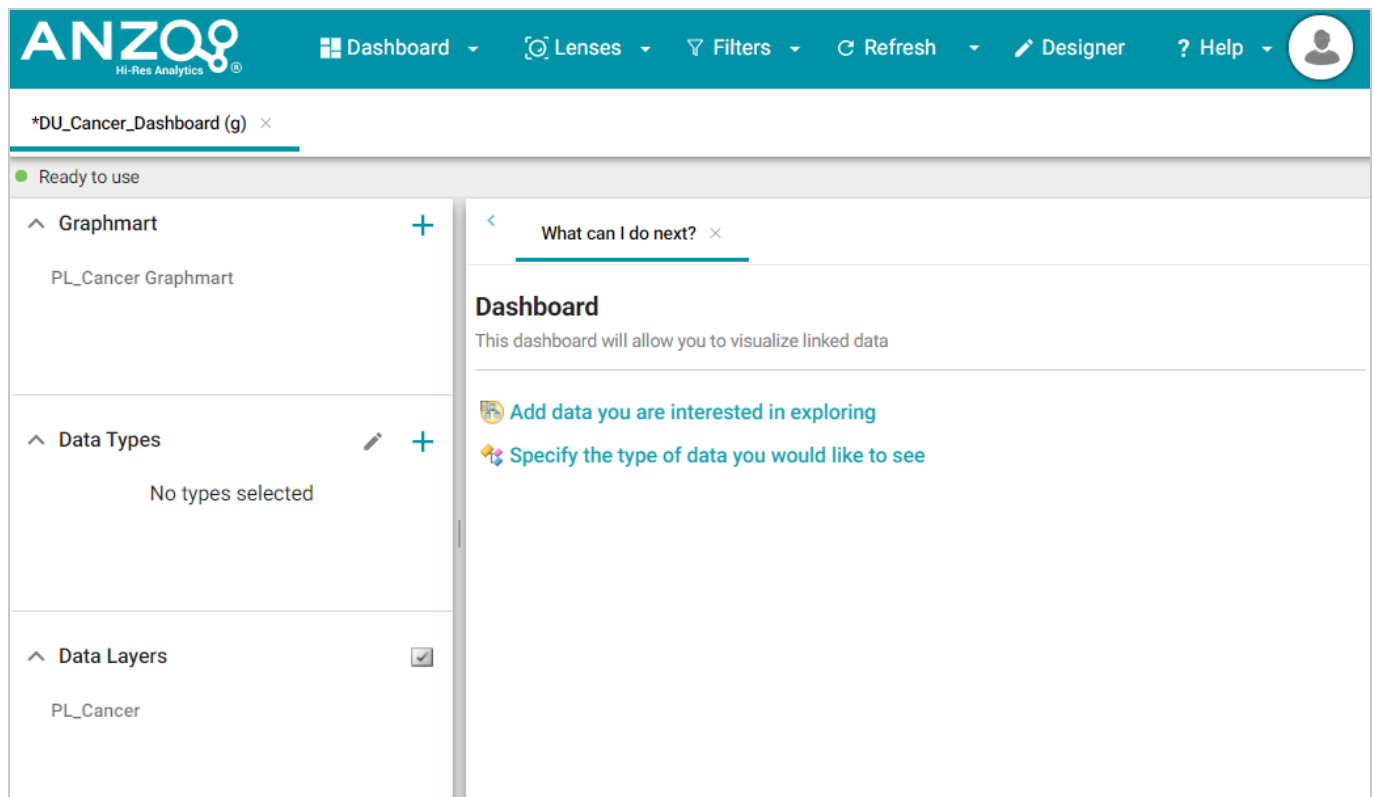


<input type="checkbox"/>	Title	Description	Status	Statements	Creator	Actions
	Books		Ready to use	156	System Administrator	
	Flights		Ready to use	265,339	System Administrator	
	Movies		Ready to use	104,166	System Administrator	
	Northwind		Ready to use	131,386	System Administrator	
	Tickets		Ready to use	3,696,548	System Administrator	

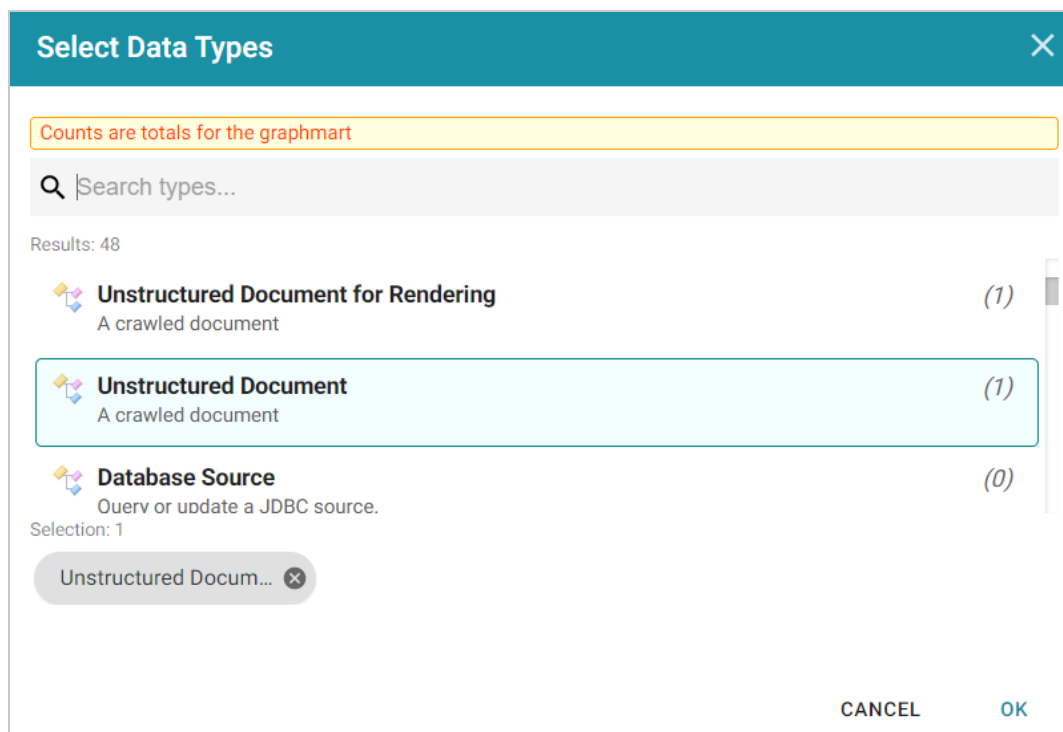
Rows per page: 25 1-5 of 5 < >

2. On the Graphmarts screen, click the name of the graphmart that contains the unstructured data. Anzo displays the graphmart overview screen.
3. Click the **Create Dashboard** button. The Hi-Res Analytics application opens and displays the Create Dashboard dialog box. Leave **Graphmart Dashboard** selected and click **Next**.
4. Type a name for the dashboard in the **Title** field and enter an optional **Description**. Then click **OK**. Anzo creates the dashboard and populates the Graphmart and Data Layers panels. For example:



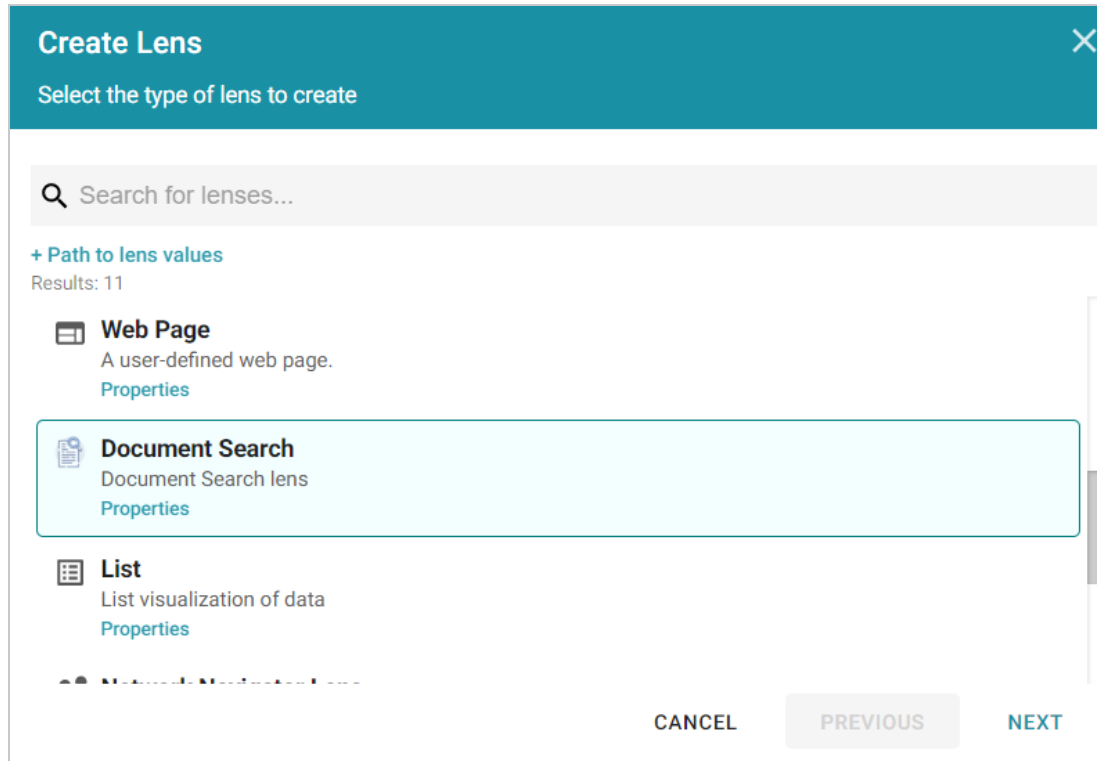


- In the Data Types panel, click the plus icon (+) to open the Select Data Types dialog box. In the dialog box, select **Unstructured Document**.



- Click **OK**. Anzo adds the data type to the Data Types panel.

- Next, click the **Lenses** button in the main toolbar and select **New** from the drop-down list. Anzo opens the Create Lens dialog box.



- In the Create Lens dialog box, select **Document Search** and then click **Next**. Anzo displays the General Information dialog box.

## Create Lens

×

Specify details about the new lens

### General Information

Title:\*

Description:

[\[+\] more...](#)

CANCELPREVIOUSFINISH

9. Type a name for the lens in the **Title** field and include an optional **Description**. Then click **Finish**. Anzo opens the Document Search Designer where you can configure the search settings or customize the style sheet, query, and HTML, if necessary. For example:

Designer

Search Settings
Query
HTML
CSS

General

Show No Results On Empty Search:

☐

Allow multi select:

☐

Synonym Expansion

Dictionary:

☐

Knowledge base:

Dataset:

Ontology:

Predicates:

☒ Preview changes

CANCEL

SAVE

10. In the Designer, change the optional search settings as needed. The list below describes each option:
- **Show No Results on Empty Search:** Determines whether documents are listed in the search results before a search is run. When enabled, the Document Search lens remains blank until a search is run.
  - **Allow Multi Select:** Determines whether a user can select multiple documents at a time in the results. When enabled, multiple documents can be selected by holding the Shift key and clicking documents in the results.
  - **Synonym Expansion Dictionary:** Determines whether to display an option for including synonyms in text searches. When enabled, the lens displays an **Include Synonyms** checkbox next to the Search field.
  - **Knowledge Base Dataset:** Enables you to include a knowledge base in the search if one exists. Click the field to select an available knowledge base.

- **Ontology:** Enables you to select a data model to use for the search.
- **Predicates:** Enables you to select specific predicates from the model.

11. Click **Save**. Anzo add the lens to the dashboard. Depending on the search settings, the lens displays the list of documents. For example:

The screenshot shows the Anzo dashboard interface. On the left, there is a sidebar with sections: 'Graphmart' (containing 'PL\_Cancer Graphmart'), 'Data Types' (containing 'Unstructured Document ( 27 )'), and 'Data Layers' (containing 'PL\_Cancer'). The main area is titled 'Designer' and 'Properties'. Below this, there are tabs: 'What can I do next?', 'Annotated Document', 'Original Document', 'Table of Documents', and 'DS'. The 'DS' tab is active. A search bar is present with the text 'Search...'. Below the search bar, there are sort options: 'Sort by: Last Modified' and 'Descending'. The results section shows '27 Results'. The first six results are displayed as a table with columns for the document title, file path, relevance score, and date.

Document Title	File Path	Relevance Score	Date
<a href="#">Growth factor receptors_ promising drug targets in cancer.pdf</a>	file:///global/nfs/data/du/docs/cancer/Growth%20factor%20receptors_%20promising%20drug%20targets%20in%20ca...	Unknown	Mon Dec 02 2...
<a href="#">Upregulation of miR-375 inhibits human liver cancer cell growth by modulating cell proliferation and apoptosis via tar...</a>	file:///global/nfs/data/du/docs/cancer/Upregulation%20of%20miR%20E2%80%91375%20inhibits%20human%20liver%2...	Unknown	Mon Dec 02 2...
<a href="#">Emergence of ERBB2 Mutation as a Biomarker and an Actionable Target in Solid Cancers.pdf</a>	file:///global/nfs/data/du/docs/cancer/Emergence%20of%20ERBB2%20Mutation%20as%20a%20Biomarker%20and%...	Unknown	Mon Dec 02 2...
<a href="#">Recognition of HER2 expression in hepatocellular carcinoma and its significance in postoperative tumor recurrence -...</a>	file:///global/nfs/data/du/docs/cancer/Recognition%20of%20HER2%20expression%20in%20hepatocellular%20carcin...	Unknown	Mon Dec 02 2...
<a href="#">Expression of ERBB2 in cancer - Summary - The Human Protein Atlas.pdf</a>	file:///global/nfs/data/du/docs/cancer/Expression%20of%20ERBB2%20in%20cancer%20-%20Summary%20-%20The...	Unknown	Mon Dec 02 2...
<a href="#">Increased Expression of c-erbB-2 in Liver is Associated with Hepatitis B x Antigen and Shorter Survival in Patients w...</a>	file:///global/nfs/data/du/docs/cancer/Increased%20Expression%20of%20c-erbB-2%20in%20Liver%20is%20Associat...	Unknown	Mon Dec 02 2...

12. To run a search, type the text to find in the **Search** field and press **Enter**. See the [Supported Search Syntax](#) section below for information about supported search syntax. Anzo finds documents that include the search value and displays the documents, snippets of text to show the context of where the matches were found, and the Elasticsearch relevance score for the match. For information about how the relevance score is calculated, see [What Is Relevance?](#) in the Elasticsearch documentation. For example:

✕ 🔍

Sort by: Relevance ▼ Descending ▼

22 Results

[Growth factor receptors\\_ promising drug targets in cancer.pdf](#)
Relevance 317.6

file:///global/nfs/data/du/docs/cancer/Growth%20factor%20receptors\_%20promising%20drug%20targets%20in%20ca... Unknown
Mon Dec 02 2...

... 12/2/2019 **Growth** factor receptors: promising drug targets in cancer <https://jcmtjournal.com/article/view/1192> 1/24 Home (<https://jcmtjournal.com/>) □ Articles (<https://jcmtjournal.com/journal/archives>) □ Article...

... □ Article Open Access | Review **Growth** factor receptors: promising drug targets in cancer Snigdha Tiash , Ezharul Hoque Chowdhury Jeffrey Cheah School of Medicine and Health Sciences, Faculty of Medicine...

Show More

[Epidermal Growth Factor Receptor \(EGFR\)-targeted Therapies in Esophagogastric Cancer.pdf](#)
Relevance 165.6

file:///global/nfs/data/du/docs/cancer/Epidermal%20Growth%20Factor%20Receptor%20%28EGFR%29-targeted%20... Unknown
Mon Dec 02 2...

... 12/2/2019 Epidermal **Growth** Factor Receptor (EGFR)-targeted Therapies in Esophagogastric Cancer [ar.iiarjournals.org/content/33/10/4139.full](http://ar.iiarjournals.org/content/33/10/4139.full) 1/21 + Anticancer Research [ar.iiarjournals.org](http://ar.iiarjournals.org) Anticancer Research...

...Research October 2013 vol. 33 no. 10 4139-4155 Epidermal **Growth** Factor Receptor (EGFR)- targeted Therapies in Esophagogastric Cancer SABARISH AYYAPPAN 1 , DHIVYA PRABHAKAR 2 and NEELESH SHARMA 1† Author...

Show More

Clicking **Show More** expands the result to display additional matches. For example:

[Growth factor receptors\\_ promising drug targets in cancer.pdf](#)
Relevance 317.6

file:///global/nfs/data/du/docs/cancer/Growth%20factor%20receptors\_%20promising%20drug%20targets%20in%20ca... Unknown
Mon Dec 02 2...

... 12/2/2019 **Growth** factor receptors: promising drug targets in cancer <https://jcmtjournal.com/article/view/1192> 1/24 Home (<https://jcmtjournal.com/>) □ Articles (<https://jcmtjournal.com/journal/archives>) □ Article...

... □ Article Open Access | Review **Growth** factor receptors: promising drug targets in cancer Snigdha Tiash , Ezharul Hoque Chowdhury Jeffrey Cheah School of Medicine and Health Sciences, Faculty of Medicine...

...Cited: 9 (<https://jcmtjournal.com/article/cit e/1192>) Mechanisms of acquired resistance to fibroblast **growth** factor receptor Download PDF (<https://jcmtjournal.com/arti cle/download/1192>) Download XML (<http...>)

...<https://www.linkedin.com/shareArticle?url=https://jcmtjournal.com/article/view/1192> javascript:; 12/2/2019 **Growth** factor receptors: promising drug targets in cancer <https://jcmtjournal.com/article/view/1192> 2/24...

...epigenetic and somatic changes deregulate the expression of **growth** factor receptors (GFRs), leading to cancer initiation and progression. Tumor cell **growth** and survival are orchestrated by clonal expansion and...

...and evasion of apoptotic signals in cancer cells. The **growth** of cells is further supported by angiogenesis and metastasis to distant organs. High expression of GFRs also contributes to the development of...

...targeting GFR have potential therapeutic applications. Keywords Carbonate apatite nanoparticles, **growth** factor receptor, monoclonal antibodies, protein kinase inhibitor, short interfering RNA, tyrosine...

...Cited: 9 (<https://jcmtjournal.com/article/cit e/1192>) Mechanisms of acquired resistance to fibroblast **growth** factor receptor Download PDF (<https://jcmtjournal.com/arti cle/download/1192>) Download XML (<http...>)

...<https://www.linkedin.com/shareArticle?url=https://jcmtjournal.com/article/view/1192> javascript:; 12/2/2019 **Growth** factor receptors: promising drug targets in cancer <https://jcmtjournal.com/article/view/1192> 3/24...

...The heterogeneous nature of cancer is characterized by continuous clonal expansion and uncontrolled **growth** of mutated cells, intravasation and extravasation of blood and lymphatic vessels, dissemination,...

Show Less

- To refine the search, alter the text in the **Search** field and press **Enter** again. You can also click highlighted terms in the search results to open a dialog box that shows the full annotated document

where the match was found. For example:



## Supported Search Syntax

This section describes the keyword search syntax that Anzo supports.

### Wildcard Characters: ? and \*

- **?:** Use a question mark (?) to represent a single wildcard character. For example, in the search **co?l**, the resulting documents will include terms like "cool" or "coal."
- **\***: Use an asterisk (\*) to represent multiple wildcard characters. For example, in the search **col\***, the resulting documents will include terms like "collect" or "color."

### Boolean Operators: +, -, OR, AND, NOT

- **+**: Use a plus (+) character to indicate mandatory matches. For example, in the search **flight +New York**, the resulting documents can include "flight" as an optional match and must include "New York."
- **-**: Use a minus (-) character to indicate a term that must not match. For example, in the search **flight +New York -Los Angeles**, the resulting documents can include "flight" as an optional match, must include "New York," and must not include "Los Angeles."
- **OR**: In the search **New York OR Los Angeles**, the resulting documents will include a match for either "New York" or "Los Angeles."
- **AND**: In the search **New York AND Los Angeles**, the resulting documents must include matches for both "New York" and "Los Angeles."

- **NOT:** In the search **New York NOT Los Angeles**, the resulting documents must include "New York" and cannot contain "Los Angeles."
- **Grouping operators:** In the search **(flight AND New York) OR Los Angeles**, the resulting documents will include "flight" and "New York" and optionally include "Los Angeles."

### Fuzzy Matches: `~n`

To search for a fuzzy match, use a tilde (~) character followed by a number to represent the number of fuzzy or incorrect characters. For example, in the search **Flgth~3**, the resulting documents could include the term "Flight."

### Regular Expressions

For example, the following search expression matches email addresses: `/([a-zA-Z0-9_\-\.]+)@([a-zA-Z0-9_\-\.]+)\.([a-zA-Z]{2,5})/`.

For more information about the regular expression syntax that Elasticsearch supports, see [Regular expression syntax](#) in the Elasticsearch documentation.

### Related Topics

[Creating Lenses](#)



# Sharing Access to a Dashboard or Lens

This topic includes reference information about dashboard and lens permissions and provides instructions for configuring permissions.

- [Dashboard Level Permissions Reference](#)
- [Lens Level Permissions Reference](#)
- [Configuring Dashboard or Lens Permissions](#)

## Dashboard Level Permissions Reference

Dashboard level permissions affect a user's ability to view, modify, delete, design, or configure dashboards and dashboard permissions. There are three predefined permission sets that can be assigned to an Anzo user or group. You also have the option to customize the set of permissions that are applied to a user or group.

The table below lists the predefined permission sets and describes the privileges that are granted for each permission that is part of the predefined set:

Set	Permission	Allows a user to:
View	Read	• Search for and open accessible dashboards.
		• Save As a new dashboard.
		• Share the dashboard.
		• View dashboard Properties.
		• View lens Properties.
Modify	In addition to the <b>Read</b> permission described above, the <b>Modify</b> set includes the <b>Write</b> and <b>Delete</b> permissions described below.	
	Write	• Use the dashboard Designer to change the dashboard.
	Delete	• Clone lenses.
		• Delete the dashboard.

Set	Permission	Allows a user to:
In addition to the <b>Read</b> , <b>Write</b> , and <b>Delete</b> permissions described above, the <b>Admin</b> set includes the <b>Manage</b> permission described below.		
<b>Admin</b>	<b>Manage</b>	<ul style="list-style-type: none"> <li>The Manage permission relates only to the Security tab. If a user has this permission, they can modify dashboard access by changing permissions for a user, group, or role.</li> </ul>

## Default Dashboard Permissions

The table below lists the predefined permission sets that are applied by default when a new dashboard is created. Besides the sysadmin user, the dashboard creator is granted **Admin** privileges by default. The Everyone role is granted **View** privileges by default. No other users, groups, or roles have dashboard permissions assigned by default.

Anzo User/Role	Applied Permission Set
Sysadmin User	Admin
Dashboard Creator	Admin
Everyone Role	View

## Lens Level Permissions Reference

Lens level permissions affect a user's ability to view, modify, delete, design, or configure lenses and lens permissions. There are three predefined lens permission sets that can be assigned to an Anzo user or group. You also have the option to customize the set of permissions that are applied. While dashboard level permissions can affect a user's ability to clone a lens, the appropriate lens level permissions are required to be able to perform functions such as deleting or redesigning a lens.

The table below lists the predefined permission sets and describes the privileges that are granted for each permission that is part of the predefined set:

Set	Permission	Allows a user to:
<b>View</b>	<b>Read</b>	<ul style="list-style-type: none"> <li>• Search for and open accessible lenses.</li> <li>• View lens Properties.</li> <li>• Export lenses.</li> </ul>
	In addition to the <b>Read</b> permission described above, the <b>Modify</b> set includes the <b>Write</b> and <b>Delete</b> permissions described below.	
	<b>Write</b>	<ul style="list-style-type: none"> <li>• Use the lens Designer to change the lens.</li> <li>• Rename the lens.</li> <li>• Clone the lens.</li> </ul>
<b>Modify</b>	<b>Delete</b>	<ul style="list-style-type: none"> <li>• Delete the lens.</li> </ul>
	In addition to the <b>Read</b> , <b>Write</b> , and <b>Delete</b> permissions described above, the <b>Admin</b> set includes the <b>Manage</b> permission described below.	
<b>Admin</b>	<b>Manage</b>	<ul style="list-style-type: none"> <li>• The Manage permission relates only to the Security tab. If a user has this permission, they can modify lens access by changing permissions for a user, group, or role.</li> </ul>

## Default Lens Permissions

The table below lists the predefined permission sets that are applied by default when a new lens is created. Besides the sysadmin user, the lens creator is granted **Admin** privileges by default. The Everyone role is granted **View** privileges by default. No other users, groups, or roles have lens permissions assigned by default.

Anzo User/Role	Applied Permission Set
Sysadmin User	Admin
Lens Creator	Admin

Anzo User/Role	Applied Permission Set
Everyone Role	View

## Configuring Dashboard or Lens Permissions

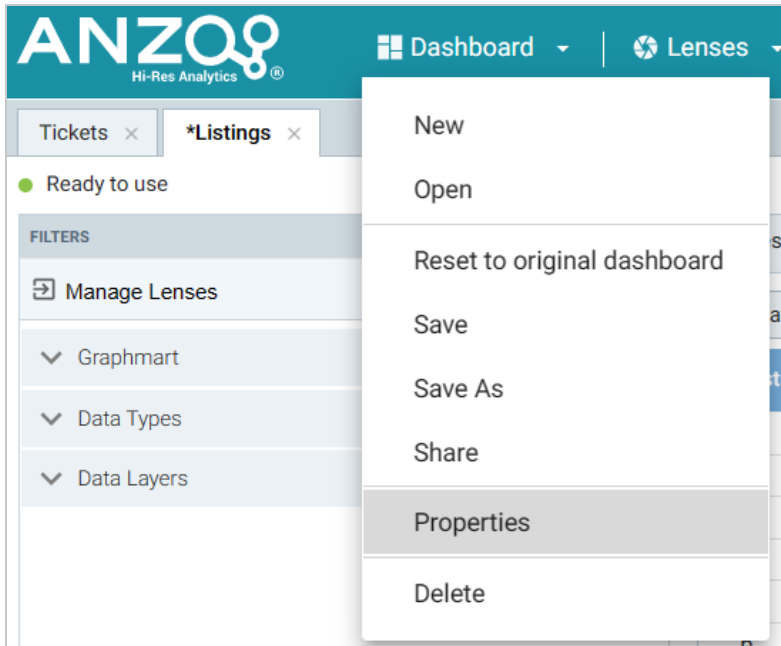
This section provides instructions for modifying dashboard or lens properties to grant or restrict access to your dashboards and lenses.

### Note

Data can be restricted at a higher level than a dashboard. Though users might have access to view your dashboards and lenses, graphmart permissions determine whether they can view the data that the dashboard displays.

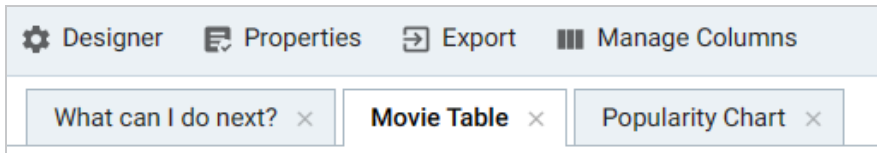
1. Open the dashboard for which you want to modify access.
2. Open the Properties dialog box for the either dashboard or for a specific lens:
  - To change access at the dashboard level, click **Dashboard** in the main toolbar and select **Properties**.

**Note** Sharing a dashboard automatically shares the lenses in that dashboard.

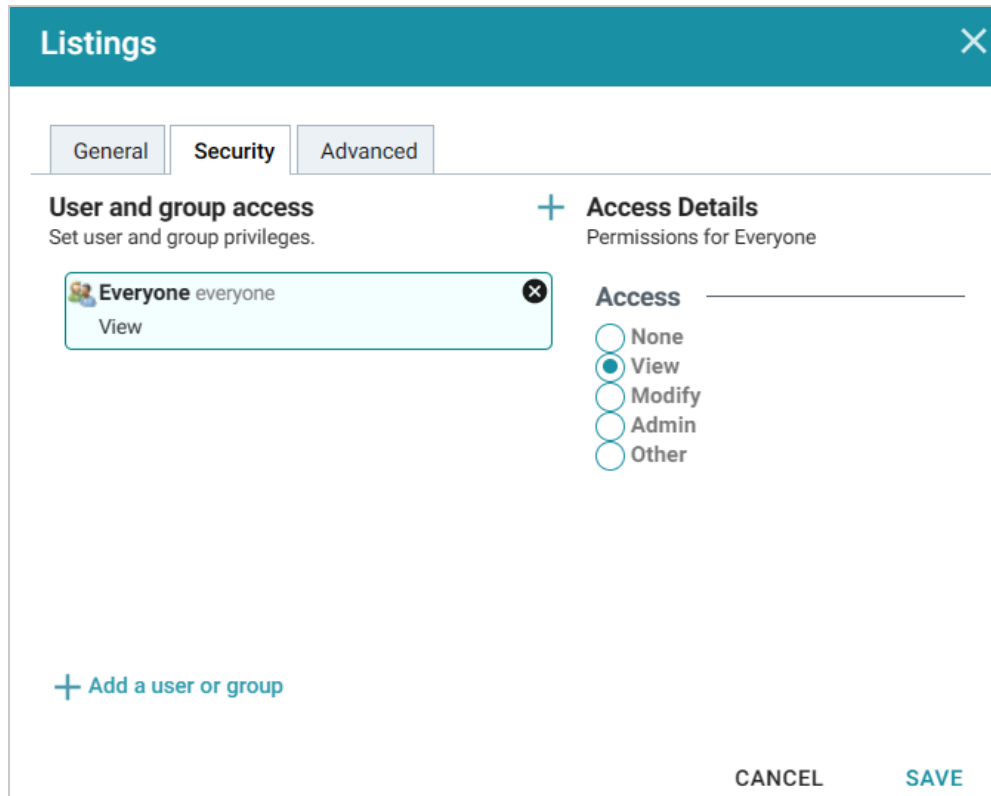


- To change access for a lens in the dashboard, click the lens to display it and then click the

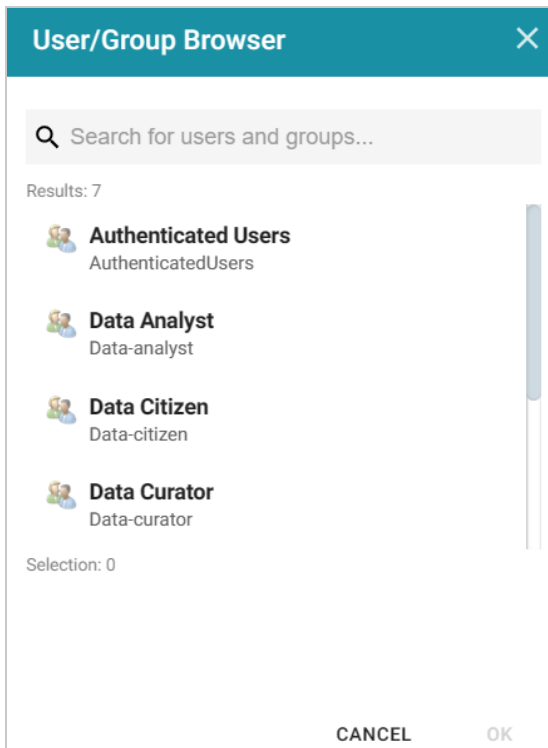
**Properties** button in the object toolbar.



3. In the Properties dialog box, click the **Security** tab. This tab lists the available groups and users who can view this dashboard or lens. The image below shows the Security tab at the dashboard level:



4. Select a user or user group to manage, and then modify any of the following options:
  - **Remove a user or group:** Click the delete icon (X) next to the user or group.
  - **Add a user or group:** Click **Add a user or group**. On the User/Group Browser dialog box, select the users or groups that you want to add. Then click **OK**.



- **Access Details:** Select the access level for the selected user or group. Refer to [Dashboard Level Permissions Reference](#) or [Lens Level Permissions Reference](#) above for details about each of the access options.
  - **None:** No permissions set for the selected dashboard or lens.
  - **View:** Grants the **View** predefined permission set for the selected dashboard or lens.
  - **Modify:** Grants the **Modify** predefined permission set for the selected dashboard or lens.
  - **Admin:** Grants the **Admin** predefined permission set for the selected dashboard or lens.
  - **Other:** Enables you to set custom access levels for the selected dashboard or lens. Select the checkboxes to enable any combination of the following permissions: Read, Write, Delete, or Manage (administrator permissions).

5. Click **Save** to save the changes.

To get a URL to your dashboard that you can send to users, click **Dashboard** in the main toolbar and select **Share**. The Share Dashboard dialog box opens and displays a URL for the dashboard. You can copy the link and send it to users.

## Related Topics

[Creating Dashboards](#)

[Creating Lenses](#)

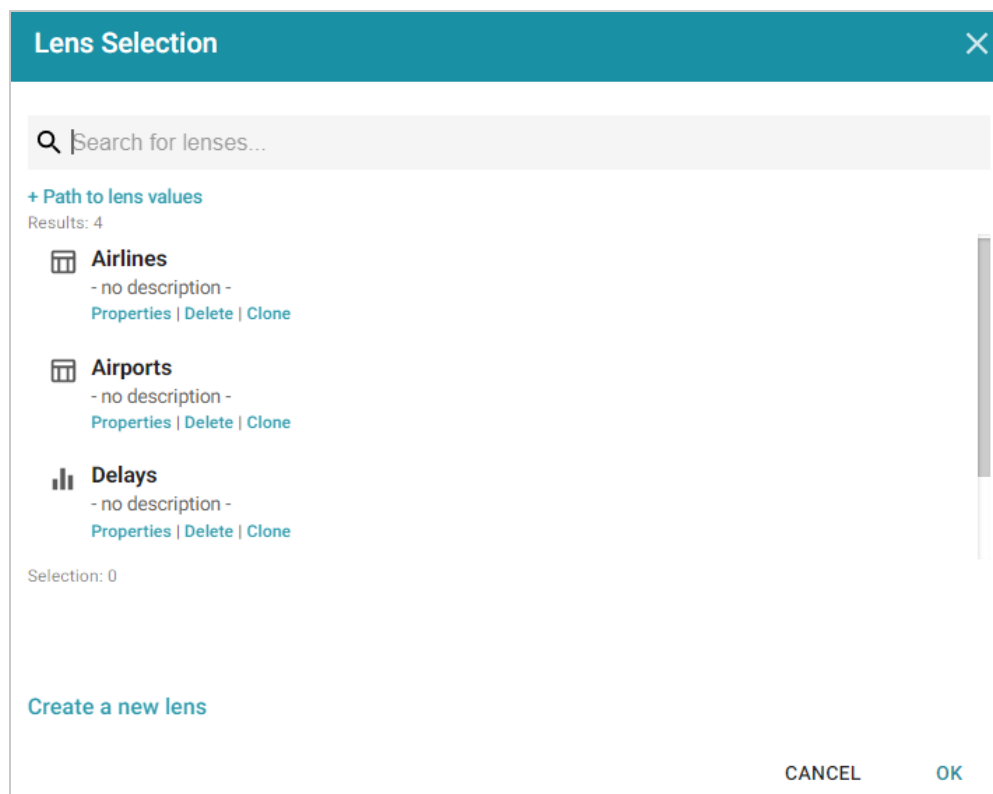
## Cloning a Lens

Cloning a lens makes a copy of the lens that can be changed without affecting the original lens or other dashboards. Follow the steps below to clone a lens.

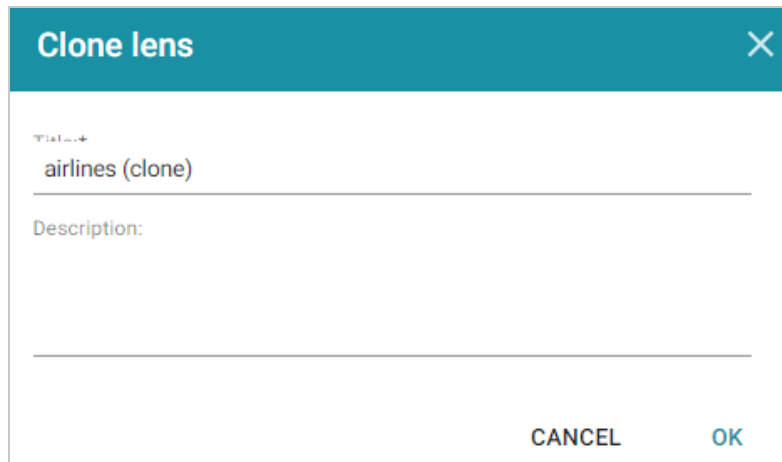
### Note

You can only clone lenses from dashboards that you have permission to modify. If you open a dashboard with read-only access, the Open Lens and Clone options are not available. To clone a lens from a read-only dashboard, save a copy of the dashboard so that you become the owner. To save a copy, click the **Dashboard** button in the main Hi-Res Analytics toolbar and select **Save As**. Then follow the procedure below to clone a lens into the dashboard that you own.

1. Open a dashboard in the Hi-Res Analytics application, then click **Lenses** in the main toolbar and select **Open**. Anzo opens the Lens Selection dialog box, which lists the lenses that are available to open. For example:



2. Click the **Clone** link for the lens that you want to clone. Anzo displays the Clone lens dialog box, and populates the Title field with the existing lens name and "(clone)." For example:

A dialog box titled "Clone lens" with a teal header bar containing a close button (X). The main area is white and contains a "Title:" label followed by a text input field containing "airlines (clone)". Below this is a "Description:" label followed by a larger text input field. At the bottom right, there are two buttons: "CANCEL" and "OK".

Clone lens

Title: airlines (clone)

Description:

CANCEL OK

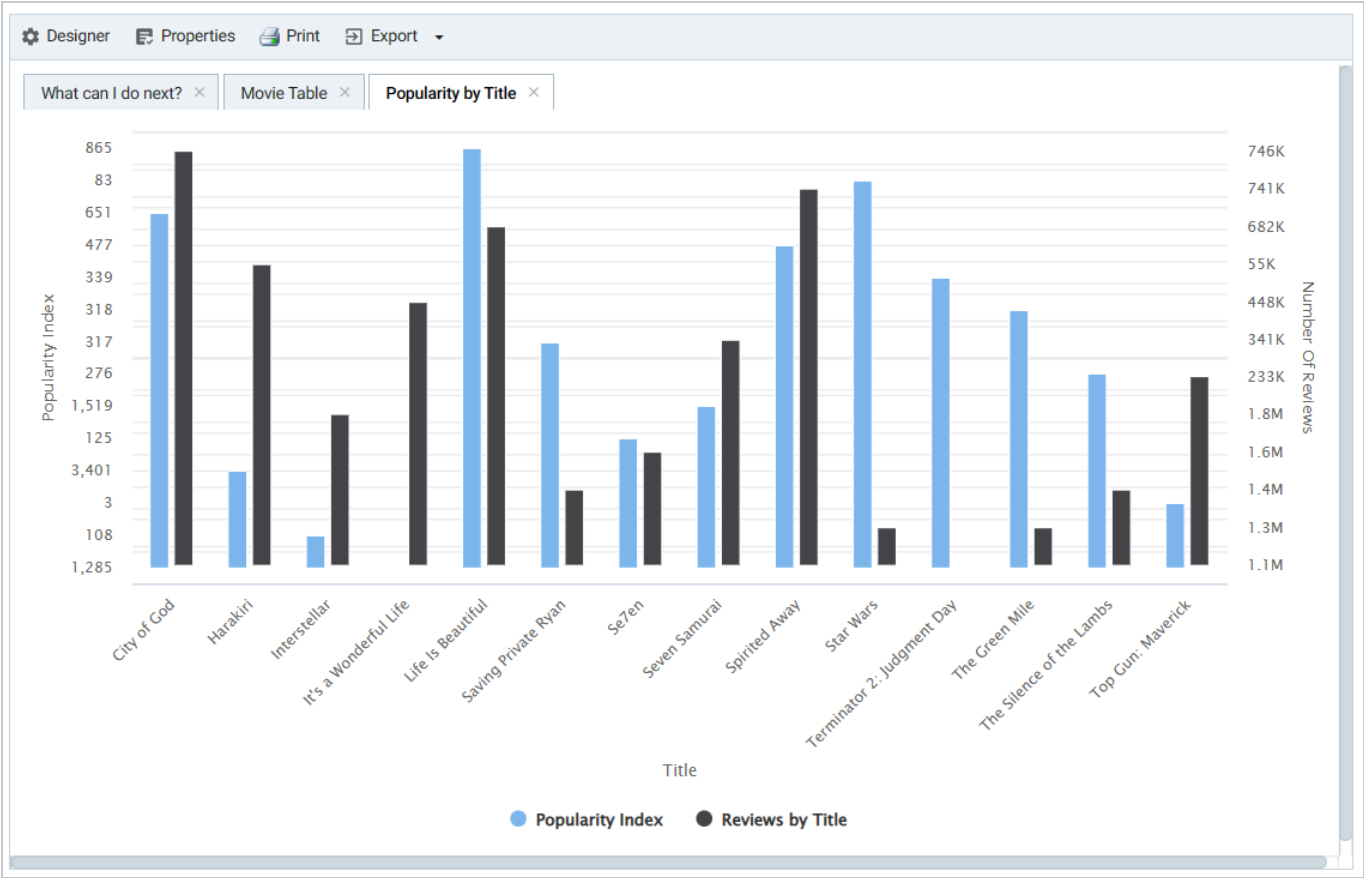
3. Modify the **Title** to name the new copy of the lens, and add or change the **Description** if necessary. Then click **OK**.
4. Anzo adds the new copy of the lens to the Lens Selection dialog box and selects it. Click **OK** to add the lens to the dashboard.



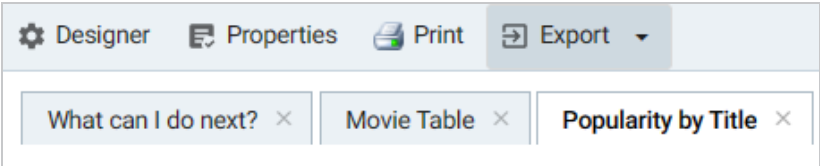
# Exporting a Lens

If you have dashboards with Table, Chart, and/or Query lenses, you can export those lenses from the Hi-Res Analytics application. Charts can be exported as images in JPEG, PNG, SVG, or PDF format. Tables can be exported to CSV or JSON files. And Query lenses can be exported to CSV files. Follow the instructions below to export a lens.

- 1. Open the dashboard that contains the lens that you want to export.
- 2. If necessary click the tab for the lens to make it active. For example, the image below shows a chart lens.



- 3. In the object toolbar for the lens, click the **Export** button.



For Query lenses, adjust the Export options as needed, and then click **Export Results**. For example:

Sort by: Value ▼
Forward ▼
Offset: 0
Limit: 100
Export Results

- If the lens is a chart, select the one of the image types from the drop-down list. Anzo creates the image as that type and downloads the file to your computer.
- If the lens is a table, the Export Options dialog box is displayed:

Export Options ×

File name:

Format:  

CSV

Multi Valued Column Separator:  

[pipe]

☒
Export Headers

Columns to Export:

Description	<input checked="" type="checkbox"/> ON
Director	<input checked="" type="checkbox"/> ON
Movie Cast	<input checked="" type="checkbox"/> ON
Number Of Reviews	<input checked="" type="checkbox"/> ON
Popularity Index	<input checked="" type="checkbox"/> ON
Rating	<input checked="" type="checkbox"/> ON
Title	<input checked="" type="checkbox"/> ON

CANCEL
OK

- In the Export Options dialog box, specify the following file options:
  - File name:** Specify a name for the file. Do not specify the file type extension.
  - Format:** Click the Format field and select **CSV** to create a .csv file or **JSON** to create a .json file.
  - Multi Valued Column Separator:** For CSV files, click this column to select the character to use as a separator in the file. This option does not apply to JSON files.
  - Export Headers:** Indicates whether to include column headers in the file. Clear the checkbox to exclude headers from the file. This option does not apply to JSON files.

- **Columns to Export:** By default, all columns are selected for export. You can clear the checkboxes next to columns that you do not want to include. In addition, you can reorder columns by dragging and dropping the rows.

7. Click **OK** to download the file to your computer.

## Deleting a Lens

This topic provides information about the permissions that are required for deleting lenses as well as instructions for deleting a lens.

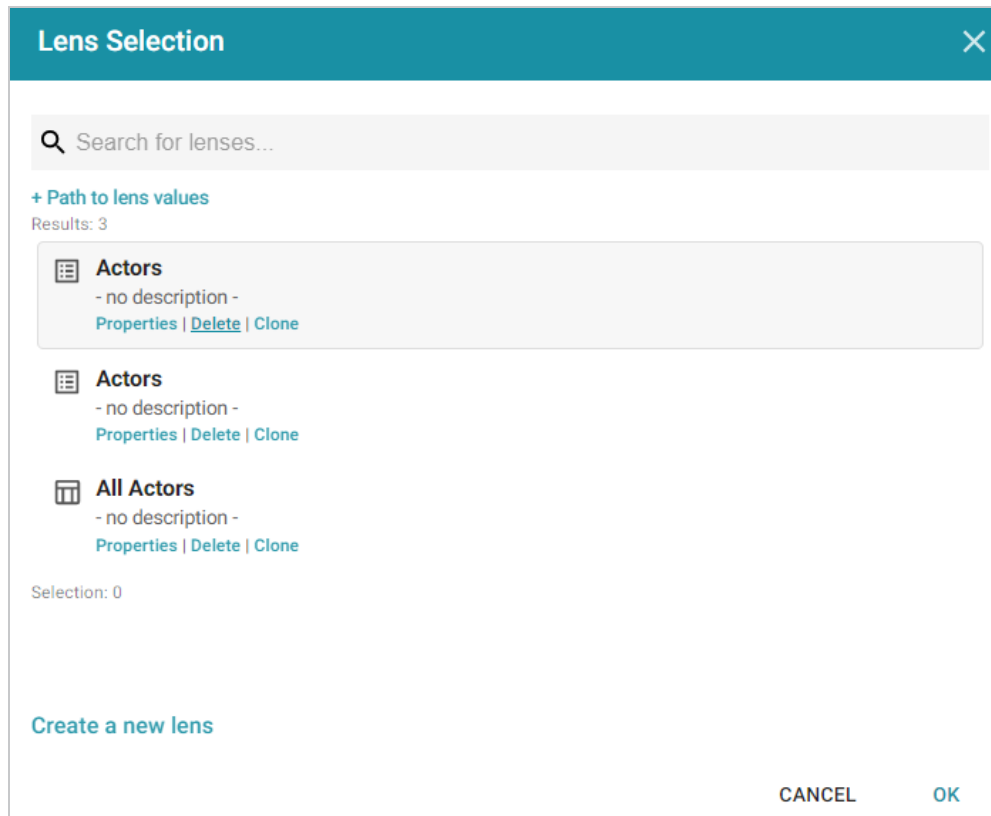
### Required Permissions for Deleting Lenses

By default, only the **sysadmin** user and **lens creator** have permission to delete a lens. To delete a lens, a user must have the **Manage** permission assigned for that lens. The Manage permission is included in the **Admin** predefined lens permission set. If lens permissions have not been changed since the lens was created, the sysadmin user and the lens creator are the only users who have permission to delete that lens. The Manage permission is also required to change lens security settings and grant privileges to other users. Users who have read access to a lens (granted through the View, Modify, or Admin lens permission sets) can view the lens security settings to identify which non-sysadmin users have permission to delete the lens. For more information, see [Sharing Access to a Dashboard or Lens](#).

### Deleting a Lens

Follow the steps below to delete a lens.

1. Open a dashboard that contains the lens that you want to delete.
2. In the Hi-Res Analytics application, click the **Lenses** menu in the main toolbar and select **Open**. The Lens Selection dialog box is displayed.
3. In the Lens Selection dialog box, find the lens that you want to delete and then click the **Delete** link for that lens. For example:



4. The application presents a confirmation message. Click **Yes** to delete the lens. The lens is removed from the Lens Selection dialog box, and you can repeat this process to delete additional lenses for which you have the required privileges.

# Accessing Data with the Query Builder

The Query Builder in the user interface provides options for accessing data in various data sources. The Query Builder includes a **Find** option that enables users to search for quads by specifying a single subject, object, predicate, or graph name. It also includes a **Query** option that enables users to write, run, and save SPARQL queries. The topics in this section provide information about accessing data using the Query Builder.

- Running SPARQL Queries in the Query Builder .....907
- Searching for Quads in the Query Builder .....916

## Running SPARQL Queries in the Query Builder

The Query Builder includes a Query tab for writing and running SPARQL queries. The query editor provides syntax assistance, type-ahead suggestions for model entity names, and automated prefix creation and query formatting for readability. It also includes the option to save queries for later use.

The Query tab supports running queries against the following data sources:

- Graphmarts and specific data layers within graphmarts
- Linked Datasets
- Data sources: Anzo System Data Source, AnzoGraph, Anzo System Tables, Data Profiling Metrics, LDAP Primary Data Source

### Note

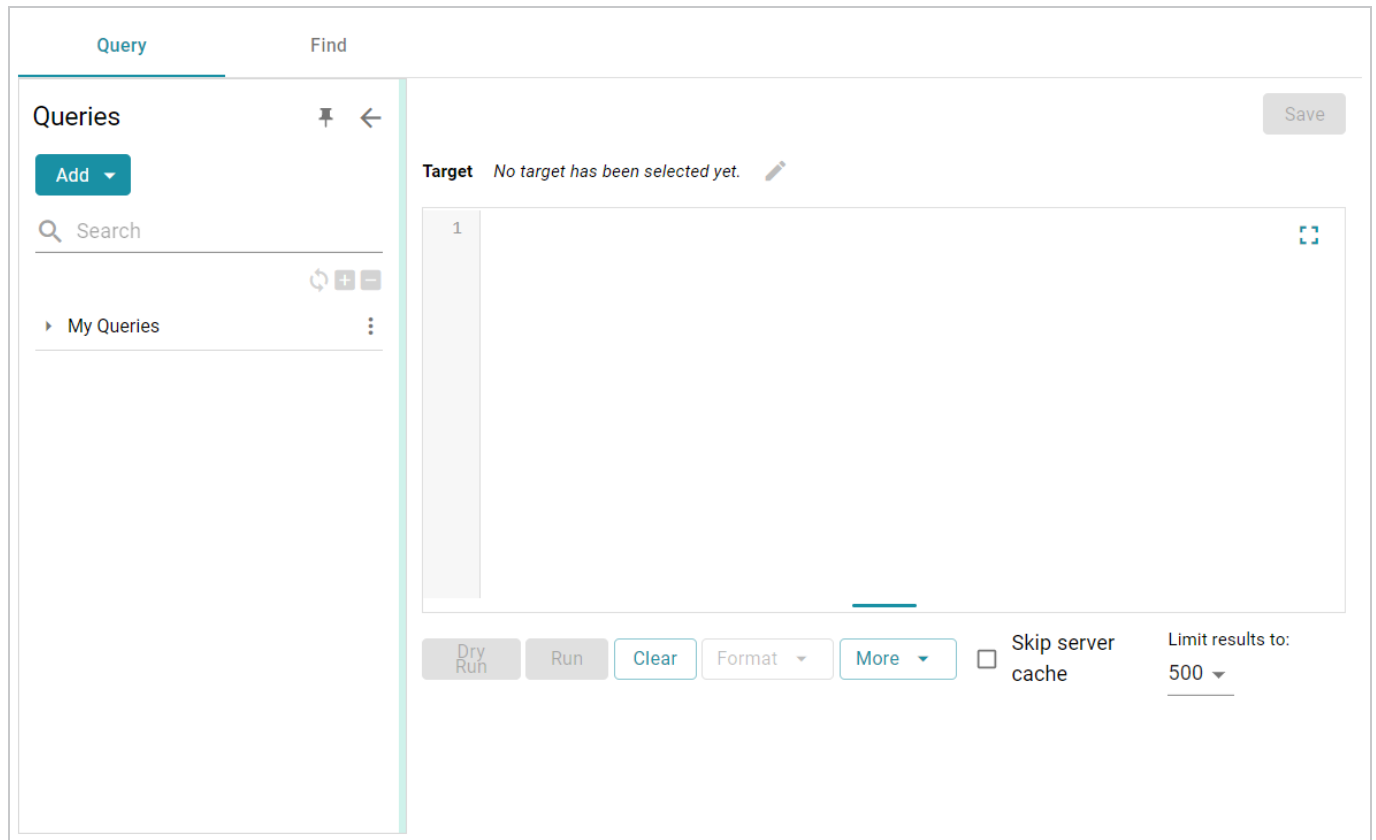
To ensure that queries perform well and do not consume too many resources on the system, keep the following guidelines in mind when developing and testing queries:

- Set a limit on the number of results to return.
- Avoid cross-product joins
- Consider using VALUES clauses instead of FILTER clauses.
- When retrieving a large number of values, use subqueries instead of OPTIONAL clauses.

For query templates and additional details about best practices, see [SPARQL Best Practices and Query Templates](#).

Follow the instructions below to write and run SPARQL queries against any of the supported data sources.

1. In the Anzo application, expand the **Access** menu and click **Query Builder**. Anzo displays the query editor.



2. At the top of the screen, click the edit (✎) icon next to the Target data source:
  - To query data that is in a graphmart, select the **Graphmart** radio button.

☒ Graphmart ☐ Linked Dataset ☐ Datasource

Select Graphmart ▼

Select Layers ▼

Click the **Select Graphmart** drop-down list and select the graphmart to query. If you want to narrow the scope of the query by selecting one or more data layers in the graphmart, click the **Select Layers** drop-down list and select the layer or layers to target.



- To query data that is in a linked dataset, select the **Linked Dataset** radio button.

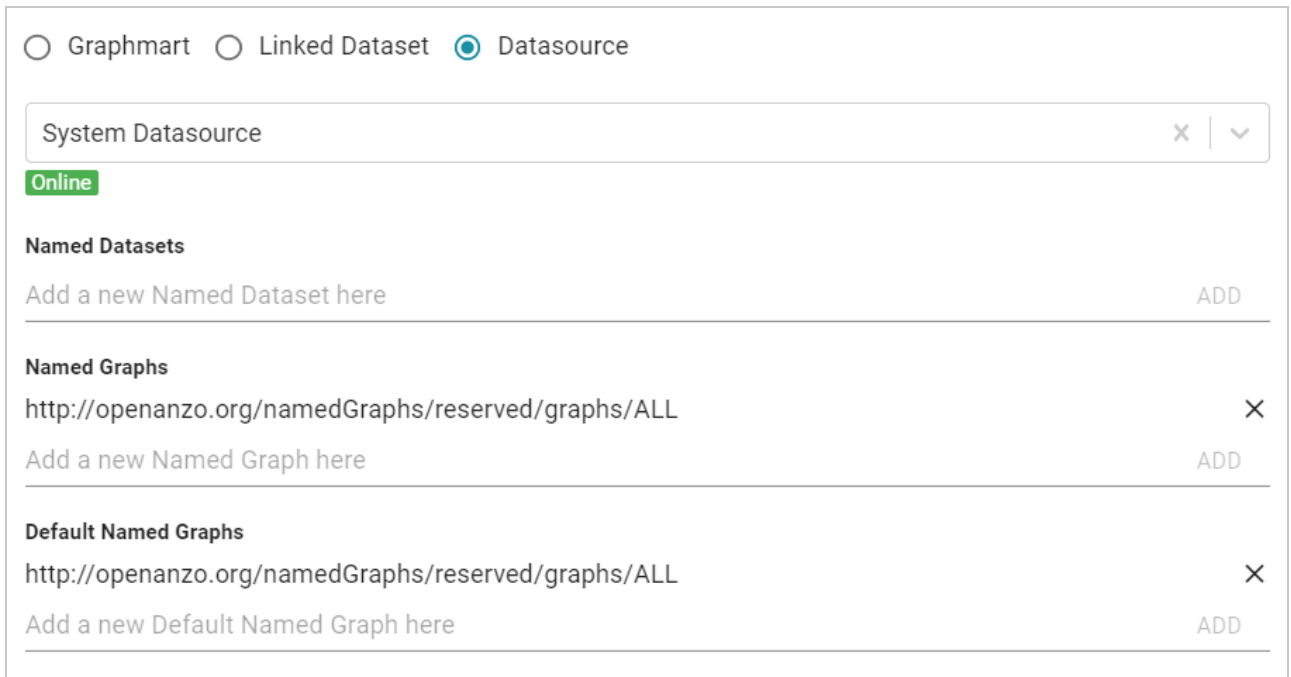


☐ Graphmart 
 ☒ Linked Dataset 
 ☐ Datasource

Select linked dataset ▼

Click the **Select linked dataset** drop-down list and select the linked dataset to query.

- To run queries against the system data source, data profiling metrics, Anzo system tables, LDAP server, or AnzoGraph, select the **Datasource** radio button.



☐ Graphmart 
 ☐ Linked Dataset 
 ☒ Datasource

System Datasource x ▼

**Online**

**Named Datasets**

Add a new Named Dataset here ADD

**Named Graphs**

http://openanzo.org/namedGraphs/reserved/graphs/ALL X

Add a new Named Graph here ADD

**Default Named Graphs**

http://openanzo.org/namedGraphs/reserved/graphs/ALL X

Add a new Default Named Graph here ADD

Click the **Datasource** drop-down list and select the target source:

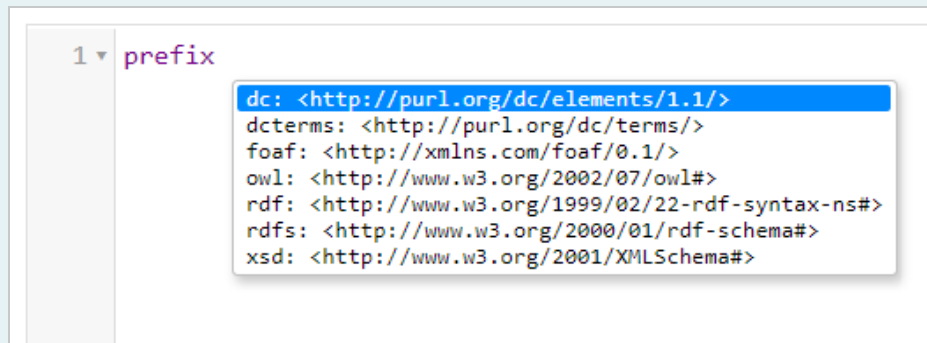
- Select **System Datasource** to search the local Anzo system volume.
- Select the name of an AnzoGraph instance to search for data in graphmarts that are loaded to that instance.
- Select **Data Profiling Metrics** to search the data metrics volume.
- Select **LDAP Primary Datasource** to search the directory server.
- Select **System Tables** to search Anzo system table data.

By default, the Named Graphs and Default Named Graphs values are set to all named graphs (<http://openanzo.org/namedGraphs/reserved/graphs/ALL>). If you want to narrow the scope of the query, you can replace the values with specific graph URIs. To list multiple graphs, separate URIs with a space.

3. When you have finished configuring the target, click outside of the dialog box to return to the Query screen.
4. In the text box below the target, compose the SPARQL query. See [SPARQL Best Practices and Query Templates](#) for tips on writing queries. For information about the supported SPARQL functions, see [Anzo Function and Formula Reference](#).

#### Tip

When adding PREFIX statements, once you type **prefix** followed by a space Anzo displays a tooltip that lists all of the global prefixes that are defined for your system. Clicking a prefix in the list inserts a PREFIX statement into the query. For example:



In addition, typing the abbreviation for a global prefix followed by a colon (:) automatically inserts the PREFIX statement into the query without opening the tooltip.

When typing entity URIs in the WHERE clause, the query builder also offers suggestions by listing the properties in the data source. You can click an item in the list to insert that entity. For example:

```

1 select *
2 where { ?s <

```

dc:  
dcterms:  
foaf:  
owl:  
rdf:  
rdfs:  
xsd:  
<http://cambridgesemantics.com/Layer/c2e76e3528d74e8da01c1d4f3e217dd2/Model#MovieActors.actorID>  
<http://cambridgesemantics.com/Layer/c2e76e3528d74e8da01c1d4f3e217dd2/Model#MovieActors.actorName>  
<http://cambridgesemantics.com/Layer/c2e76e3528d74e8da01c1d4f3e217dd2/Model#MovieActors.movieTitle>  
<http://cambridgesemantics.com/Layer/c2e76e3528d74e8da01c1d4f3e217dd2/Model#MovieComposers.movieComposerID>  
<http://cambridgesemantics.com/Layer/c2e76e3528d74e8da01c1d4f3e217dd2/Model#MovieComposers.movieComposerName>  
<http://cambridgesemantics.com/Layer/c2e76e3528d74e8da01c1d4f3e217dd2/Model#MovieComposers.movieTitle>  
<http://cambridgesemantics.com/Layer/c2e76e3528d74e8da01c1d4f3e217dd2/Model#MovieDirectors.directorName>  
<http://cambridgesemantics.com/Layer/c2e76e3528d74e8da01c1d4f3e217dd2/Model#MovieDirectors.movieDirectorID>  
<http://cambridgesemantics.com/Layer/c2e76e3528d74e8da01c1d4f3e217dd2/Model#MovieDirectors.movieTitle>  
<http://cambridgesemantics.com/Layer/c2e76e3528d74e8da01c1d4f3e217dd2/Model#MovieProducers.movieProducerID>

When a red exclamation mark icon (❗) is displayed next to a line number, you can hover the pointer over the icon to view guidance on how to resolve the issue. For example:

```

❗ 1 select count ?title

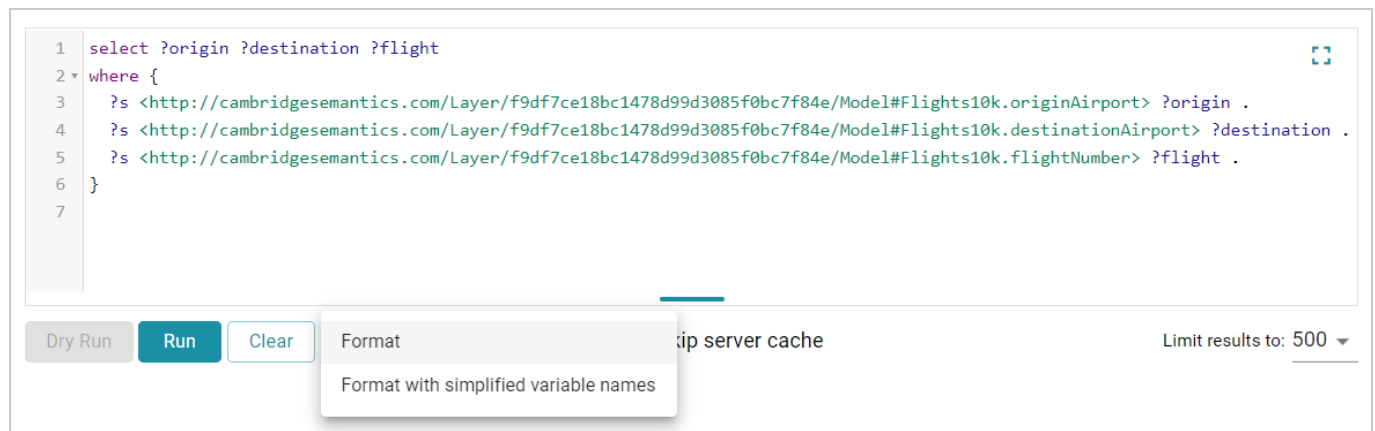
```

This line is invalid.  
Expected: DISTINCT,  
REDUCED, \*, (, VAR1,  
VAR2, TEMPLATE

5. If you want to format the query for readability, click the **Format** button and select from the following options:

- **Format:** Auto-creates prefixes, inserts URI abbreviations, and restructures the query for readability.
- **Format with simplified variable names:** Auto-creates prefixes, inserts URI abbreviations, simplifies variable names by changing them to ?\_var1, ?\_var2, ?\_varN, and restructures the query for readability.

For example, the image below shows a query before it is formatted.



After the query is formatted, prefixes and URI abbreviations are added. For example:



6. If the query is an INSERT or DELETE query, the Dry Run button becomes active. You can click **Dry Run** to do a test run of the update. In a test run, Anzo runs a version of the query where INSERT or DELETE is replaced with CONSTRUCT, and the results report the number of statements that the query affects, i.e., the number of additions or removals per graph. If the results are unexpected, you can adjust the query before clicking running the query and committing the updates.
7. If necessary, change the query limit. By default, query results are limited to 500. To adjust the limit, click the **Limit results to** drop-down list below the query editor and select a value. For example:

```
1 PREFIX Model: <http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Model#>
2
3 SELECT
4     ?origin
5     ?destination
6     ?flight
7
8 WHERE {
9     ?s Model:Flights10k\.originAirport ?origin .
10    ?s Model:Flights10k\.destinationAirport ?destination .
11    ?s Model:Flights10k\.flightNumber ?flight .
12 }
```

Dry Run

Run

Clear

Format

More

☐ Skip server cache

Limit results to:


100

500

1000

10000

8. To run the query, click **Run**. The results appear at the bottom of the screen. For example:

Results (100) Executed in 0.367 s		<input checked="" type="checkbox"/> Use prefixes	<input checked="" type="checkbox"/> Show native	Clear	Download	
origin	destination	flight	distance	airline		
LAX	PBI	2336	2330	AA		
SEA	ANC	135	1448	AS		
PDX	MSP	1670	1426	DL		
LAS	DFW	214	1055	NK		
LAS	IAH	298	1222	NK		
ANC	SEA	134	1448	AS		
SJU	JFK	304	1598	B6		
IAG	FLL	647	1176	NK		
BQN	EWR	1162	1585	UA		
BQN	MCO	1030	1129	B6		
SJU	MCO	2134	1189	B6		
ORD	PHX	602	1440	US		
GEG	SEA	695	224	AS		
HNL	ITO	102	216	HA		

#### Tip

You can click any value in the result list to copy that value to the clipboard.

9. To save the query for later use, click the **Save** button at the top of the screen. Anzo displays the New Query dialog box.

**New Query**

Title \*

Description

Folder

My Queries

x | v

Either choose a folder or create a new one (type a new name).

CANCEL

SAVE

10. In the dialog box, specify a name for the query in the **Title** field and an optional description in the **Description** field.
11. You can click the **Folder** drop-down list to select a different folder or create a new one to save the query in. Then click **Save**. The query is saved and added to the **Queries** panel on the left side of the screen.

## Related Topics

[Searching for Quads in the Query Builder](#)

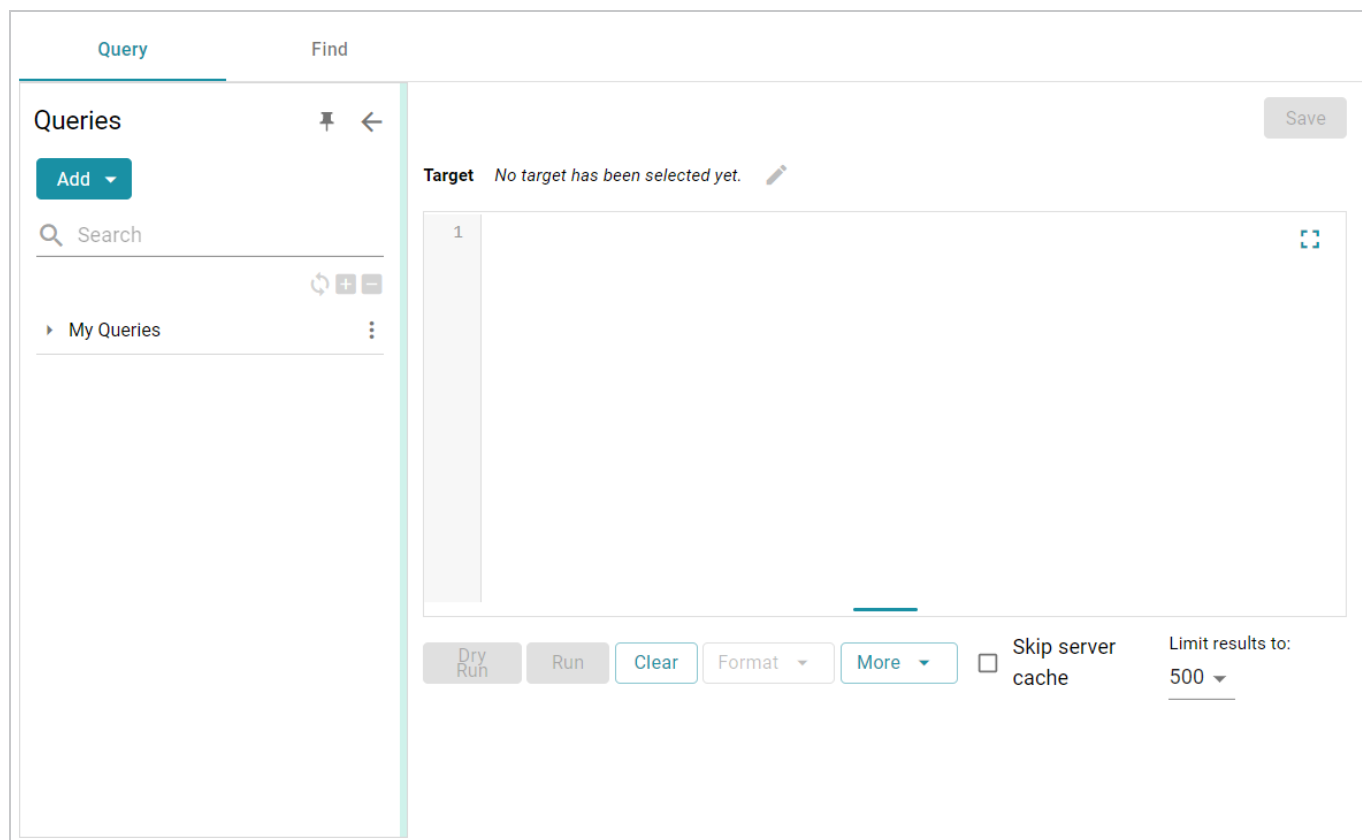
## Searching for Quads in the Query Builder

The Query Builder includes a Find tab for searching for data by specifying a single subject, object, predicate, graph name or any combination of those elements. Statements that match the search criteria are returned in quads, and the screen includes quick filters that enable users to toggle filters on and off to show or hide any of the quad elements. The Find tab supports searches against the following data sources:

- Anzo System Data Source
- AnzoGraph
- Anzo System Tables
- Data Profiling Metrics
- LDAP Primary Data Source

When finding data in the system data source, users have the option to modify or delete statements directly in the user interface. Follow the instructions below to find data in any of the supported data sources.

1. In the Anzo application, expand the **Access** menu and click **Query Builder**. Anzo displays the query editor.





2. Click the **Find** tab.

The screenshot shows the 'Find' tab in the Anzo Query Builder. At the top, there are two tabs: 'Query' and 'Find', with 'Find' being the active tab. Below the tabs, there is a section labeled 'Select a datasource:' with a dropdown menu currently showing 'System Datasource'. Below this, there are four input fields labeled 'Subject', 'Predicate', 'Object', and 'Graph'. At the bottom right of the interface, there are three buttons: 'Clear', 'Add Statements', and 'Find'.

3. Click the **Select a Datasource** drop-down list and select the data source that you want to search.
- Select **System Datasource** to search the local Anzo system volume.
  - Select the name of an AnzoGraph instance to search for data in graphmarts that are loaded to that instance.
  - Select **Data Profiling Metrics** to search the data profiling metrics volume.
  - Select **LDAP Primary Datasource** to search the directory server.
  - Select **System Tables** to search Anzo system table data.
4. Follow the guidelines below to specify the data to find in the data source:
- Specify any subject, predicate, object, or graph name in the appropriate field. You can specify a value for one field in the quad or any combination of fields.
  - Any URIs and/or literal values that you specify must match the value in the data. Partial values, wildcard characters, and regular expressions are not supported.
  - If you want to get a list of all of the statements in the data source, you can leave all of the fields blank.
5. Click **Find** to search for the statements that match the search criteria. Anzo displays the matching statements. For example:

Subject

Predicate

Object

Graph

Clear

Add Statements

Find

Results (50)

☒ Show native

☒ Subject

☒ Predicate

☒ Object

☒ Named Graph

EDIT

DELETE

<input type="checkbox"/>	Subject ↓	Predicate	Object	Named Graph
<input type="checkbox"/>	« http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Flights10k/0dWi2lJAKiRsnW1sY190ie/0000_00000/0008610 »	« http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Model#Flights10k.destinationAirport »	« BOS »	http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e
<input type="checkbox"/>	« http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Flights10k/0dWi2lJAKiRsnW1sY190ie/0000_00000/0008656 »	« http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Model#Flights10k.originAirport »	« BOS »	http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e
<input type="checkbox"/>	« http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Flights10k/0dWi2lJAKiRsnW1sY190ie/0000_00000/0008681 »	« http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Model#Flights10k.originAirport »	« BOS »	http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e
<input type="checkbox"/>	« http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Flights10k/0dWi2lJAKiRsnW1sY190ie/0000_00000/0008682 »	« http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Model#Flights10k.originAirport »	« BOS »	http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e
	« http://cambridgesemantics.com/La	« http://cambridgesemantics.co		http://cambridgeseman

6. The following options are available for working with the results:
- To filter results by showing or hiding parts of the quads in the statements, you can select or clear the following checkboxes above the results.

☒ Show native
 ☒ Subject
 ☒ Predicate
 ☒ Object
 ☒ Named Graph

Clearing a checkbox hides that part of the quad in the result list. You can display the item again by selecting the checkbox.

- To modify the search parameters, you can click any of the graph, subject, predicate, or object values in the results. The search is automatically run again using only the value that you clicked.
- If the source that you searched is the **System Datasource**, you can edit, delete, or add statements directly. See [System Datasource Options](#) below for details.

## System Datasource Options

This section provides information about editing, deleting, and adding statements on the Find screen.


## Note

Though the options described below are available for all data sources, adding, deleting, or editing statements is only successful when the data source is **System Datasource**.

- [Editing a Statement](#)
- [Deleting a Statement](#)
- [Adding a Statement](#)

## Editing a Statement

To edit a statement, click the menu icon (⋮) to the right of the statement and select **Edit**.

Results (50)		<input checked="" type="checkbox"/> Show native	<input checked="" type="checkbox"/> Subject	<input checked="" type="checkbox"/> Predicate	<input checked="" type="checkbox"/> Object	<input checked="" type="checkbox"/> Named Graph	
EDIT	DELETE						
<input type="checkbox"/>	Subject ↓	Predicate	Object	Named Graph			
<input type="checkbox"/>	« http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Flights10k/0dWi2lJAKiRsnW1sY190ie/0000_00000/0008610 »	« http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Model#Flights10k.destinationAirport »	« BOS »	http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e	<div>Edit Delete</div>		
<input type="checkbox"/>	« http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Flights10k/0dWi2lJAKiRsnW1sY190ie/0000_00000/0008656 »	« http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Model#Flights10k.originAirport »	« BOS »	http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e	⋮		

Anzo displays the Edit Statement dialog box. For example:

### Edit Statements

Subject \*

<http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Flights1

Predicate \*

<http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e/Model#

Object \*

"BOS"

Named Graph URI \*

<http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99d3085f0bc7f84e>

CANCEL

SAVE

Change any of the quad values, and then click **Save**.

**Important**  
If you edit URI values, make sure that the modified value is a valid URI.

Deleting a Statement

To delete a statement, click the menu icon (⋮) to the right of the statement and select **Delete**. Anzo displays the statement in a confirmation dialog box. For example:

Delete Statements

Subject \*

<http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99

Predicate \*

<http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99

Object \*

"BOS"

Named Graph URI \*

<http://cambridgesemantics.com/Layer/f9df7ce18bc1478d99

CANCEL

DELETE

Click **Delete** to remove the statement from the system data source.

Adding a Statement

To add a quad to the data source, click the **Add Statements** button at the top of the result list.

Subject

Predicate

Object

BOS

Add a quad containing a subject, object, predicate and graph name to the data source to search for.

Clear

Add Statements

Find

Results (50)

☒ Show native

☒ Subject

☒ Predicate

☒ Object

☒ Named Graph

Anzo displays the Add Statements dialog box.

Add Statements

×

Subject \*

Predicate \*

Object \*

Named Graph URI \*

Add

CANCELADD

Specify the new quad by adding the subject, predicate, object, and named graph URI in the appropriate fields. Each field is required. URIs must be valid, and the Named Graph URI that you specify must be present in the data source. You cannot add a new named graph. Click **Save** to add the new quad to the data source.

Related Topics

[Running SPARQL Queries in the Query Builder](#)

# Accessing Data on Demand Endpoints

The topics in this section provide information about accessing Data on Demand endpoints and using the OData API as well as the Anzo ODBC and JDBC drivers.

- [Accessing an Endpoint Programmatically](#) .....923
- [Accessing an Endpoint from an Application](#) .....927
- [OData Reference](#) .....944

## Related Topics

[Creating Data on Demand Endpoints](#)

## Accessing an Endpoint Programmatically

This topic provides guidance on accessing Data on Demand endpoints programmatically by showing some example implementations using R and Python.

- [Authentication and Data Access](#)
- [Accessing an Endpoint with R \(Through RStudio\)](#)
- [Accessing an Endpoint with Python \(Through a Linux Terminal\)](#)

### Authentication and Data Access

Connections to Data on Demand endpoints must be authenticated. Users can submit their Anzo username and password when accessing data. Ultimately the data that is available to users from OData endpoints is subject to the security and composition of the graphmart as configured in Anzo.

### Accessing an Endpoint with R (Through RStudio)

The following example shows how to connect to an OData endpoint from RStudio. The example uses the R programming language to access a Data on Demand endpoint and pull in data via a standard dataframe. New or existing R scripts can then be used with the data.

The first step in accessing data from RStudio is to prepare the R script that will construct the target URL and retrieve the resulting information via HTTP. The example script below accesses a pre-configured "Sample Data" endpoint. The script has sections for filtering the results as well as expanding the selection to include information from multiple classes:

```
require("httr")
require("jsonlite")
require("rstudioapi")

user    <- rstudioapi::showPrompt("Username", "Enter Anzo username", "sysadmin")
pw      <- rstudioapi::askForPassword(paste("Enter password for",user,sep=" "))

## Data on Demand endpoint
odata   <- "https://10.100.0.10/dataondemand/Sample-Graphmart/Sample-Data"

## Start from Probe class
startClass <- "Probe?"

## Filter results for Homo sapiens species
```

```

filterKw    <- "$filter="
filterVal   <- "Species eq 'Hs'"
urlify      <- URLEncode(filterVal)
filterStr   <- paste(filterKw,urlify,sep="")

## Select properties of interest (FeatureID) from base class
selectKw    <- "&$select="
selectVal   <- "FeatureID"
selectStr   <- paste(selectKw,selectVal,sep="")

## Select properties of interest (symbol) from Gene class
## via corresponds_to property on base Probe class
expandKw    <- "&$expand="
expandClass <- "corresponds_to"
expandProps <- "symbol"
expSelStr   <- "$select="
expandStr   <- paste(expandKw,expandClass,"(",expSelStr,expandProps,")",sep="")

## Specify format
format      <- "&$format=json"

## Generate OData URL using fragments above
url <- paste(odata,startClass,filterStr,selectStr,expandStr,format,sep="")

## Access OData endpoint
resultRaw   <- GET(url, (authenticate(user,pw, type = "basic")))
resultTxt   <- content(resultRaw, "text")
resultJson  <- fromJSON(resultTxt, flatten = TRUE)

print(url)

## Read results into dataframe
resultDataFrame <- as.data.frame(resultJson)
View(resultDataFrame)

```

Executing the above R script from RStudio results in a dataframe that represents columns from the **Probe** and **Gene** classes.



## Accessing an Endpoint with Python (Through a Linux Terminal)

Many users have existing Python scripts to use with data in Anzo or a familiarity with Python that would make exploring, retrieving, and leveraging the data easier. The following example shows how to connect to an OData endpoint by executing a Python script from a Linux terminal.

The first step in accessing data using Python is to prepare the Python script that will construct the target URL and retrieve the resulting information via HTTP. The example script below accesses a pre-configured "Sample Data" endpoint. The script has sections for filtering the results as well as expanding the selection to include information from multiple classes (the same filter and class properties that were used in the R example above).

```
import requests
import getpass
from urllib.parse import urlparse

un = getpass.getpass(prompt='Username: ')
pw = getpass.getpass(prompt='Password: ')

## OData endpoint
# Data on Demand URL
odata = 'https://10.100.0.10/dataondemand/Sample-Graphmart/Sample-Data/'

## Start from Lease class
startClass = "Probe?"

## Filter results
filterKw = "$filter="
filterVal = "Species eq 'Hs'"
urlify = urlparse(filterVal)
filterStr = filterKw + urlify.geturl()

## Select properties of interest (start date, missed payments, lease status)
from base class
selectKw = "&$select="
selectVal = "FeatureID"
selectStr = selectKw + selectVal

## Select properties of interest (name, social security number, credit score)
```

```

from Individual class
expandKw  = "&$expand="
expandClass = "corresponds_to"
expandProps = "symbol"
expSelStr  = "$select="
expandStr = expandKw + expandClass + "(" + expSelStr + expandProps + ")"

## Specify format
format = "&$format=text/csv"

## Generate OData URL using fragments above
url = odata + startClass + filterStr + selectStr + expandStr + format

## Access OData endpoint
r = requests.get(url, auth=(un, pw), verify=False)

print("URL")
print(url)
print("CONTENT")
print(r.content.decode('unicode_escape'))
print(type(r))
print(type(r.content))

```

In this example, the output is returned in CSV format (rather than JSON like the R example).

## Related Topics

[Creating Data on Demand Endpoints](#)

[Accessing an Endpoint from an Application](#)

[OData Reference](#)

## Accessing an Endpoint from an Application

Since Anzo's Data on Demand service conforms to the OData standard, any tool that supports the OData V4 REST API can access a Data on Demand endpoint to leverage data in Anzo. In addition, applications that support ODBC or JDBC APIs can use the Anzo CData ODBC or JDBC drivers to interact with Data on Demand endpoints. This capability enables users to leverage the benefits of Anzo's semantic layer, data model, and data blending capabilities in their favorite analytics tools.

This topic provides information about accessing Data on Demand endpoints from third-party applications.

- [JDBC Driver Considerations](#)
- [Authentication and Data Access](#)
- [Accessing Data via the OData API](#)
- [Downloading the Anzo ODBC and JDBC Drivers](#)

### JDBC Driver Considerations

This section describes important items to consider when using JDBC clients for accessing Data on Demand endpoints:

- [Join Performance](#)
- [Querying Multi-Valued Properties](#)
- [Working with Long Column Names](#)

### Join Performance

To join results from multiple classes, Cambridge Semantics strongly recommends using OData or SPARQL. Hi-Res Analytics and SPARQL are designed to quickly return large results from multiple classes and should be strongly considered for these use cases. Joins on large data sets are also well-supported with OData when best practices around paging are applied.

You can also join tables upstream in AnzoGraph by creating data layers. For example, you can create a view that joins the data using a CONSTRUCT query. The view becomes available as an OData table. For information about view steps, see [Constructing a View of the Data \(View Step\)](#).

In addition, Custom Data on Demand endpoints (sometimes called Table endpoints) enable you to join classes, add filters, and apply functions to properties during endpoint creation. The tables that you create are automatically translated to SPARQL queries that create views in AnzoGraph, allowing you to perform complex analytics on the graph yet generate results in the tabular format that BI tools expect.

Because the JDBC driver generates multiple OData queries and joins the results in memory, SQL queries that include JOINS on large data sets may take a very long time to complete. When using the JDBC driver, Cambridge Semantics recommends that you query one class at a time and then use the BI tool to do analytics on the returned data. For more information, see [JDBC Performance Details](#) below.

## Querying Multi-Valued Properties

Some applications do not directly support Anzo's RDF graph data structures. For example, sometimes the JDBC driver presents multi-valued properties as arrays, which are not supported by some BI tools. When creating a Data on Demand endpoint for a graphmart that includes multi-valued properties, consider denormalizing the results to expand the properties into new rows so that they can be viewed in BI tools. For more information, see [Creating Data on Demand Endpoints](#).

## Working with Long Column Names

By default, the JDBC driver creates column names based on the property labels in the data model. The property labels can be too long for some clients. For example, Informatica is limited to 128 characters. When ingesting data from a tabular source, the label is a concatenation of the table and column name. Users may need to shorten the property labels to work with JDBC clients. If the label is missing, Anzo uses the `localName` of the IRI. For information about configuring the column names to be used for a Data on Demand endpoint, see [Creating Data on Demand Endpoints](#).

## Authentication and Data Access

Connections to Data on Demand endpoints must be authenticated. Users can submit their Anzo username and password when accessing data. If your applications use single sign-on (SSO) authentication, you can also use SSO with Anzo. When using SSO, the client authenticates the user against the SSO provider and then passes the credentials to Anzo. All data is secured according to the user's SSO profile. For information about the supported SSO providers and instructions on configuring SSO access, see [Connecting to an SSO Provider](#) in the Administration Guide.

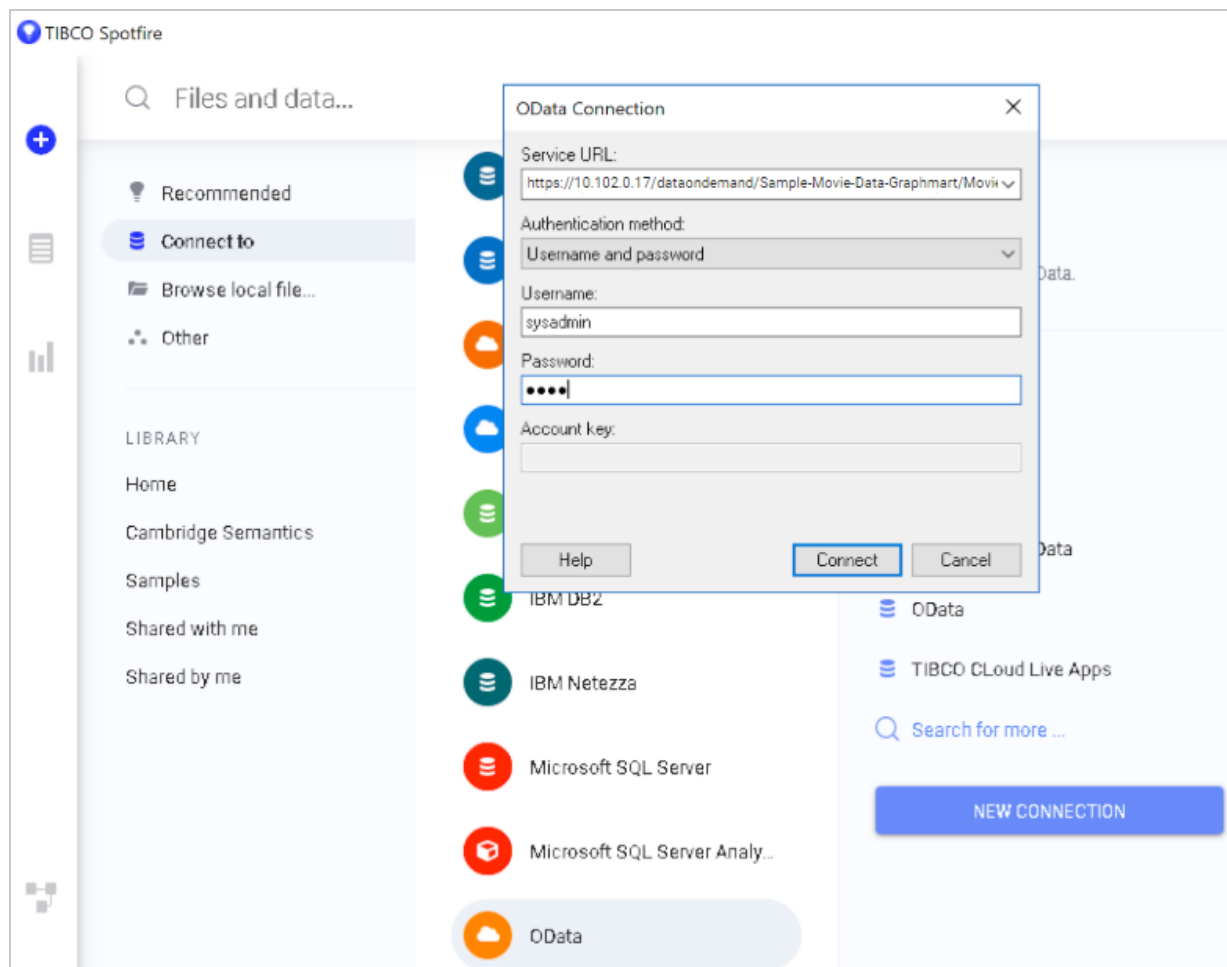
### Note

Ultimately the data that is available to users from Data on Demand endpoints is subject to the access control configuration of the graphmart in Anzo.

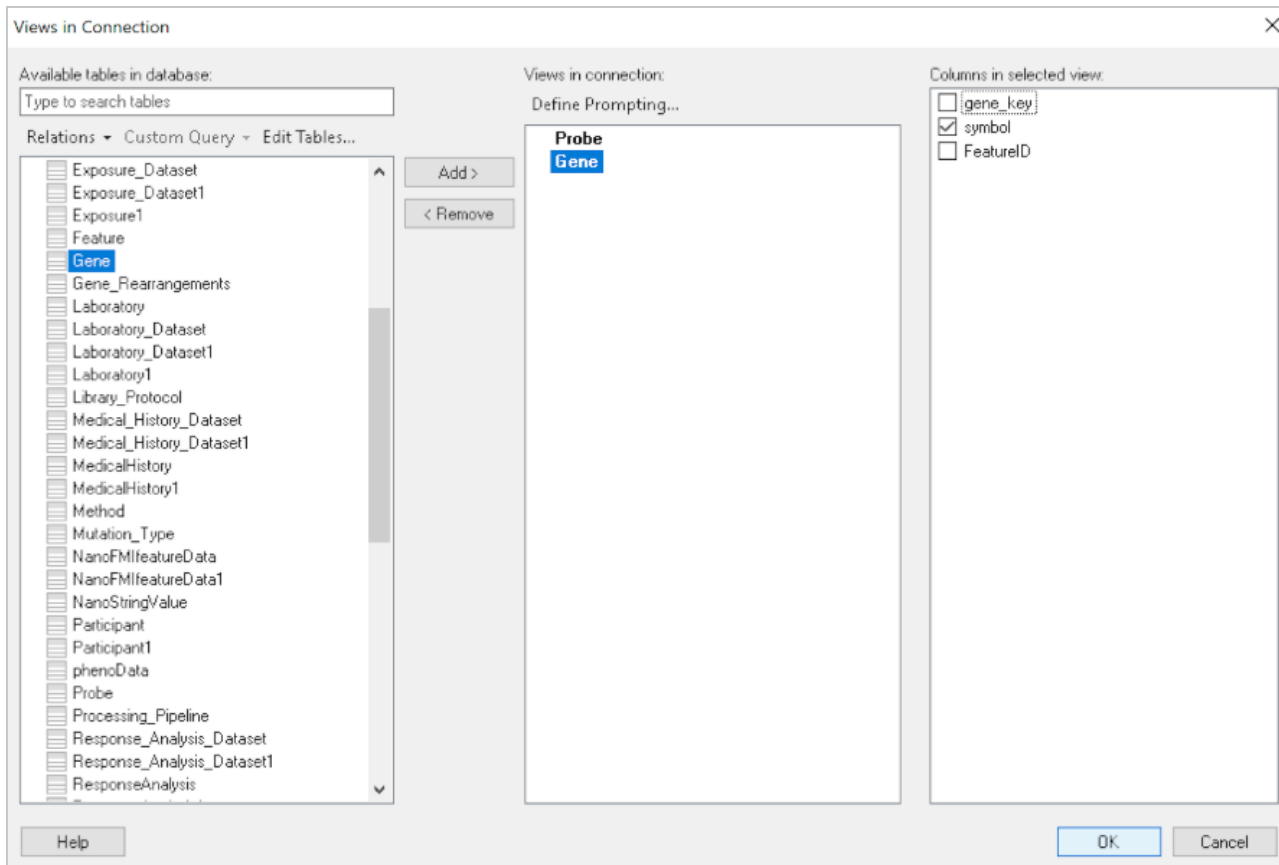
## Accessing Data via the OData API

This section provides guidance on accessing a Data on Demand endpoint from an application that supports the OData REST API. It includes an example that configures an OData connection in TIBCO Spotfire. The example steps can also be applied to OData connections in other similar business intelligence tools.

The first step is to connect to the OData endpoint using the Spotfire Data sources user interface. When setting up the OData connection, the Service URL is the OData/ODBC URL from the Data on Demand endpoint configuration details in Anzo. The OData connection uses the user's Anzo credentials for authentication.



Once the connection is established, Sportfire prompts the user to select the classes and properties to work with. In this example, the **FeatureID** property from the **Probe** class and the **symbol** property from the **Gene** class are selected:



Once the properties are chosen, the data is loaded in Spotfire and can be used to inform existing analytics and data visualizations or create new ones.

## Downloading the Anzo ODBC and JDBC Drivers

This section provides guidance on accessing Data on Demand endpoints from applications that support ODBC or JDBC APIs. Your Anzo deployment includes CData ODBC and JDBC drivers to use with applications. The first step is to retrieve the appropriate driver for your client. To download a driver, open a web browser and go to the following URL:

```
https://<Anzo_server>/installs/anzodataaccess
```

Where <Anzo\_server> is the Anzo server DNS name or IP address. The Anzo Data Access Software Installation page provides links to download each driver. For example:

## JDBC Driver

[CData JDBC Driver for Anzo - Build 21.0.81717](#)

## Install CData Driver

[CData ODBC Driver for Anzo - Build 21.0.8171 \(Windows\)](#)

[CData ODBC Driver for Anzo - Build 21.0.8171 \(Mac\)](#)

## License Details

Cdata drivers are included with Cambridge Semantics Products under OEM license agreement.

Download the appropriate driver to the client server:

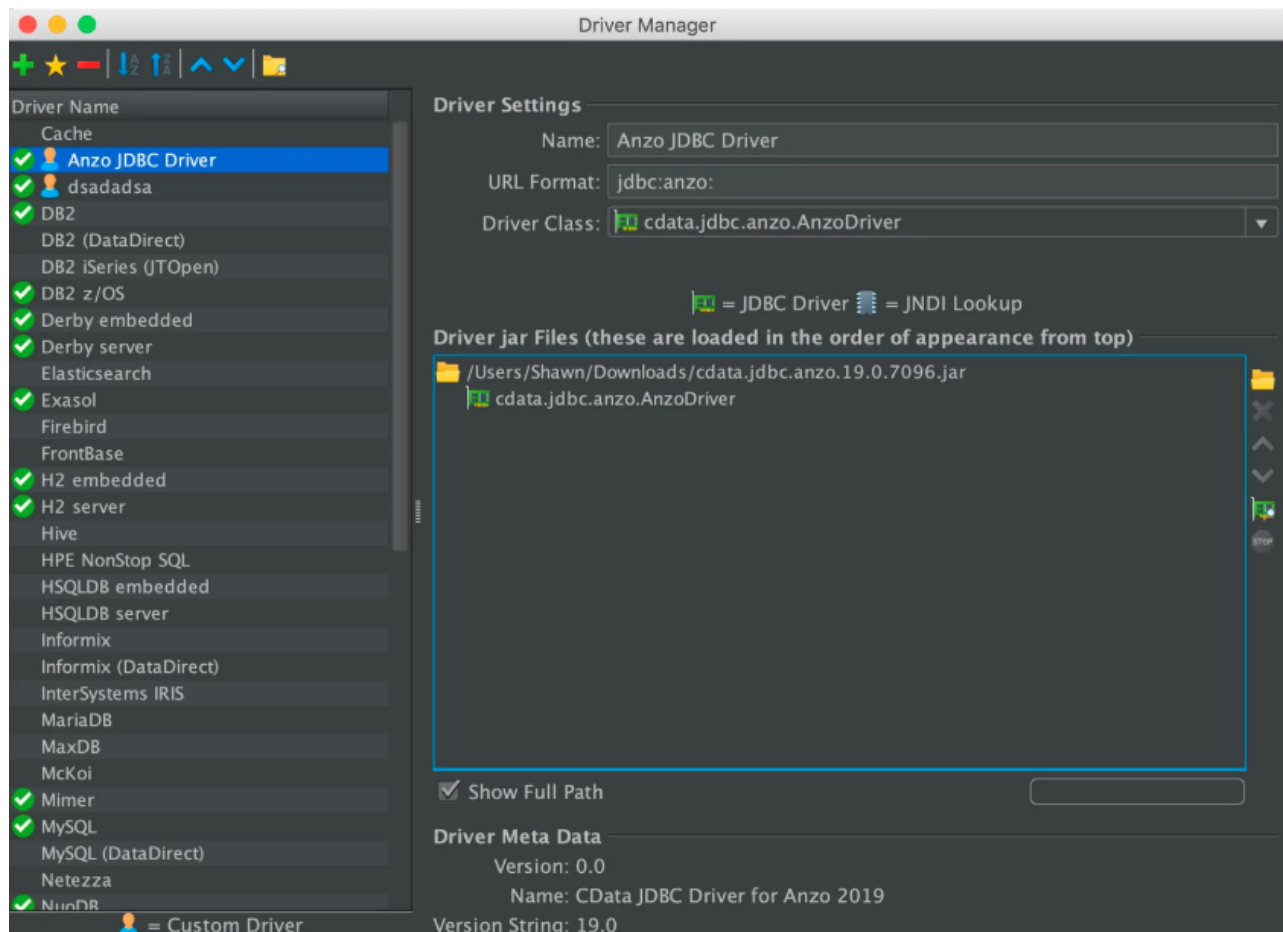
- The **CData JDBC Driver for Anzo** is used to connect to Anzo from most Java applications and database management tools.
- The **CData ODBC Driver for Anzo** for Windows or Mac is for use with applications and database management tools that support open database connectivity, such as Microsoft Excel or Tableau.

## Configuring the Driver and Connecting to the Endpoint

This section provides guidance configuring an ODBC or JDBC driver by showing examples of configuring DbVisualizer and Tableau to access a Data on Demand endpoint using Anzo's JDBC driver and configuring Power BI to access an endpoint using the ODBC driver.

### Example JDBC Setup with DbVisualizer

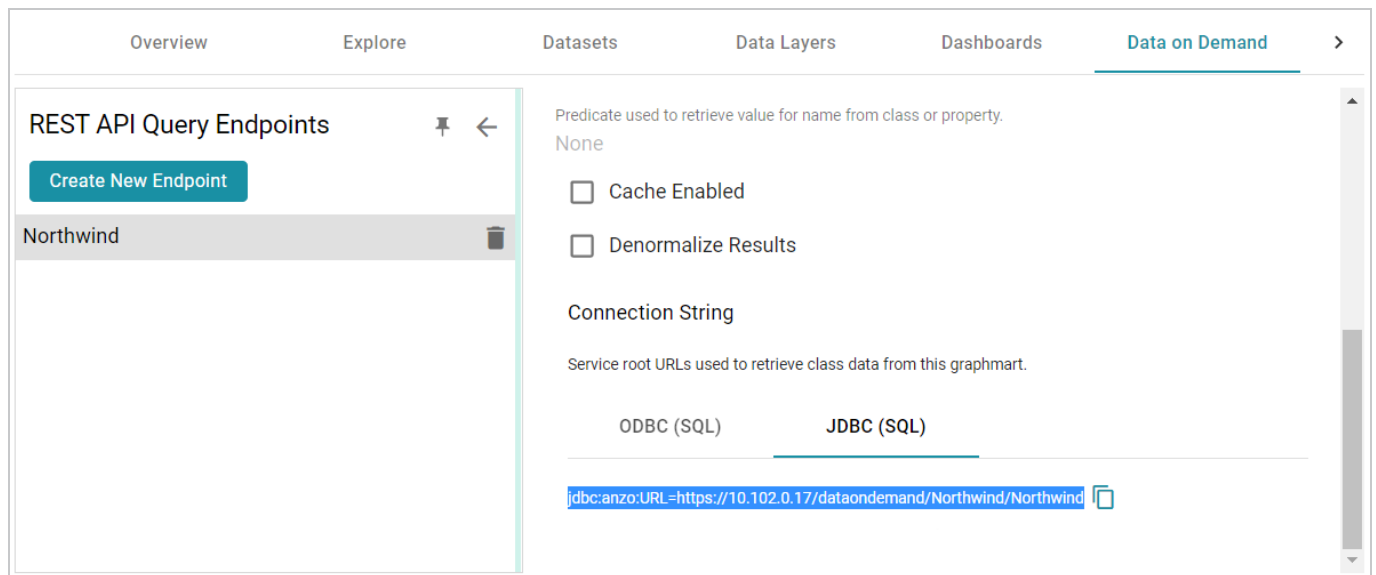
1. In DbVisualizer, go to **Tools** → **Driver Manager**.
2. In the Driver Manager, click the green plus icon to create a new driver.
3. Specify a name for the driver. For example, **Anzo JDBC Driver**.
4. In the URL Format field, specify the format **jdbc:anzo**.
5. In the **Driver File Paths** or **Driver jar Files** section of the screen, click the folder icon and then browse to and select the directory where you saved the CData JDBC Driver for Anzo `cdata.jdbc.anzo.jar` file that you downloaded to the server. DbVisualizer reads the jar and sets the Driver Class to **cdata.jdbc.anzo.AnoDriver**. For example:



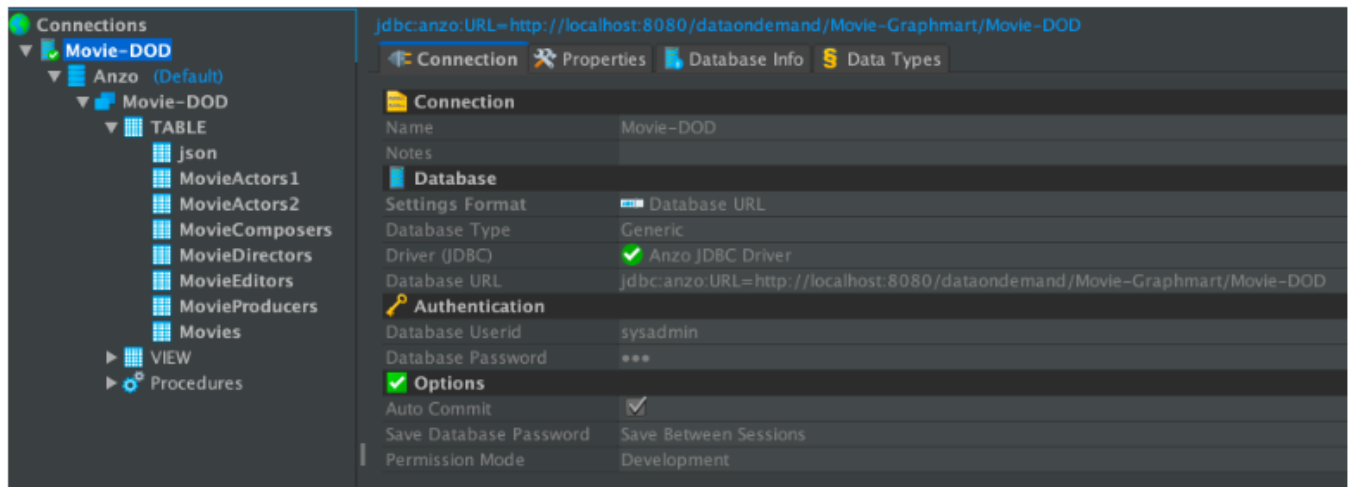
6. To connect to the endpoint in DbVisualizer, go to **Database** → **Create Database Connection**. Click **No Wizard** when prompted.
7. Specify a name for the connection in the **Name** field.
8. In the **Driver (JDBC)** field, select the Anzo JDBC driver connection.
9. In the **Database URL** field, specify the JDBC URL from the Anzo Data on Demand endpoint configuration. For example:  

```
jdbc:anzo:URL=https://10.10.0.11/dataondemand/Northwind/Northwind
```





- Under **Authentication**, enter your Anzo user ID and password. You should now be able to connect to the endpoint and view the available schemas. For example:

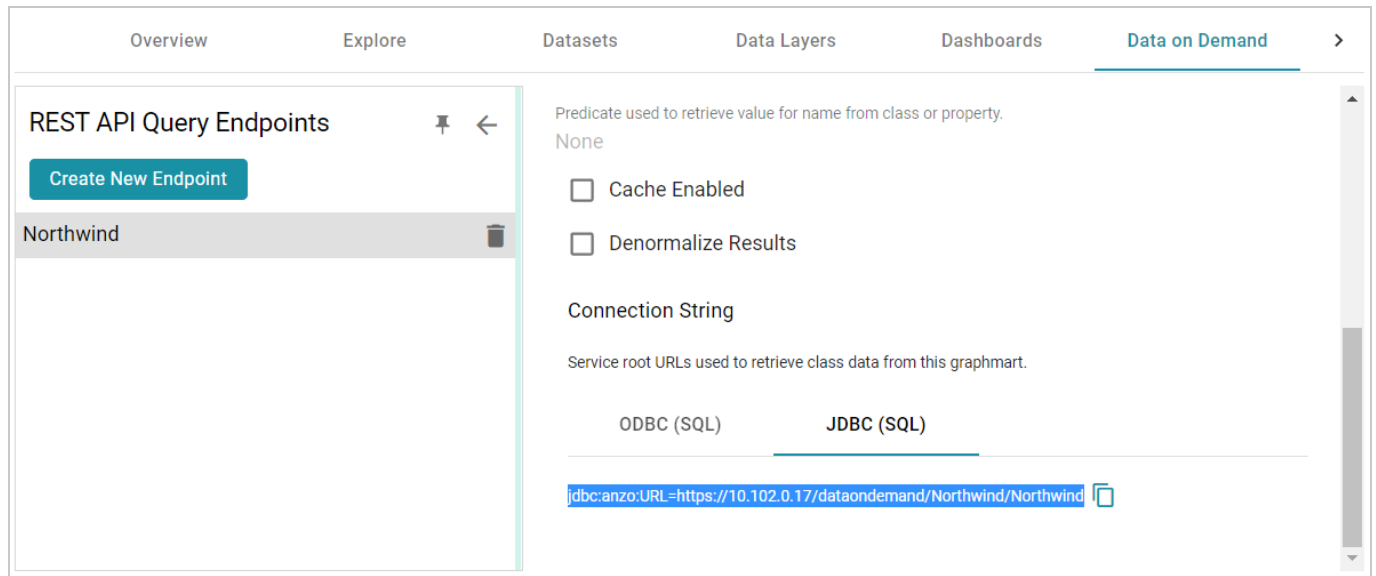


## Example JDBC Setup with Tableau

- After downloading the CData JDBC Driver for Anzo `cdata.jdbc.anzo.jar` file, place the `.jar` in the appropriate directory depending on your operating system:
  - Windows:** `C:\Program Files\Tableau\Drivers`
  - MacOS:** `~/Library/Tableau/Drivers`
  - Linux:** `/var/opt/tableau/tableau_server/data/tabsvc/vizqlserver/Datasources/`
- Restart Tableau and then go to **Add a Connection** → **To a Server**.
- Click **Other Databases (JDBC)**.

4. In the **Database URL** field, specify the JDBC URL from the Anzo Data on Demand endpoint configuration. For example:

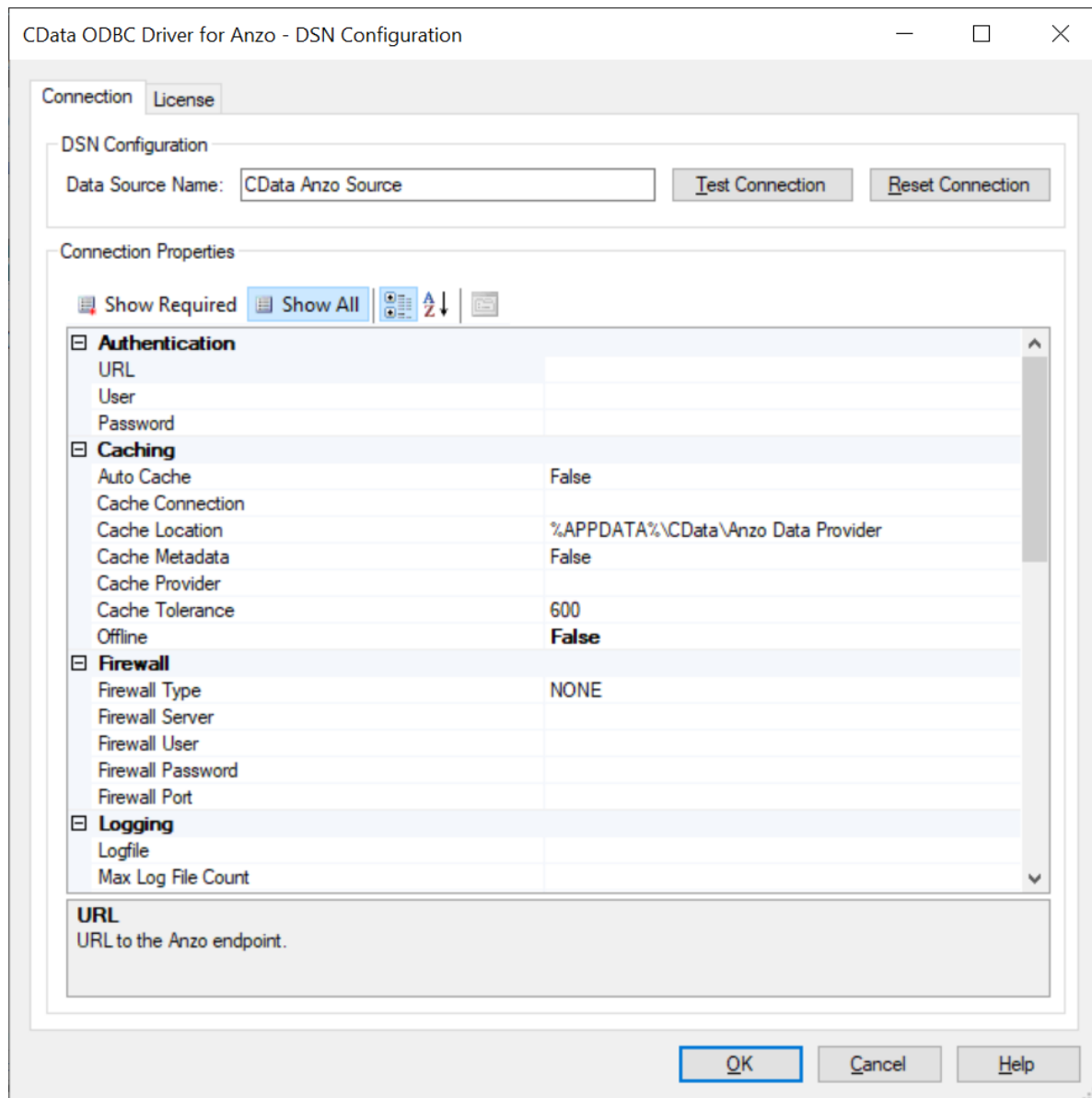
`jdbc:anzo:URL=https://10.10.0.11/dataondemand/Northwind/Northwind`



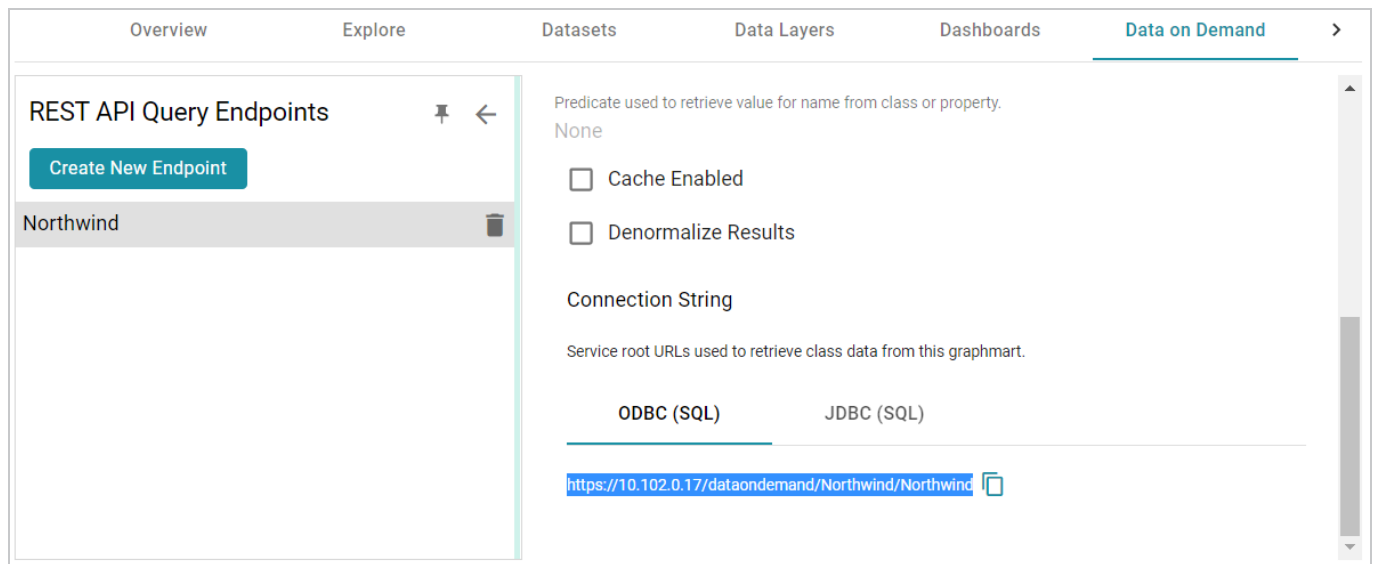
5. Enter your Anzo username and password and click **Sign In**. You should now be able to connect to the endpoint and view the available schemas.

### Example ODBC Setup with Microsoft Power BI

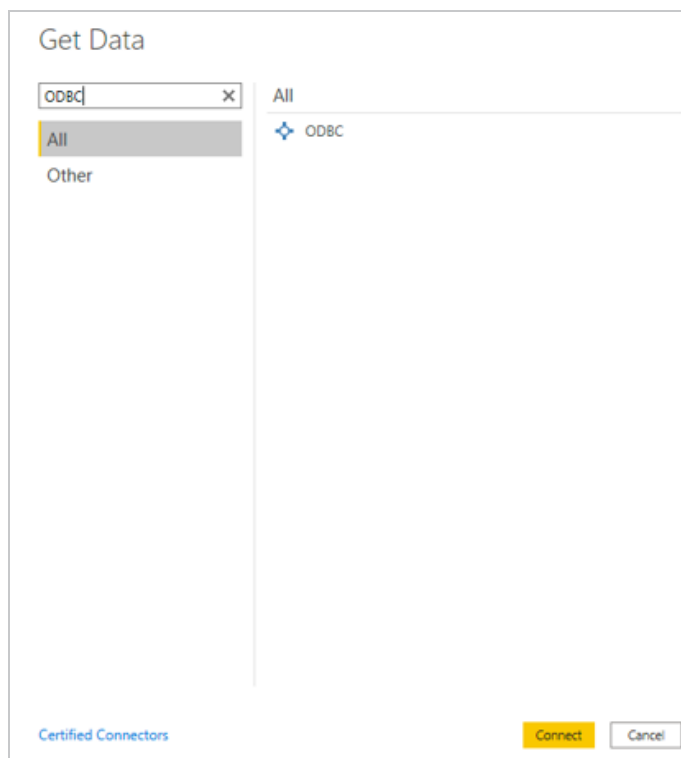
1. After downloading the Windows CData ODBC Driver for Anzo executable file, run the executable to start the installation wizard. The wizard guides you through installing the driver.
2. At the end of the installation, make sure the **Configure ODBC Data Source** checkbox is selected and click **Finish**. The wizard opens the driver's DNS Configuration screen. For example:



3. Under **Authentication** in Connection Properties, specify the **URL**, **User**, and **Password** to use for connecting to the Data on Demand endpoint. The User and Password are the Anzo username and password to use for authentication, and URL is the OData/ODBC service root URL for the endpoint. You can retrieve the URL from the Data on Demand screen for the endpoint. For example:



4. Click **OK** to save the configuration changes and close the dialog box.
5. Next, connect to the ODBC data source from Power BI. Open Power BI and click the **Get Data** button in the tool bar. In the Get Data dialog box, search for "ODBC." For example:



6. The search opens the wizard for creating an ODBC connection to a specified data source. Select **CData Anzo Source** from the drop-down list. You do not need to configure the advanced options.

From ODBC

Data source name (DSN)

CData Anzo Source

> Advanced options

OK Cancel

7. Click **OK** to create the connection. Power BI opens the Navigator screen. For example:

Navigator

Display Options ▾

- ODBC (dsn=CData Anzo Source) [1]
  - Anzo [1]
    - InitialSample [1]
      - ☐ Status\_Entry

Status\_Entry

status_entry_key	Basis	Categori
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDAyZTBiNmItYTl3OS00MD	Cumulative	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDA0Nzk5ODEtZTk5ZS00Mj	Period	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDA1OTIhYWYtNmJhNC00	Cumulative	Co
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMD8hOWI1MWQ0OGJjYy00	Cumulative	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDEwZmVhNzAtYWNjZS00	Cumulative	Co
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDEzY2RhMzQtZTVjOS00YT	Period	Co
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDEzZDliYjMtYTl0MS00YW	Cumulative	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDE2MGE0MzEtNWY5Mi00	Period	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDE2MzRjN2ItMjNiYS00ND	Period	Co
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDFkZGIwMGltZWQwZC00	Cumulative	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDIwNDZmOGUtNGQ5Mi00	Cumulative	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDIyYzcyOGltY2YzNy00ZTkz	Period	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDI0MGJlOTgtYjcyMi00YW	Cumulative	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDI3ZjgwZWQ0NGE2MC00Z	Cumulative	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDJkYTcyZDMtYWEyMy00	Cumulative	Co
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDJmODdkZDEtZWZlNy00N	Cumulative	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDMwOTk3NjUtMTMxZi00	Period	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDMxMmUzN2MtYzBjYS00	Period	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDMyZDRkMzMtZmVjYS00	Cumulative	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDM0YzQ3YmMtMDA2Zi00	Period	Ph
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDM4MjUwNDUtMWWE2MC	Cumulative	Co
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDNjM2JlZjltMGFkNS00Y2N	Period	Co
aHR0cDovL2NzaS5jb20vU3RhdHVzRW50cnkvMDNkNGRhODEtZTJhYy00M	Cumulative	Ph

Select Related Tables Load Transform Data Cancel

Under Display Options, the top level container in the view represents the ODBC driver, the **Anzo** item represents the database, and the **InitialSample** item represents the schema. Each table is represented as a table entry under the schema. In the example above there is one table. If you select a table, sample data from that table is displayed on the right side of the screen. To load table(s), select the checkbox for each table and click the **Load** button. You can also use the advanced features of Power BI to transform the data as you load it into the tool.

## JDBC Driver Quick Reference

This section provides a quick reference for JDBC driver support.

- For the complete JDBC driver documentation, see [CData JDBC Driver for Anzo](#).
- For the complete ODBC driver documentation, see [CData ODBC Driver for Anzo](#).

### SQL Compliance

The JDBC driver supports most of the standard operations for querying data. The exceptions are listed below.

- The driver does not currently support transactions.
- The driver does not support batching of SQL statements.
- The driver has support for inserting, updating, and deleting records. However, performing updates via the driver can have unexpected consequences.

For more information about SQL compliance, see the [SQL Compliance](#) section in the CData JDBC Driver documentation.

### JDBC Performance Details

By default, the JDBC driver offloads to Anzo as much of the SELECT statement processing as possible and then processes the rest of the query locally in memory.

- For joins, the driver generates multiple OData queries and joins the results in memory. As a result, SQL queries that include JOINS can take up to several minutes to complete.
- For aggregates, the driver retrieves all rows necessary to process in memory.
- For predicates, the driver determines which clauses Anzo supports and sends them to Anzo to retrieve the smallest possible superset of rows that would satisfy the query. It then filters the rest of the rows client-side.
- The driver's **SupportEnhancedSQL** setting can be disabled to limit SQL execution to only what the Anzo API supports. For more information, see the [Support Enhanced SQL](#) section in the CData JDBC Driver documentation.

#### Tip

To determine which query capabilities the driver can offload to the Anzo API, you can query the **sys\_sqlinfo** system table. The table contains information about the functionality that is supported by the connected source. For example:

```
SELECT * FROM sys_sqlinfo WHERE name='AGGREGATE_FUNCTIONS'
or name = 'COUNT' or name = 'SUPPORTED_OPERATORS' or name = 'GROUP_
BY'
or name = 'OUTER_JOINS' or name = 'OJ_CAPABILITIES' or name =
'SUBQUERIES'
or name = 'STRING_FUNCTIONS' or name = 'NUMERIC_FUNCTIONS'
or name = 'TIMEDATE_FUNCTIONS';
```

For more information, see the [sys\\_sqlinfo](#) section in the CData JDBC Driver documentation.

## Data Caching

Due to the client-side in-memory processing of aggregates and joins, the performance of queries against extremely large data sets may suffer. If this is a common use case, consider leveraging caching in the JDBC driver. If the driver maintains a local copy of the data, it reduces the number of API calls and can increase performance for long-running queries. For more information, see the [Caching Data](#) section in the CData JDBC Driver documentation.

## Supported SELECT Statement Clauses

The following list shows the supported SELECT statement clauses. For more information, see the [SELECT Statement](#) section in the CData JDBC Driver documentation.

- SELECT
- INTO
- FROM
- JOIN
- WHERE
- GROUP BY
- HAVING
- UNION
- ORDER BY
- LIMIT

## Supported Aggregate Functions

The following list shows the supported aggregate functions. For more information, see the [Aggregate Functions](#) section in the CData JDBC Driver documentation.

- COUNT
- COUNT\_DISTINCT
- AVG
- MIN
- MAX
- SUM

## Supported Joins

The following list shows the supported JOIN types. For more information, see the [JOIN Queries](#) section in the CData JDBC Driver documentation.

- **Inner Join:** Selects only the rows from both tables that match the join condition.
- **Left Join:** Selects all of the rows in the FROM table and only matching rows in the JOIN table.

## SQL Function Reference

The JDBC driver provides implementations of the following common SQL functions. For more information, see the [SQL Functions](#) section in the CData JDBC Driver documentation.

### Note

The driver interprets all function input as either column names or strings. Therefore, all string literals must be escaped with single quotes. For example, `SELECT DATENAME ( 'yy' , GETDATE ( ) ) .`

## String Functions

- ASCII(character\_expression)
- CHAR(integer\_expression)
- CHARINDEX(expressionToFind ,expressionToSearch [,start\_location ])
- CONCAT(string\_value1, string\_value2 [, string\_valueN])
- CONTAINS(expressionToSearch, expressionToFind)
- ENDSWITH(character\_expression, character\_suffix)
- FORMAT(value, format)
- FROM\_UNIXTIME(time, format, issecond)
- INDEXOF(expressionToSearch, expressionToFind [,start\_location ])
- ISNULL(check\_expression , replacement\_value)
- JSON\_AVG(json, jsonpath)



- JSON\_COUNT(json, jsonpath)
- JSON\_EXTRACT(json, jsonpath)
- JSON\_MAX(json, jsonpath)
- JSON\_MIN(json, jsonpath)
- JSON\_SUM(json, jsonpath)
- LEFT(character\_expression , integer\_expression)
- LEN(string\_expression)
- LOWER(character\_expression)
- LTRIM(character\_expression)
- NCHAR(integer\_expression)
- PATINDEX(pattern, expression)
- QUOTENAME(character\_string [, quote\_character])
- REPLACE(string\_expression, string\_pattern, string\_replacement)
- REPLICATE(string\_expression ,integer\_expression)
- REVERSE(string\_expression)
- RIGHT(character\_expression , integer\_expression)
- RTRIM(character\_expression)
- SOUNDEX(character\_expression)
- SPACE(repeatcount)
- STARTSWITH(character\_expression, character\_prefix)
- STR(float\_expression [ , integer\_length [ , integer\_decimal ] ] )
- STUFF(character\_expression , integer\_start , integer\_length , replaceWith\_expression)
- SUBSTRING(expression,integer\_start,integer\_length)
- TOSTRING(string\_value1)
- TRIM(character\_expression)
- UNICODE(ncharacter\_expression)
- UPPER(character\_expression)
- XML\_EXTRACT(xml, xpath [, separator])

## Date Functions

- CURRENT\_DATE()
- CURRENT\_TIMESTAMP()
- DATEADD(datepart , integer\_number , date [, dateformat])
- DATEDIFF(datepart , startdate , enddate )
- DATEFROMPARTS(integer\_year, integer\_month, integer\_day)
- DATENAME(datepart , date)
- DATEPART(datepart, date [,integer\_datefirst])
- DATETIME2FROMPARTS(integer\_year, integer\_month, integer\_day, integer\_hour, integer\_minute, integer\_seconds, integer\_fractions, integer\_precision)
- DATETIMEFROMPARTS(integer\_year, integer\_month, integer\_day, integer\_hour, integer\_minute, integer\_seconds, integer\_milliseconds)
- EOMONTH(start\_date [, integer\_month\_to\_add ])
- GETDATE()
- GETUTCDATE()
- ISDATE(date, [date\_format])
- SMALLDATETIMEFROMPARTS(integer\_year, integer\_month, integer\_day, integer\_hour, integer\_minute)
- SYSDATETIME()
- SYSUTCDATETIME()
- TIMEFROMPARTS(integer\_hour, integer\_minute, integer\_seconds, integer\_fractions, integer\_precision)
- YEAR(date)

## Math Functions

- ABS(numeric\_expression)
- ACOS(float\_expression)
- ASIN(float\_expression)
- ATAN(float\_expression)
- ATN2(float\_expression1 , float\_expression2)
- CEILING(numeric\_expression)

- COS(float\_expression)
- COT(float\_expression)
- DEGREES(numeric\_expression)
- EXP(float\_expression)
- EXPR(expression)
- FLOOR(numeric\_expression)
- LOG(float\_expression [, base ])
- LOG10(float\_expression)
- PI( )
- POWER(float\_expression , y)
- RADIANS(float\_expression)
- RAND([ integer\_seed ])
- ROUND(numeric\_expression , integer\_length [ ,function ])
- SIGN(numeric\_expression)
- SIN(float\_expression)
- SQRT(float\_expression)
- SQUARE(float\_expression)
- TAN(float\_expression)

## Related Topics

[Creating Data on Demand Endpoints](#)

[Accessing an Endpoint Programmatically](#)

[OData Reference](#)

# OData Reference

The Anzo Data on Demand service follows the [OData Version 4.0 specification](#), which defines the standard URL conventions and query options. This topic provides a quick reference for learning OData basics and viewing the supported string operators and output formats. It also provides some example queries.

- [OData URL Conventions](#)
- [Supported Query Operators](#)
- [Example OData Requests](#)

## OData URL Conventions

An OData service URL has three main parts:

1. The **Service Root URL** that Anzo provides. The service root URL is the metadata that describes all of the available feeds (tables).
2. The optional **Resource Path** that narrows the scope of the available data to the individual table (class) level, property level, or the schema.
3. The **Query Options** for analyzing the data.

For example, the following OData URL shows the service root from the Data on Demand screen in Anzo, a resource path that narrows the scope of the data to the Employees table (class), and query options that filter the result set to show data for the NA region only:

`https://10.100.0.10/dataondemand/Northwind-Graphmart/Northwind/Employees?$filter=contains(Region, 'NA')`

Service Root URL

Resource Path

Query Options

OData requests need to be URL-encoded. Typically you can configure programs to encode requests automatically. And browsers encode URLs that are pasted into the address bar.

## Supported Query Operators

OData query options are used to dynamically query data via the endpoint and control the amount and order of the data returned. The Data on Demand service supports the following OData query operators. See [Example OData Requests](#) below for example queries that employ the operators.

Operator	Description
\$count	Used to count the number of matching resources in the result set.

Operator	Description
<b>\$expand</b>	Used to retrieve related data and include it in the results. When you query data via OData, the default response does not include related entities. The \$expand option provides flexibility for exploring data across the data model. It allows the related information to be embedded in the response.
<b>\$filter</b>	Used to filter a result set. The expression specified with \$filter is evaluated for each resource identified by the resource path, and only items where the expression evaluates to true are included in the response.
<b>\$format</b>	Used to specify the output format for the results. The supported formats are text/CSV, JSON, and XML. For example: \$format=json
<b>\$metadata</b>	Used to return the schema, entity set, and property metadata.
<b>\$orderby</b>	Used to return results in ascending (asc) or descending (desc) order. If asc or desc is not specified, solutions are returned in ascending order.
<b>\$select</b>	Used to specify the subset of properties to include in the result set.
<b>\$skip</b>	Used to specify the number of solutions to exclude in the results. The \$top and \$skip OData query options are similar to the LIMIT and OFFSET clauses in SPARQL queries.
<b>\$top</b>	Used to limit the number of solutions that are returned.

## Example OData Requests

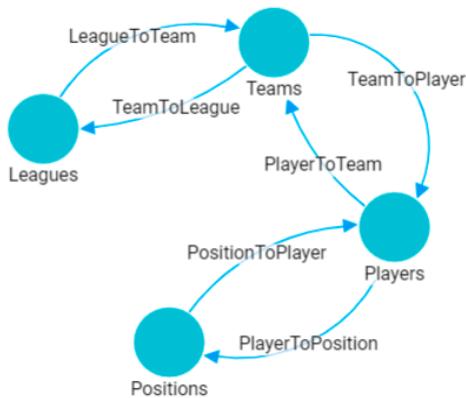
This section demonstrates the use of OData query operators by providing examples of common types of OData requests.

The examples below are run against a sample graphmart, called **LeagueGM**, that contains data about the teams and players in a small local baseball league. The Data on Demand endpoint is named **LeagueData**. The following service root URL was created by Anzo:

```
https://10.100.0.10/dataondemand/LeagueGM/LeagueData
```

For readability, the examples below abbreviate "https://10.100.0.10/dataondemand" to **dataondemand**. In addition, the examples are not URL-encoded.

The data has Leagues, Teams, Players, and Positions classes (or entities in OData). And the image below shows a graph view of the data model. To view the TriG version of the model, click [here](#).



To view the instance data for each class, you can click a link below to view the data for that class. The data is in JSON format.

 [Leagues](#)

 [Teams](#)

 [Players](#)

 [Positions](#)

## Retrieving Metadata

The request below retrieves the schema, entity set, and property metadata for the endpoint.

```
dataondemand/LeagueGM/LeagueData/$metadata
```

The results are in XML format. A snippet of the results is shown below. To view the complete response, click [here](#).

```
<?xml version="1.0" encoding="UTF-8"?>
<edmx:Edmx Version="4.0" xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx">
  <edmx:DataServices>
    <Schema xmlns="http://docs.oasis-open.org/odata/ns/edm" Namespace="Feeds">
      <EntityContainer Name="Default">
        <EntitySet Name="Leagues"
EntityType="com.cambridgesemantics.ont.autogen.LeagueDict.LeagueData.Leagues">
          <NavigationPropertyBinding Path="LeagueToTeam" Target="Teams"/>
        </EntitySet>
      </EntityContainer>
    </Schema>
  </edmx:DataServices>
</edmx:Edmx>
```

```

    <EntitySet Name="Teams"
EntityType="com.cambridgesemantics.ont.autogen.LeagueDict.LeagueData.Teams">
    <NavigationPropertyBinding Path="TeamToLeague" Target="Leagues"/>
    <NavigationPropertyBinding Path="TeamToPlayer" Target="Players"/>
</EntitySet>
    <EntitySet Name="Positions"
EntityType="com.cambridgesemantics.ont.autogen.LeagueDict.LeagueData.Positions">
    <NavigationPropertyBinding Path="PositionToPlayer" Target="Players"/>
</EntitySet>
    <EntitySet Name="Players"
EntityType="com.cambridgesemantics.ont.autogen.LeagueDict.LeagueData.Players">
    <NavigationPropertyBinding Path="PlayerToPosition"
Target="Positions"/>
    <NavigationPropertyBinding Path="PlayerToTeam" Target="Teams"/>
</EntitySet>
</EntityContainer>
</Schema>
...

```

## Counting an Entity

The request below returns the number of teams in the graphmart. Adding the resource path **Teams** to the request narrows the scope to the Teams entity (or class in Anzo).

```
dataondemand/LeagueGM/LeagueData/Teams/$count
```

### Result

```
4
```

This request returns the number of players:

```
dataondemand/LeagueGM/LeagueData/Players/$count
```

### Result

```
12
```

## Counting a Property of an Entity

The request below counts the number of players on the Al Thomas team. The request uses the `team_key` to identify the team and the `TeamToPlayer` to identify each player.

```
dataondemand/LeagueGM/LeagueData/Teams  
( 'aHR0cDovL2NzaS5jb20vVGVhbXMvMQ' ) /TeamToPlayer/$count
```

### Result

```
3
```

This request counts the number of positions played by James Smith:

```
dataondemand/LeagueGM/LeagueData/Players  
( 'aHR0cDovL2NzaS5jb20vUGxheWVycy8y' ) /PlayerToPosition/$count
```

### Result

```
2
```

## Filtering Data via Text Search

The request below filters the results to show data for the `TeamName` that equals "Black Sox." The request also returns results in JSON format:

```
dataondemand/LeagueGM/LeagueData/Teams?$filter=TeamName eq 'Black  
Sox'&$format=json
```

### Result

```
{  
  "@odata.context":  
  "https://10.100.0.10/dataondemand/LeagueGM/LeagueData/$metadata#Teams",  
  "value": [  
    {  
      "teams_key": "aHR0cDovL2NzaS5jb20vVGVhbXMvMg",  
      "TeamId": 2,  
      "teamtoleague_key": [  
        "aHR0cDovL2NzaS5jb20vTGVhZ3Vlcy8x"      ]  
    }  
  ]  
}
```



```

    ],
    "TeamName": "Black Sox",
    "teampoplayer_key": [
        "aHR0cDovL2NzaS5jb20vUGxheWVycy80",
        "aHR0cDovL2NzaS5jb20vUGxheWVycy81",
        "aHR0cDovL2NzaS5jb20vUGxheWVycy82"
    ]
  }
]
}

```

This request filters the data to find the players whose name contains "Ted."

```
dataondemand/LeagueGM/LeagueData/Players?$filter=contains(PlayerName,'Ted')
```

The request can also use "startswith" in place of contains to filter specifically for player names that start with "Ted."

```
dataondemand/LeagueGM/LeagueData/Players?$filter=startswith(PlayerName,'Ted')
```

## Result

```

{
  "@odata.context":
  "https://10.100.0.10/dataondemand/LeagueGM/LeagueData/$metadata#Players",
  "value": [
    {
      "players_key": "aHR0cDovL2NzaS5jb20vUGxheWVycy8xMA",
      "playertoposition_key": [
        "aHR0cDovL2NzaS5jb20vUG9zaXRpb25zLzM",
        "aHR0cDovL2NzaS5jb20vUG9zaXRpb25zLzI"
      ],
      "PlayerId": 10,
      "playertoteam_key": [
        "aHR0cDovL2NzaS5jb20vVGZhbmVNA"
      ],
      "PlayerName": "Ted James",
      "DefensiveRating": 92.55
    },
  ],
}

```

```

{
  "players_key": "aHR0cDovL2NzaS5jb20vUGxheWVycy84",
  "playertoposition_key": [
    "aHR0cDovL2NzaS5jb20vUG9zaXRpb25zLzI",
    "aHR0cDovL2NzaS5jb20vUG9zaXRpb25zLzEw"
  ],
  "PlayerId": 8,
  "playertoteam_key": [
    "aHR0cDovL2NzaS5jb20vVGh0bXMvMw"
  ],
  "PlayerName": "Ted Sale",
  "DefensiveRating": 77.33
}
]
}

```

## Selecting Properties and Ordering Results

The request below selects player names and their defensive ratings. The results are ordered by defensive rating in descending order so that the player with the highest defensive rating is listed first. The request also formats the results in text/csv.

```

dataondemand/LeagueGM/LeagueData/Players?$select=PlayerName,DefensiveRating&$orderby=DefensiveRating desc&$format=text/csv

```

## Result

```

PlayerName,DefensiveRating
James Smith,98.33
Alex Granderson,98.22
Matt Butler,95.66
Tim Hooper,93.43
Steve Jones,93.28
Ted James,92.55
Fred Wynn,88.68
Jared Bonds,86.34
Billy Roper,83.44
Mike Magazine,78.33

```

Ted Sale, 77.33  
Chris Underwood, 66.22

## Expanding the Results to Include Related Entities

The request below uses the \$expand operator to retrieve data from the Players entity and include the related Positions data for each player. For this example, the request limits the number of results returned to 2 players by adding \$top=2:

```
dataondemand/LeagueGM/LeagueData/Players?$expand=PlayerToPosition&$top=2
```

### Result

```
{
  "@odata.context":
  "https://10.100.0.10/dataondemand/LeagueGM/LeagueData/$metadata#Players",
  "value": [
    {
      "players_key": "aHR0cDovL2NzaS5jb20vUGxheWVycy8x",
      "playertoposition_key": [
        "aHR0cDovL2NzaS5jb20vUG9zaXRpb25zLzg"
      ],
      "PlayerId": 1,
      "playertoteam_key": [
        "aHR0cDovL2NzaS5jb20vVGVhbXMvMQ"
      ],
      "PlayerName": "Steve Jones",
      "DefensiveRating": 93.28,
      "PlayerToPosition": [
        {
          "positions_key": "aHR0cDovL2NzaS5jb20vUG9zaXRpb25zLzg",
          "PositionId": 8,
          "ShortName": "CF",
          "positiontoplayer_key": [
            "aHR0cDovL2NzaS5jb20vUGxheWVycy8xMg",
            "aHR0cDovL2NzaS5jb20vUGxheWVycy8x"
          ],
          "Description": "Centerfield"
        }
      ]
    }
  ]
}
```

```

    ]
  },
  {
    "players_key": "aHR0cDovL2NzaS5jb20vUGxheWVycy8xMA",
    "playertoposition_key": [
      "aHR0cDovL2NzaS5jb20vUG9zaXRpb25zLzI",
      "aHR0cDovL2NzaS5jb20vUG9zaXRpb25zLzM"
    ],
    "PlayerId": 10,
    "playertoteam_key": [
      "aHR0cDovL2NzaS5jb20vVGVhbXMvNA"
    ],
    "PlayerName": "Ted James",
    "DefensiveRating": 92.55,
    "PlayerToPosition": [
      {
        "positions_key": "aHR0cDovL2NzaS5jb20vUG9zaXRpb25zLzI",
        "PositionId": 2,
        "ShortName": "C",
        "positiontoplayer_key": [
          "aHR0cDovL2NzaS5jb20vUGxheWVycy84",
          "aHR0cDovL2NzaS5jb20vUGxheWVycy8xMA"
        ],
        "Description": "Catcher"
      },
      {
        "positions_key": "aHR0cDovL2NzaS5jb20vUG9zaXRpb25zLzM",
        "PositionId": 3,
        "ShortName": "1B",
        "positiontoplayer_key": [
          "aHR0cDovL2NzaS5jb20vUGxheWVycy83",
          "aHR0cDovL2NzaS5jb20vUGxheWVycy8xMA"
        ],
        "Description": "First Base"
      }
    ]
  }
]
}

```

## Related Topics

[Creating Data on Demand Endpoints](#)

[Accessing an Endpoint Programmatically](#)

[Accessing an Endpoint from an Application](#)

# Accessing Data from the SPARQL Endpoint

Anzo offers a standard HTTP(S) SPARQL endpoint for sending SPARQL requests between client applications and Anzo. The endpoint is enabled by default. This topic provides the base endpoint URL and describes the supported HTTP methods and parameters.

## Authentication

The Anzo SPARQL endpoint supports Basic Authentication. The endpoint can be configured to enable other Anzo-supported authentication methods. However, implementing alternate authentication mechanisms can have unexpected results. For more information, contact Cambridge Semantics Support.

### Note

Ultimately the data that is available to users from SPARQL endpoints depends on the access control configuration of the graphmart or linked data set as configured in Anzo.

## HTTP Methods and Options

The Anzo SPARQL endpoint accepts HTTP GET and POST methods. GET is used to retrieve data from the endpoint, and POST is used to send data to the endpoint. Update queries must use the POST method, and read queries can be submitted using GET or POST.

## Endpoint Base URL

Use the following base URL to access data in Anzo via the SPARQL endpoint. The table below describes each base URL component:

<protocol>://<hostname>:<port>/sparql/<store\_type>/<url-encoded\_dataset\_uri>

Option	Description
protocol	The protocol to use for the connection: <b>http</b> for HTTP protocol or <b>https</b> for SSL protocol.
hostname	The DNS name or IP address of the Anzo server.
port	The port for the endpoint. The port that you specify depends on the protocol that you choose. By default, the HTTP port is <b>80</b> and the HTTPS port is <b>443</b> . To view the ports

Option	Description
	that are configured for your Anzo instance, see <b>Server Settings</b> in the <b>Administration</b> menu.
<b>sparql</b>	Required keyword for the SPARQL endpoint.
<b>store_type</b>	The type of RDF store for the data. Typically users specify <b>graphmart</b> to query data that is in a graphmart. It is also possible to query the metadata for a linked data set (LDS) in the Dataset catalog. To query an LDS that is stored in a local volume, specify <b>lds</b> as the store type.
<b>url-encoded_dataset_uri</b>	<p>The URI for the graphmart or the catalog entry for the LDS. The URI must be URL-encoded using upper case hexadecimal digits. Lower case hexadecimal digits are not supported at this time.</p> <p><a href="#">How do I find the URI for a graphmart?</a></p> <p><a href="#">How do I find the catalog entry URI for a dataset?</a></p>

For example, the following base endpoint URL targets the data in a Graphmart:

```
https://10.100.10.20:8443/sparql/graphmart/http%3A%2F%2Fcambridgesemantics.com%2FGraphmart%2F1ad0ee911b834097ad7f71ee0ae1c0ff
```

The example below shows a base endpoint URL that targets a Dataset catalog entry:

```
https://10.100.10.20:8443/sparql/lds/http%3A%2F%2Fopenanzo.org%2FcatEntry(%255Bhttp%253A%252F%252Fcsi.com%252FFileBasedLinkedDataSet%252F001e517db4f0eaa9f279427e4e2a828%255D%2540%255Bhttp%253A%252F%252Fopenanzo.org%252Fdatasource%252FsystemDatasource%255D)
```

## HTTP Header Options

The HTTP header provides information related to the transfer of data between the requesting client and the SPARQL endpoint. The table below describes the supported HTTP header options. Both of the fields are optional.

Option	Description
<b>Content-</b>	The Content-Type specifies the type of request that is being sent by the client.

Option	Description
<b>Type</b>	<p>Anzo supports the following Content-Type values:</p> <ul style="list-style-type: none"> <li>• <b>application/x-www-form-urlencoded</b>: Including this value specifies that the query string will be passed as the value of a "query" or "update" HTTP parameter. This is the default value. When Content-Type is not specified, the endpoint behaves as if <code>Content-Type: application/x-www-form-urlencoded</code> is specified.</li> <li>• <b>application/sparql-query</b>: Including this value specifies that the HTTP request body includes a SPARQL read (non-update) query.</li> <li>• <b>application/sparql-update</b>: Including this value specifies that the HTTP request body includes a SPARQL update query.</li> </ul>
<b>Accept</b>	<p>The Accept field specifies the response formats that are acceptable for the server to send back to the client. You can use this field to specify the output serialization format for query results in place of the <a href="#">format</a> HTTP parameter. For details about the supported formats, see <a href="#">Format Options</a> below.</p>

## HTTP Body Parameters

The HTTP parameters in the body of the request provide the rest of the information about the request. Certain parameters are appropriate for read-only queries, SELECT and CONSTRUCT, and others are appropriate for updates, INSERT and DELETE. The tables below describe the supported parameters for query and update requests.

### Query Parameters

Parameter	Description
<b>query</b>	<p>Specifies the full read-only query string to run. If you do not specify a <a href="#">url-encoded_dataset_uri</a>, <a href="#">default-graph-uri</a> or <a href="#">named-graph-uri</a> in the request, the query string should contain the appropriate FROM clauses.</p> <p>To run an update query (INSERT or DELETE), use the <a href="#">update</a> parameter.</p>



Parameter	Description
<b>default-graph-uri</b>	Specifies a default graph URI to query. You can include this parameter multiple times in a request. When the base URL specifies a graphmart URI, you can specify a data layer URI to narrow the scope of the query to a specific data layer in the graphmart.
<b>named-graph-uri</b>	Specifies a named graph URI to query. You can include this parameter multiple times in a request. When the base URL specifies a graphmart URI, you can specify a data layer URI to narrow the scope of the query to a specific data layer in the graphmart.
<b>format</b>	Specifies the serialization format to use for the results of the query. For details about the supported formats, see <a href="#">Format Options</a> below.
<b>includeMetadataGraphs</b>	A boolean value that specifies whether to query the metadata graphs. Only valid for queries that target a linked data set (LDS) that is stored in a local volume. The default value is <b>includeMetadataGraphs=false</b> .
<b>delim</b>	Specifies a custom delimiter character to use in CSV output results. Valid only for SELECT queries where the output format is <b>text/csv</b> . This field accepts any character. When delim is not specified the default value is a , (comma).
<b>dedup</b>	A boolean value that specifies whether to deduplicate CONSTRUCT results on the client side. When dedup is not specified, the default value is <b>dedup=true</b> .
<b>serverDedup</b>	A boolean value that specifies whether to deduplicate CONSTRUCT results on the server side. When serverDedup is not specified, the default value is <b>serverDedup=true</b> .

Parameter	Description
<b>skipCache</b>	A boolean value that specifies whether to skip the reuse of any query cache that exists from a previous run of the query. When skipCache is not specified, the default value is <b>skipCache=false</b> . Anzo server's query cache should be forcibly skipped/ignored
<b>hasHeader</b>	A boolean value that specifies whether to include headers in CSV results. Valid only for SELECT queries where the output format is <b>text/csv</b> . When hasHeader is not specified, the default value is <b>hasHeader=false</b> .
<b>attachResult</b>	A boolean value that specifies whether to provide the query response as a file "attachment," i.e. the HTTP response will include the Content-Disposition of <b>attachment</b> . When attachResult is not specified, the default value is <b>attachResult=false</b> . When returning results as an attachment, you can specify a file name in <a href="#">filename</a> the parameter.
<b>filename</b>	If <a href="#">attachResult</a> is true, this parameter specifies the file name to use for the attachment, excluding the file extension. If attachResult is true and filename is not specified, the default file name is <b>QueryResult</b> .

## Format Options

The table below describes the options for specifying the serialization format of the results that the server sends back to the client. These format options, i.e., MIME types or file extensions, can be specified in the [format](#) parameter in the body of the request or in the [Accept](#) header.

### Note

When the request does not include the format parameter or Accept header, the default result format for SELECT queries is [SPARQL XML](#) (**application/sparql-results+xml**). For CONSTRUCT queries, the default format depends on whether the query includes GRAPH clauses. If no GRAPH clause is present, the default format for CONSTRUCT results is [RDF Turtle](#). If GRAPH clauses are present,

the default format is [RDF TriG](#).

Format	Accepted Values	Query Type	Description
XML	<b>application/sparql-results+xml</b> <b>application/xml</b> <b>xml</b> <b>xml2</b> <b>srx</b>	SELECT only	Returns results in <a href="#">SPARQL Query Results XML Format</a> .
	<b>application/rdf+xml</b> <b>rdf</b> <b>owl</b> <b>rdfs</b>	CONSTRUCT only	Returns results in <a href="#">RDF 1.1 XML</a> format.
JSON	<b>application/json</b> <b>json</b>	SELECT and CONSTRUCT	For SELECT queries, results are returned in <a href="#">SPARQL Query Results JSON Format</a> .  For CONSTRUCT queries, results are returned in Anzo's native JSON RDF serialization format. See <a href="#">Anzo JSON RDF Serialization</a> for details.
	<b>application/sparql-results+json</b>	SELECT only	Returns results in <a href="#">SPARQL Query Results JSON Format</a> .
CSV	<b>text/csv</b> <b>csv</b>	SELECT only	Returns results in <a href="#">SPARQL Query Results CSV Format</a> .
TriG and Gzipped TriG	<b>application/x-trig</b> <b>trig</b> <b>application/x-trigz</b>	CONSTRUCT only	CONSTRUCT queries with a GRAPH clause return <a href="#">RDF 1.1 TriG</a> by default if no format is

Format	Accepted Values	Query Type	Description
	<b>trigz</b> <b>gz</b> <b>trig.gz</b>		specified.
Turtle and Gzipped Turtle	<b>application/x-turtle</b> <b>ttl</b> <b>application/x-turtlez</b> <b>ttlz</b> <b>ttl.gz</b>	CONSTRUCT only	Returns <a href="#">RDF 1.1 Turtle</a> .  CONSTRUCT queries without a GRAPH clause return Turtle by default if no format is specified.
N-Triples	<b>text/plain</b> <b>nt</b>	CONSTRUCT only	Returns results in <a href="#">RDF 1.1 N-Triples</a> format.
Notation3 and Gzipped Notation3	<b>text/rdf+n3</b> <b>n3</b> <b>text/rdf+n3z</b> <b>n3z</b> <b>n3z.gz</b>	CONSTRUCT only	Returns results in <a href="#">RDF Notation3</a> format.
N-Quads	<b>text/x-nquads</b> <b>nq</b> <b>nquad</b> <b>nquads</b>	CONSTRUCT only	Returns results in <a href="#">RDF 1.1 N-Quads</a> format.
TriX	<b>application/trix</b> <b>trix</b>	CONSTRUCT only	Returns results in <a href="#">RDF Triples in XML</a> format.

## Update Parameters

Parameter	Description
<b>update</b>	Specifies the full update string to run. If you do not specify a <a href="#">url-encoded_dataset_uri</a> , <a href="#">using-graph-uri</a> or <a href="#">using-named-graph-uri</a> in the request, the update query should contain the appropriate

Parameter	Description
	<p>USING clauses.</p> <p>To run a non-update query (SELECT or CONSTRUCT), use the <a href="#">query</a> parameter.</p>
<b>using-graph-uri</b>	Specifies a default graph URI to update. You can include this parameter multiple times in a request. When the base URL specifies a graphmart URI, you can specify a data layer URI to narrow the scope of the update to a specific data layer in the graphmart.
<b>using-named-graph-uri</b>	Specifies a named graph URI to update. You can include this parameter multiple times in a request. When the base URL specifies a graphmart URI, you can specify a data layer URI to narrow the scope of the update to a specific data layer in the graphmart.
<b>includeMetadataGraphs</b>	A boolean value that specifies whether to query the metadata graphs. Only valid for queries that target a linked data set (LDS) that is stored in a local volume. The default value is <b>false</b> .

## Examples

The following example uses cURL to send a request that runs a SELECT query against a graphmart. Since the request does not include an Accept header or format parameter, results will be returned in SPARQL XML format.

```
curl --user sysadmin:@nz0 -c cookiejar.txt -L -v -k
http://10.100.10.20/sparql/graphmart/http%3A%2F%2Fcambridgesemantics.com%2FGrap
hmart%2F2dc579b101654ae29eb91b0c7d046ca1
--data-urlencode "query=SELECT * WHERE{ ?s ?p ?o . } LIMIT 100"
```

The following example sends a GET request that runs a SELECT query against a graphmart. The format parameter is included to format the results in text/csv serialization.

GET
https://10.102.0.17:443/sparql/graphmart/http%3A%2F%2Fcambridgesemantics.com%2FGraphmart%2Fbbf2d4b4c138403bab1c671eb6d9763...
Send
Save

Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
Code

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> format	text/csv			
<input checked="" type="checkbox"/> hasHeaders	yes			
<input checked="" type="checkbox"/> query	select * where { ?s ?p ?o } limit 100			
Key	Value	Description		

Body
Cookies (1)
Headers (14)
Test Results
Status: 200 OK
Time: 511ms
Size: 18.06 KB
Save Response

Pretty
Raw
Preview
Visualize BETA
Text

```

1 "s","p","o"
2 "http://csi.com/Shippers/1","http://www.w3.org/1999/02/22-rdf-syntax-ns#type","http://cambridgesemantics.com/ont/autogen/Fu/DB/northwind#Shippers"
3 "http://csi.com/Territories/60179","http://www.w3.org/1999/02/22-rdf-syntax-ns#type","http://cambridgesemantics.com/ont/autogen/Fu/DB/northwind#Territories"
4 "http://csi.com/Territories/31406","http://www.w3.org/1999/02/22-rdf-syntax-ns#type","http://cambridgesemantics.com/ont/autogen/Fu/DB/northwind#Territories"
5 "http://csi.com/Territories/10038","http://www.w3.org/1999/02/22-rdf-syntax-ns#type","http://cambridgesemantics.com/ont/autogen/Fu/DB/northwind#Territories"
6 "http://csi.com/Territories/98052","http://www.w3.org/1999/02/22-rdf-syntax-ns#type","http://cambridgesemantics.com/ont/autogen/Fu/DB/northwind#Territories"
7 "http://cambridgesemantics.com/ont/autogen/Fu/DB/northwind#Suppliers_Region","http://www.w3.org/1999/02/22-rdf-syntax-ns#type","http://www.w3.org/2002/07/owl#Datatype"
8 "http://cambridgesemantics.com/ont/autogen/Fu/DB/northwind#Suppliers_Region","http://www.w3.org/1999/02/22-rdf-syntax-ns#type","http://www.w3.org/2002/07/owl#Function"
9 "http://cambridgesemantics.com/ont/autogen/Fu/DB/northwind#Customers_Fax","http://www.w3.org/1999/02/22-rdf-syntax-ns#type","http://www.w3.org/2002/07/owl#DatatypePr"
10 "http://cambridgesemantics.com/ont/autogen/Fu/DB/northwind#Customers_Fax","http://www.w3.org/1999/02/22-rdf-syntax-ns#type","http://www.w3.org/2002/07/owl#Functional"
11 "http://cambridgesemantics.com/ont/autogen/Fu/DB/northwind#Suppliers_HomePage","http://www.w3.org/1999/02/22-rdf-syntax-ns#type","http://www.w3.org/2002/07/owl#Data"
12 "http://cambridgesemantics.com/ont/autogen/Fu/DB/northwind#Suppliers_HomePage","http://www.w3.org/1999/02/22-rdf-syntax-ns#type","http://www.w3.org/2002/07/owl#Funct

```

For reference, below is the URL-encoded version of the request string shown in the image above. When sending a request from a client that does not automatically encode requests, you must convert the string. Line breaks are added for readability:

```
http://10.100.10.20/sparql/graphmart/http%3A%2F%2Fcambridgesemantics.com%2F
Graphmart%2F646861d1bab54d67bc79dea94e02f3e6
?query=select%20*%20where%20%7B%3Fs%20%3Fp%20%3Fo%7D%20limit%20100
```

The example below sends a POST request that runs a SELECT query. In this example, the query is included in the body of the request and the response format is XML.

POST
https://10.102.0.17:443/sparql/graphmart/http%3A%2F%2Fcambridge semantics.com%2FGraphmart%2Fbbf2d4b4c138403...
Send
Save

Params
Authorization
Headers (11)
Body
Pre-request Script
Tests
Settings
Cookies
Code

none
form-data
x-www-form-urlencoded
raw
binary
GraphQL BETA

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	query	select * where {?s ?p ?o} limit 20			
<input checked="" type="checkbox"/>	format	xml			
	Key	Value	Description		

Body
Cookies (1)
Headers (14)
Test Results
Status: 200 OK Time: 153ms Size: 34.87 KB Save Response

Pretty
Raw
Preview
Visualize BETA
XML

```

8 <results>
9 <result>
10 <binding name='s'>
11 <uri>http://csi.com/Shippers/1</uri>
12 </binding>
13 <binding name='p'>
14 <uri>http://www.w3.org/1999/02/22-rdf-syntax-ns#type</uri>
15 </binding>
16 <binding name='o'>
17 <uri>http://cambridge semantics.com/ont/autogen/Fu/DB/northwind#Shippers</uri>
18 </binding>
19 </result>
20 <result>
21 <binding name='s'>
22 <uri>http://csi.com/Territories/60179</uri>
23 </binding>
24 </result>

```

The example below sends a GET request that runs a CONSTRUCT query. The response format is set to JSON, and the results are formatted in [Anzo JSON RDF Serialization](#).

GET
https://10.102.0.17:443/sparql/graphmart/http%3A%2F%2Fcambridge semantics.com%2FGraphmart%2Fbbf2d4b4c138403bab1c671eb6d9763...
Send
Save

Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
Code

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	query	construct { graph <http://csi.com/test> {?s ?p ?o}} where {?s ?p...			
<input checked="" type="checkbox"/>	format	json			
	Key	Value	Description		

Body
Cookies (1)
Headers (14)
Test Results
Status: 200 OK Time: 536ms Size: 6.89 KB Save Response

Pretty
Raw
Preview
Visualize BETA
JSON

```

1 {
2   {
3     "namedGraphUri": "http://csi.com/test",
4     "subject": {
5       "objectType": "uri",
6       "value": "http://cambridge semantics.com/ont/autogen/Fu/DB/northwind#Customers"
7     },
8     "predicate": "http://www.w3.org/1999/02/22-rdf-syntax-ns#type",
9     "object": {
10      "objectType": "uri",
11      "value": "http://www.w3.org/2000/01/rdf-schema#Resource"
12    }
13  },
14 }

```

The example below uses a Python script to send a request that runs a SPARQL query.

```
import requests
import urllib

server = 'https://company.anzo.com:'
port = 443
graphmart =
'http://cambridgesemantics.com/Graphmart/be4bd080c5654628b6fff90ca1b647d6'
url = server + str(port) + '/sparql/graphmart/' + urllib.quote_plus(graphmart)
#urllib.parse.quote_plus(graphmart) in Python 3

queryText = 'SELECT * WHERE {?instance a ?type .} LIMIT 10'
payload = {'query':queryText, 'format':'text/csv'}

r = requests.post(url, data = payload, auth = ('sysadmin','<pw>'))
print r.text
```

## Related Topics

[Accessing Data from the HTTP Client Interface](#)

[SPARQL Best Practices and Query Templates](#)



# Accessing Data from the HTTP Client Interface

In addition to the SPARQL HTTP(S) endpoint that enables users to send SPARQL queries to Anzo over HTTP, Anzo provides an HTTP(S) servlet that enables users to invoke Anzo client operations over HTTP. The client servlet enables external systems to interact with Anzo semantic services as well as custom services. It also enables remote servers to interact with Anzo without needing the Anzo command line interface.

## HTTP Methods and Options

The Anzo client servlet accepts HTTP GET and POST methods. GET is used for operations that retrieve data, and POST is used for update operations that add or remove data. Update operations must use the POST method, and read operations can be submitted using GET or POST.

## Client Servlet Base URL

Use the following URL to access Anzo services via the HTTP client servlet. The table below describes each URL component:

```
<protocol>://<hostname>:<port>/anzoclient/<client_operation>
```

For example:

```
https://10.100.10.20:8443/anzoclient/call
```

Option	Description
protocol	The protocol to use for the connection: <b>http</b> for HTTP protocol or <b>https</b> for SSL protocol.
hostname	The DNS name or IP address of the Anzo server.
port	The port for the endpoint. The port that you specify depends on the protocol that you choose. By default, the HTTP port is <b>80</b> and the HTTPS port is <b>443</b> . To view the ports that are configured for your Anzo instance, see <b>Server Settings</b> in the <b>Administration</b> menu.
anzoclient	Required keyword for the client servlet.

Option	Description
<b>client_operation</b>	<p>The type of Anzo client operation to invoke. The list below provides an overview of the supported operation types. For more information about the operations, see <a href="#">Client Operations</a> below.</p> <ul style="list-style-type: none"> <li>• <b>call</b>: Invokes the semantic service operation identified by the URI provided in the request (analogous to the <code>anzo call</code> CLI command)</li> <li>• <b>add</b>: Imports the specified statements to Anzo (analogous to the <code>anzo import</code> CLI command)</li> <li>• <b>remove</b>: Removes the specified statements from Anzo (analogous to the <code>anzo update -r</code> CLI command)</li> <li>• <b>get</b>: Gets the specified named graph from Anzo (analogous to the <code>anzo get</code> CLI command)</li> <li>• <b>find</b>: Finds the statements in Anzo that match the specified pattern (analogous to the <code>anzo find</code> CLI command)</li> </ul>

## Client Operations

This section provides usage information and examples for each of the Anzo client operations.

- [Call](#)
- [Add](#)
- [Remove](#)
- [Get](#)
- [Find](#)

### Call

The call operation invokes a semantic service. Identify the service to call by providing the URI for the service in the request header. The call operation is supported with HTTP GET and POST methods. When including RDF data as input to the service, the request must use the POST method.

### Call Header Options

Call operations support the following header parameters. Only the **uri** parameter is required:

- **uri:** **Required** parameter that specifies the URI of the semantic service to invoke.
- **mimeType**, **Content-Type**, or **format**: Include one of these optional parameters to specify the MIME type for the RDF serialization used in the request body as well as the response from the service. The default type is **application/json** if the header does not specify the format. For more information about the supported RDF serialization types, see [Format Options](#).

## Call Body Options

If the call operation supplies data as input to the service, include the data in the request body. The data must be serialized as specified in the request header, or **application/json** if the header does not specify a serialization type.

## Call Examples

The following cURL example uses a GET call to invoke a health check service.

```
curl https://10.100.10.20:8443/anzoclient/call \
  --user sysadmin:123 \
  --header 'uri: http://www.csi.com/service/genericIngestManager#healthCheck'
```

The example below uses a POST call to invoke a service operation. The call passes in a request data set that is serialized as RDF JSON.

```
curl https://10.100.10.20:8443/anzoclient/call \
  --header 'Content-Type: application/json' \
  --user sysadmin:123 \
  --header 'uri: http://someServiceURI#someOperation' \
  --data '{"subject" : {"objectType": "uri" ,"value" : "urn://test"},
        "predicate" : "urn://predicate",
        "object" : {"objectType": "uri" ,"value" : "urn://object"},
        "namedGraphUri" : "urn://ng"}'
```

The example below uses a POST call to invoke a service operation. The call passes in a request data set that is serialized as TriG.

```
curl https://10.100.10.20:8443/anzoclient/call \
  --header 'Content-Type: application/x-trig' \
  --user sysadmin:123 \
  --header 'uri: http://www.csi.com/service/genericIngestManager#healthCheck'
  --data '<urn://ng> { <urn://test> <urn://predicate> <urn://object> .}'
```

## Add

The add operation adds statements to the Anzo graphstore. Add is supported with the HTTP POST method. The header can include **contentType**, **Content-Type**, or **format** to specify the MIME type for the RDF serialization used in the request body as well as the response from the service. The default type is **application/json** if the header does not specify the format. For more information about the supported RDF serialization types, see [Format Options](#). The request body includes the statements to add.

### Add Examples

The following example add operation uses cURL to issue a POST call to add a statement to the graphstore. The statement is specified in Anzo JSON RDF serialization format.

```
curl https://10.100.10.20:8443/anzoclient/add \
--user sysadmin:123 \
--data '{"subject" : {"objectType": "uri" ,"value" : "urn://test"},
      "predicate" : "urn://predicate",
      "object" : {"objectType": "uri" ,"value" : "urn://object"},
      "namedGraphUri" : "urn://ng"}'
```

## Remove

The remove operation deletes statements from the Anzo graphstore. Remove is supported with the HTTP POST method. The header can include **contentType**, **Content-Type**, or **format** to specify the MIME type for the RDF serialization used in the request body as well as the response from the service. The default type is **application/json** if the header does not specify the format. For more information about the supported RDF serialization types, see [Format Options](#). The request body specifies the statements to remove.

### Remove Examples

The following example remove operation uses cURL to issue a POST call to remove a statement from the graphstore. The statement is specified in Anzo JSON RDF serialization format.

```
curl https://10.100.10.20:8443/anzoclient/remove \
--user sysadmin:123 \
--data '{"subject" : {"objectType": "uri" ,"value" : "urn://test"},
      "predicate" : "urn://predicate",
      "object" : {"objectType": "uri" ,"value" : "urn://object"},
      "namedGraphUri" : "urn://ng"}'
```

## Get

The get operation retrieves a named graph from the Anzo graphstore. The get operation is supported with HTTP GET and POST methods. The named graph URI that contains the contents to retrieve can be included as a query parameter or as a **uri** parameter in the request body. The get operation also returns the metadata graph, which is equivalent to running `anzo get -m <named_graph_uri>` with the Anzo admin CLI. The header can include **mimeType**, **Content-Type**, or **format** to specify the MIME type for the RDF serialization used to format the response from the service. The default type is **application/json** if the header does not specify the format.

## Get Examples

The following example get operation uses cURL to retrieve the contents of a named graph.

```
curl -k -XPOST https://10.100.10.20:8443/anzoclient/get --user sysadmin:123
--data-urlencode

"uri=http://cambridgesemantics.com/Graphmart/9da211618a15476daa10cead2292d8e7"
```

This example uses Python with requests:

```
import requests

url = "https://10.100.10.20:8443/anzoclient/get"
data = {"uri":
"http://cambridgesemantics.com/Graphmart/9da211618a15476daa10cead2292d8e7"}
username = "sysadmin"
password = "123"
r = requests.post(url, data=data, auth=(username, password), verify=False)
print (r.text)
```

## Find

The find operation finds the statements in the graphstore that match the pattern that is specified in the request. The find operation is supported with HTTP GET and POST methods. Header options are not applicable. The list below describes each of the supported parameters. These parameters can be included as query parameters in the URL or as parameters in the request body:

- **graph**: The named graph URI for the find pattern.
- **sub**: The subject of the find pattern.

- **pred:** The predicate of the find pattern.
- **lit:** The object of the find pattern if that object is a literal value.
- **uri:** The object URI of the find pattern if that object is a URI.
- **type:** If the object is a literal, this parameter can be used to specify the data type of the literal value.
- **lang:** If the object is a literal, this parameter can be used to specify the language of the literal value.

Results returned by the find operation are in Anzo JSON RDF serialization format. See [Anzo JSON RDF Serialization](#) below for details.

## Find Examples

The following example find operation (using the GET HTTP method) finds all of the statements in the graphstore with predicate `http://w3.org/1999/02/22-rdf-syntax-ns#type` and an object URI of `http://cambridgesemantics.com/ontologies/2009/05/LinkedData#LinkedDataSet`. The parameters are specified as query parameters in the URL.

```
curl
https://10.100.10.20:8443/anzoclient/find?pred=http://www.w3.org/1999/02/22-rdf-
syntax-
ns%23type&uri=http://cambridgesemantics.com/ontologies/2009/05/LinkedData%2
3LinkedDataSet' \
--user sysadmin:123
```

The example below finds the same statements but issues a POST call. The URL-encoded parameters are specified in the request body.

```
curl https://10.100.10.20:8443/anzoclient/find \
--user sysadmin:123
--data 'pred=http%3A%2F%2Fwww.w3.org%2F1999%2F02%2F22-rdf-syntax-
ns%23type&uri=http%3A%2F%2Fcambridgesemantics.com%2Fontologies%2F2009%2F05%2FLi
nkedData%23LinkedDataSet'
```

## Anzo JSON RDF Serialization

Anzo's JSON RDF serialization standard is straightforward but differs from the common public JSON RDF serialization standards. In Anzo JSON serialization format, a set of statements (quads) are represented as

an array of JSON objects. Each JSON object (statement) is defined as a key/value pair, where the key specifies the component of the statement, i.e., the subject, predicate, object, or namedGraphUri. Depending on the component, properties such as the component's value and data type are specified in nested objects.

The following example array shows Anzo's JSON serialization. The list below the example describes the structure.

```
[
  {
    "subject" : {
      "objectType": "uri" ,
      "value" : "urn://test"
    },
    "predicate" : "urn://predicate",
    "object" : {
      "objectType": "uri" ,
      "value" : "urn://object"
    },
    "namedGraphUri" : "urn://ng"
  },
  {
    "subject" : {
      "objectType": "uri" ,
      "value" : "urn://test"
    },
    "predicate" : "urn://predicate2",
    "object" : {
      "objectType": "literal" ,
      "value" : "test literal",
      "dataType" : "http://www.w3.org/2001/XMLSchema#string"
    },
    "namedGraphUri" : "urn://ng"
  }
]
```

- **subject** is a JSON object with two properties:
  - **objectType**: The resource type of the subject value. This is either a "uri" or "bnode" (blank node).
  - **value**: The blank node value or a string literal that specifies the URI.

- **predicate** is a string literal that specifies the predicate URI.
- **object** is a JSON object with two required properties and two optional properties:
  - **objectType**: Required property that specifies whether the object is a "uri," "literal," or "bnode."
  - **value**: Required property that specifies the string representation of the object value.
  - **dataType**: Optional property for use if the objectType is "literal." This property describes the data type of the literal value. It is a string literal of the XSD data type URI. For example:  
"http://www.w3.org/2001/XMLSchema#string"
  - **language**: Optional property for use if the objectType is "literal." This property describes the language of the literal value.
- **namedGraphUri** is a string literal that specifies the named graph URI.

## Related Topics

[Accessing Data from the SPARQL Endpoint](#)

[SPARQL Best Practices and Query Templates](#)



# Sharing Access to Artifacts

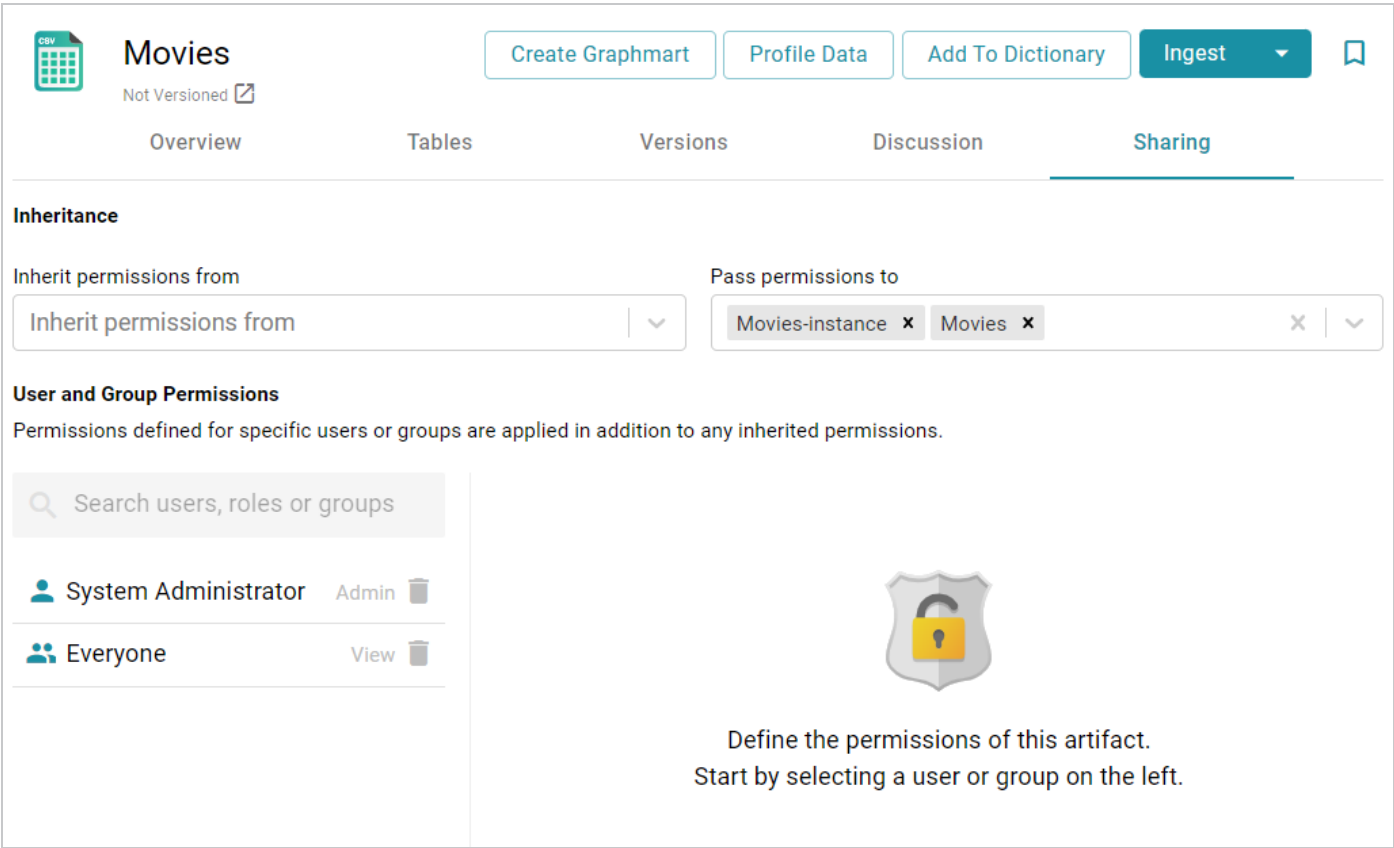
All Anzo artifacts—data sources, schemas, models, graphmarts, etc.—that you create can be shared with other groups (or users) from the **Sharing** tab in the Anzo application. This topic provides an overview of the Sharing tab and basic instructions for configuring artifact permissions.

Note

For specifics about sharing multifaceted artifacts like graphmarts that include layers and dashboards that include multiple lenses, see [Sharing Access to Graphmarts](#) and [Sharing Access to a Dashboard or Lens](#).

## Sharing Tab Overview

Access the Sharing tab by navigating to an artifact and clicking **Sharing**. For example, the image below shows the Sharing tab for a data source.



## Inheritance

The top of the screen displays the permission inheritance settings:

**Inheritance**

Inherit permissions from

Inherit permissions from | ▾

Pass permissions to

Movies-instance x Movies x x | ▾

To facilitate common workflows, the Anzo application applies logic so that artifacts in the same workflow inherit the same permissions. You can alter the inheritance configuration by choosing the artifact or artifacts that this artifact should inherit from by choosing the artifacts to **Inherit permissions from**. Since the example above is a data source and no artifacts precede the source in Anzo, the Inherit Permissions From setting is empty.

You can also configure an artifact to **Pass permissions to** other artifacts. In the example above, the Movies data source passes permissions to the Movies schema instance and the Movies graphmart.

**Pass permissions to**

Movies-instance x Movies x x | ▾

### Note


Permissions are additive. Copying permissions from multiple artifacts with differing permission levels results in the super set being acquired by the artifact that is inheriting the permissions. In addition, any permissions that are configured in the table at the bottom of the screen are also added to the set. For more conceptual information about permission inheritance, see [Permission Inheritance](#) in the Administration Guide.


## User and Group Permissions


In addition to the inheritance settings, the bottom of the screen lists the users and groups that this artifact has been shared with. For example:


## User and Group Permissions

Permissions defined for specific users or groups are applied in addition to any inherited permissions.

 System Administrator

Admin 

 Everyone

View 



Define the permissions of this artifact.  
Start by selecting a user or group on the left.

The current level of access is listed next to each name: **View**, **Modify**, or **Admin**. View, Modify, and Admin are predefined permission sets. Each predefined set selects a certain combination of six permissions. You also have the option to create a **Custom** set of permissions. Selecting a user or group from the list displays the following permissions table on the right side of the screen:

Permissions	<input checked="" type="radio"/> View	<input type="radio"/> Modify	<input type="radio"/> Admin	<input type="radio"/> Custom
Add/Edit		✓	✓	<input checked="" type="checkbox"/>
View	✓	✓	✓	<input checked="" type="checkbox"/>
Delete		✓	✓	<input checked="" type="checkbox"/>
Meta Add/Edit			✓	<input checked="" type="checkbox"/>
Meta View	✓	✓	✓	<input checked="" type="checkbox"/>
Meta Delete			✓	<input checked="" type="checkbox"/>

Typing a value in the **Search users, roles or groups** field finds and displays the users or groups that you can add to the list. The [Permission Settings](#) section below defines each of the permission sets.

## Permission Settings

The table below lists the predefined permission sets and describes the privileges that are granted for each permission that is part of the set:

Set	Permission	Allows a user to:
View	View	<ul style="list-style-type: none"> <li>• See the artifact in the Anzo application.</li> <li>• Create versions of the artifact.</li> </ul>
	Meta View	<ul style="list-style-type: none"> <li>• Relates only to an artifact's permissions. A user with Meta View can see the permissions on the Sharing tab but they cannot modify, add, or remove permissions.</li> </ul>
Modify	In addition to the <b>View</b> and <b>Meta View</b> permissions described above, the <b>Modify</b> set includes the <b>Add/Edit</b> and <b>Delete</b> permissions described below.	
	Add/Edit	<ul style="list-style-type: none"> <li>• Change an artifact, such as to rename it or edit its description.</li> <li>• Add an entity to an artifact. For example, add a schema to a data source or a layer to a graphmart.</li> </ul>
	Delete	<ul style="list-style-type: none"> <li>• Remove an entity from the artifact. For example, delete a layer from a graphmart or a schema from a data source.</li> <li>• <b>Does not give permission to remove the parent artifact.</b> For example, a user can remove a schema from a source but cannot delete the data source.</li> </ul>
Admin	In addition to the <b>View</b> , <b>Meta View</b> , <b>Add/Edit</b> , and <b>Delete</b> permissions described above, the <b>Admin</b> set includes the <b>Meta Add/Edit</b> and <b>Meta Delete</b> permissions described below.	
	Meta Add/Edit	<ul style="list-style-type: none"> <li>• Relates only to an artifact's permissions. A user with Meta Add/Edit can add permissions to a user or group. They cannot remove permissions from any user or group.</li> </ul>

Set	Permission	Allows a user to:
	Meta Delete	<ul style="list-style-type: none"> <li>Remove permissions from a user or group.</li> <li>Delete the parent artifact and its related entities.</li> </ul>

## Sharing an Artifact

Follow the instructions below to share access to an artifact.

1. In the Anzo application, navigate to the artifact that you want to share access to. Then click the **Sharing** tab.
2. If you want to change the inheritance for the artifact, use the fields at the top of the screen:
  - To apply all of the permissions from another artifact to this one, select the artifact to inherit from in the **Inherit permissions from** field.
  - To pass this artifact's permissions to other artifacts, select the artifacts to pass permissions to in the **Pass permissions to** field.

### Note


Permissions are additive. Copying permissions from multiple artifacts with differing permission levels results in the super set being acquired by the artifact that is inheriting the permissions. In addition, any permissions that are configured in the table at the bottom of the screen are also added to the set. For more conceptual information about permission inheritance, see [Permission Inheritance](#) in the Administration Guide.

3. To share access to this artifact with a particular user or group, type a value in the **Search users, roles or groups** field to find and display the user or group. The resulting list shows the current permission level that is set for each user or group in the search results. For example, the image below shows the current permissions for the IT group (None):

### User and Group Permissions

Permissions defined for specific users or groups are applied in addition to any inherited permissions.

Information Technology (IT)
None



Define the permissions of this artifact.  
Start by selecting a user or group on the left.

- Select the user or group for which you want to configure permissions. The permissions settings are displayed on the right side of the screen. For example:

### User and Group Permissions

Permissions defined for specific users or groups are applied in addition to any inherited permissions.

Information Technology (IT)
None

Information Technology (IT)
Set permission for Information Technology (IT)

Permissions
☐ View
☐ Modify
☐ Admin
☐ Custom

Add/Edit		✓	✓	<input checked="" type="checkbox"/>
View	✓	✓	✓	<input checked="" type="checkbox"/>
Delete		✓	✓	<input checked="" type="checkbox"/>
Meta Add/Edit			✓	<input checked="" type="checkbox"/>
Meta View	✓	✓	✓	<input checked="" type="checkbox"/>
Meta Delete			✓	<input checked="" type="checkbox"/>

- To assign a predefined set of permissions, click the **View**, **Modify**, or **Admin** radio button to assign that level of access to the selected user or group. Refer to [Permission Settings](#) above for details about the permissions sets. For example, the image below gives **Admin** permissions to users in the IT group:

User and Group Permissions

Permissions defined for specific users or groups are applied in addition to any inherited permissions.

info

Information Technology Admin (IT)

Information Technology (IT)

Set permission for Information Technology (IT)

Permissions

☐ View

☐ Modify

☒ Admin

☐ Custom

Add/Edit	✓	✓	✓
View	✓	✓	✓
Delete	✓	✓	✓
Meta Add/Edit		✓	✓
Meta View	✓	✓	✓
Meta Delete		✓	✓

### Note

If you want to customize the permissions, click the **Custom** radio button and then select or deselect the permissions checkboxes. To clear permissions for a user or group, click the trashcan icon (🗑️) next to the name.

Repeat the steps above to share the artifact with additional groups. Changes to permissions take effect immediately. Users do not need to log out of the application and log back in.

## Related Topics

[Sharing Access to Graphmarts](#)

[Sharing Access to a Dashboard or Lens](#)

# SPARQL Best Practices and Query Templates

To provide guidance on developing performant SPARQL queries and avoiding unexpected results, this section offers SPARQL best practices and query templates that you can use as a starting point for writing SPARQL queries in Anzo, such as in data layers, dashboard query lenses, and the Query Builder.

SPARQL Best Practices .....	981
SPARQL Query Templates .....	988



## SPARQL Best Practices

This topic provides some background information on query execution and gives general guidelines to follow to help ensure that your SPARQL queries are optimized and do not exhaust Anzo or AnzoGraph resources.

- [Query Execution Overview](#)
- [SPARQL Clause Execution Order and Optimization](#)
- [General Guidelines](#)

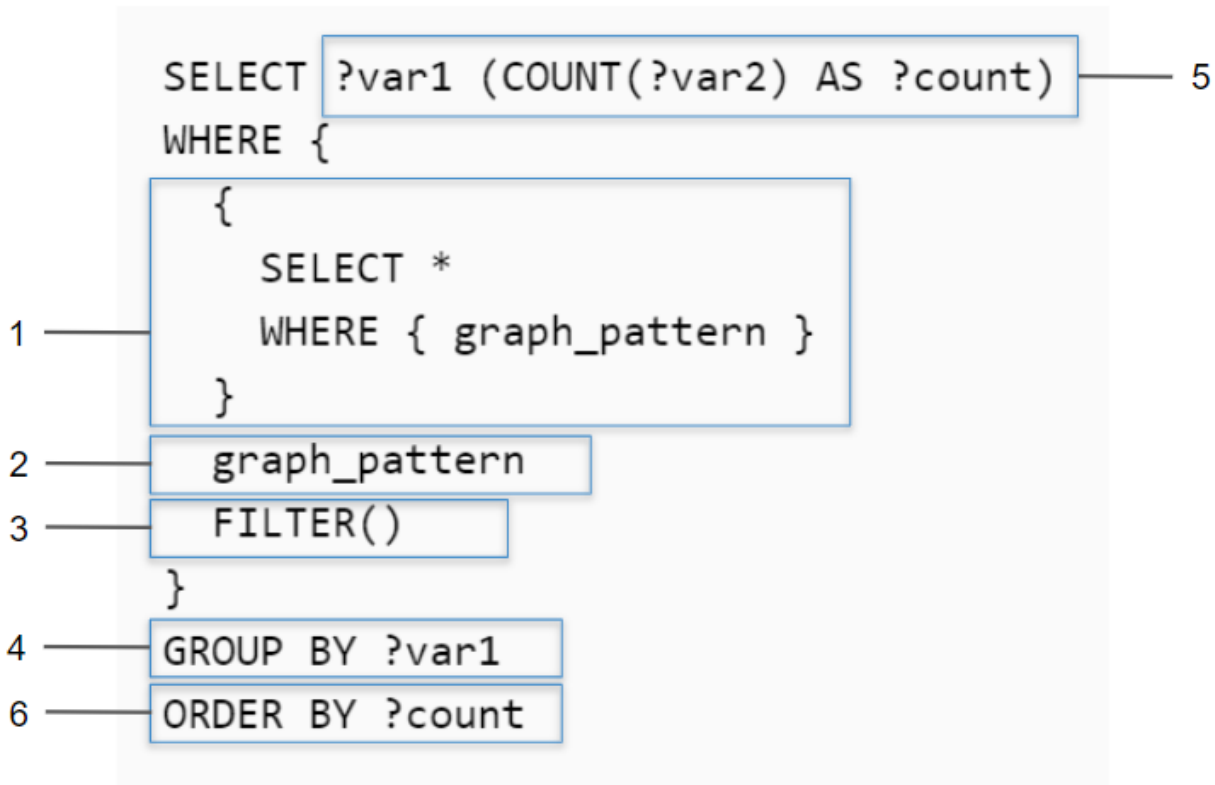
### Query Execution Overview

When you run a new query against a graphmart, AnzoGraph parses the text and then plans the query execution strategy. The planner determines the steps that are required to compute the results in an optimal way. The plan is based on the statistics about your data that is gathered during graphmart activation. After the plan is determined, code for the plan is generated, compiled, and then distributed throughout the cluster to be executed. The plan XML file and compiled code are both saved to disk so that they can be reused when the query is run again. That is why the first run of a query takes longer than subsequent runs.

In addition to the caching that AnzoGraph performs, Anzo caches the query results. If you enable the **Skip server cache** option when running a query in the Query Builder, it means that Anzo will not access the cache that is available from a previous run of the query.

### SPARQL Clause Execution Order and Optimization

The following image shows the execution order of clauses in a read (SELECT or CONSTRUCT) query. The list below describes the order.



1. First, any subselects or subqueries are performed. Note that the ordering presented in the image also applies to the clauses included in subqueries.
2. Then the graph patterns that are not included in a subquery are scanned.
3. Next, filtering is applied if a FILTER clause is included.
4. Once the WHERE clause has been processed, grouping is performed if a GROUP BY clause is included.
5. Next, the result clause is processed, including keywords like DISTINCT and any aggregation or other functions.
6. Lastly, the results are ordered when ORDER BY is specified. The ORDER BY clause is more complex when GROUP BY and LIMIT are included:  $N \log(M)$  where N is the number of grouped solutions and M is the LIMIT.

The GROUP BY and result clauses (steps 4 and 5) add linear complexity on top of the complexity in the WHERE clause. Query execution time increases proportionally to the total number of solutions found by the WHERE clause. To reduce complexity, trim the WHERE clause to have as few solutions as possible so that the expensive clauses are executed against as few solutions as possible.

## General Guidelines

This section offers guidance on developing performant queries and avoiding unexpected results.

- [Limit the Results](#)
- [Replace FILTER with VALUES or Triple Patterns](#)
- [Avoid Cross-Product Joins](#)
- [Use Subqueries when Querying Large Amounts of Data](#)

### Limit the Results

The easiest way to reduce query execution time in some cases is to apply a LIMIT statement to limit the result set to a specific number of solutions. Limiting the number of results improves performance for cases where query results are calculated and returned in a streaming fashion. Limiting results is particularly useful when results need to be ordered so that the first group of results are the only ones of interest.

#### Example: Find the sample ID and binding density for the top 10 most dense samples

```
PREFIX biom: <http://identifiers.csi.com/pharmakg/def/biomarker#>

SELECT ?sampleId ?bindingDensity
WHERE {
    ?sample a biom:Sample ;
        biom:sampleId ?sampleId ;
        biom:bindingDensity ?bindingDensity .
}
ORDER BY DESC(?bindingDensity)
LIMIT 10
```

### Replace FILTER with VALUES or Triple Patterns

While a FILTER clause is useful for narrowing down selected data per a set of requirements, only use FILTER when the logic does not lend to other operations. In many cases, replacing FILTER with a VALUES clause or a well-organized set of triple patterns increases query performance. When processing a FILTER statement, all non-filtered data must be retrieved before the FILTER can be applied. Using a VALUES clause or triple pattern, however, reduces the amount of data that is retrieved and processed after the retrieval.

#### Example 1: Inappropriate use of FILTER for a value-driven SELECT query

```
PREFIX uriRoot: <http://example.com/rootOfUri#>
```

```

SELECT ?var1 ?var2
WHERE {
    ?instanceOfClass a uriRoot:Class ;
        uriRoot:varName1 ?var1 ;
        uriRoot:varName2 ?var2 ;
        uriRoot:filteredVar ?filteredVar .

    FILTER(?filteredVar = 'COMPAREDDATA1' || ?filteredVar = 'COMPAREDDATA2' ||
?filteredVar = 'COMPAREDDATA3')
    # filteredVar is first retrieved and then run through several comparisons
}

```

### Solution 1: Use VALUES to select certain values

```

PREFIX uriRoot: <http://example.com/rootOfUris#>

SELECT ?var1 ?var2
WHERE {
    ?instanceOfClass a uriRoot:Class ;
        uriRoot:varName1 ?var1 ;
        uriRoot:varName2 ?var2 ;
        uriRoot:filteredVar ?valueVar .

    VALUES (?valueVar) {
        ('COMPAREDDATA1')
        ('COMPAREDDATA2')
        ('COMPAREDDATA3')
    }
    # selection is performed once for each entry in the VALUES clause,
    # retrieving no more data than necessary
}

```

### Example 2: Inappropriate use of FILTER for a value-driven SELECT query

```

PREFIX uriRoot: <http://example.com/rootOfUris#>

SELECT ?var1 ?filteredVar
WHERE {
    ?instanceOfClass a uriRoot:ClassName ;
        uriRoot:varName1 ?var1 ;
        uriRoot:varName2 ?var2 ;

```

```

uriRoot:filteredVar ?filteredVar .
FILTER(?filteredVar = 'COMPAREDDATA1')
# filteredVar is first retrieved and then compared
}

```

## Solution 2: Use a triple literal to select certain values

```

PREFIX uriRoot: <http://example.com/rootOfUris#>

SELECT ?var1 ?filteredVar
WHERE {
  ?instanceOfClass a uriRoot:Class ;
    uriRoot:varName1 ?var1 ;
    uriRoot:filteredVar 'COMPAREDDATA' .
  # data is only retrieved if filteredVar matches desired compared data upon
  initial retrieval
}

```

## Avoid Cross-Product Joins

When trying to gather data from multiple classes at once, it is possible to accidentally create a cross-product join, a selection that combines the selected data in a hyper-linear way rather than simply assembling the data and returning an unprocessed set.

### Example: Accidental cross-product query

```

PREFIX uriRoot: <http://example.com/rootOfUris#>

SELECT ?var1 ?var2
WHERE {
  ?instanceOfClass1 a uriRoot:Class1 ;
    uriRoot:varName1 ?var1 .
  ?instanceOfClass2 a uriRoot:Class2 ;
    uriRoot:varName2 ?var2 .
}

```

In the above example, the goal may have been to retrieve IDs from all instances of Class1 and all instances of Class2, for example, all of the participants and all of the subjects. However, the result of the query would

be every combination of participant and subject. If there are 10 participants and 5 subjects, there would be 50 results rather than 15. In large data sets, this severely affects performance and puts the system under unnecessary strain.

There are two straightforward ways to separate or parameterize data to write a more performant query:

### **Solution 1: Use UNION to replace the cross-product**

```
PREFIX uriRoot: <http://example.com/rootOfUris#>

SELECT ?commonVar
WHERE {
  {
    ?instanceOfClass1 a uriRoot:Class1 ;
      uriRoot:varName1 ?var1 .
    BIND(?var1 as ?commonVar)
  }
  UNION

  {
    ?instanceOfClass2 a uriRoot:Class2 ;
      uriRoot:varName2 ?var2 .
    BIND(?var2 as ?commonVar)
  }
  # this creates a temporary graph that is a union of two graphs
  # in each of the two graphs, the desired data is saved under the same name
}
```

### **Solution 2: Use VALUES to replace the cross-product**

```
PREFIX uriRoot: <http://example.com/rootOfUris#>

SELECT ?commonVar
WHERE {
  ?instanceOfClass a ?classURI ;
    ?propertyURI ?commonVar .

  VALUES (?classURI ?propertyURI) {
    (uriRoot:Class1 uriRoot:varName1)
    (uriRoot:Class2 uriRoot:varName2)
  }
}
```

```
}  
}
```

## Use Subqueries when Querying Large Amounts of Data

When analyzing data, there may be a need to aggregate data and then perform a selection or derivation on the resulting aggregate. In this case, it is advisable to use one or more subselects or subqueries, where a SELECT query is included inside the WHERE clause and the remainder of the WHERE clause operates on the results of that SELECT as though the data were immediately available in the graph.

### Example: Aggregate a variable and then process the aggregation

```
PREFIX uriRoot: <http://example.com/rootOfUris#>  
  
SELECT ?var1 ?var2Aggregation  
WHERE {  
  {  
    SELECT ?var1 (GROUP_CONCAT(?var2) as ?var2Aggregation)  
    WHERE {  
      ?instanceOfClass1 a uriRoot:Class1 ;  
        uriRoot:varName1 ?var1 .  
      ?instanceOfClass2 a uriRoot:Class2 ;  
        uriRoot:varName2 ?var2 .  
    }  
    GROUP BY ?var1  
  }  
  # var1 and var2Aggregation are now available for the usual processing  
  # while var2 is no longer available as it only existed within the subselect  
  
  FILTER(regex(?var2Aggregation, 'DESIREDVAR2VAL'))  
  # FILTER is used for illustrative purposes, but any processing would work  
}
```

## Related Topics

[SPARQL Query Templates](#)

## SPARQL Query Templates

This topic provides templates that you can use as a starting point for writing SPARQL queries. The templates are based on the best practices described in [SPARQL Best Practices](#).

- [Basic Data Selection](#)
- [Graph Traversal Data Selection](#)
- [Text Cleanup with REGEX](#)
- [Data Aggregation](#)
- [Applying a Filter to Selected Data](#)
- [Creating or Deriving New Variables](#)

### Basic Data Selection

The most fundamental use case for writing SPARQL queries is to select data from properties from a collection of instances. The following template and example query illustrate how to access a class in a model and return the properties on that class using their URIs.

### Abstracted Query Template

Replace the bold text to modify the query.

```
PREFIX uriRoot: <http://example.com/rootOfUri#>

# select the variables that are populated in the WHERE clause
SELECT ?var1 ?var2
WHERE {
  ?instanceOfClass a uriRoot:ClassName ;
    uriRoot:varName1 ?var1 ;
    # use a prefix to abbreviate a property URI as shown above
    # or use the full URI as shown below
    <http://example.com/rootOfUri#varName2> ?var2 .
}
```

### Example: Get the sample ID and anatomical location for each sample

```
PREFIX bm: <http://identifiers.csi.com/pharmakg/def/biomarker#>

SELECT ?sampleId ?anatomicalLocation
```



```
WHERE {
    ?sample a bm:Sample ;
        bm:sampleId ?sampleId ;
        <http://identifiers.csi.com/pharmakg/def/biomarker#fmi_anatomicalLocation>
?anatomicalLocation .
}
```

## Graph Traversal Data Selection

The graph model enables the flexibility to combine data from different classes. The following template illustrates how to traverse between classes in the data model and access data from properties on multiple classes.

### Abstracted Query Template

Replace the bold text to modify the query.

```
PREFIX uriRoot: <http://example.com/rootOfUris#>
# select the variables that are populated in the WHERE clause
SELECT ?var1 ?var2 ?varFromOtherClass
WHERE {
    ?instanceOfClass a uriRoot:ClassName ;
        uriRoot:varName1 ?var1 ;
        # use a prefix to abbreviate a property URI as shown above
        # or use the full URI as shown below
        <http://example.com/rootOfUris#varName2> ?var2 ;
        # getting data from other classes requires traversing per the model
        uriRoot:pointerToOtherClass ?instanceOfOtherClass .

    ?instanceOfOtherClass a uriRoot:OtherClassName ;
        uriRoot:varName3 ?varFromOtherClass .
}
```

## Text Cleanup with REGEX

Once data is onboarded to Anzo, it is common to encounter strings that include issues such as unintended characters, missing spaces, and inconsistent formatting. You can use regular expressions in a data layer query to manipulate those values so that they are consistent and readable in analytics against the graphmart.

The BIND clause in the query below trims any white space from before and after the string, converts the characters to upper case, and removes all non-alphanumeric characters and non-spaces. Replace the bold text as needed:

```
PREFIX : <http://csi.com/>
DELETE {
  GRAPH ${targetGraph}{
    ?s ?pred ?old_val
  }
}
INSERT {
  GRAPH ${targetGraph}{
    ?s ?pred ?new_val
  }
}
${usingSources}
WHERE {
  ?s a :Class ;
  ?pred ?old_val .

  VALUES (?pred) {
    (:property)
  }
}
BIND(TRIM(UPPER(REPLACE(?val, "[^a-zA-Z0-9[:space:]]", ""))) as ?new_val)
}
```

## Data Aggregation

Grouping data selections around a central property yields a more complete representation or summary of the data available. The following template illustrates how to use one property to act as a pivot point for collecting all the data from another property.

### Abstracted Query Template

Replace the bold text to modify the query

```
PREFIX pref: <http://example.com/rootOfUris#>

SELECT
# data can be aggregated to yield counts, concatenations of data, etc.
```

```

?instanceId GROUP_CONCAT(DISTINCT(?instanceDetail) as ?instanceDetails)
WHERE {
  # apply selection/filtering logic to narrow the aggregation
  # or get summaries of total data by applying only simple restrictions
  ?instance a pref:Class ;
    pref:instanceId ?instanceId ;
    pref:instanceDetail ?instanceDetail .
}
GROUP BY ?instanceId
# all non-aggregated variables must be grouped in GROUP BY

```

## Applying a Filter to Selected Data

Filtering the results for a query gives the ability to focus on specific aspects of the data. The following template illustrates how to restrict the total selected result set by including a filter on a variable.

## Abstracted Query Template

Replace the bold text to modify the query.

```

PREFIX pref1: <http://example.com/rootOfUris1#>
PREFIX pref2: <http://example.com/rootOfUris2#>

SELECT ?varFromClass1 ?varFromClass2 ?varFromClass3 ?filteredVar
WHERE {
  ?instance1 a pref1:Class1 ;
    pref1:varName1 ?varFromClass1 ;
    # the path on the model points from Class1 to Class2
    pref1:pointerToClass2 ?instance2 .

  ?instance2 a pref1:Class2 ;
    pref1:varName2 ?varFromClass2 .

  # models with different prefixes can still be joined
  ?instance3 a pref2:Class3 ;
    # the path on the model points from Class3 to Class2
    pref2:pointerToClass2 ?instance2 ;
    pref2:filteredVarName ?filteredVar .

  # filters use comparisons to scope the selected data

```

```
# they can use existence checks or other boolean expressions as well
FILTER(?filteredVar = 'COMPAREDDATA')
}
```

#### Tip

For optimal query performance, replace FILTER clauses. See [Replace FILTER with VALUES or Triple Patterns](#) for more information.

## Creating or Deriving New Variables

Storing intermediate or derived data within a query enables a single query to answer more complex questions. The following template illustrates how to bind a derived value to a variable. That variable is then available for selection or further manipulation.

### Abstracted Query Template

Replace the bold text to modify the query.

```
PREFIX pref1: <http://example.com/rootOfUris1#>
PREFIX pref2: <http://example.com/rootOfUris2#>
PREFIX pref3: <http://example.com/rootOfUris3#>

SELECT ?var1 ?filterVar ?var2AndVar3
WHERE {
    ?instance1 a pref1:Class1 ;
        pref1:varName1 ?var1 .

    ?filterInstance a pref2:MedicalHistory ;
        pref2:filterVarName ?filterVar ;
        # multiple traversals between classes may be necessary to link appropriate
data
        pref2:pointerToIntermediateClass ?intermediateInstance .

    ?intermediateInstance a pref2:IntermediateClass ;
        pref2:pointerToClass1 ?instance1 .

    ?instance2 a pref3:Class2 ;
        # forwards traversals tend to be more performant
        # it is still possible to identify a latter class and do a backwards
traversal
```

```
    pref3:pointerToClass1 ?instance1 ;
    pref3:varName2 ?var2 .

?instance3 a pref3:Class3 ;
    pref3:pointerToClass2 ?instance2 ;
    pref3:varName3 ?var3 .

# filters can be executed on various data types
FILTER(?filterVar < "filterData"^^xsd:filterDataType)

# binding allows population of new/derived variables
BIND(CONCAT(?var2, "--", ?var3) as ?var2AndVar3)
}
```

## Related Topics

[SPARQL Best Practices](#)

# Anzo Function and Formula Reference

This section describes the standard and advanced built-in functions that are available when working with Hi-Res Analytics dashboards or writing SPARQL queries in data layers or the Query Builder.

- String Functions .....995
- Math Functions .....1021
- Aggregate Functions .....1049
- Date and Time Functions .....1068
- Casting Functions .....1086
- Logical Functions .....1102
- Informational or Testing Functions .....1111
- Hash Functions .....1123
- Window Aggregate and Ranking Functions .....1127

## String Functions

This topic describes the Anzo functions that operate on string data types.

### Typographical Conventions

This documentation uses the following conventions in function syntax:

- **CAPS**: Although SPARQL is case-insensitive, function names and other keywords are written in uppercase for readability.
- `[ argument ]`: Brackets are used to indicate optional arguments. Arguments without brackets are required.

### Functions

- **BUSINESS\_ENTITY\_EXCLUDER**: Removes suffixes that represent business entities.
- **CONCATENATE**: Concatenates two or more strings and returns the result as a string.
- **CONCATURL**: Concatenates two or more strings and returns the result as a URI.
- **CONTAINS**: Evaluates whether the specified string contains the given pattern.
- **ENCODE\_FOR\_URI**: Encodes the specified string as a URI.
- **ESCAPEHTML**: Escapes the specified string for use in HTML.
- **FIND**: Returns the position—from left to right—of a string within another string.
- **FINDREVERSE**: Returns the position—from right to left—of a string within another string.
- **GROUP\_CONCAT**: Concatenates a group of strings into a single string.
- **GROUPOCONCAT**: Concatenates a group of strings into a single string. This function is a customizable version of **GROUP\_CONCAT**.
- **LANG**: Returns any language tags that are included with strings.
- **LANGMATCHES**: Evaluates whether a string includes a language tag that matches the specified language range.
- **LCASE**: Converts the letters in a string literal to lower case.
- **LEFT**: Returns the specified number of characters starting from the beginning (left side) of the string.
- **LEN**: Calculates the length (number of characters) in a string.
- **LEVENSHTEIN\_DIST**: Calculates the Levenshtein distance or measure of similarity between two strings.
- **LOWER**: Converts all letters in a string to lower case.

- **MD5**: Returns the MD5 checksum of a string as a hexadecimal string.
- **MID**: Returns the specified number of characters from a string, starting from a given position in the string.
- **REGEX**: Evaluates whether a string matches the specified regular expression pattern.
- **REGEXP\_SUBSTR**: Searches a string for the specified regular expression pattern and returns the substring that matches the pattern.
- **REPLACE**: Extends the REGEX function to provide the ability to find a pattern in a string and replace it with another pattern.
- **RIGHT**: Returns the specified number of characters starting from the end (right side) of the string.
- **SEARCH**: Uses text search semantics to evaluate whether the specified string matches the given pattern.
- **SHA1**: Calculates the SHA-1 digest of a string value.
- **SHA224**: Calculates the SHA-224 digest of a string value.
- **SHA256**: Calculates the SHA-256 digest of a string value.
- **SHA384**: Calculates the SHA-384 digest of a string value.
- **SHA512**: Calculates the SHA-512 digest of a string value.
- **STRAFTER**: Returns the portion of a string that comes after the specified substring.
- **STRBEFORE**: Returns the portion of a string that comes before the specified substring.
- **STRDT**: Constructs a literal value with the specified data type.
- **STREND**: Evaluates whether the specified string ends with the specified substring.
- **STRLANG**: Constructs a literal value with the specified language tag.
- **STRLEN**: Calculates the length of a string.
- **STRSTARTS**: Evaluates whether the specified string starts with the specified substring.
- **STRUUID**: Returns a string that is the result of generating a Universally Unique Identifier (UUID).
- **SUBSTITUTE**: Substitutes the existing text for the specified new text.
- **SUBSTR**: Returns a substring from a string value.
- **TOURI**: Casts a string to a URI.
- **TRIM**: Removes all spaces from a string except for any single spaces between words.
- **UCASE**: Converts all letters in a string to upper case.
- **UPPER**: Converts the letters in a string literal to upper case.



## BUSINESS\_ENTITY\_EXCLUDER

This function removes from strings the suffixes that represent business entities.

### Syntax

```
BUSINESS_ENTITY_EXCLUDER (text)
```

Argument	Data Type	Description
text	string	The string from which you want to remove business entities.

### Returns

Data Type	Description
string	The string without the business entity suffix.

## CONCATENATE

This function concatenates two or more strings and returns the result as a string.

### Syntax

```
CONCATENATE (text1, text2 [, textN ])
```

Argument	Data Type	Description
text1–N	string	The strings that you want to concatenate to form a single string.

### Returns

Data Type	Description
string	The concatenated string.

## CONCATURL

This function concatenates two or more strings and returns the result as a URI.

## Syntax

```
CONCATURL(text1, text2 [, textN ])
```

Argument	Data Type	Description
<b>text1–N</b>	string	The strings that you want to concatenate to form a URI.

## Returns

Data Type	Description
URI	The concatenated string as a URI.

## CONTAINS

This function evaluates whether the specified strings contain the given pattern. Results are grouped under "true" or "false."

## Syntax

```
CONTAINS(text, pattern)
```

Argument	Data Type	Description
<b>text</b>	string	The string value that you want to check against the specified pattern.
<b>pattern</b>	string	The string pattern that you want to look for in the supplied text.

## Returns

Data Type	Description
boolean	<code>True</code> if the strings contain the pattern and <code>false</code> if they do not.

## ENCODE\_FOR\_URI

This function encodes the specified string as a URI and returns a string in URI format.

## Syntax

```
ENCODE_FOR_URI (text)
```

Argument	Data Type	Description
text	string	The string value to encode as a URI.

## Returns

Data Type	Description
string	The string as a URI.

## ESCAPEHTML

This function escapes the specified string for use in HTML.

## Syntax

```
ESCAPEHTML (text)
```

Argument	Data Type	Description
text	string	The string value to escape for HTML.

## Returns

Data Type	Description
string	The string escaped for HTML.

## FIND

This function returns the position—from left to right—of a string within another string.

### Tip

You can use [FINDREVERSE](#) to find the character or substring position from right to left.

## Syntax

```
FIND(find_text, within_text, start_num)
```

Argument	Data Type	Description
find_text	string	The string to look for in the within_text.
within_text	string	The string to search within.
start_num	int	An integer that indicates the position to start from when looking for the find_text. The starting position is at the beginning of the within_text value and characters are counted from left to right.

## Returns

Data Type	Description
int	The character position (from left to right) where the substring starts.

## FINDREVERSE

Similar to [FIND](#), this function returns the position—from right to left—of a string within another string.

## Syntax

```
FINDREVERSE(find_text, within_text, start_num)
```

Argument	Data Type	Description
find_text	string	The string to look for in the within_text value.
within_text	string	The string to search within.
start_num	int	An integer that indicates the position to start from when looking for the find_text. The starting position is the end of the within_text value and characters are counted from right to left.

## Returns

Data Type	Description
int	The character position (from right to left) where the substring starts.

## GROUP\_CONCAT

This function concatenates a group of strings into a single string. It is a simplified version of [GROUPCONCAT](#) as it takes only one argument.

## Syntax

```
GROUP_CONCAT(text)
```

Argument	Data Type	Description
text	string	The string property whose values to concatenate into a single string.

## Returns

Data Type	Description
string	The concatenated string.

## GROUPCONCAT

This function concatenates a group of strings into a single string. Unlike [GROUP\\_CONCAT](#), this function allows for customization of the separator to use as well as the configuration of limits and options like prefixes and suffixes.

## Syntax

```
GROUPCONCAT(group1, [ group2, ..., groupN, ] group_value_separator, separator,  
serialize,  
row_limit, value_limit, delimit_blanks [, prefix ] [, suffix ] [,  
max_length ])
```

Argument	Data Type	Description
<b>group1–N</b>	string	The group(s) of strings to concatenate.
<b>group_value_separator</b>	string	The separator string to use between the groups of strings if you specified more than one <code>group</code> .
<b>separator</b>	string	The separator string to use between the values in a concatenated group of strings.
<b>serialize</b>	boolean	A boolean value that indicates whether returned values should be serialized with the value's data type.
<b>row_limit</b>	int	An integer that puts a maximum limit on the number of rows to retrieve for a group.
<b>value_limit</b>	int	An integer that puts a maximum limit on the number of values to retrieve from a group of rows.
<b>delimit_blanks</b>	boolean	A boolean value that indicates whether to delimit blanks with the <code>separator</code> value.
<b>prefix</b>	string	Optional string to add as a prefix to the resulting string.
<b>suffix</b>	string	Optional string to add as a suffix to the resulting string.
<b>max_length</b>	int	Optional integer that puts a maximum limit on the number of characters the resulting string can have.

## Returns

Data Type	Description
string	The concatenated string.

# LANG

This function returns any language tags that are included in the string. The results are grouped by each language tag or by "blank" if a value does not have a language tag.

## Syntax

```
LANG (text)
```

Argument	Data Type	Description
text	string	The string to search for language tags.

## Returns

Data Type	Description
string	The found language tags.

# LANGMATCHES

This function tests whether a string includes a language tag that matches the specified language range.

## Syntax

```
LANGMATCHES (text, language_range)
```

Argument	Data Type	Description
text	string	The string to evaluate.
language_range	string	The language tag to match in the text.

## Example

```
LANGMATCHES (LANG (?prop) , "en")
```

## Returns

Data Type	Description
boolean	<code>True</code> if strings include a language tag that matches the range and <code>false</code> if they do not.

## LCASE

This function converts the letters in a string literal to lower case.

## Syntax

```
LCASE(text)
```

Argument	Data Type	Description
<b>text</b>	string	The string literal to convert to lower case.

## Returns

Data Type	Description
string	The string with lower case letters.

## LEFT

This function returns the specified number of characters starting from the beginning (left side) of the string.

## Syntax

```
LEFT(text, num_chars)
```

Argument	Data Type	Description
<b>text</b>	string	The string from which to return the specified number of characters.
<b>num_chars</b>	int	An integer that specifies the number of characters to return, starting from the left side of the <code>text</code> .



## Returns

Data Type	Description
string	The specified number of characters from the string.

## LEN

This function calculates the length (number of characters) in a string.

## Syntax

```
LEN(text)
```

Argument	Data Type	Description
<b>text</b>	string	The string for which to calculate the length.

## Returns

Data Type	Description
int	The number of characters in the string.

## LEVENSHTEIN\_DIST

This function calculates the Levenshtein distance or measure of similarity between two strings. The distance is the number of edits required to transform the first string into the second string.

## Syntax

```
LEVENSHTEIN_DIST(text1, text2)
```

Argument	Data Type	Description
<b>text1</b>	string	The string that would be transformed into <code>text2</code> .
<b>text2</b>	string	The string to measure <code>text1</code> against.

## Returns

Data Type	Description
int	The Levenshtein distance between the strings.

## LOWER

This function converts all letters in a string to lower case.

### Syntax

```
LOWER (text)
```

Argument	Data Type	Description
text	string	The string to convert to lower case.

## Returns

Data Type	Description
string	The string with lower case letters.

## MD5

This function returns the MD5 checksum of a string as a hexadecimal string.

### Syntax

```
MD5 (text)
```

Argument	Data Type	Description
text	string	The string for which to return the MD5 checksum.

## Returns

Data Type	Description
string	The hexadecimal string.

## MID

This function returns the specified number of characters from a string, starting from a given position in the string.

### Syntax

```
MID(text, start_num, num_chars)
```

Argument	Data Type	Description
<b>text</b>	string	The string from which to return the specified characters.
<b>start_num</b>	int	An integer that indicates the starting position in the string.
<b>num_chars</b>	int	An integer that specifies the number of characters to return, starting with the character indicated by <code>start_num</code> .

### Returns

Data Type	Description
string	The specified number of characters from the string.

## REGEX

This function tests whether a string matches the specified regular expression pattern.

### Syntax

```
REGEX(text, pattern [, flags ])
```

Argument	Data Type	Description
<b>text</b>	string	The string to test against the <code>pattern</code> .
<b>pattern</b>	string	The regular expression pattern to look for in the <code>text</code> . For information about the supported regular expression syntax, see the <a href="#">Regular Expression Syntax</a>

Argument	Data Type	Description
		section of the W3C XQuery 1.0 and XPath 2.0 Functions and Operators specification.
<b>flags</b>	string	You can include one or more optional modifier flags that further define the pattern. For information about flags, see the <a href="#">Flags</a> section of the W3C Functions and Operators specification.

## Returns

Data Type	Description
boolean	<code>True</code> if the string matches the regular expression pattern and <code>false</code> if it does not.

## REGEXP\_SUBSTR

This function searches a string for the specified regular expression pattern and returns the substring that matches the pattern.

## Syntax

```
REGEXP_SUBSTR(text, pattern [, start_position ] [, nth_appearance ])
```

Argument	Data Type	Description
<b>text</b>	string	The string to test against the <code>pattern</code> .
<b>pattern</b>	string	The regular expression pattern to look for in the <code>text</code> . For information about the supported regular expression syntax, see the <a href="#">Regular Expression Syntax</a> section of the W3C XQuery 1.0 and XPath 2.0 Functions and Operators specification.
<b>start_position</b>	int	An optional integer that specifies the number of characters from the beginning of the string to start searching for matches (the default value is <b>1</b> ).
<b>nth_</b>	int	An optional integer that specifies which occurrence of the pattern to match

Argument	Data Type	Description
<b>appearance</b>		(the default value is <b>1</b> ).

## Returns

Data Type	Description
string	The substring that matches the regular expression pattern.

## REPLACE

This function extends the REGEX function to provide the ability to find a pattern in a string and replace it with another pattern. The function returns the replaced string.

## Syntax

```
REPLACE(text, pattern, replacement_pattern [, flags ])
```

Argument	Data Type	Description
<b>text</b>	string	The string to test against the <code>pattern</code> .
<b>pattern</b>	string	The regular expression pattern to look for in the <code>text</code> . For information about the supported regular expression syntax, see the <a href="#">Regular Expression Syntax</a> section of the W3C XQuery 1.0 and XPath 2.0 Functions and Operators specification.
<b>replacement_pattern</b>	string	The pattern to replace the <code>pattern</code> with.
<b>flags</b>	string	You can include one or more optional modifier flags that further define the pattern. For information about flags, see the <a href="#">Flags</a> section of the W3C Functions and Operators specification.

## Returns

Data Type	Description
string	The string that contains the replacement pattern.

## RIGHT

This function returns the specified number of characters starting from the end (right side) of the string.

## Syntax

```
RIGHT(text, num_chars)
```

Argument	Data Type	Description
<b>text</b>	string	The string from which to return the specified number of characters.
<b>num_chars</b>	int	An integer that specifies the number of characters to return, starting from the right side of the <code>text</code> .

## Returns

Data Type	Description
string	The specified characters from the string.

## SEARCH

This function uses text search semantics to evaluate whether the specified string matches the given pattern.

## Syntax

```
SEARCH(text, pattern [, required ] [, wildcard ] [, escape ])
```

Argument	Data Type	Description
<b>text</b>	string	The string to search.

Argument	Data Type	Description
<b>pattern</b>	string	The search string to look for in the <code>text</code> . Anzo automatically converts the value to a regular expression pattern that uses text search semantics.
<b>required</b>	boolean	An optional boolean value that indicates whether the <code>text</code> must include all elements of the search pattern to qualify as a match or whether matching just part of the pattern qualifies as a match.
<b>wildcard</b>	boolean	An optional boolean value that indicates whether or not to add the wildcard character <code>*</code> to the end of the search pattern.
<b>escape</b>	boolean	An optional boolean value that indicates whether or not escape all of the special characters (such as <code>+</code> , <code>-</code> , or <code> </code> ) in the <code>text</code> .

## Returns

Data Type	Description
boolean	<code>True</code> if strings match the pattern and <code>false</code> if they do not.

## SHA1

This function calculates the SHA-1 digest of a string.

## Syntax

```
SHA1(text)
```

Argument	Data Type	Description
<b>text</b>	string	The string for which to calculate the SHA-1 digest.

## Returns

Data Type	Description
string	The SHA-1 digest.

## SHA224

This function calculates the SHA-224 digest of a string.

### Syntax

```
SHA224 (text)
```

Argument	Data Type	Description
text	string	The string for which to calculate the SHA-224 digest.

### Returns

Data Type	Description
string	The SHA-224 digest.

## SHA256

This function calculates the SHA-256 digest of a string.

### Syntax

```
SHA256 (text)
```

Argument	Data Type	Description
text	string	The string for which to calculate the SHA-256 digest.

### Returns

Data Type	Description
string	The SHA-256 digest.

## SHA384

This function calculates the SHA-384 digest of a string.



## Syntax

```
SHA384(text)
```

Argument	Data Type	Description
<b>text</b>	string	The string for which to calculate the SHA-384 digest.

## Returns

Data Type	Description
string	The SHA-384 digest.

## SHA512

This function calculates the SHA-512 digest of a string.

## Syntax

```
SHA512(text)
```

Argument	Data Type	Description
<b>text</b>	string	The string for which to calculate the SHA-512 digest.

## Returns

Data Type	Description
string	The SHA-512 digest.

## STRAFTER

This function returns the portion of a string that comes after the specified substring.

## Syntax

```
STRAFTER(text, substring)
```

Argument	Data Type	Description
<b>text</b>	string	The string from which to return the characters that follow the <code>substring</code> .
<b>substring</b>	string	The string to match in the <code>text</code> . The function will return the part of the text that comes after this substring.

## Returns

Data Type	Description
string	The part of the string that comes after the substring.

## STRBEFORE

This function returns the portion of a string that comes before the specified substring.

## Syntax

```
STRAFTER(text, substring)
```

Argument	Data Type	Description
<b>text</b>	string	The string from which to return the characters that precede the <code>substring</code> .
<b>substring</b>	string	The string to match in the <code>text</code> . The function will return the part of the text that comes before this substring.

## Returns

Data Type	Description
string	The part of the string that comes before the substring.

## STRDT

This function constructs a literal value with the specified data type.

## Syntax

```
STRDT(text, datatype)
```

Argument	Data Type	Description
<b>text</b>	string	The string to add a data type specification to.
<b>datatype</b>	URI	The data type URI to add to the <code>text</code> . For example, <code>xsd:integer</code> or <code>&lt;http://www.w3.org/2001/XMLSchema#integer&gt;</code> .

## Returns

Data Type	Description
string	The typed literal value.

## STRENDS

This function evaluates whether the specified string ends with the specified substring.

## Syntax

```
STRENDS(text, substring)
```

Argument	Data Type	Description
<b>text</b>	string	The string to search for the <code>substring</code> .
<b>substring</b>	string	The string to match at the end of <code>text</code> . The function returns <code>true</code> if the text ends in the specified substring and <code>false</code> if it does not.

## Returns

Data Type	Description
boolean	<code>True</code> if strings end with the specified substring and <code>false</code> if they do not.

## STRLANG

This function constructs a literal value with the specified language tag.

### Syntax

```
STRLANG(text, language_tag)
```

Argument	Data Type	Description
<b>text</b>	string	The string to add the language tag to.
<b>language_tag</b>	string	The language tag to add to the <code>text</code> .

### Returns

Data Type	Description
string	The literal value with the language tag.

## STRLEN

This function calculates the length (in characters) of a string value.

### Syntax

```
STRLEN(text)
```

Argument	Data Type	Description
<b>text</b>	string	The string for which to return the length.

### Returns

Data Type	Description
long	The number of characters in the string.

## STRSTARTS

This function evaluates whether the specified string starts with the specified substring.

## Syntax

```
STRENDS(text, substring)
```

Argument	Data Type	Description
<b>text</b>	string	The string to search for the <code>substring</code> .
<b>substring</b>	string	The string to match at the beginning of <code>text</code> . The function returns <code>true</code> if the text starts with the specified substring and <code>false</code> if it does not.

## Returns

Data Type	Description
boolean	<code>True</code> if strings begin with the specified substring and <code>false</code> if they do not.

## STRUUID

This function returns a string that is the result of generating a Universally Unique Identifier (UUID).

## Syntax

```
STRUUID()
```

## Returns

Data Type	Description
string	The UUID.

## SUBSTITUTE

This function substitutes the existing text for the specified new text.

## Syntax

```
SUBSTITUTE(text, old_text, new_text [, instance_num ])
```

Argument	Data Type	Description
<b>text</b>	string	The string to substitute text in.
<b>old_text</b>	string	The string within the <code>text</code> to replace.
<b>new_text</b>	string	The string to replace the <code>old_text</code> with.
<b>instance_num</b>	int	An optional integer that specifies the number of <code>old_text</code> instances to replace.

## Returns

Data Type	Description
string	The string with the new text.

## SUBSTR

This function returns a substring from a string value.

## Syntax

```
SUBSTR(text, start [, length ])
```

Argument	Data Type	Description
<b>text</b>	string	The string to find the substring in.
<b>start</b>	int	An integer that specifies the number of the character in the <code>text</code> that should be the start of the substring.
<b>length</b>	int	An optional integer that specifies the total number of characters to include in the substring. If not specified, the substring will end at the end of the <code>text</code> value.

## Returns

Data Type	Description
string	The substring.

## TOURI

This function casts a string literal value to a URI.

## Syntax

```
TOURI(text)
```

Argument	Data Type	Description
<b>text</b>	string	The string literal to cast to a URI.

## Returns

Data Type	Description
URI	The literal value as a URI.

## TRIM

This function removes all spaces from a string except for any single spaces between words.

## Syntax

```
TRIM(text)
```

Argument	Data Type	Description
<b>text</b>	string	The string to trim.

## Returns

Data Type	Description
string	The string with spaces removed.

## UCASE

This function converts all letters in a string to upper case.

### Syntax

```
UPPER(text)
```

Argument	Data Type	Description
text	string	The string value to convert to upper case.

### Returns

Data Type	Description
string	The string with upper case characters.

## UPPER

This function converts all letters in a string literal to upper case.

### Syntax

```
UPPER(text)
```

Argument	Data Type	Description
text	string	The string literal to convert to upper case.

### Returns

Data Type	Description
string	The string with upper case characters.



## Math Functions

This topic describes the mathematical functions in Anzo.

### Typographical Conventions

This documentation uses the following conventions in function syntax:

- **CAPS**: Although SPARQL is case-insensitive, function names and other keywords are written in uppercase for readability.
- `[ argument ]`: Brackets are used to indicate optional arguments. Arguments without brackets are required.

### Functions

- **ABS**: Calculates the absolute value of the specified number.
- **ADD**: Adds two numeric values.
- **AVERAGEIF**: Calculates the average of the range of values that meet the specified criterion.
- **AVERAGEIFS**: Calculates the averages of the ranges of values that meet the specified criteria.
- **AVG**: Calculates the average (arithmetic mean) value for a group of numbers.
- **CEILING**: Rounds up a numeric value to the nearest integer.
- **COS**: Calculates the cosine of an angle.
- **DIVIDE**: Divides a number by another number.
- **EQUAL**: Evaluates whether two values are equal.
- **EXP**: Raises e to the specified power.
- **FACT**: Calculates the factorial of the specified number.
- **FLOOR**: Rounds down a numeric value to the nearest integer.
- **GE**: Evaluates whether one value is greater than or equal to ( $\geq$ ) another value.
- **GT**: Evaluates whether one value is greater than ( $>$ ) another value.
- **HAMMING\_DIST**: Calculates the hamming distance between two values.
- **HAVERSINE\_DIST**: Computes the haversine distance between two latitude and longitude values.
- **LE**: Evaluates whether one value is less than or equal to ( $\leq$ ) another value.
- **LN**: Calculates the natural logarithm of a double value.
- **LOG**: Calculates the specified base logarithm of a double value.

- **LOG2**: Calculates the base two logarithm of a double value.
- **LT**: Evaluates whether one value is less than (<) another value.
- **MAXVAL**: Determines the maximum value from the given literal values.
- **MINVAL**: Determines the minimum value from the given literal values.
- **MOD**: Calculates the modulo of the division between two numbers.
- **MULTIPLY**: Multiplies two number values.
- **NOT\_EQUAL**: Evaluates whether two values are not equal.
- **NPV**: Calculates the net present value of an investment.
- **NUMERIC-ADD**: Adds two numeric values.
- **NUMERIC-SUBTRACT**: Subtracts one numeric value from another numeric value.
- **PI**: Returns the value for PI.
- **POWER**: Raises the specified number to the specified power.
- **PRODUCT**: Calculates the product of a group of numbers.
- **QUOTIENT**: Calculates the quotient between two numbers.
- **RAD**: Converts to radians an angle value that is in degrees.
- **RAND**: Returns a random double value between 0 and 1.
- **RANDBETWEEN**: Returns a random integer that falls between two specified integers.
- **ROUND**: Rounds a numeric value to the nearest integer.
- **ROUNDDOWN**: Rounds a numeric value down to the specified number of digits.
- **ROUNDUP**: Rounds a numeric value up to the specified number of digits.
- **SIN**: Calculates the sine of an angle.
- **SQRT**: Calculates the square root of a number.
- **SUBTRACT**: Subtracts one RDF term from another RDF term type value.
- **SUM**: Calculates the sum of the numbers within a group.
- **SUMIF**: Calculates the sum of the range of values that meet the specified criterion.
- **SUMIFS**: Calculates the sums of the ranges of values that meet the specified criteria.
- **SUMPRODUCT**: Multiplies the numbers in a group and adds the results.
- **SUMSQ**: Calculates the square root of each number in a group and adds the results.
- **TAN**: Calculates the tangent of an angle.

## ABS

This function calculates the absolute value of the specified number.

### Syntax

```
ABS (number)
```

Argument	Data Type	Description
number	numeric	The numeric value for which to calculate the absolute value.

### Returns

Data Type	Description
number	The absolute value.

## ADD

This function adds two numeric values.

### Syntax

```
ADD (value1, value2)
```

Argument	Data Type	Description
value1	numeric	The first numeric value to add.
value2	numeric	The second numeric value to add.

### Returns

Data Type	Description
number	The result of the addition operation.

## AVERAGEIF

This function calculates the average of the range of values that meet the specified criterion.

### Tip

You can use AVERAGEIFS to specify multiple value ranges and conditions.

## Syntax

```
AVERAGEIF(values_to_test, criterion [, range_of_values ])
```

Argument	Data Type	Description
<b>values_to_test</b>	RDF term	The literal, URI, or blank node value that defines the values to test against the <code>criteria</code> .
<b>criterion</b>	RDF term	The literal, URI, or blank node value that defines the condition to test values against.
<b>range_of_values</b>	numeric	An optional number that defines the range of values to average. When omitted, <code>values_to_test</code> is used.

## Returns

Data Type	Description
number	The average value from the range of values that meet the criterion.

## AVERAGEIFS

This function calculates the averages of the ranges of values that meet the specified criteria. Unlike AVERAGEIF, this function enables you to specify multiple ranges and multiple conditions.

## Syntax

```
AVERAGEIFS(values_to_average, value_range1, criteria1,  
            value_range2, criteria2  
            [, value_rangeN, criteriaN ])
```

Argument	Data Type	Description
<b>values_to_average</b>	numeric	The numeric value that defines the overall range of values to evaluate.

Argument	Data Type	Description
<b>value_ range1–N</b>	RDF term	The literal, URI, or blank node value that defines the range of values to test against the corresponding <code>criteria</code> .
<b>criteria1–N</b>	RDF term	The literal, URI, or blank node value that defines the condition to test the corresponding <code>value_range</code> against.

## Returns

Data Type	Description
number	The average values from the ranges of values that meet the criteria.

## AVG

This function calculates the average (arithmetic mean) value for a group of numbers.

## Syntax

```
AVG (number)
```

Argument	Data Type	Description
<b>number</b>	numeric	The numeric value for which to calculate the average.

## Returns

Data Type	Description
number	The arithmetic mean of the input values.

## CEILING

This function rounds up a numeric value to the nearest integer if the value has a fractional part. `CEILING` returns the value itself if it is a whole number.

## Syntax

```
CEILING (number)
```

Argument	Data Type	Description
<b>number</b>	numeric	The numeric value to round up.

## Returns

Data Type	Description
number	The rounded up value.

## COS

This function calculates the cosine of the specified angle.

## Syntax

```
COS (angle)
```

Argument	Data Type	Description
<b>angle</b>	double	The angle in radians (double data type) to calculate the cosine for. If you have angle values in degrees, you can use <a href="#">RAD</a> to convert the degrees to radians.

## Returns

Data Type	Description
double	The cosine of the angle.

## DIVIDE

This function divides one number by another number.

## Syntax

```
DIVIDE (value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric	The number that is the dividend in the equation.
<b>value2</b>	numeric	The number to divide <code>value1</code> by.

## Returns

Data Type	Description
number	The result of the division operation.

## EQUAL

This function evaluates whether `value1` is equal to `value2`.

## Syntax

```
EQUAL(value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value2</code> .
<b>value2</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value1</code> .

## Returns

Data Type	Description
boolean	<code>True</code> if the values are equal and <code>false</code> if they are not.

## EXP

This function raises the base of the natural logarithms, *e*, to the specified power.

## Syntax

```
EXP(power)
```

Argument	Data Type	Description
<b>power</b>	numeric	The number to raise e to.

## Returns

Data Type	Description
number	E raised to the specified power.

## FACT

This function calculates the factorial of the specified number.

## Syntax

```
FACT (number)
```

Argument	Data Type	Description
<b>number</b>	int	The number for which to calculate the factorial.

## Returns

Data Type	Description
int	The factorial of the input values.

## FLOOR

This function rounds down a numeric value to the nearest integer if the value has a fractional part. FLOOR returns the value itself if it is a whole number.

## Syntax

```
FLOOR (number)
```

Argument	Data Type	Description
<b>number</b>	numeric	The numeric value to round down.



## Returns

Data Type	Description
number	The rounded down value.

## GE

This function evaluates whether value1 is greater than or equal to ( $\geq$ ) value2.

## Syntax

```
GE(value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value2</code> . This is the value that will be checked to see if it is greater than or equal to value2.
<b>value2</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value1</code> .

## Returns

Data Type	Description
boolean	True if <code>value1</code> $\geq$ <code>value2</code> . False if not.

## GT

This function evaluates whether value1 is greater than ( $>$ ) value2.

## Syntax

```
GE(value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value2</code> . This is the value that will be checked to see if it is greater than <code>value2</code> .
<b>value2</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value1</code> .

## Returns

Data Type	Description
boolean	<code>True</code> if <code>value1 &gt; value2</code> . <code>False</code> if not.

## HAMMING\_DIST

This function calculates the hamming distance between two values.

## Syntax

```
HAMMING_DIST(number1, number2)
```

Argument	Data Type	Description
<b>number1</b>	long	The first number.
<b>number2</b>	long	The second number.

## Returns

Data Type	Description
int	The hamming distance.

## HAVERSINE\_DIST

This function computes the haversine distance between two latitude and longitude values and returns the distance in kilometers. The function uses the Haversine formula, which is accurate for most purposes but assumes a spherical Earth. Since the Earth is elliptical, distances involving points near the poles will be more inaccurate than other points.

### Syntax

```
HAVERSINE_DIST(latitude1, longitude1, latitude2, longitude2)
```

Argument	Data Type	Description
<b>latitude1</b>	double	The first latitude value.
<b>longitude1</b>	double	The first longitude value.
<b>latitude2</b>	double	The second latitude value.
<b>longitude2</b>	double	The second longitude value.

### Returns

Data Type	Description
double	The distance in kilometers.

## LE

This function evaluates whether value1 is less than or equal to ( $\leq$ ) value2.

### Syntax

```
LE(value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric, boolean, dateTime, RDF	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value2</code> . This is the value that will be evaluated to see if it is less than or equal to <code>value2</code> .

Argument	Data Type	Description
	term	
<b>value2</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value1</code> .

## Returns

Data Type	Description
boolean	True if <code>value1 &lt;= value2</code> . False if not.

## LN

This function calculates the natural logarithm of a double value.

## Syntax

```
LN (number)
```

Argument	Data Type	Description
<b>number</b>	double	The double value for which to calculate the natural logarithm.

## Returns

Data Type	Description
double	The natural logarithm of the input value.

## LOG

This function calculates the specified base logarithm of a double value.

## Syntax

```
LOG (number [, base ])
```

Argument	Data Type	Description
<b>number</b>	double	The double value for which to calculate the <code>base</code> logarithm.
<b>base</b>	double	An optional double value that specifies the base for the logarithm. If omitted, base e is used.

## Returns

Data Type	Description
double	The base logarithm of the input value.

## LOG2

This function calculates the base two logarithm of a double value.

## Syntax

```
LOG2 (number)
```

Argument	Data Type	Description
<b>number</b>	double	The double value for which to calculate the base 2 logarithm.

## Returns

Data Type	Description
double	The base two logarithm of the input value.

## LT

This function evaluates whether value1 is less than (<) value2.

## Syntax

```
LT (value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, or RDF term type value to compare to <code>value2</code> . This is the value that will be evaluated to see if it is less than <code>value2</code> .
<b>value2</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, or RDF term type value to compare to <code>value1</code> .

## Returns

Data Type	Description
boolean	True if <code>value1 &lt; value2</code> . False if not.

## MAXVAL

This function determines the maximum value from the given literal values.

## Syntax

```
MAXVAL(value1 [, value2 ] [, valueN ])
```

Argument	Data Type	Description
<b>value1–N</b>	literal	A literal value from which you want to find the maximum value.

## Returns

Data Type	Description
literal	The maximum value.

## MINVAL

This function determines the minimum value from the given literal values.

## Syntax

```
MINVAL(value1 [, value2 ] [, valueN ])
```

Argument	Data Type	Description
value1–N	literal	A literal value from which you want to find the minimum value.

## Returns

Data Type	Description
literal	The minimum value.

## MOD

This function calculates the modulo or remainder of the division between two numbers.

## Syntax

```
MOD(number, divisor)
```

Argument	Data Type	Description
number	numeric	The number that is the dividend in the equation.
divisor	numeric	The number to divide the dividend by.

## Returns

Data Type	Description
number	The modulo between the input numbers.

## MULTIPLY

This function multiplies two numbers.

## Syntax

```
MULTIPLY(value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric	The first number in the multiplication equation.
<b>value2</b>	numeric	The number to multiply <code>value1</code> by.

## Returns

Data Type	Description
number	The result of the multiplication operation.

## NOT\_EQUAL

This function evaluates whether `value1` is not equal to `value2`.

## Syntax

```
NOT_EQUAL(value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value2</code> .
<b>value2</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value1</code> .

## Returns

Data Type	Description
boolean	<code>True</code> if <code>value1</code> is not equal to <code>value2</code> . <code>False</code> if they are equal.

## NPV

This function calculates the net present value of an investment by using a discount rate and a series of future payments (negative values) and income (positive values).



## Syntax

```
NPV(rate, year, value)
```

Argument	Data Type	Description
<b>rate</b>	double	A double value that defines the discount rate to use in the calculation.
<b>year</b>	double	A double value that indicates which year the <code>value</code> is in.
<b>value</b>	double	The double values that represent payments and income.

## Returns

Data Type	Description
double	The net present value.

## NUMERIC-ADD

This function adds two numeric values.

## Syntax

```
NUMERIC-ADD(value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric	The first number in the addition equation.
<b>value2</b>	numeric	The number to add to <code>value1</code> .

## Returns

Data Type	Description
number	The result of the addition operation.

## NUMERIC-SUBTRACT

This function subtracts one numeric value from another numeric value.

### Syntax

```
NUMERIC-SUBTRACT (value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric	The first number in the subtraction equation.
<b>value2</b>	numeric	The number to subtract from <code>value1</code> .

### Returns

Data Type	Description
number	The result of the subtraction operation.

## PI

This function returns the value for PI.

### Syntax

```
PI ()
```

### Returns

Data Type	Description
double	The PI value.

## POWER

This function raises the specified number to the specified power.

### Syntax

```
POWER (value, power)
```

Argument	Data Type	Description
<b>value</b>	numeric	The number to raise by the <code>power</code> .
<b>power</b>	numeric	The number to raise <code>value</code> by.

## Returns

Data Type	Description
number	The result of <code>value</code> raised to the specified <code>power</code> .

## PRODUCT

This function calculates the product of a group of numbers.

## Syntax

```
PRODUCT (number)
```

Argument	Data Type	Description
<b>number</b>	numeric	The group of numbers to multiply.

## Returns

Data Type	Description
number	The product of the group.

## QUOTIENT

This function calculates the quotient between two numbers.

## Syntax

```
QUOTIENT (numerator, denominator)
```

Argument	Data Type	Description
<b>numerator</b>	numeric	The number to divide by the <code>denominator</code> .

Argument	Data Type	Description
<b>denominator</b>	numeric	The number to divide the <code>numerator</code> by.

## Returns

Data Type	Description
long	The quotient between the input values.

## RAD

This function converts to radians an angle value that is in degrees.

## Syntax

```
RAD (angle)
```

Argument	Data Type	Description
<b>angle</b>	double	The angle value to convert to radians.

## Returns

Data Type	Description
double	The angle in radians.

## RAND

This function returns a random double value between 0 and 1.

## Syntax

```
RAND ( )
```

## Returns

Data Type	Description
double	The random value between 0 and 1.

## RANDBETWEEN

This function returns a random integer that falls between the two specified integers. The two integers are included as options to be returned.

### Syntax

```
RANDBETWEEN(low_number, high_number)
```

Argument	Data Type	Description
low_number	int	The lowest integer in the range of values.
high_number	int	The highest integer in the range of values.

#### Note

If the arguments are decimal values, Anzo returns a random integer between `CEIL(low_number)` and `FLOOR(high_number)`.

### Returns

Data Type	Description
int	The random value between the given low and high numbers.

## ROUND

This function rounds a numeric value to the nearest integer.

### Syntax

```
ROUND(number)
```

Argument	Data Type	Description
number	numeric	The number to round to the nearest integer.

## Returns

Data Type	Description
long	The rounded value.

## ROUNDDOWN

This function rounds a numeric value down to the specified number of digits.

## Syntax

```
ROUNDDOWN(number, num_digits)
```

Argument	Data Type	Description
<b>number</b>	numeric	The number to round down.
<b>num_digits</b>	int	An integer that specifies the number of digits to round down to.

## Returns

Data Type	Description
number	The rounded down value.

## ROUNDUP

This function rounds a numeric value up to the specified number of digits.

## Syntax

```
ROUNDUP(number, num_digits)
```

Argument	Data Type	Description
<b>number</b>	numeric	The number to round up.
<b>num_digits</b>	int	An integer that specifies the number of digits to round up to.

## Returns

Data Type	Description
number	The rounded up value.

## SIN

This function calculates the sine of the specified angle.

## Syntax

```
SIN (angle)
```

Argument	Data Type	Description
<b>angle</b>	double	The angle in radians to calculate the sine for. If you have angle values in degrees, you can use <a href="#">RAD</a> to convert the degrees to radians.

## Returns

Data Type	Description
double	The sine of the angle.

## SQRT

This function calculates the square root of the specified number.

## Syntax

```
SQRT (number)
```

Argument	Data Type	Description
<b>number</b>	numeric	The number for which to calculate the square root.

## Returns

Data Type	Description
double	The square root of the input value.

## SUBTRACT

This function subtracts one RDF term type (a literal value, URI, or blank node) value from another RDF term type value.

## Syntax

```
SUBTRACT(term1, term2)
```

Argument	Data Type	Description
<b>term1</b>	RDF term	The literal, URI, or blank node value that <i>term2</i> will be subtracted from.
<b>term2</b>	RDF term	The literal, URI, or blank node value to subtract from <i>term1</i> .

## Returns

Data Type	Description
REF term	The result of the subtraction operation.

## SUM

This function calculates the sum of the numbers within a group.

## Syntax

```
SUM(number)
```

Argument	Data Type	Description
<b>number</b>	numeric	The group of numbers to sum.



## Returns

Data Type	Description
number	The sum of the values in the group.

## SUMIF

This function calculates the sum of the range of values that meet the specified criterion.

### Tip

You can use [SUMIFS](#) to specify multiple value ranges and conditions.

## Syntax

```
SUMIF(values_to_test, criterion [, range_of_values ])
```

Argument	Data Type	Description
<b>values_to_test</b>	RDF term	The literal, URI, or blank node value that defines the values to test against the <code>criterion</code> .
<b>criterion</b>	RDF term	The literal, URI, or blank node value that defines the condition to test values against.
<b>range_of_values</b>	numeric	An optional number that defines the range of values to sum. When omitted, <code>values_to_test</code> is used.

## Returns

Data Type	Description
number	The sum of the range of values.

## SUMIFS

This function calculates the sums of the ranges of values that meet the specified criteria. Unlike [SUMIF](#), this function enables you to specify multiple ranges and multiple conditions.

## Syntax

```
SUMIFS(values_to_sum, value_range1, criterial1,  
       value_range2, criteria2  
       [, value_rangeN, criteriaN ])
```

Argument	Data Type	Description
<b>values_to_sum</b>	numeric	The numeric value that defines the overall range of values to evaluate.
<b>value_range1–N</b>	RDF term	The literal, URI, or blank node value that defines the range of values to test against the corresponding <code>criteria</code> .
<b>criteria1–N</b>	RDF term	The literal, URI, or blank node value that defines the condition to test the corresponding <code>value_range</code> against.

## Returns

Data Type	Description
number	The sums of the ranges of values.

## SUMPRODUCT

This function multiplies the numbers in a group and adds the results.

## Syntax

```
SUMPRODUCT (number)
```

Argument	Data Type	Description
<b>number</b>	numeric	The group of numbers to multiply and then sum the results.

## Returns

Data Type	Description
number	The sum of the product of the numbers in the group.

## SUMSQ

This function calculates the square root of each number in a group and adds the results.

## Syntax

```
SUMSQ (number)
```

Argument	Data Type	Description
<b>number</b>	numeric	The group of numbers for which to calculate the square root and then sum the results.

## Returns

Data Type	Description
number	The sum of the square root of the numbers in the group.

## TAN

This function calculates the tangent of the specified angle.

## Syntax

```
TAN (angle)
```

Argument	Data Type	Description
<b>angle</b>	double	The angle in radians to calculate the tangent for. If you have angle values in degrees, you can use <a href="#">RAD</a> to convert the degrees to radians.

## Returns

Data Type	Description
double	The tangent of the angle.

# Aggregate Functions

This topic describes the aggregate functions in Anzo.

## Typographical Conventions

This documentation uses the following conventions in function syntax:

- **CAPS**: Although SPARQL is case-insensitive, function names and other keywords are written in uppercase for readability.
- `[ argument ]`: Brackets are used to indicate optional arguments. Arguments without brackets are required.

## Functions

- **AVERAGEIF**: Calculates the average of the range of values that meet the specified criterion.
- **AVERAGEIFS**: Calculates the averages of the ranges of values that meet the specified criteria.
- **AVG**: Calculates the average (arithmetic mean) value for a group of numbers.
- **CHOOSE\_BY\_MAX**: Returns the value from a group that corresponds to the maximum value from another group.
- **CHOOSE\_BY\_MIN**: Returns the value from a group that corresponds to the minimum value from another group.
- **COUNT**: Counts the number of values that exist for a group.
- **COUNT\_DISTINCT**: Counts the number of unique values that exist for a group.
- **COUNTIF**: Counts the number of values that meet the specified criterion.
- **COUNTIFS**: Counts the number of values that meet the specified criteria.
- **GROUP\_CONCAT**: Concatenates a group of strings into a single string.
- **GROUPCONCAT**: Concatenates a group of strings into a single string. This function is a customizable version of GROUP\_CONCAT.
- **MAX**: Returns the maximum value from each group of values.
- **MEDIAN**: Returns the median number out of a group of numbers.
- **MIN**: Returns the minimum value from each group of values.
- **MODE**: Returns the mode (the value that occurs most frequently) from a group of values.
- **MODEPERCENT**: Calculates the percentage of values in a group that belong to the mode.

- **PERCENTILE\_CONT**: Calculates a percentile based on the continuous distribution of the specified group of values.
- **PERCENTILE\_DISC**: Calculates a percentile based on the discrete distribution of the specified group of values.
- **PRODUCT**: Calculates the product of a group of numbers.
- **SAMPLE**: Returns an arbitrary value from the specified group of values.
- **STDEV**: Measures the standard deviation in a group of numbers.
- **STDEVP**: Calculates the product of the standard deviation for a group of numbers.
- **SUM**: Calculates the sum of the numbers within a group.
- **SUMIF**: Calculates the sum of the range of values that meet the specified criterion.
- **SUMIFS**: Calculates the sums of the ranges of values that meet the specified criteria.
- **SUMPRODUCT**: Multiplies the numbers in a group and adds the results.
- **SUMSQ**: Calculates the square root of each number in a group and adds the results.
- **VAR**: Calculates the unbiased (sample) variance of a group of numbers.
- **VARP**: Calculates the biased (population) variance of a group of numbers.
- **WEIGHTEDAVERAGE**: Calculates the weighted average of a group of values.

### AVERAGEIF

This function calculates the average of the range of values that meet the specified criterion.

Tip

You can use AVERAGEIFS to specify multiple value ranges and conditions.

### Syntax

```
AVERAGEIF(values_to_test, criterion [, range_of_values ])
```

Argument	Data Type	Description
values_to_test	RDF term	The literal, URI, or blank node value that defines the values to test against the criteria.
criterion	RDF term	The literal, URI, or blank node value that defines the condition to test values against.

Argument	Data Type	Description
<b>range_of_values</b>	numeric	An optional number that defines the range of values to average. When omitted, <code>values_to_test</code> is used.

## Returns

Data Type	Description
number	The average value from the range of values that meet the criterion.

## AVERAGEIFS

This function calculates the averages of the ranges of values that meet the specified criteria. Unlike `AVERAGEIF`, this function enables you to specify multiple ranges and multiple conditions.

## Syntax

```
AVERAGEIFS(values_to_average, value_range1, criterial,
            value_range2, criteria2
            [, value_rangeN, criteriaN ])
```

Argument	Data Type	Description
<b>values_to_average</b>	numeric	The numeric value that defines the overall range of values to evaluate.
<b>value_range1–N</b>	RDF term	The literal, URI, or blank node value that defines the range of values to test against the corresponding <code>criteria</code> .
<b>criteria1–N</b>	RDF term	The literal, URI, or blank node value that defines the condition to test the corresponding <code>value_range</code> against.

## Returns

Data Type	Description
number	The average values from the ranges of values that meet the criteria.

## AVG

This function calculates the average (arithmetic mean) value for a group of numbers.

### Syntax

```
AVG (number)
```

Argument	Data Type	Description
<b>number</b>	numeric	The numeric value for which to calculate the average.

### Returns

Data Type	Description
number	The arithmetic mean of the input values.

## CHOOSE\_BY\_MAX

This function calculates the maximum value for one group and returns the value from another group that corresponds to the maximum from the first group.

### Syntax

```
CHOOSE_BY_MAX (test, value)
```

Argument	Data Type	Description
<b>test</b>	RDF term	The group of literal, URI, or blank node values from which to find the maximum value.
<b>value</b>	RDF term	The group of literal, URI, or blank node values from which to return the value that corresponds to the maximum value of <code>test</code> .

### Example

In a fictional ticket sales data set, the following statement returns the ID of the buyer who paid the most:

```
CHOOSE_BY_MAX (?totalPaid, ?buyerID)
```



## Returns

Data Type	Description
RDF term	The term from the <code>value</code> group that corresponds to the maximum value from the <code>test</code> group.

## CHOOSE\_BY\_MIN

This function calculates the minimum value for one group and returns the value from another group that corresponds to the minimum from the first group.

## Syntax

```
CHOOSE_BY_MIN(test, value)
```

Argument	Data Type	Description
<b>test</b>	RDF term	The group of literal, URI, or blank node values from which to find the minimum value.
<b>value</b>	RDF term	The group of literal, URI, or blank node values from which to return the value that corresponds to the minimum value of <code>test</code> .

## Example

In a fictional ticket sales data set, the following statement returns the ID of the seller who sold the least number of tickets:

```
CHOOSE_BY_MIN(?totalTickets, ?sellerID)
```

## Returns

Data Type	Description
RDF term	The term from the <code>value</code> group that corresponds to the minimum value from the <code>test</code> group.

## COUNT

This function counts the number of values that exist for a group.

## Syntax

```
COUNT (value)
```

Argument	Data Type	Description
value	RDF term	The group of literal, URI, or blank node values to count.

## Returns

Data Type	Description
long	The number of values in the group.

## COUNT\_DISTINCT

This function counts the number of unique values that exist for a group.

## Syntax

```
COUNT_DISTINCT (value)
```

Argument	Data Type	Description
value	RDF term	The group of literal, URI, or blank node values for which to count the number of distinct values.

## Returns

Data Type	Description
long	The number of unique values in the group.

## COUNTIF

This function counts the number of values that meet the specified criterion.

### Tip

You can use [COUNTIFS](#) to specify multiple conditions.

## Syntax

```
COUNTIF(values_to_test, criterion)
```

Argument	Data Type	Description
<b>values_to_test</b>	RDF term	The literal, URI, or blank node value that defines the values to test against the <code>criterion</code> .
<b>criterion</b>	RDF term	The literal, URI, or blank node value that defines the condition to test values against.

## Returns

Data Type	Description
long	The number of values that meet the criterion.

## COUNTIFS

This function counts the number of values that meet the specified criteria. Unlike [COUNTIF](#), this function enables you to specify multiple conditions.

## Syntax

```
COUNTIFS(values_to_count, criteria1 [, criteria2 ] [, criteriaN ])
```

Argument	Data Type	Description
<b>values_to_count</b>	RDF term	The literal, URI, or blank node value to compare against the criteria.
<b>criteria1–N</b>	RDF term	A literal, URI, or blank node value that defines a condition to test the <code>values_to_count</code> against.

## Returns

Data Type	Description
long	The number of values that meet the specified conditions.

## GROUP\_CONCAT

This function concatenates a group of strings into a single string. It is a simplified version of [GROUPCONCAT](#) as it takes only one argument.

## Syntax

```
GROUP_CONCAT(text)
```

Argument	Data Type	Description
text	string	The string property whose values to concatenate into a single string.

## Returns

Data Type	Description
string	The concatenated string.

## GROUPCONCAT

This function concatenates a group of strings into a single string. Unlike [GROUP\\_CONCAT](#), this function allows for customization of the separator to use as well as the configuration of limits and options like prefixes and suffixes.

## Syntax

```
GROUPCONCAT(group1, [ group2, ..., groupN, ] group_value_separator, separator,  
serialize,  
row_limit, value_limit, delimit_blanks [, prefix ] [, suffix ] [,  
max_length ])
```

Argument	Data Type	Description
<b>group1–N</b>	string	The group(s) of strings to concatenate.
<b>group_value_separator</b>	string	The separator string to use between the groups of strings if you specified more than one <code>group</code> .
<b>separator</b>	string	The separator string to use between the values in a concatenated group of strings.
<b>serialize</b>	boolean	A boolean value that indicates whether returned values should be serialized with the value's data type.
<b>row_limit</b>	int	An integer that puts a maximum limit on the number of rows to retrieve for a group.
<b>value_limit</b>	int	An integer that puts a maximum limit on the number of values to retrieve from a group of rows.
<b>delimit_blanks</b>	boolean	A boolean value that indicates whether to delimit blanks with the <code>separator</code> value.
<b>prefix</b>	string	Optional string to add as a prefix to the resulting string.
<b>suffix</b>	string	Optional string to add as a suffix to the resulting string.
<b>max_length</b>	int	Optional integer that puts a maximum limit on the number of characters the resulting string can have.

## Returns

Data Type	Description
string	The concatenated string.

## MAX

This function returns the maximum value from each group of values.

## Syntax

```
MAX(value1 [, value2 ] [, valueN ])
```

Argument	Data Type	Description
value1–N	RDF term	The group(s) of literal, URI, or blank node values for which to return the maximum value.

## Returns

Data Type	Description
RDF term	The maximum value from each group.

## MEDIAN

This function returns the median value from a group of numbers. The median is the number in the group where half of the numbers are greater than the number and half are less than the number.

## Syntax

```
MEDIAN(number)
```

Argument	Data Type	Description
number	numeric	The group of numeric values for which to calculate the median.

## Returns

Data Type	Description
number	The median for the group.

## MIN

This function returns the minimum value from each group of values.

## Syntax

```
MIN(value1 [, value2 ] [, valueN ])
```

Argument	Data Type	Description
<b>value1–N</b>	RDF term	The group(s) of literal, URI, or blank node values for which to return the minimum value.

## Returns

Data Type	Description
RDF term	The minimum value from each group.

## MODE

This function returns the mode from a group of values. The mode is the value that occurs most frequently in the group.

## Syntax

```
MODE (value)
```

Argument	Data Type	Description
<b>value</b>	RDF term	The group of literal, URI, or blank node values for which to return the mode.

## Returns

Data Type	Description
RDF term	The mode from the group.

## MODEPERCENT

This function calculates the percentage of values in a group that belong to the mode.

## Syntax

```
MODEPERCENT (value)
```

Argument	Data Type	Description
<b>value</b>	RDF term	The group of literal, URI, or blank node values for which to calculate the modepercent.

## Returns

Data Type	Description
double	The percentage of values that belong to the mode.

## PERCENTILE\_CONT

This function calculates a percentile based on the continuous distribution of the specified group of values. The returned value is interpolated and may not be equal to any of the values in the group.

## Syntax

```
PERCENTILE_CONT (percentile, value)
```

Argument	Data Type	Description
<b>percentile</b>	float	A float value that specifies the percentile to compute.
<b>value</b>	RDF term	The group of literal, URI, or blank node values for which to calculate the percentile.

## Returns

Data Type	Description
RDF term	The interpolated percentile.



## PERCENTILE\_DISC

This function calculates a percentile based on the discrete distribution of the specified group of values.

### Syntax

```
PERCENTILE_DISC (percentile, value)
```

Argument	Data Type	Description
<b>percentile</b>	float	A float value that specifies the percentile to compute.
<b>value</b>	RDF term	The group of literal, URI, or blank node values for which to calculate the percentile.

### Returns

Data Type	Description
RDF term	The percentile based on the discrete distribution of the group.

## PRODUCT

This function calculates the product of a group of numbers.

### Syntax

```
PRODUCT (number)
```

Argument	Data Type	Description
<b>number</b>	numeric	The group of numbers to multiply.

### Returns

Data Type	Description
number	The product of the group.

## SAMPLE

This function returns an arbitrary value from the specified group of values.

### Syntax

```
SAMPLE (value)
```

Argument	Data Type	Description
<b>value</b>	RDF term	The group of literal, URI, or blank node values from which to choose a sample value.

### Returns

Data Type	Description
RDF term	The arbitrary value from the group.

## STDEV

This function measures the standard deviation (amount of dispersion) of a group of numbers.

### Syntax

```
STDEV (value)
```

Argument	Data Type	Description
<b>value</b>	numeric	The numeric value that defines the set of numbers for which to measure the standard deviation.

### Returns

Data Type	Description
number	The standard deviation of the group.

## STDEVP

This function calculates the product of the standard deviation for a group of numbers.

## Syntax

```
STDEVP (value)
```

Argument	Data Type	Description
<b>value</b>	numeric	The numeric value that defines the set of numbers for which to measure the standard deviation and compute the product.

## Returns

Data Type	Description
number	The product of the standard deviation for the group.

## SUM

This function calculates the sum of the numbers within a group.

## Syntax

```
SUM (number)
```

Argument	Data Type	Description
<b>number</b>	numeric	The group of numbers to sum.

## Returns

Data Type	Description
number	The sum of the values in the group.

## SUMIF

This function calculates the sum of the range of values that meet the specified criterion.

### Tip

You can use [SUMIFS](#) to specify multiple value ranges and conditions.

## Syntax

```
SUMIF(values_to_test, criterion [, range_of_values ])
```

Argument	Data Type	Description
<b>values_to_test</b>	RDF term	The literal, URI, or blank node value that defines the values to test against the <code>criterion</code> .
<b>criterion</b>	RDF term	The literal, URI, or blank node value that defines the condition to test values against.
<b>range_of_values</b>	numeric	An optional number that defines the range of values to sum. When omitted, <code>values_to_test</code> is used.

## Returns

Data Type	Description
number	The sum of the range of values.

## SUMIFS

This function calculates the sums of the ranges of values that meet the specified criteria. Unlike [SUMIF](#), this function enables you to specify multiple ranges and multiple conditions.

## Syntax

```
SUMIFS(values_to_sum, value_range1, criterial,  
        value_range2, criteria2  
        [, value_rangeN, criteriaN ])
```

Argument	Data Type	Description
<b>values_to_sum</b>	numeric	The numeric value that defines the overall range of values to evaluate.
<b>value_range1–N</b>	RDF term	The literal, URI, or blank node value that defines the range of values to test against the corresponding <code>criteria</code> .

Argument	Data Type	Description
<b>criteria1–N</b>	RDF term	The literal, URI, or blank node value that defines the condition to test the corresponding <code>value_range</code> against.

## Returns

Data Type	Description
number	The sums of the ranges of values.

## SUMPRODUCT

This function multiplies the numbers in a group and adds the results.

## Syntax

```
SUMPRODUCT (number)
```

Argument	Data Type	Description
<b>number</b>	numeric	The group of numbers to multiply and then sum the results.

## Returns

Data Type	Description
number	The sum of the product of the numbers in the group.

## SUMSQ

This function calculates the square root of each number in a group and adds the results.

## Syntax

```
SUMSQ (number)
```

Argument	Data Type	Description
<b>number</b>	numeric	The group of numbers for which to calculate the square root and then sum

Argument	Data Type	Description
		the results.

## Returns

Data Type	Description
number	The sum of the square root of the numbers in the group.

## VAR

This function calculates the unbiased (sample) variance for a group of numbers.

## Syntax

```
VAR(value)
```

Argument	Data Type	Description
<b>value</b>	numeric	The numeric value that defines the set of numbers for which to measure the variance.

## Returns

Data Type	Description
number	The unbiased variance for the group.

## VARP

This function calculates the biased (population) variance for a group of numbers.

## Syntax

```
VARP(value)
```

Argument	Data Type	Description
<b>value</b>	number	The value that defines the set of numbers for which to measure the population variance.

## Returns

Data Type	Description
decimal	The biased variance for the group.

## WEIGHTEDAVERAGE

This function calculates the weighted average of a group of values.

## Syntax

```
WEIGHTEDAVERAGE (value, weight)
```

Argument	Data Type	Description
<b>value</b>	decimal	The decimal value that defines the group of values for which to calculate the weighted average.
<b>weight</b>	decimal	The decimal value that defines the weight to use in the calculation.

## Returns

Data Type	Description
decimal	The weighted average for the group.

## Date and Time Functions

This topic describes the date, time, and duration functions in Anzo.

### Typographical Conventions

This documentation uses the following conventions in function syntax:

- **CAPS**: Although SPARQL is case-insensitive, function names and other keywords are written in uppercase for readability.
- `[ argument ]`: Brackets are used to indicate optional arguments. Arguments without brackets are required.

### Functions

- **DATE**: Returns an `xsd:date` value based on the specified year, month, and day.
- **DATEPART**: Returns the date part of a literal string, date, long, or `dateTime` value.
- **DATETIME** (or `xsd:dateTime`): Returns a `dateTime` value from the given string, long, or `dateTime`.
- **DAY**: Returns the day of the month from the specified date.
- **DAYSFROMDURATION**: Returns the days portion of a duration value.
- **DUR\_TO\_MILLIS**: Calculates the time in milliseconds from a duration or numeric value.
- **DURATION**: Returns an `xsd:duration` value from the given numeric or duration value.
- **DURATIONFORMAT**: Converts a duration or numeric value to a string in the specified duration format.
- **DURATIONPERIODFORMAT**: Calculates the duration between the given start and end values and returns a string in the specified duration format.
- **FORMATDATE**: Converts a numeric or date value into a string with the specified date format.
- **HOURL**: Returns the hour portion of the given `dateTime` value.
- **MASKEDDATETIME**: Replaces the year, month, day, hour, minute, second, and millisecond values for the given date or `dateTime` value with the new date and time values that you specify.
- **MILLIS**: Calculates the number of milliseconds in the given date or `dateTime` value.
- **MINUTE**: Returns the minutes portion of the given `dateTime` value.
- **MONTH**: Returns the month portion of the given `dateTime` value.
- **NOW**: Returns the current server date and time.
- **NOWMILLIS**: Returns the current server date and time in epoch milliseconds.



- **PARSEDATETIME**: Attempts to convert the given string or plain literal to a date, time, or dateTime value.
- **SECOND**: Returns the seconds portion of the given dateTime value.
- **TIME**: Returns an xsd:time value based on the specified hour, minute, and second values.
- **TIMEPART**: Returns the time part of a time or dateTime value.
- **TIMEVALUE**: Converts the specified RDF term type value to an xsd:time value.
- **TIMEZONE**: Returns as a duration the timezone part of a dateTime value.
- **TODAY**: Returns today's date based on the server date.
- **TZ**: Returns as a string the timezone from a dateTime value.
- **WEEKDAY**: Returns the day of the week from a date or dateTime value.
- **WEEKNUM**: Returns the week of the year in which the given date or dateTime occurs.
- **xsd:date**: Converts the specified string, date, or dateTime value to an xsd:date.
- **YEAR**: Returns the year portion of the given dateTime value.
- **YEARMONTH**: Returns the year and month (in the format "year-month") from the specified date or dateTime value.

## DATE

This function returns an xsd:date value based on the specified year, month, and day values.

### Syntax

```
DATE (year, month, day)
```

Argument	Data Type	Description
<b>year</b>	int	An integer that represents the year.
<b>month</b>	int	An integer that represents the month.
<b>day</b>	int	An integer that represents the day.

## Returns

Data Type	Description
date	The date according to the input values.

## DATEPART

This function returns the date part of a literal string, date, long, or dateTime value.

## Syntax

```
DATEPART (value)
```

Argument	Data Type	Description
<b>value</b>	literal string, date, long, or dateTime	The literal string, date, long, or dateTime value from which to return the date.

## Returns

Data Type	Description
date	The date part of the input values.

## DATETIME (or xsd:dateTime)

This function returns a dateTime value from the given string, long, or dateTime.

## Syntax

```
DATETIME (value)
```

Argument	Data Type	Description
<b>value</b>	string, long, dateTime	The string, long, or dateTime value from which to return a dateTime.

## Returns

Data Type	Description
dateTime	The dateTime value.

## DAY

This function returns the day of the month from the specified date value.

## Syntax

```
DAY (value)
```

Argument	Data Type	Description
value	date	The date value from which to return the day of the month.

## Returns

Data Type	Description
int	The day of the month.

## DAYSFROMDURATION

This function returns the days portion of a duration value.

## Syntax

```
DAYSFROMDURATION (value)
```

Argument	Data Type	Description
value	duration	The duration value from which to return the days.

## Returns

Data Type	Description
long	The number of days in the duration.

## DUR\_TO\_MILLIS

This function calculates the time in milliseconds from a duration or numeric value.

### Syntax

```
DUR_TO_MILLIS (value)
```

Argument	Data Type	Description
<b>value</b>	duration, numeric	The duration or numeric value from which to calculate the time in milliseconds.

### Returns

Data Type	Description
long	The number of milliseconds.

## DURATION

This function returns an xsd:duration value from the given numeric or duration value.

### Syntax

```
DURATION (value)
```

Argument	Data Type	Description
<b>value</b>	duration, numeric	The duration or numeric value from which to return an xsd:duration.

### Returns

Data Type	Description
duration	The duration value.

## DURATIONFORMAT

This function converts a duration or numeric value to a string in the specified duration format.

Syntax

```
DURATIONFORMAT(value [, format ])
```

Argument	Data Type	Description
value	duration, numeric	The duration or numeric value to format.
format	string	An optional value that specifies the format to use for the resulting duration string. Anzo supports Pattern Tokens for defining the format: <ul style="list-style-type: none"><li>• <b>y</b> for year digits</li><li>• <b>M</b> for months</li><li>• <b>d</b> for days</li><li>• <b>H</b> for hours</li><li>• <b>m</b> for minutes</li><li>• <b>s</b> for seconds</li><li>• <b>S</b> for milliseconds</li><li>• <b>'text'</b> for arbitrary text content</li></ul> If <code>format</code> is not specified, <code>H:mm:ss.SSS</code> is used.

Returns

Data Type	Description
string	The duration as a string.

DURATIONPERIODFORMAT

This function calculates the duration between the given start and end dateTime or numeric values and returns a string in the specified duration format.

Syntax

```
DURATIONPERIODFORMAT(start, end [, format ])
```

Argument	Data Type	Description
<b>start</b>	dateTime, numeric	The dateTime or numeric value that is the start of the duration period.
<b>end</b>	dateTime, numeric	The dateTime or numeric value that is the end of the duration period.
<b>format</b>	string	<p>An optional value that specifies the format to use for the resulting duration string. Anzo supports Pattern Tokens for defining the format:</p> <ul style="list-style-type: none"> <li>• <b>y</b> for year digits</li> <li>• <b>M</b> for months</li> <li>• <b>d</b> for days</li> <li>• <b>H</b> for hours</li> <li>• <b>m</b> for minutes</li> <li>• <b>s</b> for seconds</li> <li>• <b>S</b> for milliseconds</li> <li>• <b>'text'</b> for arbitrary text content</li> </ul> <p>If <code>format</code> is not specified, the default is <code>'P'yyyy'Y'M'M'd'DT'H'H'm'M's.SSS'S'</code>. The default value uses 'text' patterns with Pattern Tokens, which results in a string such as <code>P1Y3M4DT1H4M44.000S</code>.</p>

## Returns

Data Type	Description
string	The duration as a string.

## FORMATDATE

This function converts a numeric or date value into a string with the specified date format.

## Syntax

```
FORMATDATE(value, format)
```

Argument	Data Type	Description
<b>value</b>	date, numeric	The date or numeric value to convert to a string in the specified date format.
<b>format</b>	string	<p>The format to use for the resulting date string. Anzo supports Pattern Tokens for defining the format:</p> <ul style="list-style-type: none"> <li>• <b>y</b> for year digits</li> <li>• <b>M</b> for months</li> <li>• <b>d</b> for days</li> <li>• <b>H</b> for hours</li> <li>• <b>m</b> for minutes</li> <li>• <b>s</b> for seconds</li> <li>• <b>S</b> for milliseconds</li> <li>• <b>'text'</b> for arbitrary text content</li> </ul>

## Returns

Data Type	Description
string	The date as a string.

## HOUR

This function returns the hour portion of the given dateTime value.

## Syntax

```
HOUR(value [, timezone ])
```

Argument	Data Type	Description
<b>value</b>	dateTime	The dateTime value from which to return the hour portion.
<b>timezone</b>	string	An optional value that specifies the timezone for the <code>value</code> .

## Returns

Data Type	Description
int	The hour.

## MASKEDDATETIME

This function replaces the year, month, day, hour, minute, second, and millisecond values for the given date or dateTime value with the new date and time values that you specify.

## Syntax

```
MASKEDDATETIME(value, year, month, day, hour, minute, second, milliseconds)
```

Argument	Data Type	Description
<b>value</b>	date, dateTime	The date or dateTime for which to replace the year, month, date, hour, minute, second, and milliseconds values.
<b>year</b>	int	The year to include in the resulting dateTime value.
<b>month</b>	int	The month to include in the resulting dateTime value.
<b>day</b>	int	The day to include in the resulting dateTime value.
<b>hour</b>	int	The hour to include in the resulting dateTime value.
<b>minute</b>	int	The minutes value to include in the resulting dateTime value.
<b>second</b>	int	The seconds value to include in the resulting dateTime value.
<b>milliseconds</b>	int	The milliseconds value to include in the resulting dateTime value.

## Returns

Data Type	Description
dateTime	The dateTime value with the specified input values.



## MILLIS

This function calculates the number of milliseconds in the given date or dateTime value.

### Syntax

```
MILLIS (value)
```

Argument	Data Type	Description
value	date, dateTime	The date or dateTime value for which to calculate the number of milliseconds.

### Returns

Data Type	Description
long	The number of milliseconds.

## MINUTE

This function returns the minutes portion of the given dateTime value.

### Syntax

```
MINUTE (value)
```

Argument	Data Type	Description
value	dateTime	The dateTime value from which to return the minutes portion.

### Returns

Data Type	Description
int	The minutes portion of the input value.

## MONTH

This function returns the month portion of the given dateTime value.

## Syntax

```
MONTH (value)
```

Argument	Data Type	Description
value	dateTime	The dateTime value from which to return the month portion.

## Returns

Data Type	Description
int	The month number.

## NOW

This function returns the current server date and time.

## Syntax

```
NOW([ timezone ])
```

Argument	Data Type	Description
timezone	string	An optional value that specifies the timezone for which to return the current dateTime.

## Returns

Data Type	Description
dateTime	The current server date and time.

## NOWMILLIS

This function returns the current server date and time in epoch milliseconds.

## Syntax

```
NOWMILLIS ()
```

## Returns

Data Type	Description
long	The current server date and time in milliseconds.

## PARSEDATETIME

This function attempts to convert the given string or plain literal to a date, time, or dateTime value.

## Syntax

```
PARSEDATETIME (value [, output_type ])
```

Argument	Data Type	Description
<b>value</b>	string, literal	The string or plain literal value to convert to a date, time, or dateTime.
<b>output_type</b>	URI	An optional URI ( <code>xsd:date</code> , <code>xsd:time</code> , or <code>xsd:dateTime</code> ) that specifies the type of value to return. If <code>output_type</code> is not specified, <code>dateTime</code> is returned.

## Returns

Data Type	Description
date, time, or dateTime	The conversion of the string to the desired type.

## SECOND

This function returns the seconds portion of the given dateTime value.

## Syntax

```
SECOND (value)
```

Argument	Data Type	Description
<b>value</b>	dateTime	The dateTime value from which to return the seconds portion.

## Returns

Data Type	Description
int	The seconds portion of the input value.

## TIME

This function returns an xsd:time value based on the specified hour, minute, and second values.

## Syntax

```
TIME(hour, minute, second)
```

Argument	Data Type	Description
<b>hour</b>	int	An integer that represents the hour.
<b>minute</b>	int	An integer that represents the minute.
<b>second</b>	int	An integer that represents the seconds.

## Returns

Data Type	Description
time	The time according to the input values.

## TIMEPART

This function returns the time part of a time or dateTime value.

## Syntax

```
TIMEPART(value)
```

Argument	Data Type	Description
<b>value</b>	time, dateTime	The time or dateTime value from which to return the time portion.

## Returns

Data Type	Description
time	The time portion of the input value.

## TIMEVALUE

This function converts the specified RDF term type value to an xsd:time value.

## Syntax

```
TIMEVALUE (value)
```

Argument	Data Type	Description
<b>value</b>	RDF term	The literal, URI, or blank node value to convert to a time value.

## Returns

Data Type	Description
time	The conversion of the term to a time value.

## TIMEZONE

This function returns the timezone part of a dateTime value as a duration. An error is returned if the input value does not include the timezone.

## Syntax

```
TIMEZONE (value)
```

Argument	Data Type	Description
<b>value</b>	dateTime	The dateTime value to return the timezone from.

## Returns

Data Type	Description
duration	The timezone in duration format.

## TODAY

This function returns today's date based on the server date.

## Syntax

```
TODAY ()
```

## Returns

Data Type	Description
date	Today's date according to the server.

## TZ

This function returns the timezone part of a dateTime value as a string.

## Syntax

```
TZ (value)
```

Argument	Data Type	Description
value	dateTime	The dateTime value to return the timezone from.

## Returns

Data Type	Description
string	The timezone as a string.

## WEEKDAY

This function returns the day of the week from a date or dateTime value.

## Syntax

```
WEEKDAY(value [, day_number_start ])
```

Argument	Data Type	Description
<b>value</b>	date, dateTime	The date or dateTime value from which to return the day of the week.
<b>day_number_start</b>	int	<p>An optional value of <b>1</b>, <b>2</b>, or <b>3</b> that defines how the days of the week are represented as numbers.</p> <ul style="list-style-type: none"><li>• <b>1</b> means Sunday is day 1. Saturday is day 7.</li><li>• <b>2</b> means Monday is day 1. Sunday is day 7.</li><li>• <b>3</b> means Monday is day 0. Sunday is day 6.</li></ul> <p>If <code>day_number_start</code> is not specified, the default value is <b>1</b>.</p>

## Returns

Data Type	Description
int	The day of the week from the input values.

## WEEKNUM

This function returns the week of the year in which the given date or dateTime occurs.

## Syntax

```
WEEKNUM(value [, day_week_begins ])
```

Argument	Data Type	Description
<b>value</b>	date, dateTime	The date or dateTime value from which to return the week number.
<b>day_week_begins</b>	int	An optional value of <b>1</b> or <b>2</b> that defines which day the weeks start on.

Argument	Data Type	Description
		<ul style="list-style-type: none"> <li>• <b>1</b> means a new week starts on Sunday.</li> <li>• <b>2</b> means a new week starts on Monday.</li> </ul> <p>If <code>day_week_begins</code> is not specified, the default value is <b>1</b>.</p>

## Returns

Data Type	Description
int	The week of the year the input value falls in.

## xsd:date

This function converts the specified string, date, or dateTime value to an xsd:date.

## Syntax

```
xsd:date (value)
```

Argument	Data Type	Description
<b>value</b>	string, date, dateTime	The string, date, or dateTime value to convert to an xsd:date.

## Returns

Data Type	Description
date	The input values converted to dates.

## YEAR

This function returns the year portion of the given dateTime value.

## Syntax

```
YEAR (value)
```



Argument	Data Type	Description
<b>value</b>	dateTime	The dateTime value to return the year from.

## Returns

Data Type	Description
int	The year portion of the input values.

## YEARMONTH

This function returns the year and month (in the format "year-month") from the specified date or dateTime value.

## Syntax

```
YEARMONTH (value)
```

Argument	Data Type	Description
<b>value</b>	literal date or dateTime	The value to return the year-month from.

## Returns

Data Type	Description
gYearMonth	The year-month from the input values.

## Casting Functions

This topic describes the functions that are available for coercing data types in Anzo.

### Typographical Conventions

This documentation uses the following conventions in function syntax:

- **CAPS**: Although SPARQL is case-insensitive, function names and other keywords are written in uppercase for readability.
- `[ argument ]`: Brackets are used to indicate optional arguments. Arguments without brackets are required.

### Functions

- **BNODE**: Creates a blank node.
- **BOOLEAN**: Casts a literal value to a boolean data type.
- **BYTE**: Casts a literal value to a byte data type.
- **CONCATURL**: Concatenates two or more strings and returns the result as a URI.
- **DATATYPE**: Returns the data type of the given value.
- **DATETIME (or xsd:dateTime)**: Returns a dateTime value from the given string, long, or dateTime.
- **DATEVALUE**: Casts a string to a date.
- **DECIMAL**: Casts a literal value to a decimal data type.
- **DOUBLE**: Casts a literal value to a double data type.
- **DURATION**: Returns an xsd:duration value from the given numeric or duration value.
- **DURATIONFORMAT**: Converts a duration or numeric value to a string in the specified duration format.
- **ENCODE\_FOR\_URI**: Encodes the specified string as a URI.
- **FLOAT**: Casts a literal value to a float data type.
- **FORMATDATE**: Casts a numeric or date value to a string in the specified date format.
- **FORMATFRACTION**: Converts a numeric value into a fraction string.
- **FORMATNUMBER**: Casts a numeric value to a string in the specified format.
- **INT**: Casts a literal value to an int data type.
- **INTEGER**: Casts a literal value to an integer data type.

- **LONG**: Casts a literal value to a long data type.
- **PARSEDATETIME**: Attempts to convert the given string or plain literal to a date, time, or dateTime value.
- **RAD**: Converts to radians an angle value that is in degrees.
- **SERIALIZE**: Creates a literal value from the string representation of the specified RDF term.
- **SHORT**: Casts a literal value to a short data type.
- **STR**: Casts an RDF term type value to a string.
- **TEXT**: Casts a numeric or dateTime value to a string in the specified format.
- **TIMEVALUE**: Converts the specified RDF term type value to an xsd:time value.
- **TOURI**: Casts a string literal value to a URI.
- **UUID**: Generates a Universally Unique Identifier (UUID).
- **xsd:date**: Converts the specified string, date, or dateTime value to an xsd:date.

## BNODE

This function creates a blank node.

### Syntax

```
BNODE([ value ])
```

Argument	Data Type	Description
value	string	An optional string value from which to create the blank node.

### Returns

Data Type	Description
blank node	The generated blank node.

## BOOLEAN

This function casts a literal value to a boolean data type.

### Syntax

```
BOOLEAN(value)
```

Argument	Data Type	Description
<b>value</b>	literal	The literal value to cast to a boolean type.

## Returns

Data Type	Description
boolean	The input cast to boolean.

## BYTE

This function casts a literal value to a byte data type.

## Syntax

```
BYTE (value)
```

Argument	Data Type	Description
<b>value</b>	literal	The literal value to cast to a byte type.

## Returns

Data Type	Description
byte	The input value cast to a byte.

## CONCATURL

This function concatenates two or more strings and returns the result as a URI.

## Syntax

```
CONCATURL(text1, text2 [, textN ])
```

Argument	Data Type	Description
<b>text1–N</b>	string	The strings that you want to concatenate to form a URI.

## Returns

Data Type	Description
URI	The concatenated string as a URI.

## DATATYPE

This function returns the data type of the given literal value.

## Syntax

```
DATATYPE (value)
```

Argument	Data Type	Description
value	literal	The literal value for which to return the data type.

## Returns

Data Type	Description
data type URI	The data type.

## DATETIME (or xsd:dateTime)

This function returns a dateTime value from the given string, long, or dateTime.

## Syntax

```
DATETIME (value)
```

Argument	Data Type	Description
value	string, long, dateTime	The string, long, or dateTime value from which to return a dateTime.

## Returns

Data Type	Description
dateTime	The dateTime value.

## DATEVALUE

This function casts the given string value to a date.

## Syntax

```
DATEVALUE (value)
```

Argument	Data Type	Description
value	string	The string value from which to return the date.

## Returns

Data Type	Description
date	The string cast to a date.

## DECIMAL

This function casts a literal value to a decimal data type.

## Syntax

```
DECIMAL (value)
```

Argument	Data Type	Description
value	literal	The literal value to convert to a decimal type.

## Returns

Data Type	Description
decimal	The literal value cast to a decimal type.

## DOUBLE

This function casts a literal value to a double data type.

### Syntax

```
DOUBLE (value)
```

Argument	Data Type	Description
value	literal	The literal value to convert to a double type.

### Returns

Data Type	Description
double	The literal value cast to a double type.

## DURATION

This function returns an xsd:duration value from the given numeric or duration value.

### Syntax

```
DURATION (value)
```

Argument	Data Type	Description
value	duration, numeric	The duration or numeric value from which to return an xsd:duration.

### Returns

Data Type	Description
duration	The duration value.

## DURATIONFORMAT

This function converts a duration or numeric value to a string in the specified duration format.

## Syntax

```
DURATIONFORMAT(value [, format ])
```

Argument	Data Type	Description
<b>value</b>	duration, numeric	The duration or numeric value to format.
<b>format</b>	string	An optional value that specifies the format to use for the resulting duration string. Anzo supports Pattern Tokens for defining the format: <ul style="list-style-type: none"><li>• <b>y</b> for year digits</li><li>• <b>M</b> for months</li><li>• <b>d</b> for days</li><li>• <b>H</b> for hours</li><li>• <b>m</b> for minutes</li><li>• <b>s</b> for seconds</li><li>• <b>S</b> for milliseconds</li><li>• <b>'text'</b> for arbitrary text content</li></ul> If <code>format</code> is not specified, <code>H:mm:ss.SSS</code> is used.

## Returns

Data Type	Description
string	The duration as a string.

## ENCODE\_FOR\_URI

This function encodes the specified string as a URI and returns a string in URI format.

## Syntax

```
ENCODE_FOR_URI(text)
```



Argument	Data Type	Description
<b>text</b>	string	The string value to encode as a URI.

## Returns

Data Type	Description
string	The string as a URI.

## FLOAT

This function casts a literal value to a float data type.

## Syntax

```

FLOAT (value)

```

Argument	Data Type	Description
<b>value</b>	literal	The literal value to convert to a float type.

## Returns

Data Type	Description
float	The literal value cast to a float type.

## FORMATDATE

This function converts a numeric or date value into a string with the specified date format.

## Syntax

```

FORMATDATE (value, format)

```

Argument	Data Type	Description
<b>value</b>	date, numeric	The date or numeric value to convert to a string in the specified date format.

Argument	Data Type	Description
<b>format</b>	string	<p>The format to use for the resulting date string. Anzo supports Pattern Tokens for defining the format:</p> <ul style="list-style-type: none"> <li>• <b>y</b> for year digits</li> <li>• <b>M</b> for months</li> <li>• <b>d</b> for days</li> <li>• <b>H</b> for hours</li> <li>• <b>m</b> for minutes</li> <li>• <b>s</b> for seconds</li> <li>• <b>S</b> for milliseconds</li> <li>• <b>'text'</b> for arbitrary text content</li> </ul>

## Returns

Data Type	Description
string	The date as a string.

## FORMATFRACTION

This function converts a numeric value into a fraction string.

## Syntax

```
FORMATFRACTION(value [, tolerance ] [, whole_number ])
```

Argument	Data Type	Description
<b>value</b>	numeric	The numeric value to convert to fraction text.
<b>tolerance</b>	double	An optional double value that specifies the precision of the fraction. The default value is 0.0001. The resulting fractional representation is the original value + or - the <code>tolerance</code> . The smaller the tolerance, the more precise the fraction is. For example, 399/800 vs. 1/2.

Argument	Data Type	Description
<b>whole_number</b>	boolean	An optional boolean value that specifies whether to include whole numbers in the result. For example, if <code>true</code> , the result would be formatted like <code>1 2/3</code> instead of <code>5/3</code> .

## Returns

Data Type	Description
string	The fraction string.

## FORMATNUMBER

This function casts a numeric value to a string in the specified format.

## Syntax

```
TEXT(value, format)
```

Argument	Data Type	Description
<b>value</b>	numeric	The numeric value to convert to a string.
<b>format</b>	string	A text string that specifies the format to follow when converting the <code>value</code> to a string. Anzo supports <a href="#">Java Decimal Format</a> .

## Returns

Data Type	Description
string	The numeric value as a string.

## INT

This function casts a literal value to an int data type.

## Syntax

```
INT(value)
```

Argument	Data Type	Description
<b>value</b>	literal	The literal value to convert to an int type.

## Returns

Data Type	Description
int	The literal value cast to an int type.

## INTEGER

This function casts a literal value to an integer data type.

## Syntax

```
INTEGER(value)
```

Argument	Data Type	Description
<b>value</b>	literal	The literal value to convert to an integer type.

## Returns

Data Type	Description
integer	The literal value cast to an integer type.

## LONG

This function casts a literal value to a long data type.

## Syntax

```
LONG(value)
```

Argument	Data Type	Description
<b>value</b>	literal	The literal value to convert to a long type.

## Returns

Data Type	Description
long	The literal value cast to a long type.

## PARSEDATETIME

This function attempts to convert the given string or plain literal to a date, time, or dateTime value.

## Syntax

```
PARSEDATETIME (value [, output_type ])
```

Argument	Data Type	Description
<b>value</b>	string, literal	The string or plain literal value to convert to a date, time, or dateTime.
<b>output_type</b>	URI	An optional URI ( <code>xsd:date</code> , <code>xsd:time</code> , or <code>xsd:dateTime</code> ) that specifies the type of value to return. If <code>output_type</code> is not specified, <code>dateTime</code> is returned.

## Returns

Data Type	Description
date, time, or dateTime	The conversion of the string to the desired type.

## RAD

This function converts to radians an angle value that is in degrees.

## Syntax

```
RAD (angle)
```

Argument	Data Type	Description
<b>angle</b>	double	The angle value to convert to radians.

## Returns

Data Type	Description
double	The angle in radians.

## SERIALIZE

This function creates a literal value from the string representation of the specified RDF term type value.

## Syntax

```
SERIALIZE (value)
```

Argument	Data Type	Description
<b>value</b>	RDF term	The literal, URI, or blank node value for which to generate a plain literal.

## Returns

Data Type	Description
string	The string representation of the input term.

## SHORT

This function casts a literal value to a short data type.

## Syntax

```
SHORT (value)
```

Argument	Data Type	Description
<b>value</b>	literal	The literal value to convert to a short type.

## Returns

Data Type	Description
short	The literal value cast to a short type.

## STR

This function casts the specified RDF term type value to a string.

## Syntax

```
STR(value)
```

Argument	Data Type	Description
value	RDF term	The literal, URI, or blank node value to convert to a string.

## Returns

Data Type	Description
string	The term cast to a string type.

## TEXT

This function casts a numeric or dateTime value to a string in the specified format.

## Syntax

```
TEXT(value, format)
```

Argument	Data Type	Description
value	numeric, dateTime	The numeric or datetime value to convert to a string.
format	string	A text string that specifies the format to follow when converting the <code>value</code> to a string. For numeric values, Anzo supports <a href="#">Java Decimal Format</a> . For dateTime values, <a href="#">Simple Date Format</a> is supported.

## Returns

Data Type	Description
string	The string in the specified format.

## TIMEVALUE

This function converts the specified RDF term type value to an xsd:time value.

## Syntax

```
TIMEVALUE (value)
```

Argument	Data Type	Description
value	RDF term	The literal, URI, or blank node value to convert to a time value.

## Returns

Data Type	Description
time	The conversion of the term to a time value.

## TOURI

This function casts a string literal value to a URI.

## Syntax

```
TOURI (text)
```

Argument	Data Type	Description
text	string	The string literal to cast to a URI.

## Returns

Data Type	Description
URI	The literal value as a URI.



## UUID

This function generates a Universally Unique Identifier (UUID).

### Syntax

```
UUID ()
```

### Returns

Data Type	Description
URI	The UUID.

## xsd:date

This function converts the specified string, date, or dateTime value to an xsd:date.

### Syntax

```
xsd:date (value)
```

Argument	Data Type	Description
value	string, date, dateTime	The string, date, or dateTime value to convert to an xsd:date.

### Returns

Data Type	Description
date	The input values converted to dates.

## Logical Functions

This topic describes the logical functions in Anzo.

### Typographical Conventions

This documentation uses the following conventions in function syntax:

- **CAPS**: Although SPARQL is case-insensitive, function names and other keywords are written in uppercase for readability.
- `[ argument ]`: Brackets are used to indicate optional arguments. Arguments without brackets are required.

### Functions

- **AND**: Evaluates two logical expressions and returns true if both expressions are true.
- **BOUND**: Evaluates whether an RDF term type is bound.
- **CASE**: Evaluates a series of conditions for the specified value and returns the matching result.
- **COALESCE**: Evaluates a number of expressions and returns the results for the first expression that is bound and does not raise an error.
- **EQUAL**: Evaluates whether two values are equal.
- **IF**: Evaluates a condition and returns the specified result depending on the outcome of the test.
- **IFERROR**: Synonym for COALESCE.
- **IN**: Evaluates whether the specified RDF term is found in any of the given test values.
- **NOT**: Evaluates whether the specified logical expression is not true.
- **NOT\_EQUAL**: Evaluates whether two values are not equal.
- **NOT\_IN**: Evaluates whether the specified RDF term is not found in any of the given test values.
- **OR**: Evaluates two logical expressions and returns true if at least one of the expressions is true.
- **PARTITIONINDEX**: Returns the zero-based index of the bucket in which the specified value falls.
- **SAMETERM**: Evaluates whether two RDF term type values are the same.
- **UNBOUND**: Extends the SPARQL UNDEF functionality to enable users to include an undefined value as a function argument.

## AND

This function evaluates two logical expressions. If both expressions are true, the function returns `true`. If one or both arguments are false, the function returns `false`.

### Syntax

```
AND(logical_expression1, logical_expression2)
```

Argument	Data Type	Description
<code>logical_expression1</code>	evaluates to boolean	The first logical expression to evaluate.
<code>logical_expression2</code>	evaluates to boolean	The second logical expression to evaluate.

### Returns

Data Type	Description
boolean	<code>True</code> if both conditions are true and <code>false</code> if either condition is false.

## BOUND

This function evaluates whether the specified RDF term has a value bound to it.

### Syntax

```
BOUND(term)
```

Argument	Data Type	Description
<code>term</code>	RDF term	The literal, URI, or blank node value to evaluate.

### Returns

Data Type	Description
boolean	<code>True</code> if the term is bound and <code>false</code> if it is not.

## CASE

This function evaluates a series of conditions for the specified value and returns the matching result. CASE acts like an IF-THEN-ELSE statement.

### Syntax

```
CASE(test_value, condition1, [ condition2 ] [, conditionN ]  
      result1 [, result2 ] [, resultN ] [, default ])
```

Argument	Data Type	Description
<b>test_value</b>	RDF term	The literal, URI, or blank node value to compare to the list of conditions (condition1-N).
<b>condition1-N</b>	RDF term	The conditions to be evaluated in the order that they are specified. Once a condition evaluates to true, the corresponding result is returned and the remaining conditions are not evaluated.
<b>result1-N</b>	RDF term	The results to return for the specified conditions.
<b>default</b>	RDF term	An optional value to be returned if none of the specified conditions pass.

### Returns

Data Type	Description
RDF term	The specified result according to the evaluation of the conditions.

## COALESCE

This function evaluates a number of expressions and returns the results for the first expression that is bound and does not raise an error.

### Syntax

```
COALESCE(expression1 [, expression2 ] [, expressionN ] )
```

Argument	Data Type	Description
<b>expression1–N</b>	RDF term	The literal, URI, or blank node expressions to evaluate.

## Returns

Data Type	Description
RDF term	The result of the first expression that is bound and does not error.

## EQUAL

This function evaluates whether value1 is equal to value2.

## Syntax

```
EQUAL(value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to value2.
<b>value2</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to value1.

## Returns

Data Type	Description
boolean	<code>True</code> if the values are equal and <code>false</code> if they are not.

## IF

This function evaluates a condition and returns the specified result depending on the outcome of the test. If the condition evaluates to true, the first result is returned. If the condition evaluates to false, the second result is returned. And if the condition results in an error, the third result is returned.

## Syntax

```
IF(logical_expression, true_result, false_result [, error_result ])
```

Argument	Data Type	Description
<b>logical_expression</b>	evaluates to boolean	The condition that evaluates to true or false.
<b>true_result</b>	RDF term	The value that defines the result to return if the condition evaluates to true.
<b>false_result</b>	RDF term	The value that defines the result to return if the condition evaluates to false.
<b>error_result</b>	RDF term	An optional value that defines the result to return if the condition evaluates to an error. If the condition results in an error and <code>error_result</code> is not specified, <code>logical_expression(error)</code> is returned.

## Returns

Data Type	Description
RDF term	The result based on the evaluation of the condition.

## IFERROR

This function is a synonym for [COALESCE](#).

## IN

This function evaluates whether the specified RDF term type value is found in any of the given test values.

## Syntax

```
IN(term, test_value1 [, test_value2 ] [, test_valueN])
```

Argument	Data Type	Description
<b>term</b>	RDF term	The literal, URI, or blank node value to look for in the test values.
<b>test_value1–N</b>	RDF term	The literal, URI, or blank node values to look for the specified <code>term</code> in.

## Returns

Data Type	Description
boolean	<code>True</code> if the given term is found in the test values and <code>false</code> if it is not.

## NOT

This function evaluates whether the specified logical expression is not true.

## Syntax

```
NOT(logical_expression)
```

Argument	Data Type	Description
<b>logical_expression</b>	evaluates to boolean	The condition to evaluate.

## Returns

Data Type	Description
boolean	<code>True</code> if the condition is false and <code>false</code> if it is true.

## NOT\_EQUAL

This function evaluates whether value1 is not equal to value2.

## Syntax

```
NOT_EQUAL(value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value2</code> .
<b>value2</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value1</code> .

## Returns

Data Type	Description
boolean	True if value1 is not equal to value2. False if they are equal.

## NOT\_IN

This function evaluates whether the specified RDF term type value is not found in any of the given test values.

## Syntax

```
IN(term, test_value1 [, test_value2 ] [, test_valueN])
```

Argument	Data Type	Description
term	RDF term	The literal, URI, or blank node value to look for in the test values.
test_value1– N	RDF term	The literal, URI, or blank node values in which to look for the specified term.

## Returns

Data Type	Description
boolean	True if the given term is not in the test values and false if it is found in the test values.

## OR

This function evaluates two logical expressions. If at least one expression is true, the function returns `true`. If both expressions are false, the function returns `false`.

## Syntax

```
OR(logical_expression1, logical_expression2)
```

Argument	Data Type	Description
logical_expression1	evaluates to boolean	The first logical expression to evaluate.



Argument	Data Type	Description
<b>logical_expression2</b>	evaluates to boolean	The second logical expression to evaluate.

## Returns

Data Type	Description
boolean	<code>True</code> if one or both conditions are true and <code>false</code> if both conditions are false.

## PARTITIONINDEX

This function returns the zero-based index of the bucket in which the specified value falls. Buckets start at the specified `start` value and are sized according to the specified `interval`. The first bucket is `[start, start+interval)`. That means it is closed on the low end and open on the high end. `PARTITIONINDEX` returns less than 0 if the value does not fall into any bucket, such as when the given `value` is less than `start` or if the comparison is indeterminate for date and time data types.

## Syntax

```
PARTITIONINDEX(value, start, interval)
```

Argument	Data Type	Description
<b>value</b>	literal	The literal value for which to determine the zero-based index.
<b>start</b>	literal	The literal value that indicates the start of the first bucket.
<b>interval</b>	literal	The literal value that specifies the size of the bucket.

## Returns

Data Type	Description
long	The zero-based index of the bucket in which the specified value exists.

## SAMETERM

This function evaluates whether two RDF term type values are the same.

## Syntax

```
SAMETERM(term1, term2)
```

Argument	Data Type	Description
<b>term1</b>	RDF term	The first literal, URI, or blank node value to compare.
<b>term2</b>	RDF term	The literal, URI, or blank node value to compare to <i>term1</i> .

## Returns

Data Type	Description
boolean	<code>True</code> if the terms are the same and <code>false</code> if they are not.

## UNBOUND

This function is like the SPARQL UNDEF keyword but extends that functionality to enable users to include an undefined value as a function argument, as UNDEF is only supported in VALUES clauses.

## Syntax

```
UNBOUND()
```

## Example

The following example statement incorporates UNBOUND to return null if the specified condition (`?x > 5`) fails:

```
BIND(IF(?x > 5 , "Win", UNBOUND()) as ?testResult)
```

In this case, `?testResult` is bound if `?x` is greater than 5. If `?x` is not greater than 5, `?testResult` is not bound.

## Returns

Data Type	Description
RDF term	The specified result according to the evaluation of the condition.

## Informational or Testing Functions

This topic describes the functions in Anzo that retrieve information from your values and let you ask questions about them or test whether the values match expectations.

### Typographical Conventions

This documentation uses the following conventions in function syntax:

- **CAPS**: Although SPARQL is case-insensitive, function names and other keywords are written in uppercase for readability.
- `[ argument ]`: Brackets are used to indicate optional arguments. Arguments without brackets are required.

### Functions

- **CONTAINS**: Evaluates whether the specified string contains the given pattern.
- **GE**: Evaluates whether one value is greater than or equal to ( $\geq$ ) another value.
- **GT**: Evaluates whether one value is greater than ( $>$ ) another value.
- **ISBLANK**: Evaluates whether the given RDF term is a blank node.
- **ISDATATYPE**: Evaluates whether the given literal value is typed as the specified data type.
- **ISERROR**: Tests whether the given RDF term evaluates to an error. **Only valid in queries against the Anzo System Datasource or other volume.**
- **ISIRI**: Evaluates whether the given RDF term is an IRI.
- **ISLITERAL**: Evaluates whether the given RDF term is a literal value.
- **ISNUMERIC**: Evaluates whether the given RDF term is a numeric literal value.
- **ISURI**: Evaluates whether the given RDF term is a URI.
- **LANG**: Returns any language tags that are included with strings.
- **LANGMATCHES**: Evaluates whether a string includes a language tag that matches the specified language range.
- **LE**: Evaluates whether one value is less than or equal to ( $\leq$ ) another value.
- **LOCALNAME**: Retrieves the local name from the given URI.
- **LT**: Evaluates whether one value is less than ( $<$ ) another value.
- **METADATAGRAPHURI**: Returns the metadata graph URI for the given URI.

- **NAMESPACE**: Retrieves the namespace for the specified URI.
- **SAMETERM**: Evaluates whether two RDF term type values are the same.

## CONTAINS

This function evaluates whether the specified strings contain the given pattern. Results are grouped under "true" or "false."

### Syntax

```
CONTAINS(text, pattern)
```

Argument	Data Type	Description
<b>text</b>	string	The string value that you want to check against the specified pattern.
<b>pattern</b>	string	The string pattern that you want to look for in the supplied text.

### Returns

Data Type	Description
boolean	<code>True</code> if the strings contain the pattern and <code>false</code> if they do not.

## GE

This function evaluates whether value1 is greater than or equal to ( $\geq$ ) value2.

### Syntax

```
GE(value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value2</code> . This is the value that will be checked to see if it is greater than or equal to <code>value2</code> .
<b>value2</b>	numeric,	The number, boolean, datetime, literal, URI, or blank node value to

Argument	Data Type	Description
	boolean, dateTime, RDF term	compare to <code>value1</code> .

## Returns

Data Type	Description
boolean	True if <code>value1 &gt;= value2</code> . False if not.

## GT

This function evaluates whether `value1` is greater than (`>`) `value2`.

## Syntax

```
GE(value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value2</code> . This is the value that will be checked to see if it is greater than <code>value2</code> .
<b>value2</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value1</code> .

## Returns

Data Type	Description
boolean	True if <code>value1 &gt; value2</code> . False if not.

## ISBLANK

This function evaluates whether the given RDF term value is a blank node. It returns `true` if it is a blank node or `false` if it is not.

### Syntax

```
ISBLANK (value)
```

Argument	Data Type	Description
<b>value</b>	RDF term	The literal, URI, or blank node value to test and determine if it is a blank node.

### Returns

Data Type	Description
boolean	<code>True</code> if the term is a blank node and <code>false</code> if it is not.

## ISDATATYPE

This function evaluates whether the given literal value is typed as the specified data type. It returns `true` if the value is typed as the provided type or `false` if it is not.

### Syntax

```
ISDATATYPE (value, datatype_uri)
```

Argument	Data Type	Description
<b>value</b>	literal	The literal value that you want to test against the <code>datatype_uri</code> .
<b>datatype_uri</b>	URI	The URI for the data type that you want to test the <code>value</code> against.

### Returns

Data Type	Description
boolean	<code>True</code> if the literal is typed as specified data type and <code>false</code> if it is not.

## ISERROR

This function tests whether the given RDF term evaluates to an error. It returns `true` if it does evaluate to an error or `false` if it does not.

### Note

The ISERROR function is only for use in queries that are run against the Anzo System Datasource or other volume. It is invalid for graphmart queries.

## Syntax

```
ISERROR(term)
```

Argument	Data Type	Description
<b>term</b>	RDF term	The literal, URI, or blank node value to evaluate for an error.

## Returns

Data Type	Description
boolean	<code>True</code> if the term evaluates to an error and <code>false</code> if it does not.

## ISIRI

This function evaluates whether the given RDF term type value is an IRI. It returns `true` if the value is an IRI or `false` if it is not.

## Syntax

```
ISIRI(term)
```

Argument	Data Type	Description
<b>term</b>	RDF term	The literal, URI, or blank node value to evaluate whether it is an IRI.

## Returns

Data Type	Description
boolean	<code>True</code> if the term is an IRI and <code>false</code> if it is not.

## ISLITERAL

This function evaluates whether the given RDF term type value is a literal value. It returns `true` if the value is a literal or `false` if it is not.

## Syntax

```
ISLITERAL (term)
```

Argument	Data Type	Description
<b>term</b>	RDF term	The literal, URI, or blank node value to evaluate whether it is a literal.

## Returns

Data Type	Description
boolean	<code>True</code> if the term is a literal value and <code>false</code> if it is not.

## ISNUMERIC

This function evaluates whether the given RDF term type value is a numeric literal. It returns `true` if the value is a numeric literal or `false` if it is not.

## Syntax

```
ISNUMERIC (term)
```

Argument	Data Type	Description
<b>term</b>	RDF term	The literal, URI, or blank node value to evaluate whether it is a numeric literal.



## Returns

Data Type	Description
boolean	<code>True</code> if the term is a numeric literal and <code>false</code> if it is not.

## ISURI

This function evaluates whether the given RDF term type value is a URI. It returns `true` if the value is a URI or `false` if it is not.

## Syntax

```
ISURI (term)
```

Argument	Data Type	Description
<b>term</b>	RDF term	The literal, URI, or blank node value to evaluate whether it is a URI.

## Returns

Data Type	Description
boolean	<code>True</code> if the term is a URI and <code>false</code> if it is not.

## LANG

This function returns any language tags that are included in the string. The results are grouped by each language tag or by "blank" if a value does not have a language tag.

## Syntax

```
LANG (text)
```

Argument	Data Type	Description
<b>text</b>	string	The string to search for language tags.

## Returns

Data Type	Description
string	The found language tags.

## LANGMATCHES

This function tests whether a string includes a language tag that matches the specified language range.

## Syntax

```
LANGMATCHES(text, language_range)
```

Argument	Data Type	Description
<b>text</b>	string	The string to evaluate.
<b>language_range</b>	string	The language tag to match in the <code>text</code> .

## Example

```
LANGMATCHES(LANG(?prop), "en")
```

## Returns

Data Type	Description
boolean	<code>True</code> if strings include a language tag that matches the range and <code>false</code> if they do not.

## LE

This function evaluates whether `value1` is less than or equal to (`<=`) `value2`.

## Syntax

```
LE(value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric,	The number, boolean, datetime, literal, URI, or blank node value to

Argument	Data Type	Description
	boolean, dateTime, RDF term	compare to <code>value2</code> . This is the value that will be evaluated to see if it is less than or equal to <code>value2</code> .
<b>value2</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, literal, URI, or blank node value to compare to <code>value1</code> .

## Returns

Data Type	Description
boolean	<code>True</code> if <code>value1 &lt;= value2</code> . <code>False</code> if not.

## LOCALNAME

This function retrieves the local name from the given URI.

## Syntax

```
LOCALNAME(uri)
```

Argument	Data Type	Description
<b>uri</b>	URI	The URI from which to retrieve the local name.

## Returns

Data Type	Description
string	The local name.

## LT

This function evaluates whether `value1` is less than (`<`) `value2`.

## Syntax

```
LT(value1, value2)
```

Argument	Data Type	Description
<b>value1</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, or RDF term type value to compare to <code>value2</code> . This is the value that will be evaluated to see if it is less than <code>value2</code> .
<b>value2</b>	numeric, boolean, dateTime, RDF term	The number, boolean, datetime, or RDF term type value to compare to <code>value1</code> .

## Returns

Data Type	Description
boolean	True if <code>value1 &lt; value2</code> . False if not.

## METADATAGRAPHURI

This function returns the metadata graph URI for the specified URI.

## Syntax

```
METADATAGRAPHURI(uri)
```

Argument	Data Type	Description
<b>uri</b>	URI	The URI for which you want to return the corresponding metadata graph URI.

## Returns

Data Type	Description
URI	The metadata graph URI.

## NAMESPACE

This function retrieves the namespace for the given URI.

## Syntax

```
NAMESPACE(uri)
```

Argument	Data Type	Description
uri	URI	The URI from which to retrieve the namespace.

## Returns

Data Type	Description
string	The namespace.

## SAMETERM

This function evaluates whether two RDF term type values are the same.

## Syntax

```
SAMETERM(term1, term2)
```

Argument	Data Type	Description
term1	RDF term	The first literal, URI, or blank node value to compare.
term2	RDF term	The literal, URI, or blank node value to compare to <code>term1</code> .

## Returns

Data Type	Description
boolean	<code>True</code> if the terms are the same and <code>false</code> if they are not.

## Hash Functions

This topic describes the hash functions in Anzo.

### Typographical Conventions

This documentation uses the following conventions in function syntax:

- **CAPS**: Although SPARQL is case-insensitive, function names and other keywords are written in uppercase for readability.
- `[ argument ]`: Brackets are used to indicate optional arguments. Arguments without brackets are required.

### Functions

- [MD5](#): Returns the MD5 checksum of a string as a hexadecimal string.
- [SHA1](#): Calculates the SHA-1 digest of a string value.
- [SHA224](#): Calculates the SHA-224 digest of a string value.
- [SHA256](#): Calculates the SHA-256 digest of a string value.
- [SHA384](#): Calculates the SHA-384 digest of a string value.
- [SHA512](#): Calculates the SHA-512 digest of a string value.

### MD5

This function returns the MD5 checksum of a string as a hexadecimal string.

### Syntax

```
MD5 (text)
```

Argument	Data Type	Description
<b>text</b>	string	The string for which to return the MD5 checksum.

### Returns

Data Type	Description
string	The hexadecimal string.

## SHA1

This function calculates the SHA-1 digest of a string.

### Syntax

```
SHA1 (text)
```

Argument	Data Type	Description
text	string	The string for which to calculate the SHA-1 digest.

### Returns

Data Type	Description
string	The SHA-1 digest.

## SHA224

This function calculates the SHA-224 digest of a string.

### Syntax

```
SHA224 (text)
```

Argument	Data Type	Description
text	string	The string for which to calculate the SHA-224 digest.

### Returns

Data Type	Description
string	The SHA-224 digest.

## SHA256

This function calculates the SHA-256 digest of a string.



## Syntax

```
SHA256(text)
```

Argument	Data Type	Description
<b>text</b>	string	The string for which to calculate the SHA-256 digest.

## Returns

Data Type	Description
string	The SHA-256 digest.

## SHA384

This function calculates the SHA-384 digest of a string.

## Syntax

```
SHA384(text)
```

Argument	Data Type	Description
<b>text</b>	string	The string for which to calculate the SHA-384 digest.

## Returns

Data Type	Description
string	The SHA-384 digest.

## SHA512

This function calculates the SHA-512 digest of a string.

## Syntax

```
SHA512(text)
```

Argument	Data Type	Description
<b>text</b>	string	The string for which to calculate the SHA-512 digest.

## Returns

Data Type	Description
string	The SHA-512 digest.

## Window Aggregate and Ranking Functions

Window aggregates operate on a particular partition or window of the result set. Unlike grouped aggregate functions that group the result set and return a single row, window aggregates retain the resulting rows and return a value for each row. For example, using the grouped aggregate SUM function to add the total number of tickets sold in a year returns one value: the total number of tickets sold for the year. By using WINDOW\_SUM, the results could be partitioned by month so that the query returns 12 values: the sum of the number of tickets sold in each month of the year. This topic describes the window aggregate functions in Anzo.

### Typographical Conventions

This documentation uses the following conventions in function syntax:

- **CAPS:** Although SPARQL is case-insensitive, function names and other keywords are written in uppercase for readability.
- `[ argument ]`: Brackets are used to indicate optional arguments. Arguments without brackets are required.

### Functions

- [WINDOW\\_AVG](#): Calculates the average value of each group of values.
- [WINDOW\\_COUNT](#): Counts the number of values in each group of values.
- [WINDOW\\_MAX](#): Calculates the maximum value of each group of values.
- [WINDOW\\_MIN](#): Calculates the minimum value of each group of values.
- [WINDOW\\_NTILE](#): Divides the rows in the partition into the specified number of ranked groups and returns the group that each value belongs to.
- [WINDOW\\_PERCENTILE](#): Divides the rows in the partition into 100 ranked groups and returns the group that each value belongs to.
- [WINDOW\\_PERCENTILE\\_CONT](#): Calculates a percentile based on the continuous distribution of the specified groups of values.
- [WINDOW\\_PERCENTILE\\_DISC](#): Calculates a percentile based on the discrete distribution of the specified groups of values.
- [WINDOW\\_PRODUCT](#): Calculates the product of each group of values.
- [WINDOW\\_QUARTILE](#): Divides the rows in the partition into four ranked groups and returns the group that each value belongs to.
- [WINDOW\\_SUM](#): Calculates the sum of each group of values.

## WINDOW\_AVG

This function calculates the average value of each group of values.

### Syntax

```
WINDOW_AVG(value [, partition_over ] [, order_by ] [, order ]  
            [, start_frame_type ] [, start_frame_value ]  
            [, end_frame_type ] [, end_frame_value ])
```

Argument	Data Type	Description
<b>value</b>	numeric	Required argument that defines the groups of values to operate on.
<b>partition_over</b>	variable	Optional argument that partitions the results into groups of rows. If you do not include <code>partition_over</code> , the partition becomes the entire set identified by <code>value</code> .
<b>order_by</b>	variable	Optional argument that defines the order or sequence of rows within each partition.
<b>order</b>	boolean	Optional argument that controls whether the order is ascending or descending. When <code>true</code> , the order is ascending. When <code>false</code> , the order is descending.
<b>start_frame_type</b>	string	<p>When <code>order_by</code> is specified, the optional <code>start_frame_type</code>, <code>start_frame_value</code>, <code>end_frame_type</code>, and <code>end_frame_value</code> arguments can be included to refine the set of rows to include in the partitions or groups.</p> <p>The <code>start_frame_type</code> argument defines the starting row of the partition and can be one of the following values:</p> <ul style="list-style-type: none"><li>• CURRENT ROW</li><li>• UNBOUNDED PRECEDING</li><li>• PRECEDING <code>start_frame_value</code></li><li>• FOLLOWING <code>start_frame_value</code></li></ul>

Argument	Data Type	Description
<b>start_frame_value</b>	int	Optional argument that specifies the starting row based on the <code>start_frame_type</code> value.
<b>end_frame_type</b>	string	This argument defines the ending row of the partition and can be one of the following values: <ul style="list-style-type: none"> <li>• CURRENT ROW</li> <li>• UNBOUNDED FOLLOWING</li> <li>• PRECEDING <code>end_frame_value</code></li> <li>• FOLLOWING <code>end_frame_value</code></li> </ul>
<b>end_frame_value</b>	int	Optional argument that specifies the ending row based on the <code>end_frame_type</code> value.

## Returns

Data Type	Description
int	The average values.

## WINDOW\_COUNT

This function counts the number of values in each group of values.

## Syntax

```
WINDOW_COUNT(value [, partition_over ] [, order_by ] [, order ]
              [, start_frame_type ] [, start_frame_value ]
              [, end_frame_type ] [, end_frame_value ])
```

Argument	Data Type	Description
<b>value</b>	numeric	Required argument that defines the groups of values to operate on.
<b>partition_over</b>	variable	Optional argument that partitions the results into groups of rows. If you do not include <code>partition_over</code> , the partition becomes the entire set

Argument	Data Type	Description
		identified by <code>value</code> .
<b>order_by</b>	variable	Optional argument that defines the order or sequence of rows within each partition.
<b>order</b>	boolean	Optional argument that controls whether the order is ascending or descending. When <code>true</code> , the order is ascending. When <code>false</code> , the order is descending.
<b>start_frame_type</b>	string	<p>When <code>order_by</code> is specified, the optional <code>start_frame_type</code>, <code>start_frame_value</code>, <code>end_frame_type</code>, and <code>end_frame_value</code> arguments can be included to refine the set of rows to include in the partitions or groups.</p> <p>The <code>start_frame_type</code> argument defines the starting row of the partition and can be one of the following values:</p> <ul style="list-style-type: none"> <li>• <code>CURRENT ROW</code></li> <li>• <code>UNBOUNDED PRECEDING</code></li> <li>• <code>PRECEDING start_frame_value</code></li> <li>• <code>FOLLOWING start_frame_value</code></li> </ul>
<b>start_frame_value</b>	int	Optional argument that specifies the starting row based on the <code>start_frame_type</code> value.
<b>end_frame_type</b>	string	<p>This argument defines the ending row of the partition and can be one of the following values:</p> <ul style="list-style-type: none"> <li>• <code>CURRENT ROW</code></li> <li>• <code>UNBOUNDED FOLLOWING</code></li> <li>• <code>PRECEDING end_frame_value</code></li> <li>• <code>FOLLOWING end_frame_value</code></li> </ul>
<b>end_frame_value</b>	int	Optional argument that specifies the ending row based on the <code>end_</code>

Argument	Data Type	Description
<b>value</b>		<code>frame_type</code> value.

## Returns

Data Type	Description
int	The counts of values.

## WINDOW\_MAX

This function calculates the maximum value of each group of values.

## Syntax

```
WINDOW_MAX(value [, partition_over ] [, order_by ] [, order ]
            [, start_frame ] [, start_frame_type ] [, start_frame_value ]
            [, end_frame_type ] [, end_frame_value ])
```

Argument	Data Type	Description
<b>value</b>	numeric	Required argument that defines the groups of values to operate on.
<b>partition_</b> <b>over</b>	variable	Optional argument that partitions the results into groups of rows. If you do not include <code>partition_over</code> , the partition becomes the entire set identified by <code>value</code> .
<b>order_by</b>	variable	Optional argument that defines the order or sequence of rows within each partition.
<b>order</b>	boolean	Optional argument that controls whether the order is ascending or descending. When <code>true</code> , the order is ascending. When <code>false</code> , the order is descending.
<b>start_frame_</b> <b>type</b>	string	When <code>order_by</code> is specified, the optional <code>start_frame_type</code> , <code>start_frame_value</code> , <code>end_frame_type</code> , and <code>end_frame_value</code> arguments can be included to refine the set of rows to include in the partitions or

Argument	Data Type	Description
		<p>groups.</p> <p>The <code>start_frame_type</code> argument defines the starting row of the partition and can be one of the following values:</p> <ul style="list-style-type: none"> <li>• CURRENT ROW</li> <li>• UNBOUNDED PRECEDING</li> <li>• PRECEDING <code>start_frame_value</code></li> <li>• FOLLOWING <code>start_frame_value</code></li> </ul>
<b>start_frame_value</b>	int	Optional argument that specifies the starting row based on the <code>start_frame_type</code> value.
<b>end_frame_type</b>	string	<p>This argument defines the ending row of the partition and can be one of the following values:</p> <ul style="list-style-type: none"> <li>• CURRENT ROW</li> <li>• UNBOUNDED FOLLOWING</li> <li>• PRECEDING <code>end_frame_value</code></li> <li>• FOLLOWING <code>end_frame_value</code></li> </ul>
<b>end_frame_value</b>	int	Optional argument that specifies the ending row based on the <code>end_frame_type</code> value.

## Returns

Data Type	Description
int	The maximum values.

## WINDOW\_MIN

This function calculates the minimum value of each group of values.



## Syntax

```
WINDOW_MIN(value [, partition_over ] [, order_by ] [, order ]  
            [, start_frame_type ] [, start_frame_value ]  
            [, end_frame_type ] [, end_frame_value ])
```

Argument	Data Type	Description
<b>value</b>	numeric	Required argument that defines the groups of values to operate on.
<b>partition_</b> <b>over</b>	variable	Optional argument that partitions the results into groups of rows. If you do not include <code>partition_over</code> , the partition becomes the entire set identified by <code>value</code> .
<b>order_by</b>	variable	Optional argument that defines the order or sequence of rows within each partition.
<b>order</b>	boolean	Optional argument that controls whether the order is ascending or descending. When <code>true</code> , the order is ascending. When <code>false</code> , the order is descending.
<b>start_frame_</b> <b>type</b>	string	<p>When <code>order_by</code> is specified, the optional <code>start_frame_type</code>, <code>start_frame_value</code>, <code>end_frame_type</code>, and <code>end_frame_value</code> arguments can be included to refine the set of rows to include in the partitions or groups.</p> <p>The <code>start_frame_type</code> argument defines the starting row of the partition and can be one of the following values:</p> <ul style="list-style-type: none"><li>• CURRENT ROW</li><li>• UNBOUNDED PRECEDING</li><li>• PRECEDING <code>start_frame_value</code></li><li>• FOLLOWING <code>start_frame_value</code></li></ul>
<b>start_frame_</b> <b>value</b>	int	Optional argument that specifies the starting row based on the <code>start_frame_type</code> value.

Argument	Data Type	Description
<b>end_frame_type</b>	string	This argument defines the ending row of the partition and can be one of the following values: <ul style="list-style-type: none"> <li>• CURRENT ROW</li> <li>• UNBOUNDED FOLLOWING</li> <li>• PRECEDING end_frame_value</li> <li>• FOLLOWING end_frame_value</li> </ul>
<b>end_frame_value</b>	int	Optional argument that specifies the ending row based on the end_frame_type value.

## Returns

Data Type	Description
int	The minimum values.

## WINDOW\_NTILE

This function divides the rows in the partition into the specified number of ranked groups and returns the group that each value belongs to.

## Syntax

```
WINDOW_NTILE(ntile, value, order_by [, partition_over ])
```

Argument	Data Type	Description
<b>ntile</b>	int	Required argument that specifies the number of ranking groups.
<b>value</b>	numeric	Required argument that defines the groups of values to operate on.
<b>order_by</b>	variable	Required argument that defines the order or sequence of rows within each partition.
<b>partition_</b>	variable	Optional argument that partitions the results into groups of rows. If you do

Argument	Data Type	Description
<b>over</b>		not include <code>partition_over</code> , the partition becomes the entire set identified by <code>value</code> .

## Returns

Data Type	Description
int	The group that the values belong to.

## WINDOW\_PERCENTILE

This function divides the rows in the partition into 100 ranked groups and returns the group that each value belongs to.

## Syntax

```
WINDOW_PERCENTILE(value, order_by [, partition_over ])
```

Argument	Data Type	Description
<b>value</b>	numeric	Required argument that defines the groups of values to operate on.
<b>order_by</b>	variable	Required argument that defines the order or sequence of rows within each partition.
<b>partition_over</b>	variable	Optional argument that partitions the results into groups of rows. If you do not include <code>partition_over</code> , the partition becomes the entire set identified by <code>value</code> .

## Returns

Data Type	Description
int	The group that the values belong to.

## WINDOW\_PERCENTILE\_CONT

This function calculates a percentile based on the continuous distribution of the specified groups of values. The returned value is interpolated and may not be equal to any of the values in the group.

### Syntax

```
WINDOW_PERCENTILE_CONT(percentile, value, order_by [, partition_over ])
```

Argument	Data Type	Description
<b>percentile</b>	int	Required argument that specifies the percentile for the calculation.
<b>value</b>	numeric	Required argument that defines the groups of values to operate on.
<b>order_by</b>	variable	Required argument that defines the order or sequence of rows within each partition.
<b>partition_over</b>	variable	Optional argument that partitions the results into groups of rows. If you do not include <code>partition_over</code> , the partition becomes the entire set identified by <code>value</code> .

### Returns

Data Type	Description
int	The interpolated percentiles.

## WINDOW\_PERCENTILE\_DISC

This function calculates a percentile based on the discrete distribution of the specified groups of values.

### Syntax

```
WINDOW_PERCENTILE_DISC(percentile, value, order_by [, partition_over ])
```

Argument	Data Type	Description
<b>percentile</b>	int	Required argument that specifies the percentile for the calculation.

Argument	Data Type	Description
<b>value</b>	numeric	Required argument that defines the groups of values to operate on.
<b>order_by</b>	variable	Required argument that defines the order or sequence of rows within each partition.
<b>partition_over</b>	variable	Optional argument that partitions the results into groups of rows. If you do not include <code>partition_over</code> , the partition becomes the entire set identified by <code>value</code> .

## Returns

Data Type	Description
int	The percentiles based on the discrete distribution of the groups.

## WINDOW\_PRODUCT

This function calculates the product of each group of values.

## Syntax

```
WINDOW_PRODUCT(value [, partition_over ] [, order_by ] [, order ]
                [, start_frame_type ] [, start_frame_value ]
                [, end_frame_type ] [, end_frame_value ])
```

Argument	Data Type	Description
<b>value</b>	numeric	Required argument that defines the groups of values to operate on.
<b>partition_over</b>	variable	Optional argument that partitions the results into groups of rows. If you do not include <code>partition_over</code> , the partition becomes the entire set identified by <code>value</code> .
<b>order_by</b>	variable	Optional argument that defines the order or sequence of rows within each partition.

Argument	Data Type	Description
<b>order</b>	boolean	Optional argument that controls whether the order is ascending or descending. When <code>true</code> , the order is ascending. When <code>false</code> , the order is descending.
<b>start_frame_type</b>	string	<p>When <code>order_by</code> is specified, the optional <code>start_frame_type</code>, <code>start_frame_value</code>, <code>end_frame_type</code>, and <code>end_frame_value</code> arguments can be included to refine the set of rows to include in the partitions or groups.</p> <p>The <code>start_frame_type</code> argument defines the starting row of the partition and can be one of the following values:</p> <ul style="list-style-type: none"> <li>• <code>CURRENT ROW</code></li> <li>• <code>UNBOUNDED PRECEDING</code></li> <li>• <code>PRECEDING start_frame_value</code></li> <li>• <code>FOLLOWING start_frame_value</code></li> </ul>
<b>start_frame_value</b>	int	Optional argument that specifies the starting row based on the <code>start_frame_type</code> value.
<b>end_frame_type</b>	string	<p>This argument defines the ending row of the partition and can be one of the following values:</p> <ul style="list-style-type: none"> <li>• <code>CURRENT ROW</code></li> <li>• <code>UNBOUNDED FOLLOWING</code></li> <li>• <code>PRECEDING end_frame_value</code></li> <li>• <code>FOLLOWING end_frame_value</code></li> </ul>
<b>end_frame_value</b>	int	Optional argument that specifies the ending row based on the <code>end_frame_type</code> value.

## Returns

Data Type	Description
int	The products of the groups.

## WINDOW\_QUARTILE

This function divides the rows in the partition into four ranked groups and returns the group that each value belongs to.

## Syntax

```
WINDOW_QUARTILE(value, order_by [, partition_over ])
```

Argument	Data Type	Description
<b>value</b>	numeric	Required argument that defines the groups of values to operate on.
<b>order_by</b>	variable	Required argument that defines the order or sequence of rows within each partition.
<b>partition_over</b>	variable	Optional argument that partitions the results into groups of rows. If you do not include <code>partition_over</code> , the partition becomes the entire set identified by <code>value</code> .

## Returns

Data Type	Description
int	The group that the values belong to.

## WINDOW\_SUM

This function calculates the sum of each group of values.

## Syntax

```
WINDOW_SUM(value [, partition_over ] [, order_by ] [, order ]  
            [, start_frame_type ] [, start_frame_value ])
```

```
[, end_frame_type ] [, end_frame_value ])
```

Argument	Data Type	Description
<b>value</b>	numeric	Required argument that defines the groups of values to operate on.
<b>partition_ over</b>	variable	Optional argument that partitions the results into groups of rows. If you do not include <code>partition_over</code> , the partition becomes the entire set identified by <code>value</code> .
<b>order_by</b>	variable	Optional argument that defines the order or sequence of rows within each partition.
<b>order</b>	boolean	Optional argument that controls whether the order is ascending or descending. When <code>true</code> , the order is ascending. When <code>false</code> , the order is descending.
<b>start_frame_ type</b>	string	<p>When <code>order_by</code> is specified, the optional <code>start_frame_type</code>, <code>start_frame_value</code>, <code>end_frame_type</code>, and <code>end_frame_value</code> arguments can be included to refine the set of rows to include in the partitions or groups.</p> <p>The <code>start_frame_type</code> argument defines the starting row of the partition and can be one of the following values:</p> <ul style="list-style-type: none"> <li>• CURRENT ROW</li> <li>• UNBOUNDED PRECEDING</li> <li>• PRECEDING <code>start_frame_value</code></li> <li>• FOLLOWING <code>start_frame_value</code></li> </ul>
<b>start_frame_ value</b>	int	Optional argument that specifies the starting row based on the <code>start_frame_type</code> value.
<b>end_frame_ type</b>	string	This argument defines the ending row of the partition and can be one of the following values:



Argument	Data Type	Description
		<ul style="list-style-type: none"> <li>• CURRENT ROW</li> <li>• UNBOUNDED FOLLOWING</li> <li>• PRECEDING end_frame_value</li> <li>• FOLLOWING end_frame_value</li> </ul>
end_frame_value	int	Optional argument that specifies the ending row based on the end_frame_type value.

## Returns

Data Type	Description
int	The sum of each group.

## Example

The example below first creates data by running the following INSERT DATA query in a graphmart Query Step:

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ex: <http://example.com/csi#>
INSERT DATA {
  GRAPH ${targetGraph} {
    ex:sale1 ex:date "2014-05-23T10:20:13"^^xsd:dateTime ; ex:volume 15 .
    ex:sale2 ex:date "2014-06-23T10:20:13"^^xsd:dateTime ; ex:volume 3 .
    ex:sale3 ex:date "2014-06-23T10:22:13"^^xsd:dateTime ; ex:volume 35 .
    ex:sale4 ex:date "2014-07-23T10:20:13"^^xsd:dateTime ; ex:volume 66 .
    ex:sale5 ex:date "2014-09-23T10:20:13"^^xsd:dateTime ; ex:volume 19 .
    ex:sale6 ex:date "2014-11-23T10:20:13"^^xsd:dateTime ; ex:volume 33 .
    ex:sale7 ex:date "2014-12-23T10:20:13"^^xsd:dateTime ; ex:volume 12 .
  }
}
```

The following query against the new data uses the WINDOW\_SUM function to return the total volume of sales for each month:

```

PREFIX ex: <http://example.com/csi#>
SELECT DISTINCT ?Month ?Total_Volume
WHERE {
  {
    SELECT
      ?Month (WINDOW_SUM(?o, ?Month) AS ?Total_Volume)
    WHERE {
      ?s ex:volume ?o .
      ?s ex:date ?date .
      BIND (MONTH(?date) AS ?Month)
    }
  }
}

```

The query returns the following results:

Month	Total_Volume
-----+	-----
5	15
6	38
7	66
9	19
11	33
12	12

# FAQ

This topic provides answers to frequently asked questions and includes references to more detailed information.

- [What is an Anzo Data Store?](#)
- [What is the difference between a Dataset Pipeline and an ETL Pipeline?](#)
- [How do I update Anzo if a file in my CSV data source changes?](#)
- [How do I duplicate a mapping?](#)
- [How do I associate a model with an existing dataset?](#)
- [How do I download a model?](#)
- [How do I see which models are included in a graphmart?](#)
- [How do I find the URI for a graphmart?](#)
- [How do I find the URI for a data layer in a graphmart?](#)
- [How do I find the URI for a dataset?](#)
- [How do I find the catalog entry URI for a dataset?](#)
- [How do I clear the data components from the Managed Edition of a dataset?](#)
- [What is the difference between the graphmart reload and refresh options?](#)
- [What happens to the existing data in an FLDS when I run an incremental ETL job?](#)

## What is an Anzo Data Store?

An Anzo Data Store, also known as a graph data source, is a designated directory on the File Store where Anzo can save the RDF files that are generated during the ETL process. All installations require at least one Anzo Data Store. You can create one data store and configure all pipelines to write to that store (each ETL run automatically creates a new sub-directory under the data store) or you can create multiple data stores to use for different Datasets. For information about creating data stores, see [Creating an Anzo Data Store](#) in the Administration Guide.

---

## What is the difference between a Dataset Pipeline and an ETL Pipeline?

Dataset Pipelines are used to ingest data into Anzo. They produce new Datasets in the Dataset catalog and generate RDF files for loading data to AnzoGraph. All auto-ingested pipelines are Dataset Pipelines. For more information, see [Creating a Dataset Pipeline](#).

---

ETL pipelines do not ingest data into Anzo; they are used to ingest data into a file or another database. ETL pipelines do not generate a new Dataset entry in the catalog or produce RDF load files for AnzoGraph. When you create mappings for ETL pipelines, you define a file schema or database as the target. For more information, see [Creating an ETL Pipeline](#).

## How do I update Anzo if a file in my CSV data source changes?

If the data in a CSV file changes, the way that you update the dataset in Anzo depends on the type of changes that were made to the file and the file system where the file is hosted. The table below provides guidance on the steps to take to update Anzo based on the type of content updates and the file location.

### Note

The instructions below assume that the updated file has the same name and location as the file that was originally uploaded.


Update Type	File Location	Update Process
Added, deleted, or changed <b>rows</b> – Columns did not change	Uploaded from your computer and copied to the base upload path as described in <a href="#">Setting the Default Base File Store Path for File Uploads</a> in the Administration Guide.	<ol style="list-style-type: none"><li>1. Replace the file on the file store with the updated version of the file.</li><li>2. Re-publish the job for this file or the entire pipeline to update the existing data set.</li><li>3. Reload any graphmarts that include the updated data set and then refresh the affected Hi-Res Analytics dashboards to view the updated data.</li></ol>
	Selected from the File Store.	<ol style="list-style-type: none"><li>1. Replace the file on the file store with the updated version of the file.</li><li>2. Re-publish the job for this file or the entire pipeline to update the existing data set.</li><li>3. Reload any graphmarts that include the</li></ol>


Update Type	File Location	Update Process
		updated data set and then refresh the affected Hi-Res Analytics dashboards to view the updated data.
Added, deleted, or changed <b>columns</b> and rows	Uploaded from your computer and copied to the base upload path as described in <a href="#">Setting the Default Base File Store Path for File Uploads</a> in the Administration Guide.	<ol style="list-style-type: none"> <li>1. Replace the file on the file store with the updated version of the file.</li> <li>2. In the Anzo application, view the CSV data source that contains the file to update. On the Tables tab, select the checkbox next to the file to re-import. Then click the <b>Import Selected</b> button to import the updated file.</li> <li>3. Click <b>Ingest</b> and re-ingest the data source.</li> <li>4. Publish the pipeline to update the existing data set.</li> <li>5. Reload any graphmarts that include the updated data set and then refresh the affected Hi-Res Analytics dashboards to view the updated data.</li> </ol>
	Selected from the File Store.	<ol style="list-style-type: none"> <li>1. Replace the file on the file store with the updated version of the file.</li> <li>2. In the Anzo application, view the CSV data source that contains the file to update. On the Tables tab, select the checkbox next to the file to re-import. Then click the <b>Import Selected</b> button to import the updated file.</li> <li>3. Click <b>Ingest</b> and re-ingest the data source.</li> </ol>

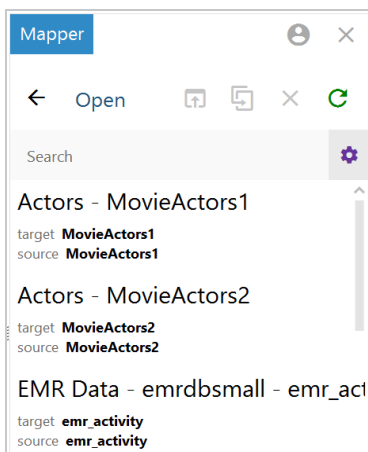
Update Type	File Location	Update Process
		<ol style="list-style-type: none"> <li>4. Publish the pipeline to update the existing data set.</li> <li>5. Reload any graphmarts that include the updated data set and then refresh the affected Hi-Res Analytics dashboards to view the updated data.</li> </ol>

## How do I duplicate a mapping?

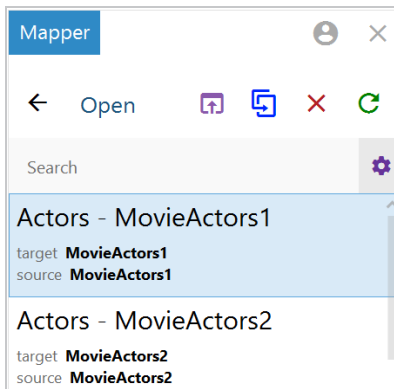
Using the Anzo for Office Excel plugin, users can duplicate mappings to use as a template for a new mapping. Follow the instructions below to duplicate a mapping.

1. In Excel, open the Anzo Mapper tool and connect to the Anzo server.
2. In the Mapper menu, click the folder icon (  ) to list the mappings that are available to open.

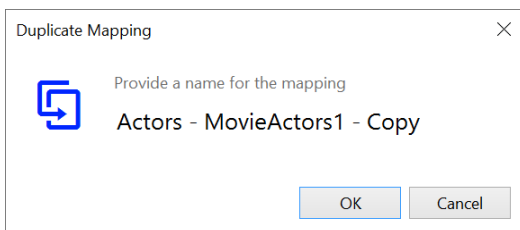
By default, the mapping tool lists only the mappings that you created. To display additional mappings, such as auto-generated files, type a term in the **Search** field, and then click the cog icon (  ) to display the files. For example:



3. Click the mapping that you want to duplicate. Selecting a mapping activates the buttons at the top of the screen. For example:



4. Click the **Duplicate** icon (📄) to copy the selected mapping. Anzo displays the Duplicate Mapping dialog box. For example:



5. Edit the mapping name and then click **OK** to create the duplicate. The new mapping is added to the list of mappings that are available to open.

---

## How do I associate a model with an existing dataset?

Follow the instructions below to associate a model that is in Anzo with an onboarded data set.

1. In the Anzo application, expand the **Blend** menu and click **Datasets**. Anzo displays the Datasets catalog, which lists the existing data sets.
2. Click the data set that you want to add a model to. Anzo displays the Explore screen for the data set.
3. Click the **Overview** tab. Under the Description field, click **Advanced** to display the advanced options. For example:

Overview

Explore

Description

None

Advanced ▲

Data Location

/nfs/data/store/LoadParquet\_c4ae7/

Models

None

- Click the edit icon (✎) for the **Models** field to open the Models drop-down list, and then select the model that you want to use for this data set. To include a system model, select the **Include System Data** checkbox. To select multiple models, click the drop-down list again and select another model.
- When you have finished selecting models, click the checkmark icon (✓) to save the change and associate the model or models with the data set.

## How do I download a model?

Follow the instructions below to download a data model to your computer.

- In the Anzo application, click **Model**. Anzo displays the Manage Data Model Working Set screen. For example:

Manage Data Model Working Set

▼

Search

≡

≡

✓

Add Model ▼

<input type="checkbox"/>	Title	Class #	Description	Updated Date ↑	Actions
<input checked="" type="checkbox"/>	Movies Layer Model	4	The managed model for 'Movies Layer' in the 'Movies' graphmart.	Nov 1, 2022	
<input checked="" type="checkbox"/>	Flights Layer Model	1	The managed model for 'Flights Layer' in the 'Flights' graphmart.	Nov 1, 2022	
<input checked="" type="checkbox"/>	DB-QA Layer Model	3	The managed model for 'DB-QA Layer' in the 'Books' graphmart.	Nov 2, 2022	
<input checked="" type="checkbox"/>	DB-QA Layer Model	27	The managed model for 'DB-QA Layer' in the 'Northwind' graphmart.	Nov 2, 2022	
<input checked="" type="checkbox"/>	Tickets Layer Model	6	The managed model for 'Tickets Layer' in the 'Tickets' graphmart.	Nov 2, 2022	
<input checked="" type="checkbox"/>	Data Toolkit	32	Defines the interface to the Data Toolkit service.		
<input checked="" type="checkbox"/>	SKOS Vocabulary	4	An RDF vocabulary for describing the basic structure and content of concept schem...		

Rows per page: 25 ▼

1-7 of 7

<

>

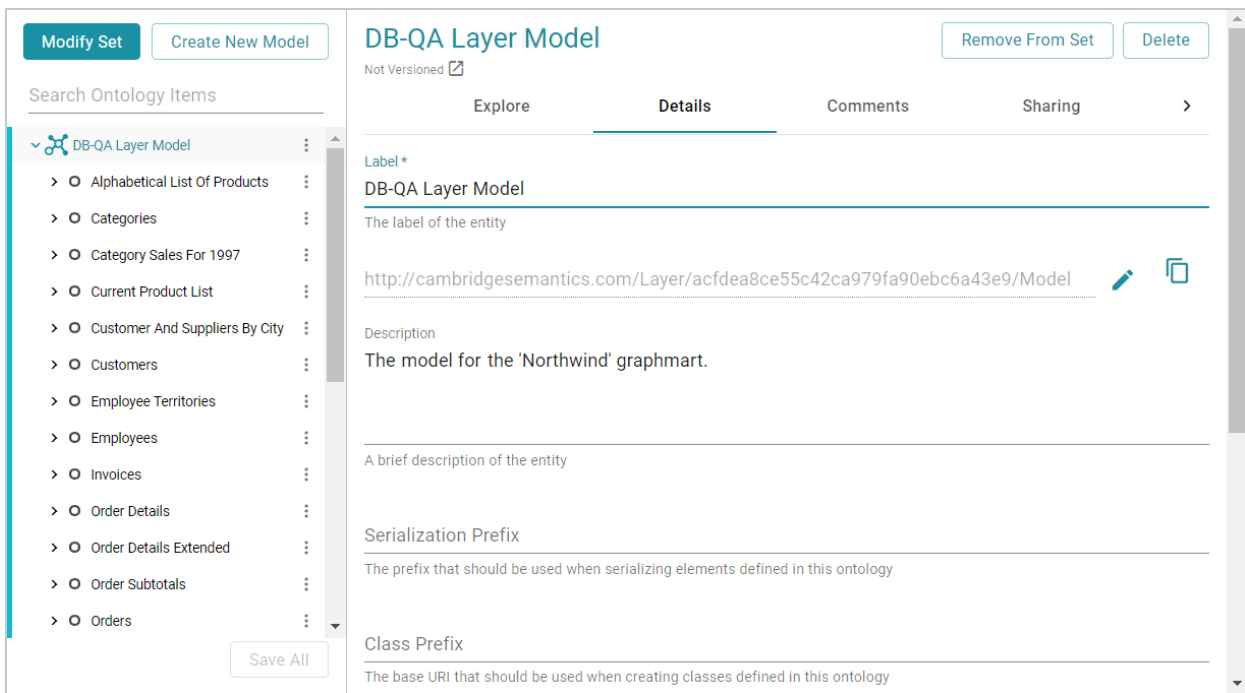
UPLOAD MODELS

CANCEL
 

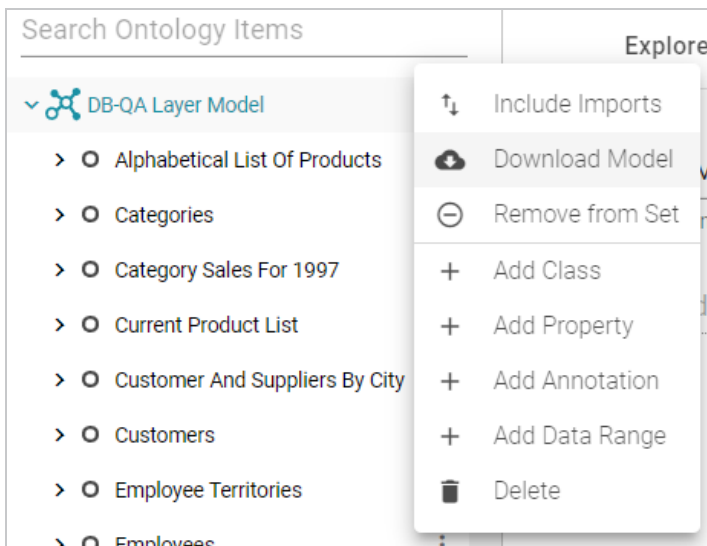
OK

- On the Manage Working Set screen, select the checkbox next to the model that you want to download, and then click **OK**. Anzo opens the selected model in the editor. For example:





3. Open the model menu by clicking the menu icon (⋮) to the right of the model name. Then select **Download Model**.



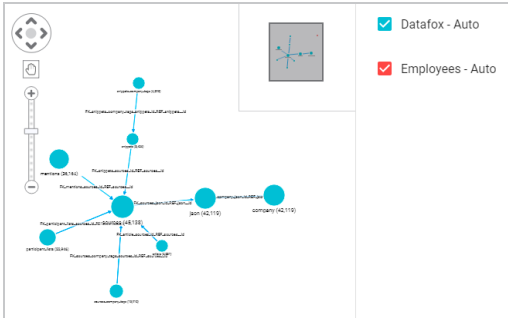
Anzo displays the Download Model dialog box:



In the bottom right corner of the graph view in the center of the screen, there are three icons:



3. To view the associated Models, click the contents icon (≡) on the right. For example, the image below shows a Graphmart with two Models:



---

## How do I find the URI for a graphmart?

Anzo displays graphmart details on the Overview screen for the graphmart. Follow the steps below to view and copy a Graphmart URI.

1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays the Graphmarts screen.
2. In the list of Graphmarts, click the name of the Graphmart for which you want to view or copy the URI. Anzo displays the Graphmart Overview. For example:

3. View the Graphmart URI in the statistics section on the right side of the screen. For example:

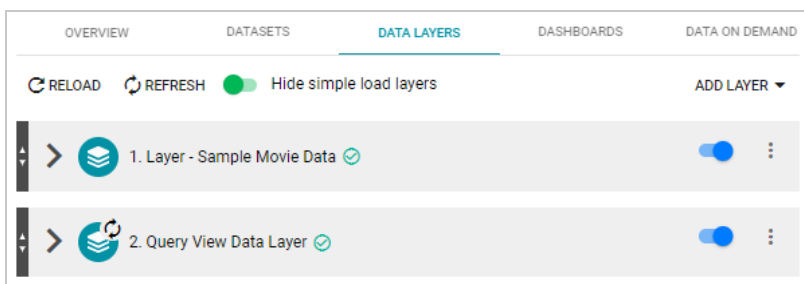
You can click the clipboard icon (📋) to copy the URI to your clipboard.

You can also copy a URL-encoded version of the Graphmart URI from the address bar in the browser when viewing the Graphmart Overview. For example:

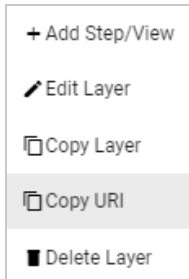
## How do I find the URI for a data layer in a graphmart?

You can retrieve a graph or Data Layer URI on the Data Layers screen for a Graphmart. Follow the steps below to copy a graph URI.

1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays the Graphmarts screen.
2. In the list of Graphmarts, click the name of the Graphmart that contains the Data Layer whose URI you want to copy. Anzo displays the Graphmart Overview.
3. Click the **Data Layers** tab. Anzo displays the Data Layers in the Graphmart. Each layer is a graph. For example:



4. To copy the URI for a layer to your clipboard, click the menu icon (⋮) for the layer and click **Copy URI**.

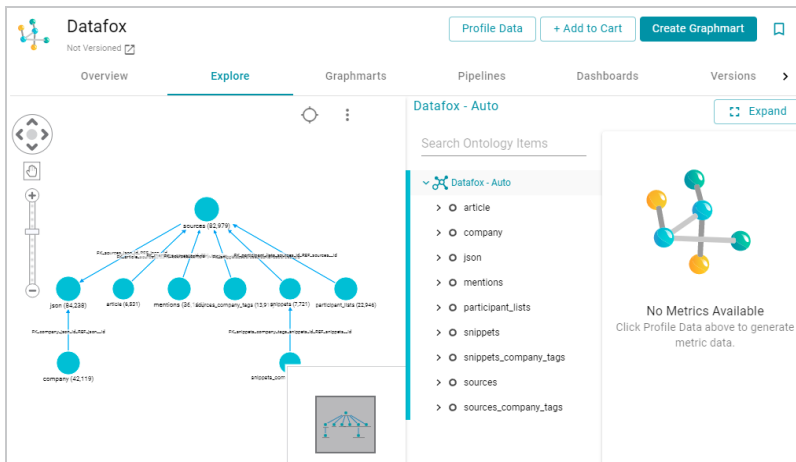


---

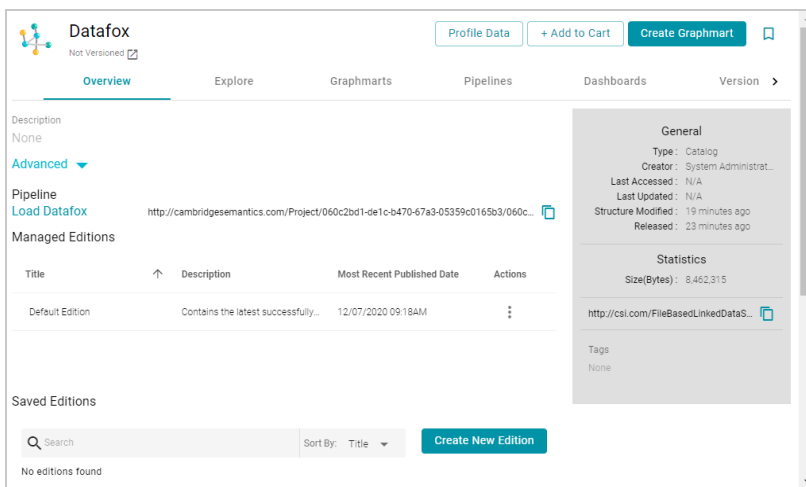
## How do I find the URI for a dataset?

Follow the steps below to view and copy the URI for a Dataset in the catalog.

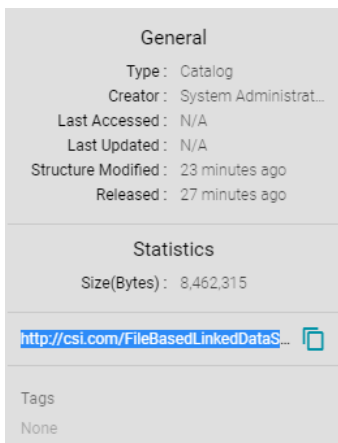
1. In the Anzo application, expand the **Blend** menu and click **Datasets**. Anzo displays the Dataset catalog, which lists the onboarded Datasets.
  2. Click the name of the Dataset for which you want to copy the URI. Anzo displays the Explore tab. For example:
-



3. Click the **Overview** tab to view the general information for the Dataset. For example:



4. In the statistics section on the right side of the screen, click the clipboard icon (📋) to copy the URI to your clipboard. For example, the image below shows the URI highlighted:



## How do I find the catalog entry URI for a dataset?

To query from a remote client (such as over the SPARQL endpoint) a linked data set (LDS) that is stored in a local volume, you need to specify the catalog entry URI for that LDS as the target data set. The catalog entry URI uniquely identifies an LDS because it encodes both the LDS and its data source (local volume) in the URI. Follow the steps below to find the catalog entry for an LDS.

1. First, retrieve the URI for the LDS whose catalog entry URI you want to find. For instructions, see [How do I find the URI for a dataset?](#) above.
2. Next, open the Find tab in the Query Builder. In the Anzo application, expand the **Access** menu and click **Query Builder**. Then click the **Find** tab. The Find screen opens and the **System Datasource** is selected as the target data source.

The screenshot shows the 'Find' tab in the Anzo Query Builder. At the top, there are two tabs: 'Query' and 'Find', with 'Find' being the active tab. Below the tabs, there is a section labeled 'Select a datasource:' with a dropdown menu showing 'System Datasource'. Below this, there are four input fields labeled 'Subject', 'Predicate', 'Object', and 'Graph'. At the bottom right, there are three buttons: 'Clear', 'Add Statements', and 'Find'.

3. If the LDS is in a different volume, click the **Source** drop-down list and select the appropriate volume. Typically, linked data sets are stored in the system volume.
4. Paste in the **Object** field the LDS URI that you copied in the first step. Then click **Find**. Anzo returns the set of quads for which the LDS URI is the object. For example:

The screenshot shows the results of a query in the Anzo Query Builder. The table has four columns: 'Subject', 'Predicate', 'Object', and 'Named Graph'. The 'Object' column contains the LDS URI that was entered in the previous step. The table shows 14 results.

Subject	Predicate	Object	Named Graph
<http://openanzo.org/datasets#NamedGraphs>	<http://openanzo.org/ontologies/2008/07/Anzo#namedGraph>	<http://csi.com/FileBasedLinkedDataSet/001e517db4f0eae9f279427e4e2a828>	
<http://openanzo.org/catEntry(%5Bhttp%3A%2F%2Fcsi.com%2FFileBasedLinkedDataSet%2F001e517db4f0eae9f279427e4e2a828%5D%40%5Bhttp%3A%2F%2Fopenanzo.org%2Fdatasource%2FsystemDatasource%5D)>	<http://cambridgesemantics.com/ontologies/2009/05/LinkedData#linkedDataset>	<http://csi.com/FileBasedLinkedDataSet/001e517db4f0eae9f279427e4e2a828>	
<http://openanzo.org/catEntry(%5Bhttp%3A%2F%2Fcsi.com%2FFileBasedLinkedDataSet%2F001e517db4f0eae9f279427e4e2a828%5D%40%5Bhttp%3A%2F%2Fopenanzo.org%2Fdatasource%2FsystemDatasource%5D)>	<http://cambridgesemantics.com/ontologies/2009/05/LinkedData#dataset>	<http://csi.com/FileBasedLinkedDataSet/001e517db4f0eae9f279427e4e2a828>	
<http://csi.com/FileBasedLinkedDataSet/001e517db4f0eae9f279427e4e2a828>	<http://cambridgesemantics.com/ontologies/2009/05/LinkedData#dataset>	<http://csi.com/FileBasedLinkedDataSet/001e517db4f0eae9f279427e4e2a828>	
<http://cambridgesemantics.com/registries/LinkedDataSets>	<http://openanzo.org/ontologies/2008/07/Anzo#defaultNamedGraph>	<http://csi.com/FileBasedLinkedDataSet/001e517db4f0eae9f279427e4e2a828>	

5. In the **Subject** field in the results, look for a URI that begins with **http://openanzo.org/catEntry**. The value is the catalog entry URI for the LDS. For example:

```
« <http://openanzo.org/catEntry(%5Bhttp%3A%2F%2Fcsi.com%2FFileBasedLinkedDataSet%2F001e517db4f0eaea9f279427e4e2a828%5D%40%5Bhttp%3A%2F%2Fopenanzo.org%2Fdatasource%2FsystemDatasource%5D)> »
```

6. Copy the entire URI. This is the URI to use as the target data source for SPARQL endpoint queries against the LDS. For more information about the SPARQL endpoint, see [Accessing Data from the SPARQL Endpoint](#).

---

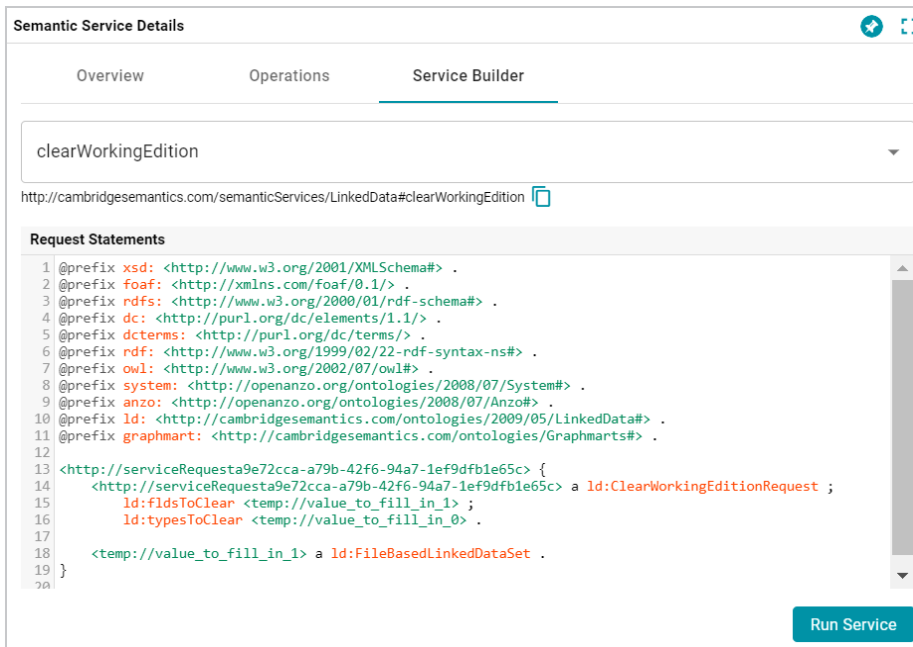
## How do I clear the data components from the Managed Edition of a dataset?

Follow the instructions below if you want to clear out all of the existing components from the Managed Edition so that the Edition is recreated from scratch the next time the pipeline is published.

**Note** Permission to **Manage Semantic Services** is required to complete this task.

1. First, copy the URI of the Dataset for which you want to clear the Managed Edition. [How do I find the URI for a dataset?](#)
2. Next, In the Administration application, expand the **Monitoring & Diagnostics** menu and select **Semantic Services**.
3. Search for the **LinkedDataService** and view its details. Then click the **Service Builder** tab in Semantic Service Details.
4. Click the **Please Select an Operation** field and select **clearWorkingEdition** from the drop-down list. The Request Statements for the service call are populated:





5. Toward the bottom of the request, replace the **<temp://value\_to\_fill\_in\_1>** placeholder URI with the URI for the Dataset.

```

<http://serviceRequesta9e72cca-a79b-42f6-94a7-1ef9dfb1e65c> {
  <http://serviceRequesta9e72cca-a79b-42f6-94a7-1ef9dfb1e65c> a
ld:ClearWorkingEditionRequest ;
  ld:fldsToClear <temp://value_to_fill_in_1> ;
  ld:typesToClear <temp://value_to_fill_in_0> .

  <temp://value_to_fill_in_1> a ld:FileBasedLinkedDataSet .
}

```

For example:

```

<http://serviceRequesta9e72cca-a79b-42f6-94a7-1ef9dfb1e65c> {
  <http://serviceRequesta9e72cca-a79b-42f6-94a7-1ef9dfb1e65c> a
ld:ClearWorkingEditionRequest ;
  ld:fldsToClear
<http://csi.com/FileBasedLinkedDataSet/ee8d3d5792fd218a03b70fdf850
b6a4c> ;
  ld:typesToClear <temp://value_to_fill_in_0> .
}

```

```
<http://csi.com/FileBasedLinkedDataSet/ee8d3d5792fd218a03b70fdf850
b6a4c> a ld:FileBasedLinkedDataSet .
}
```

6. Comment out the `ld:typesToClear <temp://value_to_fill_in_0>` line. For example:

```
<http://serviceRequesta9e72cca-a79b-42f6-94a7-1ef9dfb1e65c> {
  <http://serviceRequesta9e72cca-a79b-42f6-94a7-1ef9dfb1e65c> a
ld:ClearWorkingEditionRequest ;
  ld:fldsToClear
<http://csi.com/FileBasedLinkedDataSet/ee8d3d5792fd218a03b70fdf850
b6a4c> ;
  # ld:typesToClear <temp://value_to_fill_in_0> .

<http://csi.com/FileBasedLinkedDataSet/ee8d3d5792fd218a03b70fdf850
b6a4c> a ld:FileBasedLinkedDataSet .
}
```

7. Click the **Run Service** button to clear the Edition. Anzo returns a response such as the following example when the request is processed:

```
@prefix n-1060687345:
<http://openanzo.org/ClearWorkingEditionResponse/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ld:
<http://cambridgesemantics.com/ontologies/2009/05/LinkedData#> .
@prefix ss:
<http://openanzo.org/ontologies/2008/07/SemanticService#> .

n-1060687345:effbda5b-ce9c-4190-abb1-6f6efd07f5d8 {
  n-1060687345:effbda5b-ce9c-4190-abb1-6f6efd07f5d8
ld:wasWorkingEditionCleared
  "true"^^<http://www.w3.org/2001/XMLSchema#boolean> .
  n-1060687345:effbda5b-ce9c-4190-abb1-6f6efd07f5d8 rdf:type
ld:ClearWorkingEditionResponse .
  n-1060687345:effbda5b-ce9c-4190-abb1-6f6efd07f5d8 rdf:type
```

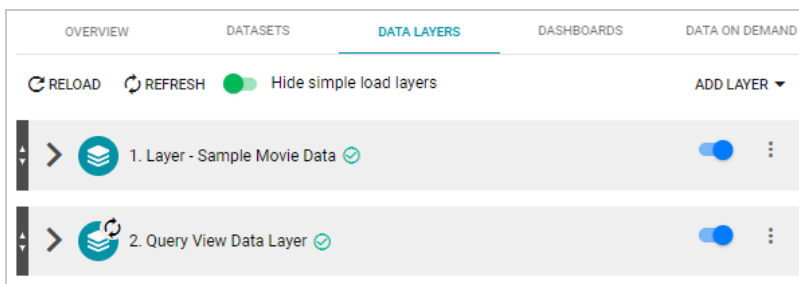
```
ss:ServiceResponse
}
```

Now, if you browse the Managed Edition for the Dataset, you will see that the Edition does not contain any Jobs or Data Components. The next time this Dataset's pipeline is published, the Managed Edition will be repopulated.

---

## What is the difference between the graphmart reload and refresh options?

When you make modifications to data layers in a graphmart, Anzo displays **Reload** and **Refresh** buttons on the top of the Data Layers screen. For example:



The Refresh option becomes available when changes have been made to one or more data layers. Clicking **Refresh** resets (deletes from AnzoGraph) and reloads only the data layers that have changed. Clicking **Reload** resets and reloads the entire graphmart to AnzoGraph, including the data layers that have not changed.

---

## What happens to the existing data in an FLDS when I run an incremental ETL job?

When you publish a pipeline that includes a job that onboards data incrementally, Anzo preserves the previously onboarded data by moving the existing RDF files to a hidden directory within the FLDS. The new RDF files for the current pipeline are then written to the FLDS. Since hidden directories are not loaded to AnzoGraph, only the most current data is loaded into memory. For information about onboarding data incrementally, see [Creating an Incremental Schema](#).