



5.2 Deployment and User Guide

Last Updated: 4/10/2023



Table of Contents

	2
Anzo 5.2 Deployment & User Guide	14
Deployment Guide	15
Deploying Anzo	16
Anzo Requirements	16
Installing Anzo	22
Installing the Anzo for Office Plugin	30
Upgrading Anzo	30
Uninstalling Anzo	32
Deploying the Shared File System	33
Deploying a Static AnzoGraph Cluster	35
AnzoGraph Requirements	35
Sizing Guidelines for In-Memory Storage	41
Sizing Guidelines for Disk-Based Storage (Preview)	48
Installing AnzoGraph	49
Prepare the AnzoGraph Host Servers	50
Install AnzoGraph	53
Complete the Post-Installation Configuration	57
Upgrading AnzoGraph	66
Uninstalling AnzoGraph	67
Deploying a Static Anzo Unstructured Cluster	68

Anzo Unstructured Overview	68
Anzo Unstructured Data Onboarding Process	73
Anzo Unstructured Requirements	75
Installing Anzo Unstructured	77
Deploy the Leader Node	78
Deploy the Worker Nodes	79
Installing and Configuring Elasticsearch	85
Upgrading Anzo Unstructured	87
Using K8s for Dynamic Deployments of Anzo Components	89
Kubernetes Concepts	89
Anzo K8s Requirements	90
Compute Resource Planning	92
Deploying the K8s Infrastructure	94
Amazon EKS Deployments	95
Google Kubernetes Engine Deployments	129
Azure Kubernetes Service Deployments	160
User Guide	195
Onboarding Structured Data	196
Adding Data Sources and Schemas	196
Creating a Database Data Source	196
Creating a CSV Data Source	222
Creating a JSON Data Source	227

Creating an XML Data Source	234
Creating a SAS Data Source	238
Creating a Parquet Data Source and Ingesting the Data	244
Assigning Primary Keys in an Onboarded Schema	249
Creating or Changing Foreign Keys	252
Managing Data Source Metadata	256
Creating a Metadata Dictionary	256
Configuring Data Source Categories	266
Generating a Source Data Profile	269
Ingesting Data	273
Ingesting a New Data Source	273
Re-Ingesting an Updated Data Source	278
Ingesting a Data Source with a Metadata Dictionary	283
Working with Mappings	289
Creating a New Mapping	290
Configuring Mappings to Ingest a Subset of the Source Data	299
Transforming Data in Mappings	303
Supported Mapping Functions	317
Working with Pipelines	336
Creating a Dataset Pipeline	336
Creating an ETL Pipeline	341
Publishing a Pipeline or Subset of Jobs	341

Incremental Pipeline Reference	343
Onboarding Unstructured Data	346
Creating an Unstructured Pipeline	346
Running an Unstructured Pipeline	361
Modeling Data	363
Introduction to Models	363
Model Requirements and Recommendations	365
Uploading a Model to Anzo	369
Creating a Model	371
Editing a Model	373
Opening Models in the Editor	374
Changing Model Components	376
Class Editor Reference	378
Property Editor Reference	380
Setting Class Instance URI Patterns	381
Downloading a Model	384
Blending Data	388
Working with Datasets	388
Adding a Dataset to the Dataset Catalog	388
Adding an Empty Dataset for an Export Step	393
Managing Dataset Editions	395
Configuring Dataset Categories	405

Generating a Dataset Data Profile	408
Working with Graphmarts	411
Creating a New Graphmart	411
Copying a Graphmart	416
Graphmart Settings Reference	417
Generating a Graphmart Data Profile	419
Adding a Dataset to a Graphmart	421
Working with Data Layers	424
Sharing Access to Graphmarts	512
Creating Data on Demand Endpoints	521
Blending Data from Remote Sources (Preview)	533
Introduction to the Graph Data Interface	533
Reading Remote Source Metadata	534
Reading or Ingesting Remote Instance Data	548
Sharing Access to Artifacts	561
Accessing and Analyzing Data	567
Accessing Data with Hi-Res Analytics Dashboards	567
Introduction to Hi-Res Analytics	568
Getting Started: Exploring and Visualizing Data	571
Creating Dashboards	581
Creating a Lens	605
Creating Dashboard Filters	608

Combining Data from Multiple Classes	649
Calculating Values in Lenses and Filters	654
Searching for Text in Unstructured Documents	658
Sharing Access to a Dashboard or Lens	663
Exporting a Lens	669
Deleting a Lens	670
Function and Formula Reference	671
Lens Type Reference	696
Accessing Data on Demand Endpoints	719
Accessing an Endpoint Programmatically	720
Accessing an Endpoint from an Application	723
OData Reference	736
Accessing Data with the Query Builder	744
Running SPARQL Queries in the Query Builder	744
Searching for Quads in the Query Builder	749
Accessing Data from the SPARQL Endpoint	754
Accessing Data from the HTTP Client Interface	764
SPARQL Query Templates and Best Practices	770
SPARQL Best Practices	775
Versioning and Migrating Artifacts	780
Creating and Restoring Versions of Artifacts	780
Creating a Backup Version of an Artifact	780

Restoring a Backup Version of an Artifact	782
Exporting an Artifact	784
Making Values Replaceable on Export	788
Importing Exported Versions of Artifacts	789
Graph Data Storage Reference	793
Exploring Data Provenance	795
Administration Guide	798
Accessing the Administration Application	799
Anzo Server Administration	800
Starting and Stopping Anzo	800
Changing Anzo Server Settings	801
Managing Certificates	810
Using a Signed Certificate	810
Adding a Certificate to the Trust Store	814
Updating the Server License	815
Updating the License Key	815
Licensing and User Account Best Practices	816
Managing Volumes	817
Creating a New Volume	817
Mounting an Existing Volume	819
Uploading a Plugin	820
Advanced Configuration of Semantic Services	821

Setting a Base File Store Path for File Uploads	821
Enabling and Configuring the System Monitor Service	822
Normalizing LDAP Names	824
Routing Hi-Res Analytics to a Custom URL	825
Separating Audit Logs by Type of Event	827
Limiting the Age (and Size) of Audit Logs	828
Configuring a User Inactivity Timeout	829
Reporting on Binary Store Access Events	830
Configuring the Max Page Size for OData Feeds	830
Scanning the Whole CSV File on Import	831
Including Views as Schemas for Database Data Sources	832
Limiting the Number of Anzo Unstructured Status Journals	833
Connection Administration	834
Connecting to a File Store	834
Creating an Anzo Data Store	839
Connecting to AnzoGraph	842
Connecting to Elasticsearch	848
Connecting to an ETL Engine	849
Configuring a Spark ETL Engine	850
Configuring a Sparkler Engine	853
Limiting Job Concurrency on a Remote Sparkler Engine	857
Connecting to a Cloud Location	858

Importing the NFS Configuration	858
Creating a Cloud Location	860
User Management	863
User Management and Access Control Concepts	863
User Management Concepts	863
Artifact Access Control Concepts	867
Connecting to a Directory Server	871
Adding Directory Users and Groups to Anzo	877
Enabling Self-Authorization for Directory Users	881
Connecting to an SSO Provider	882
Creating and Managing Roles	902
Creating an Internal Anzo User	906
Predefined Anzo Roles and Permissions	908
Role Permissions Reference	914
Managing Default Access Policies	922
Monitoring and Diagnostics	930
Managing Anzo Logging	930
Introduction to Anzo Logging	930
Adding the Recommended Log Packages	939
Retrieving AnzoGraph Diagnostic Files	949
Monitoring AnzoGraph	951
System Query Audit	955

AnzoGraph Server Administration	959
Starting and Stopping AnzoGraph	959
Configuring AnzoGraph for Kerberos Authentication	961
Using the AnzoGraph CLI	962
Changing AnzoGraph Configuration Settings	967
Relocating AnzoGraph Directories	968
Using AnzoGraph Persistence (Preview)	969
Ignoring Missing Graphs	970
Changing the Default FROM Clause Behavior	971
Managing the Automatic Restart Feature	972
Enabling Paged Data Mode (Preview)	975
AnzoGraph System Settings Reference	977
Generating Diagnostic Files with the System Manager	983
Anzo Admin CLI	986
Setting up the Admin CLI	986
Querying Graphmart Data	991
Accessing a Graph's Metadata	993
Specifying an Output Format	993
Developer Guide	995
Deploying the Anzo Java SDK	996
Troubleshooting	1004
Getting Information from the Anzo Log Files	1005

Viewing the Current Stack in a Browser	1006
Error Message Reference	1008
Anzo Error Messages	1008
AnzoGraph Error Messages	1009
FAQ	1010

Anzo 5.2 Deployment & User Guide

Welcome to the Anzo 5.2 Deployment & User Guide! This guide provides deployment instructions, administration and configuration information, and instructions for using Anzo 5.2 components.

- [Deployment Guide](#)
- [User Guide](#)
- [Administration Guide](#)
- [Developer Guide](#)
- [Troubleshooting](#)
- [FAQ](#)

Deployment Guide

The Deployment Guide provides hardware and software requirements and installation instructions for Anzo and all of the components in the platform. Once you install Anzo, the AnzoGraph, Anzo Unstructured, Spark, and Elasticsearch components can be deployed on "static" clusters, where the software is installed on pre-configured hardware, VMs, or cloud instances, or they can be deployed dynamically in a Kubernetes (K8s) cluster. When the K8s infrastructure is deployed, Anzo can launch the components on-demand and then deprovision the resources when the components are not in use. This guide includes instructions for deploying the components on static clusters or as dynamic, K8s-based applications.

- [Deploying Anzo](#)
- [Deploying the Shared File System](#)
- [Deploying a Static AnzoGraph Cluster](#)
- [Deploying a Static Anzo Unstructured Cluster](#)
- [Using K8s for Dynamic Deployments of Anzo Components](#)

Deploying Anzo

The topics in this section provide details about the Anzo server requirements and give instructions for installing, upgrading, and uninstalling the software.

- [Anzo Requirements](#)
- [Installing Anzo](#)
- [Installing the Anzo for Office Plugin](#)
- [Upgrading Anzo](#)
- [Uninstalling Anzo](#)

Anzo Requirements

This page provides important guidelines to follow when choosing the hardware and software for Anzo host servers.

- [Hardware Requirements](#)
- [Software Requirements](#)
- [Firewall Requirements](#)
- [File Storage Requirements](#)
- [Standalone Spark Server Requirements](#)

Hardware Requirements

The following guidelines apply to individual Anzo servers within production and development environments. Your Cambridge Semantics Customer Success manager can help you identify an overall Anzo and AnzoGraph deployment configuration that is appropriate for your solution and use cases.

- [Production Environments](#)
- [Development Environments](#)

Production Environments

Component	Minimum	Recommended	Description
RAM	64 GB	128+ GB	The Anzo system data source is a disk-based graph store (called a Journal or Volume). When the system source is queried, Anzo swaps the data from disk to memory on demand. Choosing a host server with more RAM increases the performance of system queries because the OS can store the journal data in its file cache, avoiding the need for Anzo to swap data from disk to memory. In addition, RAM is required to hold intermediate results for join queries.
Disk Space: Anzo Install Path	100 GB	500+ GB	The Anzo server installation disk needs to have enough space to store the Anzo system data source, Anzo log files, any plugins, and the Anzo client. In addition, if the local Sparkler compiler and Spark ETL engine are used on the Anzo server, consider that the disk size also needs to be sufficient for hosting all of the job-related .jar files.
Disk Space: Shared File System	500 GB	1+ TB	The shared file system stores all of the RDF data and ETL files that are shared between Anzo and all AnzoGraph, Anzo Unstructured, Spark, and Elasticsearch servers. For more information, see File Storage Requirements below.
vCPU	16	32	Once you provision sufficient RAM, performance depends on CPU capabilities. Keep in mind that you are provisioning for both a production database and a busy application server. A greater number of cores and high clock speed can make a dramatic difference in performance when there are many concurrent Anzo users.
Architecture	64-bit	64-bit	Anzo is supported only on 64-bit architecture.

Development Environments

Component	Minimum	Recommended	Description
RAM	32 GB	64+ GB	These RAM guidelines assume that the development environment is intended to host smaller data volumes than the production environment and support one or two Anzo users at a time. For development environments with large data volumes and multiple concurrent users, increase the RAM amount.
Disk Space: Anzo Install Path	100 GB	500+ GB	The Anzo server installation disk needs to have enough space to store the Anzo system data source, Anzo log files, any plugins, and the Anzo client. In addition, if the local Sparkler compiler and Spark ETL engine are used on the Anzo server, consider that the disk size also needs to be sufficient for hosting all of the job-related .jar files.
Disk Space: Shared File System	500 GB	1+ TB	Typically the development environment mounts the same shared file system as the production environment.
vCPU	8	16	Like the RAM guidelines, these vCPU guidelines assume that the development environment is intended to host smaller data volumes than the production environment and support one or two Anzo users at a time. For development environments with large data volumes and multiple concurrent users, increase the number of vCPU.
Architecture	64-bit	64-bit	Anzo is supported only on 64-bit architecture.

Software Requirements

This section lists the software requirements for Anzo servers and client workstations. It also includes important service account information and lists the supported single sign-on providers.

Note

Do not run any other software, including anti-virus software, on the same server as Anzo. Additional software may be run in a development environment with the expectation of lowered Anzo performance. Cambridge Semantics strongly recommends that you do not run additional software on the Anzo server in a production environment.

Component	Minimum	Recommended	Guidelines
Operating System (Anzo Server)	RHEL/CentOS 6	RHEL/CentOS 7.9	Cambridge Semantics recommends that you tune the ulimits for your Linux distribution to increase the limits for certain resources. See Configure User Resource Limits for more information.
Microsoft Excel (Client Workstation)	Excel 2003	Excel 2007+	The Anzo for Office data integration mapping tool plugin requires Microsoft Excel.
Web Browser (Client Workstation)	Firefox 62+ Chrome 74+ Safari 12+ Chromium-Based	Chrome 90+	Use the latest versions of web browsers, especially if you are using a Chromium-based browser, as some older versions will not work with the Anzo user interface components.
Enterprise-Level Anzo Service User Account	N/A	N/A	It is important to work with your IT organization to create an Anzo service user account at the enterprise level. The service user account needs to be associated with a central directory server (LDAP) so that it is available across Anzo environments and is managed in accordance with the permissions policies of your company. For more information, see Anzo Service Account Requirements below.

Anzo Service Account Requirements

For consistent and appropriate access management across current and future Anzo environments, it is important for the IT organization to create an enterprise-level, LDAP-managed Anzo service user account. The service account should be used when installing and running Anzo and all of the components in the platform, such as AnzoGraph,

Spark, Elasticsearch, and Anzo Unstructured clusters. The service account should not have root user privileges but does need the following access:

- The account must have read and write permissions for the Anzo component installation directories. The default Anzo server installation directory is `/opt/Anzo`.
- The account must have read and write access to the shared file store, such as the NFS mount location, where all Anzo components will read and write files during the data onboarding processes. For more information about the shared file system requirements, see [Deploying the Shared File System](#).

Important

Set the Anzo account User ID (UID) and Group ID (GID) to **1000**. For integration between Anzo applications, it is important that the owner of files that are written to the shared file store is UID 1000, especially if you are considering Kubernetes-based deployments of Anzo applications.

- The account must have a home directory on the Anzo host server.

Supported Single Sign-On Providers

Anzo supports the following single sign-on (SSO) protocols:

- Basic SSO
- Facebook OAuth
- JSON Web Tokens (JWT)
- Kerberos
- OpenID Connect (OIDC)
- Security Assertion Markup Language (SAML)
- Spring Security OAuth2

For information about configuring SSO access, see [Connecting to an SSO Provider](#).

Firewall Requirements

The table below lists the TCP ports to open on the Anzo host.

Port	Description	Access Needed...
61616	Anzo port used by the software development kit (SDK) and command line interface (CLI)	<ul style="list-style-type: none">• Between Anzo and users.
61617	Anzo SSL port used by the SDK and CLI	<ul style="list-style-type: none">• Between Anzo and users.
8022	Anzo SSH service port	<ul style="list-style-type: none">• Between Anzo and users.

Port	Description	Access Needed...
8945	Anzo Administration service port	<ul style="list-style-type: none"> Between Anzo and users
8946	Anzo Administration service SSL port	<ul style="list-style-type: none"> Between Anzo and users.
80	Application HTTP port	<ul style="list-style-type: none"> Between Anzo and users.
443	Application HTTPS port.	<ul style="list-style-type: none"> Between Anzo and users.
3389	LDAP port	<ul style="list-style-type: none"> Between Anzo and the LDAP server.
9393 (optional)	Optional Java Management Extensions (JMX) port. Enable this port if you want to connect to Anzo from a JMX client.	<ul style="list-style-type: none"> Between Anzo and the JMX client.
9394 (optional)	Optional JMX SSL port. Enable this port if you want to make a secure connection to Anzo from a JMX client.	<ul style="list-style-type: none"> Between Anzo and the JMX client.
5700	The Anzo protocol (gRPC) port for secure communication between AnzoGraph and Anzo For more information about the communication between Anzo and AnzoGraph, see Firewall Requirements in AnzoGraph Server Requirements.	<ul style="list-style-type: none"> Between Anzo and the AnzoGraph leader server.
5600	AnzoGraph's SSL system management port	<ul style="list-style-type: none"> Between Anzo and the AnzoGraph leader server.

File Storage Requirements

Anzo needs to have read and write access to a file storage system that can be shared between Anzo and all AnzoGraph, Anzo Unstructured, ETL Engine, and Elasticsearch servers. The supported storage systems are NFS, Hadoop Distributed File Systems (HDFS), File Transfer Protocol (FTP or FTPS) systems, Google Cloud Platform (GCP) storage, and Amazon Simple Cloud Storage Service (S3). In almost all cases, organizations create an NFS to mount to all of the servers in the Anzo environment. Mounted network file systems offer the best support and performance for reading and writing files.

Note

For details and guidance on choosing the file system, see [Deploying the Shared File System](#).

Standalone Spark Server Requirements

Anzo includes an embedded Spark ETL engine to integrate data from various sources. Depending on your server configuration, the embedded engine might not be sufficient for ingesting very large amounts of data. To support ingestion of large data sets, you can install standalone ingestion servers. The table below lists the recommended configuration for standalone Spark servers.

Component	Recommendation
Available RAM	100+ GB
Disk Space	200+ GB
vCPU	16+

Related Topics

[Installing Anzo](#)

[Deploying the Shared File System](#)

Installing Anzo

This topic provides instructions for installing Anzo. For information about server requirements, see [Anzo Requirements](#).

1. [Complete the Pre-Installation Configuration](#)
2. [Install and Configure Anzo](#)
3. [Complete the Post-Installation Configuration](#)

Complete the Pre-Installation Configuration

- [Make Sure the Anzo Service User Account is Created](#)
- [Configure User Resource Limits](#)

Make Sure the Anzo Service User Account is Created

Important

It is important to work with your IT organization to ensure that an Anzo service user account is created at the enterprise level. The user account needs to be associated with a central directory server (LDAP) so that it is available for installing and running Anzo components across environments. For more information, see [Anzo](#)

Service Account Requirements.

If necessary, you can create a temporary user account on the Anzo host server. Note that creating the account locally can cause issues when migrating Anzo or integrating with a central LDAP server. The service account should meet the following requirements:

- The service account should not have root-user privileges.
- The account must have read and write permissions for the Anzo installation directory. The default installation directory is `/opt/Anzo`.
- The account must have read and write access to the shared file store, such as the NFS mount location, where Anzo will read and write files during the data onboarding processes.

Note

If your organization will use Anzo Unstructured with Elasticsearch to onboard unstructured data, it is especially important to install and run Anzo as a non-root user. Elasticsearch cannot be run by a root user, but it must have access to the data that Anzo writes on the shared file store. When Anzo is run as root the data that it generates is owned by root and Elasticsearch cannot access it.

Configure User Resource Limits

Cambridge Semantics recommends that you tune the user resource limits (ulimits) for your Linux distribution to increase the limits for the following resources:

- Increase the limit for the following resources to at least **65535**:
 - open files (nofile)
 - max user processes (nproc)
- Increase the limit for the following resources to **infinity**:
 - address space (as)
 - CPU time (cpu)
 - file locks (locks)
 - file size (fsize)
 - max memory size (memlock)

To view the current ulimits, run `ulimit -a`. To permanently change ulimits, modify the `/etc/security/limits.conf` file. For information, see [How to set ulimit values](#) in the RHEL support documentation.

Note

Typically, as part of post-installation configuration, a systemd service is set up to start and stop the Anzo process. When systemd starts a process, however, it uses the limits that are defined in the systemd service rather than the limits in `/etc/security/limits.conf`. In addition to changing the ulimits in `limits.conf`, it is important to set the limits in the Anzo service. The service file contents shown in [Configure and Start the Anzo Service](#) includes the recommended ulimit settings.

Install and Configure Anzo

Follow the instructions below to install Anzo. These instructions assume that you have copied the Anzo installation script to the server.

Important Complete the steps below as the Anzo service user.

1. If necessary, run the following command to become the Anzo service user:

```
# su name
```

Where *name* is the name of the service user. For example:

```
# su anzo
```

2. If necessary, run the following command to make the Anzo installation script executable:

```
chmod +x script_name
```

3. Run the following command to start the installation wizard:

```
./script_name
```

The script unpacks the JRE and then waits for input before starting the installation.

4. Press **Enter** to start the installation.
5. Review the software license agreement. Press **Enter** to scroll through the terms. At the end of the agreement, type **1** to accept the terms or type **2** to disagree and stop the installation.
6. Specify the components to install. Item 1 is the Anzo server; item 2 is the Anzo command line client. To install both components, accept the default value by pressing **Enter**. Or type **1** to install only the server or **2** to install only the command line client, then press **Enter**.
7. Specify the path and directory for the Anzo installation. Press **Enter** to accept the default installation path or type an alternate path and then press **Enter**.
8. Indicate whether you want the installer to create symlinks. Press **Enter** for yes or type **n** and press **Enter** for no.
9. If you chose to let the installer create symlinks, specify the directory to create the symlinks in. Press **Enter** to accept the default path or type an alternate path and then press **Enter**.

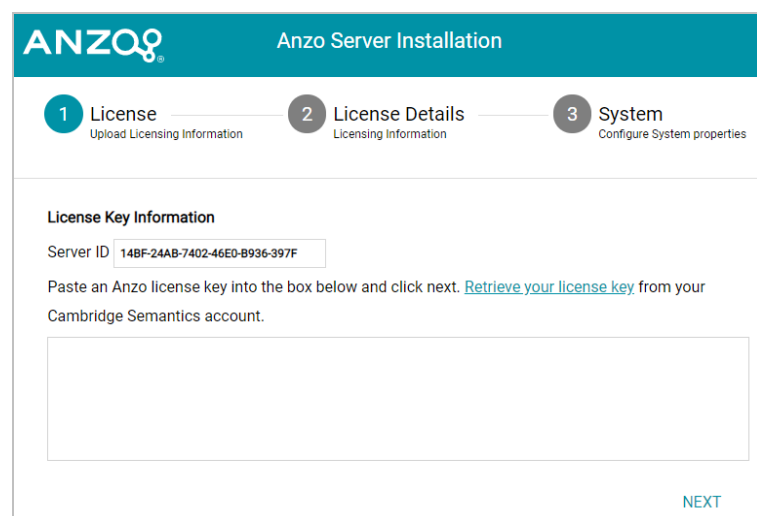
10. Specify the maximum amount of memory (in MB) that the server can use and then press **Enter**. The installation wizard lists the total RAM available. To meet the minimum memory requirement, the wizard chooses 1/4 of the total memory as the default value. Cambridge Semantics recommends that you allocate at least 1/2 of the total memory to Anzo.

The wizard installs the components that you selected and then asks if you want to start the Anzo services.

11. Press **Enter** to start the Anzo services. When prompted, open a browser and go to the following URL to open the license administration wizard.

```
http://<hostname>:8945/
```

Where <hostname> is the Anzo server DNS name or IP address. The License Key Information screen appears. For example:



The screenshot shows the 'ANZO' logo and 'Anzo Server Installation' header. Below is a progress bar with three steps: 1. License (Upload Licensing Information), 2. License Details (Licensing Information), and 3. System (Configure System properties). The 'License' step is active. The main content area is titled 'License Key Information' and contains a 'Server ID' field with the value '14BF-24AB-7402-46E0-B936-397F'. Below this is a text prompt: 'Paste an Anzo license key into the box below and click next. [Retrieve your license key](#) from your Cambridge Semantics account.' A large text input box is provided for the license key. A 'NEXT' button is located at the bottom right.

12. Paste your license key into the box provided and then click **Next**. If necessary, you can obtain the license key by clicking **Retrieve your license key** and logging in to your Cambridge Semantics account.
13. The wizard displays your license details. Review the details and then click **Next**. The wizard displays the System Configuration screen. For example:

ANZO Anzo Server Installation

1 License Upload Licensing Information 2 License Details Licensing Information 3 System Configure System properties

System Configuration

Set the Anzo Server system properties

System userid
sysadmin

System password

Verify System password

Advanced Configuration

Storage Directory:
/opt/Anzo51/Server/data

BACK FINISH

14. On the left side of the screen, specify the password to use for the system administrator, **sysadmin**, in the **System Password** and **Verify System Password** fields.

Important

Do not change the system administration user ID. It must be **sysadmin**. The sysadmin user account has permission to access all Anzo features in the main Anzo application as well as administrative features in the Administration application. In addition, the sysadmin user has read and write access to all of the artifacts (Data Sources, Models, Pipelines, etc.) that are created by all Anzo users. For more information about the account, see [System Administrator](#).

15. On the right side of the screen under Advanced Configuration, the **Storage Directory** setting is displayed. This setting configures the binary store location. By default Anzo stores binary data in <install_path>/Server-/data. You can change the location by typing a new path and directory.
16. Click **Finish**. The wizard starts configures and restarts the server. The process may take several minutes. Once the server is running, the browser displays the Anzo login screen. Before logging in, there is one more configuration step to complete. Some of the Anzo services will not have started properly because they could not bind to the default HTTP/S ports. The default Anzo HTTP port is 80 and the HTTPS port is 443. Since non-root users cannot access ports below 1000, Anzo services will not be able to access the default ports when Anzo is run by the new service user. The Anzo port settings need to be changed to the non-root ports 8080 and 8443:
 - a. On the Anzo server, run the following command to make an SSH connection to the Anzo Command Console as the **sysadmin** user:

```
ssh sysadmin@localhost -p 8022
```

- b. When prompted, specify the password for the sysadmin user and log in to the Anzo OSGI Command Console.

- c. At the OSGI prompt, run the commands below, followed by exit to exit the console:

```
osgi> httpPort 8080
```

```
osgi> httpsPort 8443
```

```
osgi> exit
```

17. Run the following command to restart Anzo and complete the port configuration:

```
./<install_path>/Server/AnzoServer restart
```

18. When Anzo starts, open the Anzo user interface by going to the following URL in your browser:

```
https://<hostname>
```

Where <hostname> is the Anzo server DNS name or IP address.

Complete the Post-Installation Configuration

This section provides instructions for completing post-installation tasks.

- [Route Anzo HTTP/S Ports to Non-Root Ports for User Access](#)
- [Change the Local Spark Engine Callback URL to the Non-Root Port](#)
- [Configure and Start the Anzo Service](#)

Route Anzo HTTP/S Ports to Non-Root Ports for User Access

This section provides instructions for configuring the firewall to forward HTTP requests to port 8080 and HTTPS requests to port 8443 so that users can access Anzo without having to specify the new HTTP/S ports.

Note Root user privileges are required to complete this task.

To re-route Anzo ports using the iptables interface

Run the following commands to route the Anzo ports via the iptables interface:

```
# iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080
# iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 443 -j REDIRECT --to-port 8443
# iptables-save > /etc/sysconfig/iptables
```

To re-route Anzo ports using the firewalld interface

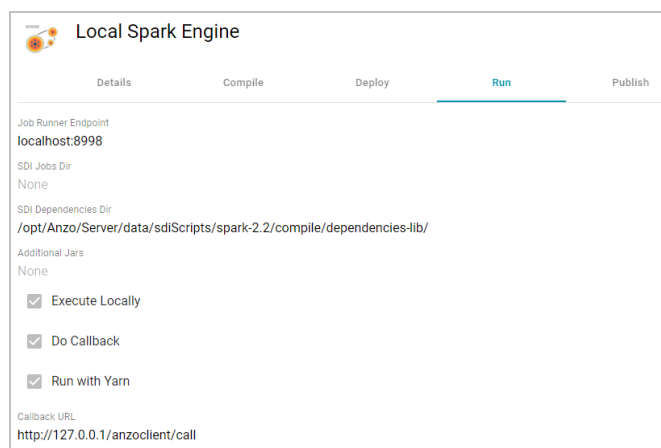
Run the following commands to route the Anzo ports via the firewalld interface:

```
# firewall-cmd --permanent --add-forward-port=port=443:proto=tcp:toport=8443
# firewall-cmd --permanent --add-forward-port=port=80:proto=tcp:toport=8080
# firewall-cmd --reload
```

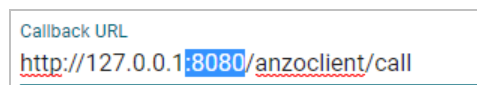
Change the Local Spark Engine Callback URL to the Non-Root Port

If you plan to use the pre-configured local Anzo Spark ETL engine to run pipelines, the callback URL for the engine must be configured to bind to the new Anzo HTTP port. Follow the instructions below to change the callback URL.

1. In the Administration application, expand the **Connections** menu and click **ETL Engine Config**.
2. On the ETL Engine Config screen, click the **Local Spark Engine** to view the configuration details for the engine.
3. Click the **Run** tab. Anzo displays the Run screen. For example:



4. At the bottom of the screen, click the edit icon (✎) next to the **Callback URL** field (hover your pointer over the field to display the edit icon). Then edit the callback URL value to specify the HTTP port at the end of the IP address. For example:



5. Click the check mark icon (✓) to save the change.

Configure and Start the Anzo Service

Cambridge Semantics recommends that you configure an Anzo service for starting Anzo automatically as the service user. Follow the instructions below to implement and start the service.

Note Root user privileges are required to complete this task.

1. Create a file called **anzo-server.service** in the `/usr/lib/systemd/system` directory. For example:

```
# vi /usr/lib/systemd/system/anzo-server.service
```

2. Add the following contents to `anzo-server.service`. Placeholder values are shown in **bold**:

```
[Unit]
Description=Service for Anzo server.
```



```

After=syslog.target network.target local-fs.target remote-fs.target nss-lookup.target

[Service]
Type=simple
RemainAfterExit=yes
LimitCPU=infinity
LimitNOFILE=65536
LimitAS=infinity
LimitNPROC=65536
LimitMEMLOCK=infinity
LimitLOCKS=infinity
LimitFSIZE=infinity
ExecStart=/install_path/Server/AnzoServer start
ExecStop=/install_path/Server/AnzoServer stop
User=service_user_name
Group=service_user_name

[Install]
WantedBy=default.target

```

Where **install_path** is the Anzo installation path and directory and **service_user_name** is the name of the Anzo service user. For example:

```

[Unit]
Description=Service for Anzo server.
After=syslog.target network.target local-fs.target remote-fs.target nss-lookup.target

[Service]
Type=simple
RemainAfterExit=yes
LimitCPU=infinity
LimitNOFILE=65536
LimitAS=infinity
LimitNPROC=65536
LimitMEMLOCK=infinity
LimitLOCKS=infinity
LimitFSIZE=infinity
ExecStart=/opt/Anzo/Server/AnzoServer start
ExecStop=/opt/Anzo/Server/AnzoServer stop
User=anzo
Group=anzo

[Install]
WantedBy=default.target

```

3. Save and close the file, and then run the following commands to start and enable the new service:

```
# systemctl start anzo-server.service
```

```
# systemctl enable anzo-server.service
```

The client displays a message such as the following:

```
Created symlink from /etc/systemd/system/default.target.wants/anzo-server.service to  
/usr/lib/systemd/system/anzo-server.service.
```

Once the service is enabled, Anzo should be running. Any time you start and stop Anzo, run the following `systemctl` commands: `sudo systemctl stop anzo-server` and `sudo systemctl start anzo-server`.

Tip

For an introduction to Anzo concepts, an overview of the user interface, basic setup steps, and instructions for building a sample solution from scratch, see the [Getting Started Guide](#).

Related Topics

[Upgrading Anzo](#)

[Installing the Anzo for Office Plugin](#)

[User Guide](#)

Installing the Anzo for Office Plugin

After installing Anzo, you can access the installation package for the Anzo for Microsoft Office plugin. Anzo for Office includes the data integration mapping tool which enables you to map relationships between schemas and models as well as apply various transformations to the source data.

To access the installations that are included with your license, go to the following URL:

```
http://<Anzo_server>/installs
```

Where `<Anzo_server>` is the Anzo server DNS name or IP address. Follow the instructions onscreen to download and install the plugin.

Related Topics

[Deploying Anzo](#)

Upgrading Anzo

Before you upgrade Anzo, Cambridge Semantics recommends that you make a backup copy of the current Anzo installation in case you have issues and need to revert to the original version. There are three commonly used methods for backing up Anzo:

- Some users choose to make a copy of the Anzo system volume or journal, `<install_path>/Server-/data/journal/anzo.jnl`. If you keep a copy of `anzo.jnl`, you can restore the original Anzo version by reinstalling that release and then copying the backed up journal file into the installation.
- Some users choose to copy or create a tarball of the entire Anzo installation directory, `<install_path>/Anzo`. A backup of the directory can be large, however, and you might want to remove log files to reduce the overall size of the directory before copying or compressing it. If you keep a copy of `<install_path>/Anzo`, you can restore that version by uninstalling the new version and moving the backed up directory to the original installation location.
- Some users choose to take a snapshot of the application disk.

Follow the instructions below to upgrade Anzo.

Important

Complete the steps below as the Anzo service user. When Anzo is initially installed, a server ID is generated based on a number of system properties, including the user account that runs the installation script. The Anzo server license is tied to that server ID. If Anzo is re-installed (for instance, during an upgrade) by a different user account, a new server ID is generated and the existing license will no longer be valid for the installation. For more information, see [Licensing and User Account Best Practices](#).

1. Stop the existing Anzo server if it is running. Then copy the new Anzo installation script to the server and run the following command to make the script executable:

```
chmod +x <file_name>
```

2. Run the following command to start the installation wizard and perform the upgrade:

```
./<file_name>
```

The wizard unpacks the JRE and then waits for input before starting the upgrade.

3. Press **Enter** to start the upgrade. The wizard detects the existing installation and asks if you want to update it.
4. Press **Enter** to update the existing installation.
5. Review the software license agreement. Press **Enter** to scroll through the terms. At the end of the agreement, type **1** and press **Enter** to accept the terms or type **2** and press **Enter** to disagree and stop the update.
6. Specify the components to install. Accept the default entry by pressing **Enter**. Or type **1** to install only the server components or **2** to install only the command line client, then press **Enter**.
7. Specify the maximum amount of memory (in MB) that the server can use and then press **Enter**. The wizard lists the amount of memory you have dedicated to the existing Anzo installation. You can type a different value if necessary, and then press **Enter**. The wizard starts the upgrade and then asks if you want to start the server automatically when the upgrade completes.

8. Press **Enter** to start Anzo when the upgrade completes. If you do not want to start the server, type **n** and then press **Enter**. The setup wizard completes the upgrade process.

Related Topics

[Installing Anzo](#)

[Updating the Server License](#)

Uninstalling Anzo

This topic provides instructions for uninstalling Anzo.

Important Complete the steps below as the Anzo service user.

1. Run the following command to begin the uninstall process:

```
./<install_path>/uninstall
```

2. Press **Enter** to confirm that you want to uninstall Anzo. The wizard asks if you want to clear the Anzo installation directory and user and configuration files.
3. Press **Enter** if you want the wizard to remove the entire Anzo installation directory as well as all configuration and user files. Type **n** and then press **Enter** if you do not want the wizard to remove the installation directory.

The wizard uninstalls Anzo.

Related Topics

[Installing Anzo](#)

Deploying the Shared File System

Anzo and all of its remote applications must be able to access files on a shared file system. Anzo, AnzoGraph, Anzo Unstructured, Spark, and Elasticsearch servers need to share storage so that they can read and/or write the source data ingestion files, RDF load files, ETL job files, Elasticsearch indexes, and other supporting files.

While Anzo supports file connections to Network File Systems (NFS), Hadoop Distributed File Systems (HDFS), File Transfer Protocol (FTP or FTPS) systems, Google Cloud Platform (GCP) storage, and Amazon Simple Cloud Storage Service (S3), some object stores, like Amazon S3, are sufficient for long-term storage but do not offer POSIX support. Other storage systems, such as FTP, often have poor file transfer performance.

Note

For the best read and write performance, Cambridge Semantics strongly recommends that you deploy an NFS and then mount it to each of the AnzoGraph, Anzo Unstructured, Elasticsearch, and Spark servers that make up the Anzo platform.

Important

If you plan to set up Kubernetes (K8s) integration for dynamic deployments of Anzo components, an NFS is required. Other file and object stores are not supported for K8s deployments at this time.

NFS Guidelines

This section describes the key recommendations to follow when creating an NFS for the Anzo platform:

- Use NFS Version 4 or later.
- Provision SSD disk types for the best performance.
- When determining the size of the NFS, consider your workload and use cases. There needs to be enough storage space available for any source data files, ETL job files, generated RDF data files, Elasticsearch indexes, and any other files that you plan to store on the NFS. In addition, consider that cloud-based NFS servers often have better performance if you over-provision resources. When using a cloud-based VM for your NFS, it can be beneficial to provision more CPU, disk space, and RAM than required to store your artifacts.
- For integration between Anzo applications, the Anzo service account must have read and write access to the NFS. In addition, it is important to set the Anzo account User ID (UID) and Group ID (GID) to **1000** so that the owner of files that are written to the shared file store is UID 1000. For more information about the user account requirements, see [Anzo Service Account Requirements](#).

Note

If you are unable to map the Anzo service account UID and GID to 1000, you can modify **anonuid** and **anongid** in the NFS server export table to map all requests to 1000. To do so, add the following line to

/etc/exports on the NFS server:

```
<mount_point> *(insecure,rw,sync,no_root_squash) x.x.x.x(rw,all_  
squash,anonuid=1000,anongid=1000)
```

For example:

```
/global/nfs/data *(insecure,rw,sync,no_root_squash) x.x.x.x(rw,all_  
squash,anonuid=1000,anongid=1000)
```

Related Topics

[Deploying Anzo](#)

[Connecting to a File Store](#)

[Deploying a Static AnzoGraph Cluster](#)

[Deploying a Static Anzo Unstructured Cluster](#)

[Using K8s for Dynamic Deployments of Anzo Components](#)

Deploying a Static AnzoGraph Cluster

The topics in this section provide instructions for deploying a static AnzoGraph cluster. This section includes the hardware and software requirements for AnzoGraph host servers, provides guidelines for sizing a cluster, and gives instructions for installing, upgrading, and uninstalling AnzoGraph.

Tip

For instructions on setting up Kubernetes infrastructure so that AnzoGraph clusters can be launched on-demand, see [Using K8s for Dynamic Deployments of Anzo Components](#).

- [AnzoGraph Requirements](#)
- [Sizing Guidelines for In-Memory Storage](#)
- [Sizing Guidelines for Disk-Based Storage \(Preview\)](#)
- [Installing AnzoGraph](#)
- [Upgrading AnzoGraph](#)
- [Uninstalling AnzoGraph](#)

AnzoGraph Requirements

This topic lists the minimum requirements and recommendations to follow for setting up static AnzoGraph host servers and cluster environments.

- [Hardware Requirements](#)
- [Software Requirements](#)
- [Firewall Requirements](#)

Hardware Requirements

The following guidelines apply to individual AnzoGraph servers. Your Cambridge Semantics Customer Success manager can help you identify an overall AnzoGraph deployment configuration that is appropriate for your solution and use cases.

Component	Minimum	Recommended	Guidelines
RAM	16 GB (for small-scale testing only)	200+ GB	<p>AnzoGraph needs enough RAM to store data, intermediate query results, and run the server processes. Cambridge Semantics recommends that you allocate 3 to 4 times as much RAM as the planned data size. Do not overcommit RAM on a VM or on the hypervisor/container host.</p> <div> <p>Tip</p> <p>For more information about determining the server and cluster size that is ideal for hosting AnzoGraph, see Sizing Guidelines for In-Memory Storage.</p> </div>
Disk Space & Type	20 GB HDD	200+ GB SSD	<p>AnzoGraph requires 10 GB for internal requirements. The amount of additional disk space required for any load file staging, data persistence, or logs depends on the size of the data to be loaded. For persistence, Cambridge Semantics recommends that you have twice as much disk space on the local AnzoGraph file system as RAM on the server.</p>
vCPU	2	32	<p>Once you provision sufficient RAM and a high-performing I/O subsystem, performance depends on CPU capabilities. A greater number of cores can make a dramatic difference in the performance of file loads and concurrent queries.</p> <div> <p>Note</p> <p>Intel processors are preferred, but AnzoGraph is supported on newer Epyc AMD processors. AnzoGraph does not run on older AMD processors.</p> </div>

Component	Minimum	Recommended	Guidelines
Networking	10gbE	20+gbE	<p>Not applicable for single server installations. Since AnzoGraph is high performance computing (HPC) Massively Parallel Processing (MPP) OLAP engine, inter-cluster communications bandwidth dramatically affects performance. AnzoGraph clusters require optimal network bandwidth.</p> <div> <p>Important</p> <p>All servers in a cluster must be in the same network. Make sure that all instances are in the same VLAN, security group, or placement group.</p> </div> <p>In a switched network, make sure that all NICs link to the same Top Of Rack or Full-Crossbar Modular switch. If possible, enable SR-IOV and other HW acceleration methods and dedicated layer 2 networking that guarantees bandwidth.</p>
Shared File System	N/A	N/A	<p>The Anzo file store (shared file system) must be accessible from each AnzoGraph server in the cluster. For more information about the shared file system, see Deploying the Shared File System.</p>

Clusters and Virtual Environments

AnzoGraph requires that all elements of the infrastructure provide the same quality of service (QoS). Do not run AnzoGraph on the same server as any other software, including anti-virus software, except when in single-server mode and with an expectation of lowered performance. Providing the same QoS is especially important when using AnzoGraph in a clustered configuration. If any of the servers in the cluster perform additional processing, the cluster becomes unbalanced and may perform poorly. A single poor performing server degrades the other servers to the same performance level. **All nodes require the same hardware specification and configuration.** Also use static IP addresses or make sure that DHCP leases are persistent.

To ensure the maximum and most reliable QoS for CPU, memory, and network bandwidth, do not co-locate other virtual machines or containers (such as Docker containers) on the same hypervisor or container host. For hypervisor-managed VMs, configure the hypervisor to reserve the available memory for the AnzoGraph server. For clusters,

make sure there is enough physical RAM to support all of the AnzoGraph servers, and reserve the memory via the hypervisor.

In addition, running memory compacting services such as Kernel Same-page Merging (KSM) impacts CPU QoS significantly and does not benefit AnzoGraph. Live migrations also impact the performance of VMs while they get migrated. While live migration can provide value for planned host maintenance, AnzoGraph performance may be impacted if live migrations occur frequently. For more information about Kernel Same-page Merging, see https://en.wikipedia.org/wiki/Kernel_same-page_merging.

Tip

Advanced configurations may benefit from CPU pinning on the hypervisor host and disabling CPU hyper-threading. For more information about CPU pinning, see https://en.wikipedia.org/wiki/Processor_affinity. For information about hyper-threading, see <https://en.wikipedia.org/wiki/Hyper-threading>.

Cambridge Semantics can provide benchmarks to establish relative cluster performance metrics and validate the environment.

Software Requirements

The table below lists the software requirements for AnzoGraph servers. Instructions for installing each of the required software components are included in the AnzoGraph installation instructions. See [Deploying a Static AnzoGraph Cluster](#) for more information.

Component	Minimum	Recommended	Guidelines
Operating System	RHEL 7.5, CentOS 7.5	RHEL 7.9, CentOS 7.9	Cambridge Semantics recommends that you tune the ulimits for your Linux distribution to increase the limits for certain resources. See Configure User Resource Limits for more information. <div> Note AnzoGraph is not supported on RHEL/CentOS 8 at this time. </div>
GNU Compiler Collection	N/A	Installed	Install the latest version of the GCC tools for your operating system. GCC installation instructions are included in Prepare the AnzoGraph Host Servers .

Component	Minimum	Recommended	Guidelines
OpenJDK 11	N/A	Installed	AnzoGraph uses a Java client interface to access connected data sources for data profiling, remote sources for data blending, and Elasticsearch for unstructured pipelines. Java Development Kit version 11 is required for using the Java client. OpenJDK installation instructions are included in Prepare the AnzoGraph Host Servers .
bzip2	N/A	Installed	Required for unpacking the AnzoGraph tool set during installation. BZIP2 installation instructions are included in Prepare the AnzoGraph Host Servers .
Enterprise-Level Anzo Service User Account	N/A	N/A	It is important to work with your IT organization to create an Anzo service user account at the enterprise level. The service user account needs to be associated with a central directory server (LDAP) so that it is available across Anzo environments and is managed in accordance with the permissions policies of your company. For more information, see Anzo Service Account Requirements .

Optional Software

Program	Description
vim	Editor for creating or changing files.
sudo	Enables users to run programs with alternate security privileges.
net-tools	Networking utilities.

Program	Description
psutil	Python system and process utilities for retrieving information on running processes and system usage.
tuned	Linux system service to apply tuning.
wget	Utility for downloading files over a network.
Google SDK	For virtual servers on Google Cloud Engine (GCE). Command line tool to enable syncing of data from Google storage. You can download the latest version from Google: https://cloud.google.com/sdk/ .

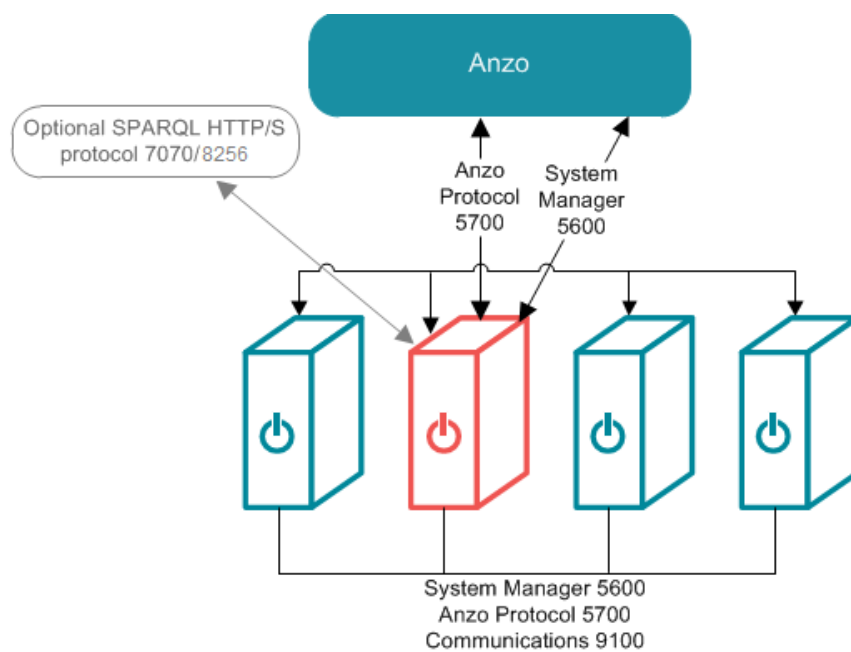
Firewall Requirements

AnzoGraph servers communicate via TCP/IP sockets. AnzoGraph communicates with Anzo via the secure, encrypted, gRPC-based Anzo protocol. Since AnzoGraph is SPARQL-compliant, you also have the option to use standard SPARQL HTTP/S protocol for communication.

Important

For AnzoGraph clusters, all servers in the cluster must be in the same network. Make sure that all instances are in the same VLAN, security group, or placement group.

Open the TCP ports listed in the table below. This image shows a visual representation of the communication ports:



Port	Description	Access Needed...
5700	The Anzo protocol (gRPC) port for secure communication between AnzoGraph and Anzo.	<ul style="list-style-type: none"> Between Anzo and the AnzoGraph leader server. Between all AnzoGraph servers in the cluster. Available for AnzoGraph on single node installations.
5600	AnzoGraph's SSL system management port.	<ul style="list-style-type: none"> Between Anzo and the AnzoGraph leader server. Between all AnzoGraph servers in the cluster. Available for AnzoGraph on single node installations.
9100	AnzoGraph's internal fabric communications port.	<ul style="list-style-type: none"> Between all AnzoGraph servers in a cluster. Available for AnzoGraph on single node installations.
7070 (optional)	Optional SPARQL service HTTP port to enable if you want to give external applications access to AnzoGraph over HTTP.	<ul style="list-style-type: none"> Between external applications and the AnzoGraph leader server.
8256 (optional)	Optional SPARQL service HTTPS port to enable if you want to give external applications SSL access to AnzoGraph and/or use the command line interface, azgi.	<ul style="list-style-type: none"> Between external applications and the AnzoGraph leader server.

Related Topics

[Sizing Guidelines for In-Memory Storage](#)

[Sizing Guidelines for Disk-Based Storage \(Preview\)](#)

[Installing AnzoGraph](#)

Sizing Guidelines for In-Memory Storage

This topic provides guidance on determining the server and cluster size that is ideal for hosting AnzoGraph, depending on the characteristics of your data.

- [Memory Sizing Guidelines](#)
- [Analyzing Data Characteristics in Load Files](#)
- [Cluster Sizing Guidelines](#)

Memory Sizing Guidelines

Since AnzoGraph is a high-performance, in-memory database, it is important to consider the amount of memory needed to store the data that you plan to load. Estimating the amount of memory your workload requires can help you decide what size server to use and whether to use multiple servers. The sections below describe the key points to consider about memory usage and AnzoGraph.

- [Data at rest should remain below 50% of the total memory](#)
- [AnzoGraph reserves 20% of the memory for the OS](#)
- [Memory usage can be high during loads](#)
- [Memory usage depends on data characteristics](#)

Data at rest should remain below 50% of the total memory

The data loaded into memory should not consume more than 50% of the total available memory on the instance or across a cluster. **Ideally, the data at rest should use only 25%-30% of the available memory** because query processing and intermediate results can temporarily consume a very large amount of RAM.

AnzoGraph reserves 20% of the memory for the OS

To avoid unexpected shutdowns by the Linux operating system, the default AnzoGraph configuration leaves 20% of memory available for the OS; AnzoGraph will not use more than 80% of the total available memory. Account for this memory buffer in sizing calculations.

Memory usage can be high during loads

During the load streaming process, before duplicates are pruned and triples are moved to their final storage blocks, memory usage temporarily increases and potentially doubles, particularly if the data includes many string values.

Memory usage depends on data characteristics

Memory usage varies significantly depending on the makeup of the data, such as the data types and sizes of literal values, and the complexity of the queries that you run. Triple storage ranges anywhere from 12 bytes per triple to 1 megabyte for a triple that stores pages of text from an unstructured document. For example:

- Triples with integer objects like the following example require about 16 bytes to store in memory.

```
<http://csi.com/resource/person1> <http://csi.com/resource/age> 50
```

- Triples made up of URIs like the following example require about 18 bytes to store in memory.

```
<http://csi.com/resource/person1> <http://csi.com/resource/friend>
<http://csi.com/resource/person100>
```

- Triples with user-defined data types (UDTs) like the following example also require about 18 bytes to store in memory.

```
<http://csi.com/resource/person1> <http://csi.com/resource/height> "5'8"^^height
```

- Triples with dateTime values like the following example require about 20 bytes to store in memory.

```
<http://www.wikidata.org/entity/Q65949130>
<http://www.wikidata.org/prop/direct/P585>
"1995-01-01T00:00:00Z"^^<http://www.w3.org/2001/XMLSchema#dateTime> .
```

- Triples with long strings like the following example require about 700 bytes to store in memory.

```
<http://dbpedia.org/resource/Keanu_Reeves> <http://dbpedia.org/ontology/abstract>
"Keanu Charles Reeves
(/ker'ɑ:nu:/ kay-AH-noo; born September 2, 1964) is a Canadian actor, producer,
director and musician.
Reeves is best known for his acting career, beginning in 1985 and spanning more than
three decades.
He gained fame for his starring role performances in several blockbuster films
including comedies
from the Bill and Ted franchise (1989-1991), action thrillers Point Break (1991) and
Speed (1994),
and the science fiction-action trilogy The Matrix (1999-2003). He has also appeared in
dramatic
films such as Dangerous Liaisons (1988), My Own Private Idaho (1991), and Little
Buddha (1993),
as well as the romantic horror Bram Stoker's Dracula (1992)."
```

The table below provides estimates for the number of triples that you can load and query with commonly configured amounts of available RAM. The table also lists the number of triples that could be stored if a data set comprised the example triples above.

Note

The examples below show the number of triples at rest and consider that the data should not consume more than 50% of the available RAM.

RAM	General Estimate	Examples
16 GB	Up to about 100 million triples	<p>Considering that the data at rest should use less than 8 GB RAM, a server with 16 GB total RAM could store:</p> <ul style="list-style-type: none"> • About 12 million 700-byte triples like the Keanu Reeves example above. • About 475 million 18-byte URI triples like the example above.
32 GB	Up to about 200 million triples	<p>Considering that the data at rest should use less than 16 GB RAM, a server with 32 GB total RAM could store:</p> <ul style="list-style-type: none"> • About 24 million 700-byte triples like the Keanu Reeves example above. • About 850 million 20-byte triples like the dateTime example above.
64 GB	Up to about 400 million triples	<p>Considering that the data at rest should use less than 32 GB RAM, a server with 64 GB total RAM could store:</p> <ul style="list-style-type: none"> • About 48 million 700-byte triples like the Keanu Reeves example above. • About 1.7 billion 20-byte triples.
128 GB	Up to about 800 million triples	<p>Considering that the data at rest should use less than 64 GB RAM, a server with 128 GB total RAM could store:</p> <ul style="list-style-type: none"> • About 96 million 700-byte triples like the Keanu Reeves example above. • About 3.4 billion 20-byte triples.
256 GB	Up to about 1.5 billion triples	<p>Considering that the data at rest should use less than 128 GB RAM, a server with 256 GB total RAM could store:</p> <ul style="list-style-type: none"> • About 192 million 700-byte triples like the Keanu Reeves example above. • About 6.8 billion 20-byte triples.
480 GB	Up to about 3 billion triples	<p>Considering that the data at rest should use less than 240 GB RAM, a server with 480 GB total RAM could store:</p> <ul style="list-style-type: none"> • About 368 million 700-byte triples like the Keanu Reeves example above. • About 12 billion 20-byte triples.

Analyzing Data Characteristics in Load Files

AnzoGraph enables you to perform pre-load analysis on file-based linked data sets without actually loading the data into memory. You can use this method to run statistical queries, such as counting the number of triples or returning a list of the unique subjects and predicates. Performing a "dry run" of a data load enables you to analyze data set characteristics to help with tasks such as memory sizing. Since the data remains on disk, you can use this method to capture statistics about a large data set without having to deploy an AnzoGraph cluster that has enough memory to store all of the data.

Important Considerations for Analyzing Load Files

- Since AnzoGraph scans the files on disk, queries run much slower than they do when run against data in memory. Consider performance when deciding how many files to query at once and how complex to make the queries.
- Though the pre-load feature does not use memory for storing data, queries that you run against files do consume memory. The server must have sufficient memory available to use for these intermediate query results.
- Unlike loads into the database, pre-load analysis does not prune duplicate triples. Statistics returned for load file queries may differ somewhat from the statistics returned after the data is loaded.

Analysis Query Syntax

Use the following query syntax to analyze load files :

```
SELECT <expression>
FROM EXTERNAL <URI>
[ FROM EXTERNAL <URI> ]
WHERE { <triple_patterns> }
```

Option	Description
SELECT <expression>	The SELECT clause specifies an expression that returns statistical results such as a count of the total number of triples or the number of distinct predicates. Queries that return values for a specific property may return an error.
FROM EXTERNAL <URI>	<p>The URI in the FROM clause specifies the location of the load file or directory of files. For example, this URI specifies a single file:</p> <pre><file:/data/load/values.ttl></pre> <p>This example specifies a directory of files:</p> <pre><dir:/data/store/LoadDBNorthwind/rdf.ttl.gz></pre>

For example, the following query analyzes the files in the `rdf.ttl.gz` directory for an FLDS. The query counts the total number of triples in the files:

```
SELECT (count (*) as ?triples)
FROM EXTERNAL <dir:/nfs/data/store/LoadGHIB_f5886/rdf.ttl.gz>
WHERE { ?s ?p ?o . }
```

```
triples
-----
143704445
1 rows
```

Assessing Memory Requirements Based on File Analysis

Although the memory required to load and perform queries on specific data sets will vary based on the size and type of data contained in a data set as well as the type of queries run, you can still obtain a reasonable estimate for the amount of memory you will need to store data set by using the equation below:

$$\text{total_triples} \times \text{avg_triple_size} + \text{total_chars} = \text{size_estimate}(\text{bytes})$$

Follow the steps below to calculate the values to use in the equation:

1. [Count the total number of triples in the files](#)
2. [Determine the average triple size](#)
3. [Count the number of characters for all strings](#)
4. [Calculate the size estimate](#)

Count the total number of triples in the files

As shown in the example above, the following query counts the total number of triples in FLDS load files:

```
SELECT (count (*) as ?triples)
FROM EXTERNAL <dir:/nfs/data/store/LoadGHIB_f5886/rdf.ttl.gz>
WHERE { ?s ?p ?o . }
```

```
triples
-----
143704445
1 rows
```

Determine the average triple size

The [Memory usage depends on data characteristics](#) section above shows some example triples and their estimated size. If you are familiar with the data in the files, you may be able to determine the average size based on the examples. Otherwise, Cambridge Semantics recommends using 30 bytes as the average triple size.

Count the number of characters for all strings

For ASCII characters, AnzoGraph uses about 1-byte of memory to store each character. Counting the number of characters in the load files provides a good estimate of the number of bytes required to store the strings in your data.

```
SELECT (SUM(IF(DATATYPE(?o)=<http://www.w3.org/2001/XMLSchema#string>,
              (STRLEN(?o)),0)) as ?char_count)
FROM EXTERNAL <URI>
WHERE {?s ?p ?o}
```

For example, the following query returns the number of characters in the strings for the FLDS referenced above:

```
SELECT (SUM(IF(DATATYPE(?o)=<http://www.w3.org/2001/XMLSchema#string>,
              (STRLEN(?o)),0)) as ?char_count)
FROM EXTERNAL <dir:/nfs/data/store/LoadGHIB_f5886/rdf.ttl.gz>
WHERE {?s ?p ?o}
```

```
char_count
-----
684348190
1 rows
```

Calculate the size estimate

Once you have counted the triples, determined the average triple size, and counted the characters, use the formula below to estimate the amount of memory needed to store the data at rest:

```
total_triples x avg_triple_size + total_chars = size_estimate(bytes)
```

For example:

```
143,704,445 x 30 + 684,348,190 = 4,995,481,540 bytes
```

This example FLDS requires roughly 5 GB of memory to store the data.

Cluster Sizing Guidelines

When your workload size requires using a cluster, do not create clusters with fewer than 4 nodes. When using a single node, data gets redistributed in memory without using the network. If you add 1 or 2 more nodes to create a 2- or 3-node cluster, data then gets distributed over the network. The CPU gain from the additional 1 or 2 nodes does not outweigh the performance degradation from the network. Using at least 4 nodes significantly reduces the network degradation and provides a near-linear performance benefit when compared to a single node.

Related Topics

[AnzoGraph Requirements](#)

[Deploying a Static AnzoGraph Cluster](#)

Sizing Guidelines for Disk-Based Storage (Preview)

For fast performance and scalability, AnzoGraph stores all data in memory. If persistence is enabled, data is saved to disk as a backup and so that graphs are automatically reloaded into memory when AnzoGraph is restarted, but queries do not access the data on disk since all of the data is cached in memory. And accessing data in memory is much faster than retrieving data from disk.

When deploying large memory-optimized servers for fast query performance is not feasible, however, AnzoGraph can be configured to operate as a disk-based graph database. In this configuration (called "Paged Data"), data is loaded to AnzoGraph, converted to AnzoGraph's internal storage format, and persisted to disk without being retained in memory. Data is then paged into memory from disk as requested for analytic operations. For details about database operations in paged data mode, see [Enabling Paged Data Mode \(Preview\)](#).

Note

The Paged Data feature is available as a **Preview** release in **2.3.x** versions of AnzoGraph, 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 table below lists the disk and memory sizing requirements and guidelines to follow if you are considering enabling disk-based storage.

Hardware Requirements

Component	Recommendation	Guidelines
RAM	100+ GB	<ul style="list-style-type: none">Though all graph data is stored on disk, RAM is required to hold intermediate results when performing computations and joins.Having more RAM available for paged data caching can reduce the frequency with which AnzoGraph swaps data from disk to memory. More data can remain paged in memory for access during query execution.The amount of data you can expect to be able to store is about 3X the size of RAM. For example, with 200 GB of RAM, you can load and query about 600 GB of data on disk.

Component	Recommendation	Guidelines
Disk Size	500+ GB	The disk size should be at least 4X the size of the data at rest. For example, loading 1 TB of data requires a 4 TB disk to support paging operations.
Disk Type	SSD	The speed of the disk that hosts the persisted data has an impact on query performance. For the best performance, store the persistence directory on a fast disk, such as SSD. You can relocate the default persistence directory from the AnzoGraph file system to a separate location. See Relocating AnzoGraph Directories for more information.
CPU	32	A greater number of multi-core CPU with a high clock speed can make a dramatic difference in the performance of paged data queries. <div data-bbox="680 987 1250 1165"> <p>Note</p> <p>Intel processors are preferred, but AnzoGraph is supported on newer Epyc AMD processors. Older AMD processors are not supported.</p> </div>

Note For software and firewall requirements, see [AnzoGraph Requirements](#).

Ultimately, queries perform significantly slower when data is stored on disk versus in memory. If fast performance is a requirement, data should be stored in-memory, and configuring AnzoGraph for paged data operations should not be considered. For more information, see [Enabling Paged Data Mode \(Preview\)](#).

Related Topics

[AnzoGraph Requirements](#)

[Enabling Paged Data Mode \(Preview\)](#)

Installing AnzoGraph

The topics in this section guide you through installing AnzoGraph on a single server or on multiple servers in a cluster. If you are installing AnzoGraph for the first time on a new host server, make sure that you complete each of the procedures below to perform the prerequisite configuration of the host servers, install the AnzoGraph software, and then complete the post-installation configuration and start the AnzoGraph services.

1. [Prepare the AnzoGraph Host Servers](#)
2. [Install AnzoGraph](#)
3. [Complete the Post-Installation Configuration](#)

Prepare the AnzoGraph Host Servers

Before deploying AnzoGraph, follow the instructions below to install the required software packages on each AnzoGraph host server. In addition to listing the software dependencies, this topic also includes important information about configuring user resource limits, ensuring that AnzoGraph is installed as the appropriate user, and recording the cluster IP addresses that are needed during the install process.

Tip For information about host server requirements, see [AnzoGraph Requirements](#).

- [Install GNU Compiler Collection \(GCC\)](#)
- [Install BZIP2](#)
- [Install OpenJDK 11](#)
- [Configure User Resource Limits](#)
- [Unset Linux Proxy Variables](#)
- [Use the Anzo Service User Account when Installing AnzoGraph](#)
- [Note the IP Addresses of the Cluster Servers](#)

Install GNU Compiler Collection (GCC)

AnzoGraph requires the latest version of the GCC tools for your operating system. Run the following command to install GCC:

```
sudo yum install gcc
```

Note

Specifically, AnzoGraph requires the **glibc**, **glibc-devel**, and **gcc-c++** libraries. Typically, when you install GCC by running `yum install gcc`, those libraries are included as part of the package. In rare cases, depending on the host server configuration, installing GCC excludes certain libraries. If AnzoGraph fails to start and you receive a "Compilation failed" message, it may indicate that some of the required libraries are missing. To install the missing libraries, run the following command:

```
sudo yum install glibc glibc-devel gcc-c++
```

Install BZIP2

BZIP2 is required for unpacking the AnzoGraph tool set during installation. Run the following command to install bzip2:

```
sudo yum install bzip2
```

Install OpenJDK 11

AnzoGraph uses a Java client interface, called the Graph Data Interface (GDI), to access Data Sources when Data Source Profiling is performed or when data from remote endpoints is blended into Graphmarts. AnzoGraph also uses the Java client to communicate with Elasticsearch when Anzo Unstructured Graphmarts are deployed. Java Development Kit version 11 is required for using the Java client. Follow the instructions below to install OpenJDK on all servers in the cluster.

1. Run the following command to install OpenJDK 11:

```
sudo yum install java-11-openjdk
```

Note

Do not set the \$JAVA_HOME variable to use the JDK installation at this time. AnzoGraph's system management daemon requires JAVA_HOME, and it is set as part of the post-installation configuration ([Complete the Post-Installation Configuration](#)). In addition, the Java plugin is deployed after AnzoGraph is installed.

2. If your organization uses Anzo Unstructured, test the connection between the AnzoGraph server and Elasticsearch. Make sure that Elasticsearch is running and then run the following telnet command:

```
telnet <Elasticsearch_server_IP> <port>
```

By default, the port range for Elasticsearch requests (http.port) is 9200-9300. If port 9200 is not available when Elasticsearch is started, Elasticsearch tries 9201 and so on until it finds an accessible port. Specify the HTTP request port that Elasticsearch is using.

For more information about the Graph Data Interface, see [Blending Data from Remote Sources \(Preview\)](#).

Configure User Resource Limits

Cambridge Semantics recommends that you tune the user resource limits (ulimits) for your Linux distribution to increase the limits for the following resources. Tune ulimits on all servers in the cluster.

- Increase the **open files (nofile)** limit to at least **4096**.
- Increase the limit for the following resources to **infinity**:
 - address space (as)
 - CPU time (cpu)
 - file locks (locks)
 - file size (fsize)

- max memory size (memlock)
- max user processes (nproc)

To view the current ulimits, run `ulimit -a`. To permanently change ulimits, modify the `/etc/security/limits.conf` file. For more information, see [How to set ulimit values](#) in the RHEL support documentation.

Note

Typically, as part of post-installation configuration, systemd services are set up to start and stop the AnzoGraph processes. When systemd starts a process, however, it uses the limits that are defined in the systemd service rather than the limits in `/etc/security/limits.conf`. In addition to changing the ulimits in `limits.conf`, it is important to set the limits in the AnzoGraph system management service. The service file contents shown in [Configure the AnzoGraph System Management Service](#) includes the recommended ulimit settings.

Unset Linux Proxy Variables

Make sure that the Linux environment variables `http_proxy` and `https_proxy` are not set on the servers. The Anzo gRPC protocol cannot make connections to the database when proxies are enabled.

Use the Anzo Service User Account when Installing AnzoGraph

Important

Because AnzoGraph offers features such as user-defined extensions, it is not secure software certified and should not be installed or run as the root user. In addition, since AnzoGraph accesses the data that Anzo writes on the shared file store, it is important to install and run AnzoGraph with the same service account that runs Anzo. For more information, see [Anzo Service Account Requirements](#).

Note the IP Addresses of the Cluster Servers

If you are installing AnzoGraph in a clustered setup, make note of the IP addresses for each of the servers in the cluster. The installation wizard will prompt you to enter the IP addresses during the installation. In addition, choose one server to be the leader server.

Once all of the prerequisites are in place, proceed to [Install AnzoGraph](#) for instructions on installing AnzoGraph.

Related Topics

[AnzoGraph Requirements](#)

[Install AnzoGraph](#)

Install AnzoGraph

This topic provides instructions for using the installer to install AnzoGraph on a single server or cluster. Before installing AnzoGraph, make sure that the prerequisites are configured. See [Prepare the AnzoGraph Host Servers](#) for details.

- [Installing AnzoGraph on a Single Server](#)
- [Installing AnzoGraph on a Cluster](#)

Installing AnzoGraph on a Single Server

Follow the instructions below to install AnzoGraph on a single server.

Important Complete the following steps as the Anzo service user.

1. If necessary, run the following command to become the Anzo service user:

```
su <name>
```

Where *name* is the name of the service user. For example:

```
su anzo
```

2. If necessary, run the following command to make the AnzoGraph installation script executable:

```
chmod +x <script_name>
```

3. Run the following command to start the installation wizard:

```
./<script_name>
```

The script unpacks the JRE and then waits for input before starting the installation.

4. Press **Enter** to proceed with the installation. The wizard displays the AnzoGraph license agreement.
5. Review the license agreement. Press **Enter** to scroll through the terms. At the end of the agreement, type **1** to accept the terms or type **2** to disagree and stop the installation.
6. The wizard prompts you to specify which components to install. Specify **1** (AnzoGraph) and press **Enter**.
7. Specify the path and directory for the AnzoGraph installation. Press **Enter** to accept the default installation path or type an alternate path and then press **Enter**.
8. At the server installation type prompt, accept the default option **1 (Standalone)** and press **Enter**.
9. Indicate whether this installation is for use with Anzo. Press **Enter** for **Yes**. Answering yes configures AnzoGraph to use the settings that are optimal for Anzo. Answering no configures the settings that are optimal for AnzoGraph standalone use.

10. Set up the AnzoGraph admin user. Type a username to use for authentication. Anzo will use this username to connect to AnzoGraph. Then press **Enter**.
11. Type a password for the Anzo username and press **Enter**.

Note

Some special characters, such as \$ and *, are treated as parameters in bash. When typing a password, avoid or escape special characters to remove their special meaning to the command line. For more information, see [Quoting](#) in the Bash Reference Manual.

12. Configure any additional AnzoGraph settings. If Cambridge Semantics Support provided custom settings to use for your configuration, type the supplied values and then press **Enter**.

Tip

The AnzoGraph CLI, **azgi**, makes an SSL connection to AnzoGraph on the SPARQL HTTPS port. SSL protocol is disabled by default, however. If you want to be able to use azgi, you can enable SSL protocol by specifying the following value in this prompt: `enable_ssl_protocol=true`. Note that enabling SSL protocol also makes the HTTPS port available to external applications. You may want to check that firewall rules are in place to block external access before enabling SSL protocol. For azgi usage information, see [Using the AnzoGraph CLI](#).

13. The wizard extracts the AnzoGraph files and completes the installation. Proceed to [Complete the Post-Installation Configuration](#) to complete the initial configuration and start the database.

Installing AnzoGraph on a Cluster

Follow the instructions in this section to install AnzoGraph on multiple servers in a cluster. There are two steps in the process:

1. [Install AnzoGraph on the Compute Servers](#)
2. [Install AnzoGraph on the Leader Server](#)

Install AnzoGraph on the Compute Servers

Follow the instructions below to install AnzoGraph on each compute server.

Important Complete the following steps as the Anzo service user.

1. If necessary, run the following command to become the Anzo service user:

```
su <name>
```

Where <name> is the name of the service user. For example:

```
su anzo
```

2. If necessary, run the following command to make the AnzoGraph installation script executable:

```
chmod +x <script_name>
```

3. Run the following command to start the installation wizard:

```
./<script_name>
```

The script unpacks the JRE and then waits for input before starting the installation.

4. Press **Enter** to proceed with the installation. The wizard displays the AnzoGraph license agreement.
5. Review the license agreement. Press **Enter** to scroll through the terms. At the end of the agreement, type **1** to accept the terms or type **2** to disagree and stop the installation.
6. The wizard prompts you to specify which components to install. Specify **1** (AnzoGraph) and press **Enter**.
7. Specify the path and directory for the AnzoGraph installation. Specify the same location on each server. Press **Enter** to accept the default installation path or type an alternate path and then press **Enter**.
8. At the server installation type prompt, specify option **3 (Cluster Compute)** and press **Enter**.
9. Indicate whether this installation is for use with Anzo. Press **Enter** for **Yes**. Answering yes configures AnzoGraph to use the settings that are optimal for Anzo. Answering no configures the settings that are optimal for AnzoGraph standalone use.
10. Type a comma-separated list of the IP addresses for each server in the cluster. Type the leader server IP address first, followed by each compute IP address. For example, on a cluster with 4 servers where 192.168.2.1 is the leader server:

```
192.168.2.1,192.168.2.2,192.168.2.3,192.168.2.4
```

Important

Make sure that you enter this value exactly the same, with IP addresses in the same order, during the installation on each server.

11. After typing the list of IP addresses, press **Enter**. The wizard extracts the AnzoGraph files and completes the installation.
12. Repeat the steps above to install AnzoGraph on each compute server. Then proceed to [Install AnzoGraph on the Leader Server](#) below.

Install AnzoGraph on the Leader Server

Follow the instructions below to install AnzoGraph on the leader server.

Important Complete the steps below as the Anzo service user.

1. If necessary, run the following command to become the Anzo service user:

```
# su <name>
```

Where <name> is the name of the service user. For example:

```
# su anzo
```

2. If necessary, run the following command to make the AnzoGraph installation script executable:

```
chmod +x <script_name>
```

3. Run the following command to start the installation wizard:

```
./<script_name>
```

The script unpacks the JRE and then waits for input before starting the installation.

4. Press **Enter** to proceed with the installation. The wizard displays the AnzoGraph license agreement.
5. Review the license agreement. Press **Enter** to scroll through the terms. At the end of the agreement, type **1** to accept the terms or type **2** to disagree and stop the installation.
6. The wizard prompts you to specify which components to install. Specify **1** (AnzoGraph) and press **Enter**.
7. Specify the path and directory for the AnzoGraph installation. Specify the same location as the compute server installations. Press **Enter** to accept the default installation path or type an alternate path and then press **Enter**.
8. At the server installation type prompt, specify option **2 (Cluster Leader)** and press **Enter**.
9. Indicate whether this installation is for use with Anzo. Press **Enter** for **Yes**. Answering yes configures AnzoGraph to use the settings that are optimal for Anzo. Answering no configures the settings that are optimal for AnzoGraph standalone use.
10. Set up the AnzoGraph admin user. Type a username to use for authentication. Anzo will use this username to connect to AnzoGraph. Then press **Enter**.
11. Type a password for the Anzo username and press **Enter**.

Note

Some special characters, such as \$ and *, are treated as parameters in bash. When typing a password, avoid or escape special characters to remove their special meaning to the command line. For more information, see [Quoting](#) in the Bash Reference Manual.

12. Type a comma-separated list of the IP addresses for each server in the cluster. Type the leader server IP address first, followed by each compute IP address. For example, on a cluster with 4 servers where 192.168.2.1 is the leader server:

```
192.168.2.1,192.168.2.2,192.168.2.3,192.168.2.4
```

Important

Make sure that you enter this value exactly the same, with IP addresses in the same order, as the compute servers.

13. After typing the list of IP addresses, press **Enter**. Configure any additional AnzoGraph settings. If Cambridge Semantics Support provided custom settings to use for your configuration, type the supplied values and then press **Enter**.

Tip

The AnzoGraph CLI, **azgi**, makes an SSL connection to AnzoGraph on the SPARQL HTTPS port. SSL protocol is disabled by default, however. If you want to be able to use **azgi**, you can enable SSL protocol by specifying the following value in this prompt: `enable_ssl_protocol=true`. Note that enabling SSL protocol also makes the HTTPS port available to external applications. You may want to check that firewall rules are in place to block external access before enabling SSL protocol. For **azgi** usage information, see [Using the AnzoGraph CLI](#).

14. The wizard extracts the AnzoGraph files and completes the installation. Proceed to [Complete the Post-Installation Configuration](#) to complete the initial cluster configuration and start AnzoGraph.

Related Topics

[Complete the Post-Installation Configuration](#)

[Prepare the AnzoGraph Host Servers](#)

Complete the Post-Installation Configuration

Once AnzoGraph is installed, there are additional, critical tasks to complete to ensure that AnzoGraph is configured to support all of the Anzo functionality. In addition is it important to set up AnzoGraph services to run as the Anzo service user so that AnzoGraph can access the data that other platform components write to the shared file system. Follow the instructions in the steps below to complete the post-installation configuration.

1. [Deploy the Graph Data Interface Java Plugin](#)
2. [Deploy Optional Drivers for Accessing Database Sources](#)
3. [Configure and Start the AnzoGraph Services](#)

Deploy the Graph Data Interface Java Plugin

The Graph Data Interface (GDI) Java Plugin is a .jar file that is provided by Cambridge Semantics Customer Success. A separate, optional Logging plugin is also provided to enable reporting for GDI usage. Follow the instructions below to deploy the plugins and configure logging.

Note

Java Development Kit version 11 is required for using the GDI. If OpenJDK 11 is not installed, see [Install OpenJDK 11](#) for instructions.

1. Download the following .jar files provided by Cambridge Semantics. Place the downloaded files on the AnzoGraph leader server:
 - gdi-<version>.jar
 - logging-<version>.jar
2. Copy the two files to the <install_path>/lib/udx directory on the leader server.
3. Next, run the following command to change the owner of the files to the anzograph user:

```
chown anzograph:anzograph -R install_path/lib/udx
```

For example:

```
chown anzograph:anzograph -R /opt/anzograph/lib/udx
```

4. If you want to enable logging for the GDI, create a file called **log.config** in the <install_path>/lib/udx directory. Then add the following contents to log.config:

```
@level=WARN
@file=location_to_create_log_file/udx.log
@file.color=false
@udx=true
@stderr=false
@stderr.color=false
@stdout=false
@stdout.color=false
com.cambridgesemantics.anzo.*=INFO
com.cambridgesemantics.anzo.datatoolkit.*=TRACE
com.cambridgesemantics.anzograph.*=INFO
com.cambridgesemantics.anzograph.datatoolkit.*=TRACE
org.openanzo.*=ERROR
```

For example:

```
@level=WARN
@file=/opt/anzograph/internal/udx.log
@file.color=false
@udx=true
@stderr=false
@stderr.color=false
@stdout=false
@stdout.color=false
```

```
com.cambridgesemantics.anzo.*=INFO
com.cambridgesemantics.anzo.datatoolkit.*=TRACE
com.cambridgesemantics.anzograph.*=INFO
com.cambridgesemantics.anzograph.datatoolkit.*=TRACE
org.openanzo.*=ERROR
```

Once the AnzoGraph services are configured and the database is started (as described in [Configure and Start the AnzoGraph Services](#) below), the new plugins are enabled. On a cluster, the leader broadcasts the .jar file to the compute servers.

Note

The GDI natively supports reading or ingesting data from CSV and TSV, JSON, XML, Parquet, and SAS (SAS Transport XPT and SAS7BDAT) files as well as HTTP/REST endpoints. You can extend the service to access relational databases by adding JDBC drivers to the `<install_path>/lib/udx` directory. See [Deploy Optional Drivers for Accessing Database Sources](#) below for more information.

Tip

The `<install_path>/lib/udx` directory on the leader node is a user-managed directory rather than an AnzoGraph-managed directory like `<install_path>/bin` or `<install_path>/internal`. Users can place JDBC drivers and Java or C++ extensions in the `lib/udx` directory any time. Each time the database is started, AnzoGraph scans that directory, saves a copy of its contents to the `<install_path>/internal/extensions` directory, and then broadcasts the `internal/extensions` contents from the leader node to the compute nodes. Each restart clears `internal/extensions` and AnzoGraph rescans `lib/udx` to reload `internal/extensions` with the latest plugins.

Deploy Optional Drivers for Accessing Database Sources

To extend the Graph Data Interface (GDI) service to access relational databases, JDBC drivers can also be deployed to AnzoGraph. If AnzoGraph will access relational Data Sources for Data Source Profiling or GDI queries, copy the same drivers that you use for Anzo to the `<install_path>/lib/udx` directory on the AnzoGraph leader server. The leader also broadcasts any driver .jar files to the compute servers when the database is started.

Configure and Start the AnzoGraph Services

Once the Graph Data Interface client and any other optional drivers are deployed, the last step is to configure the AnzoGraph services and start the database. There are three processes involved in the initial startup of AnzoGraph. And subsequent starts involve one or more of these steps depending on the state of AnzoGraph and the servers:

1. The first process involves the configuration of the Linux kernel and it applies to all servers in the cluster. The default kernel configuration for the following settings is not optimal for AnzoGraph:

- **transparent_hugepage**: Transparent Huge Pages (THP) are enabled by default and can degrade AnzoGraph performance. THP should be disabled for AnzoGraph.
- **max_map_count**: By default, the maximum number of memory map areas that a process can use is 65535. Since AnzoGraph is memory intensive, it may reach the maximum map count and be shut down by the operating system. AnzoGraph requires a **max_map_count** value of **2097152**.

At startup, AnzoGraph checks these settings and returns a warning if the values are not suitable. You are required to make the kernel changes or configure AnzoGraph to start with non-optimal configurations. The AnzoGraph deployment includes a script (`<install_path>/bin/azg_system_config`) that makes the required kernel configuration changes. Superuser privileges are required to make the changes, however, and each time the host server is rebooted the script must be run again because the kernel configuration reverts to the defaults.

What does `azg_system_config` do?

The script runs the following commands to disable THP:

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
```

```
echo never > /sys/kernel/mm/transparent_hugepage/defrag
```

The script runs this command to increase the `max_map_count` value:

```
sysctl -w vm.max_map_count=2097152
```

2. The second process involves the AnzoGraph system management daemon, **azgmgrd**. This very lightweight program runs on all servers in the cluster and manages AnzoGraph communication between the nodes. It must be running to start the database, but it typically does not need to be restarted unless you are upgrading AnzoGraph or the host servers are rebooted. It does not need to be stopped and started each time the database is restarted.
3. The third process involves starting the database with the system manager. Starting the database is done only on the leader server. The leader connects to the system managers on the compute servers and starts the database across the cluster.

To ensure that the right account/permissions are used to perform the three steps above (i.e., the root user makes the kernel changes and the Anzo service account starts the system management daemon and the database) whenever the host server is rebooted, Cambridge Semantics recommends that you configure services to run the AnzoGraph startup steps. This section provides instructions for configuring the three services.

Note Root user privileges are required to complete the tasks below.

1. [Configure the Linux Kernel Configuration Service](#)
2. [Configure the AnzoGraph System Management Service](#)
3. [Configure the AnzoGraph Database Service](#)

Important

On clusters, configure the first two services, the Linux kernel configuration service and the AnzoGraph system management service, on all servers in the cluster. Configure the AnzoGraph database service only on the leader node. For single-server deployments, configure all three services on the server.

Configure the Linux Kernel Configuration Service

On each server in the cluster, follow the instructions below to set up a service to apply the Linux kernel configuration changes any time the AnzoGraph host server is restarted.

Note

If making the kernel changes is not possible, you can set the `os_allow_alternate_vm_config` value to `true` in the AnzoGraph settings file. This setting enables AnzoGraph to start with non-optimal Linux configurations. See [Changing AnzoGraph Configuration Settings](#) for instructions.

1. Run the following command to copy the AnzoGraph system configuration script, **azg_system_config**, to the root directory:

```
# cp <install_path>/bin/azg_system_config /root/
```

For example:

```
# cp /opt/anzograph/bin/azg_system_config /root/
```

2. Run the following command to remove "sudo" from the **azg_system_config** script:

```
# sed -i 's/sudo//g' /root/azg_system_config
```

3. Create a file called **azg_system_config.service** in the `/usr/lib/systemd/system` directory. For example:

```
# vi /usr/lib/systemd/system/azg_system_config.service
```

4. Add the following contents to **azg_system_config.service**:

```
[Unit]
Description=Configure Linux for AnzoGraph

[Service]
Type=oneshot
ExecStart=/root/azg_system_config
```

```
[Install]
WantedBy=multi-user.target
```

5. Save and close the file.
6. Run the following commands to start and enable the new service:

```
# systemctl start azg_system_config.service
```

```
# systemctl enable azg_system_config.service
```

7. Repeat this process on all of the compute servers and the leader server.

Configure the AnzoGraph System Management Service

On each server in the cluster, follow the instructions below to set up a service that starts the AnzoGraph system management daemon (azgmgrd) as the Anzo service user if the host server is restarted.

1. Create a file called **azgmgrd.service** in the `/usr/lib/systemd/system` directory. For example:

```
# vi /usr/lib/systemd/system/azgmgrd.service
```

2. Add the following contents to **azgmgrd.service**. The placeholder values are shown in **bold**:

```
[Unit]
Description=AnzoGraph communication service
# depends on NetworkManager-wait-online.service enabled
Wants=network-online.target
After=network-online.target

[Service]
Type=forking
RemainAfterExit=yes
Restart=on-failure
RestartSec=60s
# The PID file is optional but recommended so that systemd
# can identify the main process of the daemon
# PIDFile=/var/run/azgmgrd.pid
WorkingDirectory=install_path
StandardOutput=syslog
StandardError=syslog
LimitCPU=infinity
LimitNOFILE=4096
LimitAS=infinity
LimitNPROC=infinity
LimitMEMLOCK=infinity
LimitLOCKS=infinity
```

```

LimitFSIZE=infinity
User=Anzo_service_user
UMask=0022
Environment=PATH=/sbin:/bin:/usr/sbin:/usr/bin:/install_path/bin:/install_path/tools/bin
Environment=JAVA_HOME=/usr/lib/jvm/jre-11
ExecStart=/install_path/bin/azgmgrd /install_path/
CPUAccounting=false
MemoryAccounting=false

[Install]
WantedBy=multi-user.target

```

Where **install_path** is the AnzoGraph installation path and directory and **Anzo_service_user** is the name of the Anzo service user. For example:

```

[Unit]
Description=AnzoGraph communication service
# depends on NetworkManager-wait-online.service enabled
Wants=network-online.target
After=network-online.target

[Service]
Type=forking
RemainAfterExit=yes
Restart=on-failure
RestartSec=60s
# The PID file is optional but recommended so that systemd
# can identify the main process of the daemon
# PIDFile=/var/run/azgmgrd.pid
WorkingDirectory=/opt/anzograph
StandardOutput=syslog
StandardError=syslog
LimitCPU=infinity
LimitNOFILE=4096
LimitAS=infinity
LimitNPROC=infinity
LimitMEMLOCK=infinity
LimitLOCKS=infinity
LimitFSIZE=infinity
User=anzo
UMask=0022

Environment=PATH=/sbin:/bin:/usr/sbin:/usr/bin:/opt/anzograph/bin:/opt/anzograph/tools/bin
Environment=JAVA_HOME=/usr/lib/jvm/jre-11

```

```

ExecStart=/opt/anzograph/bin/azgmgrd /opt/anzograph/
CPUAccounting=false
MemoryAccounting=false

[Install]
WantedBy=multi-user.target

```

3. Save and close the file.
4. Run the following commands to start and enable the new service:

```
# systemctl start azgmgrd.service
```

```
# systemctl enable azgmgrd.service
```

5. Repeat this process on all of the compute servers and the leader server.

Configure the AnzoGraph Database Service

On the leader server only, follow the instructions below to set up a service that will start AnzoGraph as the Anzo service user. This service is configured to run after the system management daemon is started.

1. Create a file called **anzograph.service** in the `/usr/lib/systemd/system` directory. For example:

```
# vi /usr/lib/systemd/system/anzograph.service
```

2. Add the following contents to `anzograph.service`. The placeholder values are shown in **bold**:

```

[Unit]
Description=AnzoGraph database service
After=azgmgrd.service
Wants=azgmgrd.service

[Service]
Type=oneshot
RemainAfterExit=yes
RestartSec=60s
# The PID file is optional but recommended so that systemd
# can identify the main process of the daemon
# PIDFile=/var/run/anzograph.pid
WorkingDirectory=install_path
StandardOutput=syslog
StandardError=syslog
User=Anzo_service_user
UMask=0022
Environment=PATH=/sbin:/bin:/usr/sbin:/usr/bin:install_path/bin:install_
path/tools/bin
ExecStart=install_path/bin/azgctl -start

```

```
ExecStop=/install_path/bin/azgctl -stop

[Install]
WantedBy=multi-user.target
```

For example:

```
[Unit]
Description=AnzoGraph database service
After=azgmgrd.service
Wants=azgmgrd.service

[Service]
Type=oneshot
RemainAfterExit=yes
RestartSec=60s
# The PID file is optional but recommended so that systemd
# can identify the main process of the daemon
# PIDFile=/var/run/anzograph.pid
WorkingDirectory=/opt/anzograph
StandardOutput=syslog
StandardError=syslog
User=anzo
UMask=0022

Environment=PATH=/sbin:/bin:/usr/sbin:/usr/bin:/opt/anzograph/bin:/opt/anzograph/tools
/bin
ExecStart=/opt/anzograph/bin/azgctl -start
ExecStop=/opt/anzograph/bin/azgctl -stop

[Install]
WantedBy=multi-user.target
```

3. Save and close the file.
4. Run the following commands to start and enable the new service:

```
# systemctl start anzograph.service
```

```
# systemctl enable anzograph.service
```

Once the services are in place and enabled, AnzoGraph should be running. Any time you start and stop the database, run the following `systemctl` commands on the leader node: `sudo systemctl stop anzograph` and `sudo systemctl start anzograph`. You do not need to stop and start `azgmgrd`.

For instructions on configuring the connection to AnzoGraph in the Anzo application, see [Connecting to AnzoGraph](#).

Related Topics

[Connecting to AnzoGraph](#)

Upgrading AnzoGraph

A key area of growth in AnzoGraph 2.1, 2.2, and 2.3 releases is the development and support of custom, user-managed extensions, such as the Graph Data Interface for virtualization and Elasticsearch support. Most AnzoGraph releases include revisions to the API and prepackaged extensions.

Because of the frequency of updates and because the extensions directory (`<install_path>/lib/udx`) is user-managed rather than AnzoGraph- or installer-controlled, Cambridge Semantics recommends that you uninstall the existing version and install the new version instead of upgrading in-place.

Note

Since AnzoGraph is stateless when used with Anzo and Anzo manages all of your data, removing the existing AnzoGraph files does not impact Anzo or your graphmarts.

Follow the instructions below to back up any custom files and remove the AnzoGraph directory before installing a new version.

Important Complete the steps below as the Anzo service user.

1. First, run the following commands to stop the database and the system management daemon. On a cluster, run these commands on the leader node:

```
sudo systemctl stop anzograph
```

```
sudo systemctl stop azgmgrd
```

2. Next, if you have custom configuration settings, make a backup copy of the `<install_path>/-config/settings.conf` file on the leader node. Make sure that you choose a backup location that is outside of the AnzoGraph installation path.

After installing the new version of AnzoGraph, you can overwrite the new `settings.conf` file with the backup copy.

3. If you have custom JDBC drivers or user-defined extensions in the `<install_path>/lib/udx` directory, make sure those are also backed up in a separate location.

After installing the new version of AnzoGraph, you can place the custom files back into the `<install_path>/lib/udx` directory on the leader node.

4. Remove the AnzoGraph directory from the file system. You can remove AnzoGraph by deleting the installation directory or by running the `<install_path>/uninstall` script and following the prompts to remove the directory. On a cluster, remove the AnzoGraph directory on all nodes.

Once AnzoGraph has been uninstalled, follow the appropriate installation instructions in [Install AnzoGraph](#) to install the new version of AnzoGraph.

Related Topics

[Install AnzoGraph](#)

Uninstalling AnzoGraph

This topic provides instructions for uninstalling AnzoGraph. On clusters, complete steps 2 through 4 below on each server in the cluster.

Important Complete the steps below as the Anzo service user.

1. First, make sure the database and system management daemon processes are stopped. Run the following commands to stop the services. On a cluster, run these commands on the leader server:

```
sudo systemctl stop anzograph
```

```
sudo systemctl stop azgmgrd
```

2. Next, run the following command to begin the uninstall process:

```
./install_path/uninstall
```

3. Press **Enter** to confirm that you want to uninstall AnzoGraph. The wizard asks if you want to clear the AnzoGraph installation directory and user and configuration files. Cambridge Semantics recommends that you remove all installation and configuration files.
4. Press **Enter** if you want the wizard to remove the entire AnzoGraph installation directory as well as all configuration and user files. Type **n** and then press **Enter** if you do not want the wizard to remove the installation directory.

The wizard uninstalls AnzoGraph.

Related Topics

[Deploying a Static AnzoGraph Cluster](#)

Deploying a Static Anzo Unstructured Cluster

If your organization plans to onboard unstructured data to Anzo, additional infrastructure is required for running unstructured pipelines. This section provides instructions for deploying a static Anzo Unstructured (AU) cluster. The topics include an overview of the AU infrastructure, details about the requirements and recommendations, and instructions for installing the software components with the AU installer.

Tip

For instructions on setting up the Kubernetes infrastructure so that AU clusters can be launched on-demand, see [Using K8s for Dynamic Deployments of Anzo Components](#).

- [Anzo Unstructured Overview](#)
- [Anzo Unstructured Data Onboarding Process](#)
- [Anzo Unstructured Requirements](#)
- [Installing Anzo Unstructured](#)
- [Installing and Configuring Elasticsearch](#)
- [Upgrading Anzo Unstructured](#)

Anzo Unstructured Overview

One of Anzo's differentiators as a leading enterprise knowledge graph and data integration platform is its treatment of unstructured data as a first-class citizen in the knowledge graph. Anzo onboards unstructured data—sources that contain text, such as PDFs, text messages, or text snippets embedded in structured data—directly into the knowledge graph using configurable, scalable unstructured data pipelines. These pipelines generate a graph model for the unstructured text and extracted metadata, and they create connections in the graph between these elements and related entities so that the data can be fully integrated into the knowledge graph. In addition, the pipelines build an Elasticsearch index that can be used for highly performant, fully-integrated search queries that look across both free-text and semantic relationships within the knowledge graph.

The following sections provide an overview of the key features of Anzo's unstructured data integration capabilities.

- [Support for a Variety of Sources](#)
- [Text Processing and Annotation](#)
- [Text Indexing and Searching](#)
- [Scalability and Progress Tracking](#)

Support for a Variety of Sources

Anzo's onboarding pipelines can process unstructured text from a large variety of data sources and formats.

Configurable **crawlers** determine what unstructured text a given onboarding pipeline will process. The crawlers can

locate and extract text from files of a variety of formats, including PDFs, emails, HTML files, and Microsoft Word documents.

Anzo's unstructured onboarding pipelines can also be configured to crawl the knowledge graph itself for unstructured content to index and annotate—whether the graph contains free-text directly or references to locations of documents. When combined with Anzo's data virtualization capabilities (see [Blending Data from Remote Sources \(Preview\)](#) for more information), this presents a flexible and powerful framework to rapidly process unstructured data and bring it into a knowledge graph from practically any source or repository in a modern data ecosystem. Anzo's data virtualization capabilities allow users to pull directly into the graph up-to-date structured file metadata from document repositories or unstructured text data stored in external systems. The resulting graph can then be seamlessly passed on as an input to unstructured processing pipelines.

Text Processing and Annotation

As a baseline, unstructured pipelines in Anzo extract basic metadata about each document that they process, such as file location, file size, title, author, etc., and store this metadata within the knowledge graph according to a standardized graph model. The pipelines generate HTML versions of the document that can be rendered in a browser, and references to the document's original binary are maintained in the graph. With this, unstructured content and its associated metadata can be connected and queried alongside any other information stored in the knowledge graph.

Beyond this baseline processing capability, Anzo enables more advanced annotation of unstructured text. Built-in, configurable annotators allow Anzo's unstructured pipelines to pull out facts or references in the text as annotations. Anzo adds the unstructured text data as well as these extracted annotations to the knowledge graph, where they are described by a graph model (ontology) that is dynamically generated by the onboarding pipeline. Additionally, the unstructured pipelines align the annotation spans to the source text and include highlights of the annotated text in the rendered HTML version of the document. Once in the knowledge graph, the unstructured annotation data can easily be discovered, explored, and connected alongside basic document data as well as any other enterprise data in the graph.

The image below shows an HTML rendering of a document and its highlighted annotations in an Anzo Hi-Res Analytics dashboard:

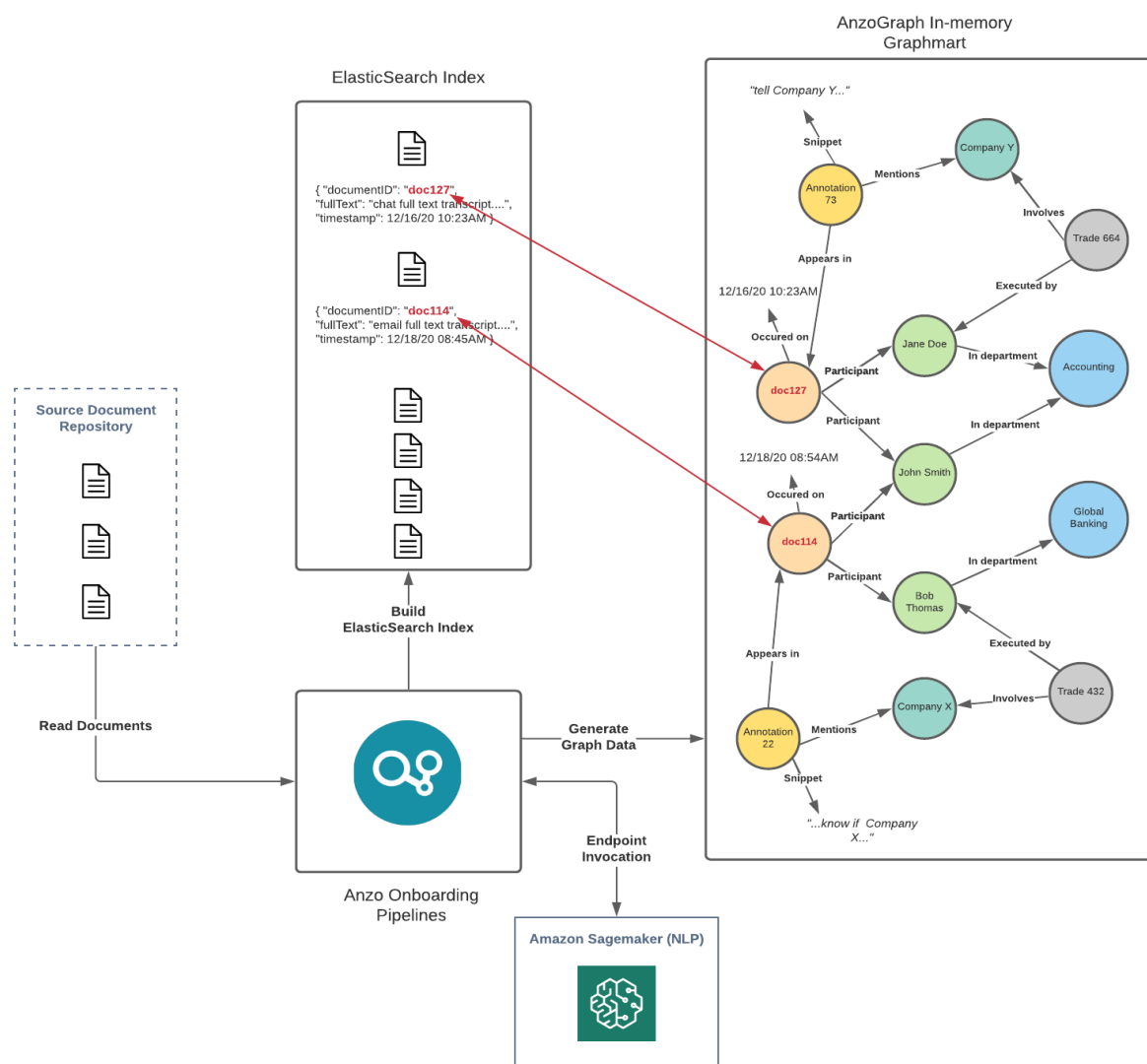
The screenshot displays the Anzo 5.2 user interface. On the left, a search results pane shows documents related to 'Diffusions-Tensor-Bildgebung' from Wikipedia. The main pane on the right is titled 'Table of Documents' and lists several documents with columns for File Name, Document Title, Source, Author, Last Modified, Summary, and Intake Time. Below this, an 'Annotated Document' pane shows a hierarchical diagram of memory systems. The diagram is divided into 'Sensory M.' and 'Long-term M.'. 'Sensory M.' includes 'Short-term M.' (which is further divided into 'Central' and 'Peripheral') and 'Long-term M.'. 'Long-term M.' is divided into 'Declarative M. (explicit)' and 'Nondeclarative M. (implicit)'. 'Declarative M. (explicit)' is further divided into 'Semantic M. Concept M.' and 'Episodic M. Autobiographic M.'. 'Nondeclarative M. (implicit)' is divided into 'Perceptual M.' and 'Procedural M.'. The diagram also includes 'Phonological Loop', 'Visuospatial Sketchpad', and 'Working M.'.

Anzo's built-in annotators offer annotation capabilities based on pattern matching and taxonomies or dictionaries of terms that already exist in the knowledge graph. Anzo's unstructured pipelines also offer a flexible and agnostic extension framework to support integration with external NLP engines that can provide domain-specific or ML-driven text processing capabilities (for example, Amazon Sagemaker, spaCy NER, Amazon Comprehend, etc.). With simple configurations, Anzo's pipelines provide unstructured plaintext to these external components, and then bring their output back into the knowledge graph, dynamically generating a graph model and connecting the extracted annotations to the document metadata and related entities. This can serve not only as an effective way to integrate state-of-the-art NLP insights alongside related data in a knowledge graph, but also as a flexible and transparent paradigm for validation and analysis of ML-driven NLP development.

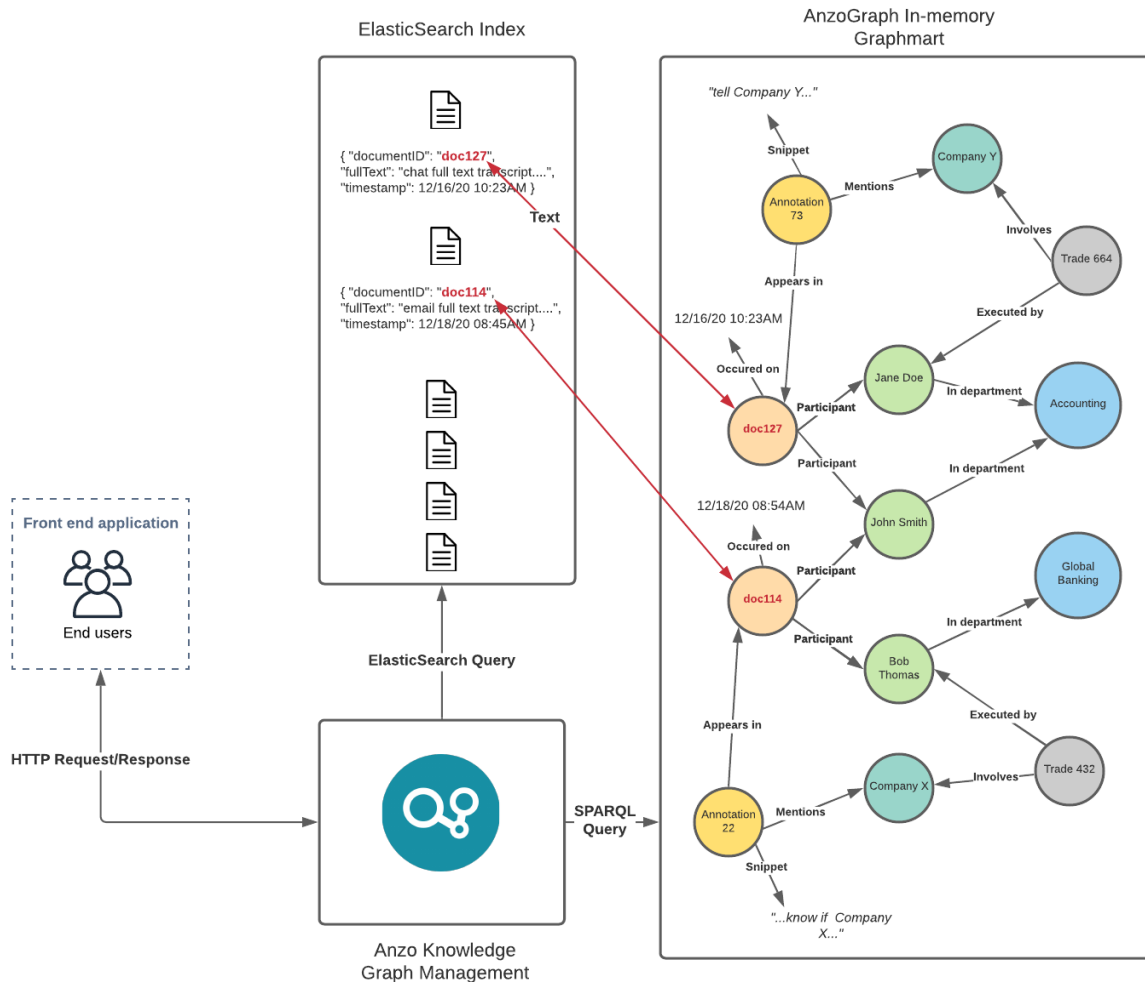
Text Indexing and Searching

Natively, Anzo's unstructured pipelines create an Elasticsearch index of all unstructured files onboarded to Anzo. These indexes contain references to URIs of related entities in the knowledge graph so that the indexed data be joined directly against the rich and highly connected knowledge graph. When coupled with AnzoGraph's native Elasticsearch SPARQL extension, this allows a truly state-of-the-art integration. Users can leverage AnzoGraph's MPP engine and seamlessly execute queries that combine scalable, performant free-text search alongside complex, semantic queries against the graph. Both elements of the query are computed in a highly parallelized manner, resulting in unmatched query performance. This integration can serve as a strong and flexible foundation for advanced, complex modern search applications.

The diagram below shows an overview of Anzo Unstructured's Elasticsearch integration during pipeline processing:



The following diagram shows an overview of Anzo Unstructured's Elasticsearch integration during querying and analysis:

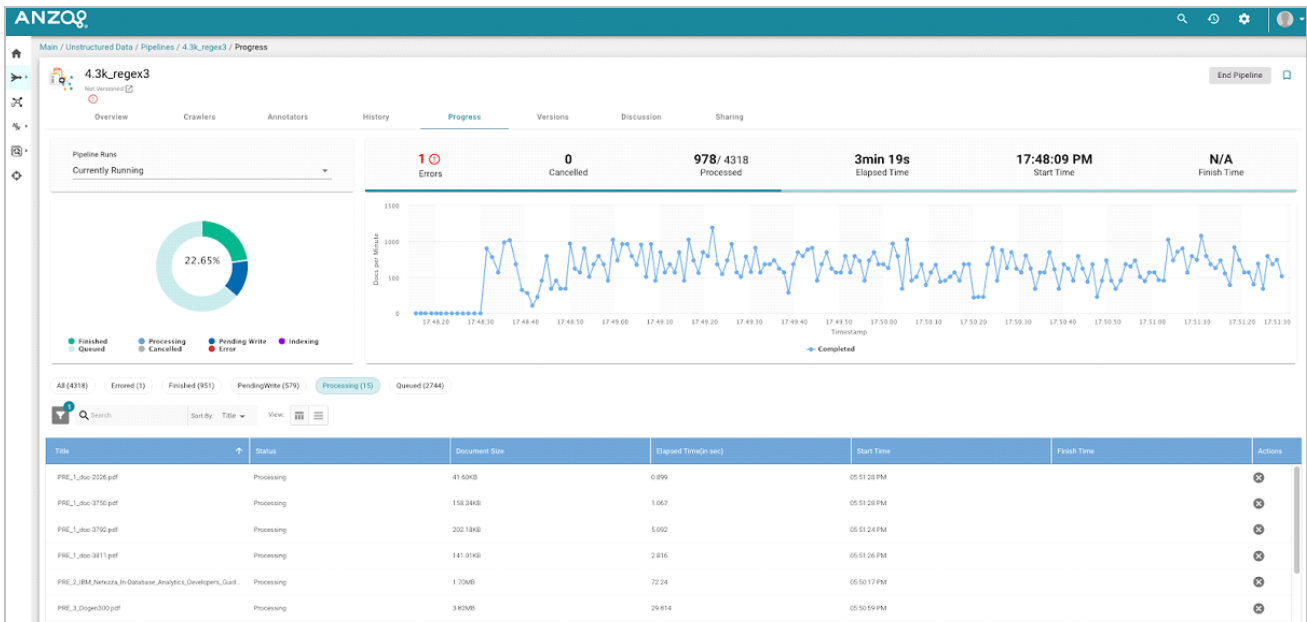


Scalability and Progress Tracking

Anzo's unstructured pipelines run using a highly distributed and performant microservice cluster built using [Akka](#). Worker nodes, which perform text processing in parallel, can be scaled out and up to increase the processing throughput of the pipeline. With this parallelization and scalability, Anzo's pipelines are capable of processing tens of thousands of unstructured documents per minute. The pipeline processing services can be deployed alongside Anzo on standard hardware or cloud instances, or they can be spun up dynamically using Anzo's native Kubernetes integration (see [Using K8s for Dynamic Deployments of Anzo Components](#) for more information).

To track the progress of unstructured data pipelines, Anzo offers a user interface that reports fine-grained status information about each document and its processing status, as well as any issues encountered in processing. The user interface also shows global statistics about a given pipeline run, including overall processing throughput, percentage complete, time elapsed, etc. This reporting module gives system administrators a centralized view of processing progress and an easy way to oversee the pipeline as it operates.

The image below shows Anzo's reporting interface on unstructured pipeline progress:



For more information about unstructured pipeline processing and the resulting artifacts, see [Anzo Unstructured Data Onboarding Process](#).

Related Topics

[Anzo Unstructured Data Onboarding Process](#)

[Anzo Unstructured Requirements](#)

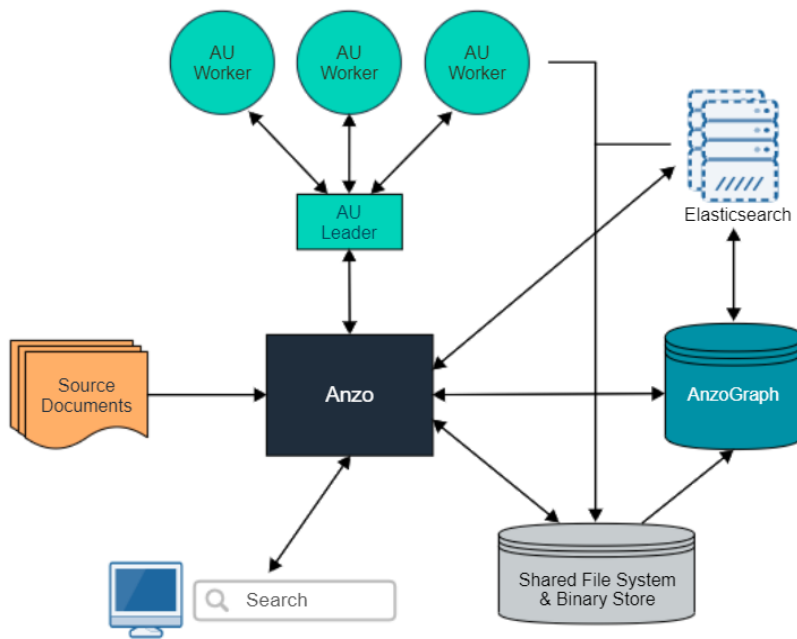
[Installing Anzo Unstructured](#)

[Installing and Configuring Elasticsearch](#)

Anzo Unstructured Data Onboarding Process

Anzo onboards unstructured data through pipelines that run in a distributed environment where a cluster of Worker nodes process the incoming documents and generate output artifacts for Anzo. This topic provides an overview of the Anzo Unstructured (AU) pipeline process and infrastructure.

The diagram below provides a high level overview of the Anzo platform architecture with integration of AU and Elasticsearch. The description below the diagram describes the unstructured data onboarding process and resulting artifacts.



When an unstructured pipeline is run, an Anzo crawler service streams data to a pipeline service. The pipeline service reads the stream of files and constructs the appropriate request payloads—one request per document to process. Anzo sends the requests to the AU leader instance, and the leader queues the requests and distributes them to the AU worker server instances to process in parallel. When each worker instance processes a document, it creates a temporary output artifact on the shared file system. The artifact includes:

- An RDF file that describes the text annotations and general metadata about the processed document.
- A binary store artifact for Anzo.
- A JSON artifact that contains a reference to the extracted text of the document. Elasticsearch uses this artifact to generate the document index.

When the AU workers have processed all of the documents, Anzo completes the following post-processing steps:

- Consolidate the RDF artifacts from the workers and create a file-based linked data set (FLDS) for loading to AnzoGraph.
- Read the JSON artifacts and instruct the Elasticsearch server to build an index with the text extracted from the documents. A snapshot of the index is saved on the file system with the FLDS. Any time a graphmart that includes that FLDS is loaded to an AnzoGraph instance, Anzo loads the corresponding snapshot into the Elasticsearch server that is associated with the AnzoGraph connection.

When the post-processing is finished, the pipeline service finalizes the FLDS metadata to store in its catalog. The new unstructured data set becomes available in the Dataset catalog, and it can be added to a Graphmart and loaded to AnzoGraph for use in Hi-Res Analytics dashboards.

Related Topics

[Anzo Unstructured Overview](#)

[Anzo Unstructured Requirements](#)

[Installing Anzo Unstructured](#)

[Installing and Configuring Elasticsearch](#)

[Upgrading Anzo Unstructured](#)

Anzo Unstructured Requirements

The Anzo Unstructured (AU) infrastructure is highly customizable and scalable. The number, size, and configuration of the servers in the environment depends on your unstructured data size, pipeline workload, and performance expectations. This topic provides guidance on determining the infrastructure to deploy as well as the requirements for each of the AU components. For an introduction to the AU architecture and pipeline process, see [Anzo Unstructured Data Onboarding Process](#).

AU requires two programs that are installed separately from Anzo:

- An Anzo Unstructured cluster for processing the incoming data. See [Anzo Unstructured Cluster Requirements and Recommendations](#).
- Elasticsearch for indexing and searching unstructured document contents. See [Elasticsearch Requirements and Recommendations](#).

Anzo Unstructured Cluster Requirements and Recommendations

An Anzo Unstructured (AU) cluster consists of one Leader instance and one or more Worker instances. Cambridge Semantics provides an installation script for installing the AU software. In an AU cluster:

- The **Leader** instance is a lightweight program and is typically installed on the Anzo host server.
- The **Worker** instances require significant resources to process the unstructured documents and are typically installed on dedicated servers.

Consider the size of your unstructured data workload when deploying Worker host servers. Each Worker instance can have multiple server instances to process documents. The table below lists the requirements for Anzo Unstructured Worker servers:

Component	Requirement
Operating System	RHEL/CentOS 7.5+ Cambridge Semantics recommends that you tune the ulimits for your Linux distribution to increase the limits for certain resources. See Configure User Resource Limits for more information.

Component	Requirement
CPU	4+ cores
RAM	16+ GB
Disk Space	10+ GB
File System	The Anzo file store (shared file system) must be accessible from each AU server in the cluster. For more information about the shared file system, see Deploying the Shared File System .

Note

Do not run any other software, including anti-virus software, on the Anzo Unstructured Worker servers. Additional programs running on the Worker nodes may severely impact the performance of Unstructured Pipelines.

For instructions on installing Anzo Unstructured, see [Installing Anzo Unstructured](#).

Elasticsearch Requirements and Recommendations

Anzo Unstructured uses the Elasticsearch engine to build an index after an unstructured pipeline runs and for running searches on unstructured data that is onboarded to Anzo. When choosing an Elasticsearch host server, consider the following information:

- Generating the index is a lightweight operation compared to document search operations. If you have a light unstructured data workload and do not perform text searches on large amounts of data, installing an Elasticsearch engine on the Anzo host server might be sufficient.
- If you onboard a large number of unstructured documents and plan to perform text searches across a large amount of data, Cambridge Semantics recommends that you install Elasticsearch on a dedicated server.

The table below list the Elasticsearch server requirements:

Component	Requirement
Elasticsearch Version	7.10.2
	<div> Note Elasticsearch versions 7.11 and 7.12 are not supported at this time. </div>

Component	Requirement
CPU	8+ cores
RAM	64+ GB
Disk Space	100+ GB
Ports	By default, the port range for Elasticsearch requests (http.port) is 9200-9300 . If port 9200 is not available when Elasticsearch is started, Elasticsearch tries 9201 and so on until it finds an accessible port. The Anzo server and the AnzoGraph leader server need to be able to access Elasticsearch on the HTTP request port that Elasticsearch uses.
File System	The Anzo file store (shared file system) must be accessible from each Elasticsearch server. For more information about the shared file system, see Deploying the Shared File System .

For instructions on installing Elasticsearch, see [Installing and Configuring Elasticsearch](#).

Related Topics

[Anzo Unstructured Overview](#)

[Anzo Unstructured Data Onboarding Process](#)

[Installing Anzo Unstructured](#)

[Installing and Configuring Elasticsearch](#)

[Upgrading Anzo Unstructured](#)

Installing Anzo Unstructured

This topic provides instructions for deploying an Anzo Distributed Unstructured cluster.

Tip See [Anzo Unstructured Requirements](#) for details about server requirements.

1. [Complete the Pre-Installation Configuration](#)
2. [Deploy the Leader Node](#)
3. [Deploy the Worker Nodes](#)
4. [Complete the Post-Installation Configuration](#)

Complete the Pre-Installation Configuration

- [Configure User Resource Limits](#)
- [Use the Anzo Service User Account when Installing AU](#)

Configure User Resource Limits

Before installing Anzo Unstructured, Cambridge Semantics recommends that you tune the user resource limits (ulimits) for your Linux distribution to increase the limits for the following resources. Tune ulimits on all AU host servers in the cluster:

- Increase the limit for the following resources to at least **65535**:
 - open files (nofile)
 - max user processes (nproc)
- Increase the limit for the following resources to **infinity**:
 - address space (as)
 - CPU time (cpu)
 - file locks (locks)
 - file size (fsize)
 - max memory size (memlock)

To view the current ulimits, run `ulimit -a`. To permanently change ulimits, modify the `/etc/security/limits.conf` file. For information, see [How to set ulimit values](#) in the RHEL support documentation.

Note

Typically, as part of post-installation configuration, a systemd service is set up to start and stop the Leader and Worker processes. When systemd starts a process, however, it uses the limits that are defined in the systemd service rather than the limits in `/etc/security/limits.conf`. In addition to changing the ulimits in `limits.conf`, it is important to set the limits in the Leader and Worker services. The service file contents shown in [Complete the Post-Installation Configuration](#) includes the recommended ulimit settings.

Use the Anzo Service User Account when Installing AU

Important

Since the Anzo Unstructured cluster will access the shared file store, it is important to install and run the software with the same service account that runs Anzo. For more information, see [Anzo Service Account Requirements](#).

Deploy the Leader Node

Follow the instructions below to deploy the Anzo Distributed Unstructured (DU) Leader node.

1. Make sure that the Leader host server has access to the Anzo shared file system and meets the requirements in [Anzo Unstructured Cluster Requirements and Recommendations](#).
2. Copy the Anzo DU installation script to the Leader server and then run the following command to make the script executable:

```
chmod +x <script_name>
```

3. If necessary, run the following command to become the Anzo service user:

```
su <name>
```

Where <name> is the name of the service user. For example:

```
su anzo
```

4. Run the following command to start the installation wizard:

```
./<script_name>
```

The script unpacks the JRE and then waits for input before starting the installation.

5. Press **Enter** to start the installation.
6. Review the software license agreement. Press **Enter** to scroll through the terms. At the end of the agreement, type **1** to accept the terms or type **2** to disagree and stop the installation.
7. At the prompt that asks which components to install, type **1** (Leader) and then press **Enter**.
8. Specify the directory to install Anzo DU. Press **Enter** to accept the default installation path or type an alternate path and then press **Enter**.
9. The wizard prompts for the IP address of this leader instance. The wizard defaults to the IP address of the server. Press **Enter** to accept the default value. If necessary, type a different IP address, and then press **Enter**.
10. The wizard prompts for any additional leader node IP addresses. Typically there is one leader node and this value is specified as the same IP address as the previous step. If you set up additional leader nodes for redundancy, however, enter a comma separated list of the alternate nodes. Otherwise, accept the default value and press **Enter**.
11. Specify the maximum amount of memory (in MB) that this leader instance can use. The install wizard lists the total RAM available and chooses 1/2 of the total memory as the default value. Adjust the value as needed or accept the default value and then press **Enter**.
12. The wizard proceeds to install Anzo DU according to the values that you specified. Proceed to [Deploy the Worker Nodes](#) to install the Worker instances.

Deploy the Worker Nodes

Follow the instructions below to deploy the Anzo Distributed Unstructured (DU) Worker nodes.

1. Make sure that the Worker host servers have access to the Anzo shared file system and meet the requirements in [Anzo Unstructured Cluster Requirements and Recommendations](#).
2. Copy the Anzo DU installation script to each of the Worker servers and then run the following command to make the script executable:

```
chmod +x <script_name>
```

3. If necessary, run the following command to become the Anzo service user:

```
su <name>
```

Where <name> is the name of the service user. For example:

```
su anzo
```

4. Run the following command to start the installation wizard:

```
./<script_name>
```

The script unpacks the JRE and then waits for input before starting the installation.

5. Press **Enter** to start the installation.
6. Review the software license agreement. Press **Enter** to scroll through the terms. At the end of the agreement, type **1** to accept the terms or type **2** to disagree and stop the installation.
7. At the prompt that asks which components to install, type **2** (Worker) and then press **Enter**.
8. Specify the directory to install Anzo DU. Press **Enter** to accept the default installation path or type an alternate path and then press **Enter**.
9. The wizard prompts for the IP address to use for this worker node. The wizard defaults to the IP address of the server. Press **Enter** to accept the default value. If necessary, type a different IP address, and then press **Enter**.
10. The wizard prompts you to specify the maximum number of service instances for this Worker node. Each service instance processes one unstructured document at a time. The default value is 2 instances. Press **Enter** to accept the default or specify another value and then press **Enter**.
11. Specify the port to use for this Worker. The wizard defaults to port **2552**. Press **Enter** to accept the default value or type a different port and then press **Enter**.
12. The wizard prompts you to enter the IP address of the Leader node. Specify the IP address for the Leader instance that you deployed in the procedure above. If you deployed multiple Leader nodes, specify each Leader's IP address in a comma separated list.
13. Specify the maximum amount of memory (in MB) that this Worker instance can use. The install wizard lists the total RAM available and chooses 1/2 of the total memory as the default value. Adjust the value as needed or accept the default value and then press **Enter**.

The wizard proceeds to install Anzo DU according to the values that you specified.

- Repeat the steps above for each Worker instance in the cluster.

Once the Leader and all of the Worker nodes are installed, proceed to [Complete the Post-Installation Configuration](#) to complete the initial configuration and start the software.

Note

If you upgraded the Anzo Unstructured software, make sure that you restart the Leader and Worker applications. In addition, restart the following two services in Anzo:

- Anzo Server Akka Cluster Integration
- Anzo Unstructured Distributed

Complete the Post-Installation Configuration

Once the Anzo Unstructured (AU) cluster is installed, Cambridge Semantics recommends that you set up Leader and Worker services to ensure that AU runs as the Anzo service user and can access the data that other platform components write to the shared file system. Follow the instructions in the steps below to configure the services.

Note Root user privileges are required to complete these tasks.

- [Configure and Start the Leader Service](#)
- [Configure and Start the Worker Service](#)

Configure and Start the Leader Service

Follow the instructions below to create and start the Leader service.

- On the Leader server, create a file called **anzo-du-leader.service** in the `/usr/lib/systemd/system` directory. For example:

```
# vi /usr/lib/systemd/system/anzo-du-leader.service
```

- Add the following contents to `anzo-du-leader.service`. Placeholder values are shown in **bold**:

```
[Unit]
Description=Service for Distributed Unstructured Leader
After=syslog.target network.target local-fs.target remote-fs.target nss-lookup.target

[Service]
Type=forking
RemainAfterExit=yes
LimitCPU=infinity
LimitNOFILE=65536
LimitAS=infinity
LimitNPROC=65536
```

```

LimitMEMLOCK=infinity
LimitLOCKS=infinity
LimitFSIZE=infinity
ExecStart=/install_path/leader start
ExecStop=/install_path/leader stop
User=service_user_name
Group=service_user_name

[Install]
WantedBy=default.target

```

Where **install_path** is the Anzo DU installation path and directory and **service_user_name** is the name of the Anzo service user. For example:

```

[Unit]
Description=Service for Distributed Unstructured Leader
After=syslog.target network.target local-fs.target remote-fs.target nss-lookup.target

[Service]
Type=forking
RemainAfterExit=yes
LimitCPU=infinity
LimitNOFILE=65536
LimitAS=infinity
LimitNPROC=65536
LimitMEMLOCK=infinity
LimitLOCKS=infinity
LimitFSIZE=infinity
ExecStart=/opt/AnzoDU/leader start
ExecStop=/opt/AnzoDU/leader stop
User=anzo
Group=anzo

[Install]
WantedBy=default.target

```

3. Save and close the file, and then run the following commands to start and enable the new service:

```

# systemctl start anzo-du-leader.service

# systemctl enable anzo-du-leader.service

```

Once the service is enabled, the Leader should be running. Any time you start and stop the Leader, run the following **systemctl** commands: `sudo systemctl stop anzo-du-leader` and `sudo systemctl start anzo-du-leader`.

Configure and Start the Worker Service

Follow the instructions below to create and start the Worker service. Complete the steps below on each Worker node in the cluster.

1. Create a file called **anzo-du-worker.service** in the `/usr/lib/systemd/system` directory. For example:

```
# vi /usr/lib/systemd/system/anzo-du-worker.service
```

2. Add the following contents to `anzo-du-worker.service`. Placeholder values are shown in **bold**:

```
[Unit]
Description=Service for Distributed Unstructured Worker
After=syslog.target network.target local-fs.target remote-fs.target nss-lookup.target

[Service]
Type=forking
RemainAfterExit=yes
LimitCPU=infinity
LimitNOFILE=65536
LimitAS=infinity
LimitNPROC=65536
LimitMEMLOCK=infinity
LimitLOCKS=infinity
LimitFSIZE=infinity
ExecStart=/install_path/worker start
ExecStop=/install_path/worker stop
User=service_user_name
Group=service_user_name

[Install]
WantedBy=default.target
```

Where **install_path** is the Anzo DU installation path and directory and **service_user_name** is the name of the Anzo service user. For example:

```
[Unit]
Description=Service for Distributed Unstructured Worker
After=syslog.target network.target local-fs.target remote-fs.target nss-lookup.target

[Service]
Type=forking
RemainAfterExit=yes
LimitCPU=infinity
LimitNOFILE=65536
LimitAS=infinity
```

```

LimitNPROC=65536
LimitMEMLOCK=infinity
LimitLOCKS=infinity
LimitFSIZE=infinity
ExecStart=/opt/AnzoDU/worker start
ExecStop=/opt/AnzoDU/worker stop
User=anzo
Group=anzo

[Install]
WantedBy=default.target

```

3. Save and close the file, and then run the following commands to start and enable the new service:

```

# systemctl start anzo-du-worker.service

# systemctl enable anzo-du-worker.service

```

4. Repeat the steps above for each Worker server.

Once the service is enabled, the Worker should be running. Any time you start and stop a Worker, run the following **systemctl** commands: `sudo systemctl stop anzo-du-worker` and `sudo systemctl start anzo-du-worker`.

After deploying an Anzo Unstructured cluster, you do not need to perform additional configuration in Anzo to connect to the cluster. The connection is configured automatically based on the values specified during installation. You can view the Distributed Pipeline options in **Server Settings** in the Administration application. For more information, see [Configure Network Connections to an Anzo Distributed Unstructured Cluster](#).

Important

Any time the AU Leader instance is restarted, the following two services must be restarted in Anzo:

- Anzo Server Akka Cluster Integration
- Anzo Unstructured Distributed

To restart a service:

1. In the Administration application, expand the **Servers** menu and click **Advanced Configuration**.
2. On the Advanced Configuration screen, click the **I understand and accept the risk** button to view the Anzo bundles.
3. In the **Search** field at the top of the screen, start typing the name of the service that you want to restart. When the service appears in the list onscreen, click the service name to view the details.
4. At the top of the screen, click **Stop Bundle**. Then click **Start Bundle** when the start option becomes available.

Related Topics

[Anzo Unstructured Requirements](#)

[Installing and Configuring Elasticsearch](#)

[Upgrading Anzo Unstructured](#)

Installing and Configuring Elasticsearch

This topic provides instructions for deploying Elasticsearch for use in the Anzo Unstructured environment.

Important

Elasticsearch cannot be run as the root user and must have read and write access to the Anzo file store. Therefore, it is important to install and run Elasticsearch as the Anzo service user, otherwise unstructured pipelines will fail due to permissions errors. For more information, see [Anzo Service Account Requirements](#).

1. Make sure that the Elasticsearch host server has access to the Anzo shared file system and meets the requirements in [Elasticsearch Requirements and Recommendations](#).
2. Become the Anzo service user before proceeding. If necessary, create the user on the server. For more information, see [Make Sure the Anzo Service User Account is Created](#).
3. Download Elasticsearch version 7.10.2 from the Elasticsearch [Past Releases website](#). Docker images are also available from the [Docker @ Elastic](#) website. Follow the Elasticsearch documentation to install the software.

Note Elasticsearch versions 7.11 and 7.12 are not supported at this time.

4. Configure Elasticsearch to save snapshots to the Anzo shared file system.
 - For a mounted file system, such as NFS, uncomment the Path setting, **path.repo**, in `<elasticsearch_install_path>/config/elasticsearch.yml` and specify the path and directory for the mounted file system:

```
path.repo: /<path>/<directory>
```

For example:

```
path.repo: /opt/anzoshare
```

- For S3, see [S3 Repository Plugin](#) in the Elasticsearch documentation for information about installing the S3 repository plugin. Then see [Client Settings](#) for instructions on configuring the S3 client.
- For HDFS, see [Hadoop HDFS Repository Plugin](#) in the Elasticsearch documentation for information about installing the HDFS repository plugin. Then see [Hadoop Security](#) for information about configuring Kerberos authentication.

5. Configure the amount of memory that Elasticsearch can use. By default, Elasticsearch is configured to use a maximum heap size of 1 GB. Cambridge Semantics recommends that you increase the amount to 50% of the memory that is available on the server. To change the configuration, open the `<elasticsearch_install_path>/config/jvm.options` file in an editor. At the top of the file, modify the **Xms** and **Xmx** values to replace the **1** with the new value. For example:

```
# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space

-Xms15g
-Xmx15g
```

6. If you want to secure the Elasticsearch instance, follow the instructions in [Configuring security in Elasticsearch](#) in the Elasticsearch documentation.

Important

If you set up SSL authentication with a trusted certificate, make sure that you add the certificate to the Anzo trust store. For instructions, see [Adding a Certificate to the Trust Store](#).

7. When the configuration is complete, run the following command to start Elasticsearch:

```
./<install_path>/bin/elasticsearch
```

For more information about starting Elasticsearch, see [Starting Elasticsearch](#) in the Elasticsearch documentation. For information about configuring Elasticsearch to start automatically as the Anzo user, see [Configuring an Elasticsearch Service](#) below.

Once this Elasticsearch instance is configured and running, follow the instructions in [Connecting to Elasticsearch](#) to connect Anzo to this instance.

Configuring an Elasticsearch Service

Cambridge Semantics recommends that you configure an Elasticsearch service for starting Elasticsearch automatically as the Anzo service user. Follow the instructions below to implement the service.

Note Root user privileges are required to complete this task.

1. Create a file called **es.service** in the `/usr/lib/systemd/system` directory. For example:

```
# vi /usr/lib/systemd/system/es.service
```

2. Add the following contents to es.service:

```
[Unit]
Description=elasticsearch
Wants=network-online.target
After=network-online.target
[Service]
Type=oneshot
ExecStart=/sbin/runuser -l Anzo_service_user /install_path/elasticsearch-
7.10.2/bin/elasticsearch
[Install]
WantedBy=multi-user.target
```

Where **Anzo_service_user** is the name of the Anzo service user, and **install_path** is the path to Elasticsearch.

For example:

```
[Unit]
Description=elasticsearch
Wants=network-online.target
After=network-online.target
[Service]
Type=oneshot
ExecStart=/sbin/runuser -l anzo /opt/elasticsearch-7.10.2/bin/elasticsearch
[Install]
WantedBy=multi-user.target
```

3. Save and close the file, and then run the following commands to start and enable the new service:

```
# systemctl enable es.service

# systemctl status es.service

# systemctl start es.service
```

Once the service is in place, Elasticsearch should be stopped and started via systemctl. For example, `systemctl stop es` and `systemctl start es`.

Related Topics

[Anzo Unstructured Requirements](#)

[Installing Anzo Unstructured](#)

[Connecting to Elasticsearch](#)

Upgrading Anzo Unstructured

The steps to upgrade the Anzo Unstructured (AU) software are the same as the installation instructions in [Installing Anzo Unstructured](#). When you update the existing installation, each prompt defaults to the value that is specified for

the current deployment. You can press **Enter** through the prompts to retain the existing settings. The last step in the process, however, asks if you want to overwrite files in the `<AnzoDU_install_path>/etc` directory that have been modified. Cambridge Semantics recommends that you choose **ya (Yes To All)** to overwrite all files in that directory so that important options from the version you are upgrading to are deployed to your environment. If you have customized files in the `etc` directory, create a backup copy of the directory before starting the upgrade so that you can refer to the backup files when customizing the new version.

Important

When upgrading the AU software, the Leader and Worker applications must be upgraded at the same time using the same installer so that the software versions are identical across the cluster. You cannot upgrade the Worker nodes without upgrading the Leader and vice versa.

After the upgrade, make sure that you restart the Leader and Worker applications as well as the following Anzo services:

- Anzo Server Akka Cluster Integration
- Anzo Unstructured Distributed

Related Topics

[Installing Anzo Unstructured](#)

[Installing and Configuring Elasticsearch](#)

Using K8s for Dynamic Deployments of Anzo Components

Anzo integrates with Amazon Elastic Kubernetes Service (EKS), Google Kubernetes Engine (GKE), and Azure Kubernetes Service (AKS) services to offer Kubernetes-based, dynamic deployments of AnzoGraph, Anzo Unstructured with Anzo Agent, Spark, and Elasticsearch.

The Kubernetes (K8s) integration automates the provisioning and deprovisioning of the resources and applications that support onboarding and accessing data in Anzo. In a K8s-based environment, Anzo users can activate pre-configured environments on-demand without needing specific technical, cloud platform, or infrastructure deployment skills. In addition, right-sized clusters are automatically created and deleted, avoiding the need to keep instances running indefinitely and reducing the overall cost of maintaining the applications.

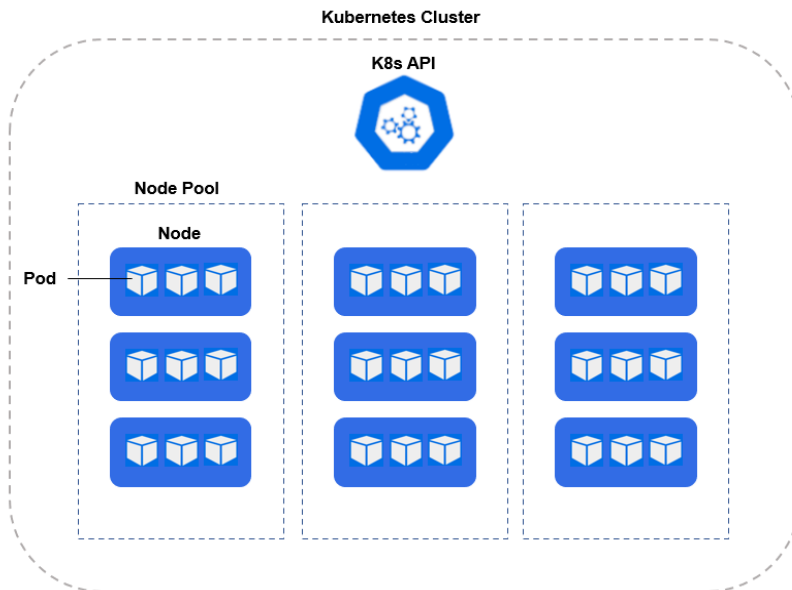
The topics in this section provide an overview of K8s concepts, general requirements for integrating K8s with Anzo, and guidance on choosing the compute instances that are ideal for hosting the Anzo applications. This section also includes instructions on deploying and configuring all of the K8s infrastructure for each of the supported cloud service providers.

- [Kubernetes Concepts](#)
- [Anzo K8s Requirements](#)
- [Compute Resource Planning](#)
- [Deploying the K8s Infrastructure](#)

Kubernetes Concepts

To set up the Kubernetes (K8s) infrastructure needed to integrate with Anzo, you use scripts that are supplied by Cambridge Semantics and the API for your preferred cloud service provider (CSP) to deploy a K8s cluster. The cluster includes a K8s API server, which manages all communication for the cluster.

In the cluster, you create a number of node pools or node groups. A **node pool** or **node group** is a group of nodes within a cluster that all have the same configuration. Different node pools are designed based on machine types and specific properties to be set on each **node**. The nodes are tuned to host a particular type of pod. A **pod** is an instance of an application, i.e., a container of images. The diagram below shows a high level view of a K8s cluster:



For example, an AnzoGraph node pool contains the type of nodes that are suitable for running pods with AnzoGraph images.

Node pools can be configured so that they are **static** or **autoscaling**. In static node pools, the nodes are deployed in the K8s cluster and remain provisioned even if they do not run an application. If a node pool is configured with an autoscaler, nodes are not deployed unless resources are requested. When the resources are no longer in use, the autoscaler deprovisions the nodes.

For more information about node pools and other requirements, see [Anzo K8s Requirements](#).

Related Topics

[Anzo K8s Requirements](#)

[Compute Resource Planning](#)

[Deploying the K8s Infrastructure](#)

Anzo K8s Requirements

This section gives an overview of the general infrastructure requirements for Anzo K8s integration. Additional software, network infrastructure, and permission-related requirements are included in the deployment instructions for each of the cloud service providers.

- [Supported Kubernetes Versions](#)
- [File Storage Requirements](#)
- [Node Pool Requirements](#)
- [Container Registry Requirements](#)

Supported Kubernetes Versions

The table below shows the supported Kubernetes (K8s) versions by Cloud Service Provider (CSP):

CSP	K8s v1.17	K8s v1.18	K8s v1.19
Amazon EKS	✓	✓	✓
Google GKE	✓	✓	✓
Azure AKS		✓	✓

File Storage Requirements

A network file system (NFS) is required for shared file storage between Anzo and the dynamic applications. You are required to create the file system. However, Anzo automatically mounts the NFS to the nodes when AnzoGraph, Anzo Unstructured, Spark, or Elasticsearch pods are deployed. See [Deploying the Shared File System](#) for more information.

Node Pool Requirements

There are three types of node pools or node groups that you are required to configure for integration with Anzo. In addition to the scripts for creating and configuring the K8s cluster, Cambridge Semantics supplies configuration files to use as templates for defining the policies for each type of node pool. The node pools can be configured as static or autoscaling.

Operator Node Pool

An Operator node pool is tuned to run operator pods. Operator pods manage the application pods and control the K8s resources of the applications that are deployed in the node pools. There is one operator for each application: AnzoGraph, Elasticsearch, Anzo Agent and Anzo Unstructured, and Spark. Anzo deploys and manages the operator pods. With the help of the operators, Anzo orchestrates the provisioning and deprovisioning of the application nodes and pods. Since the operators in the Operator node pool are required to be active at all times, operator pods are designed to be very small and use very few resources. They can be deployed on standard, small-sized cloud instances.

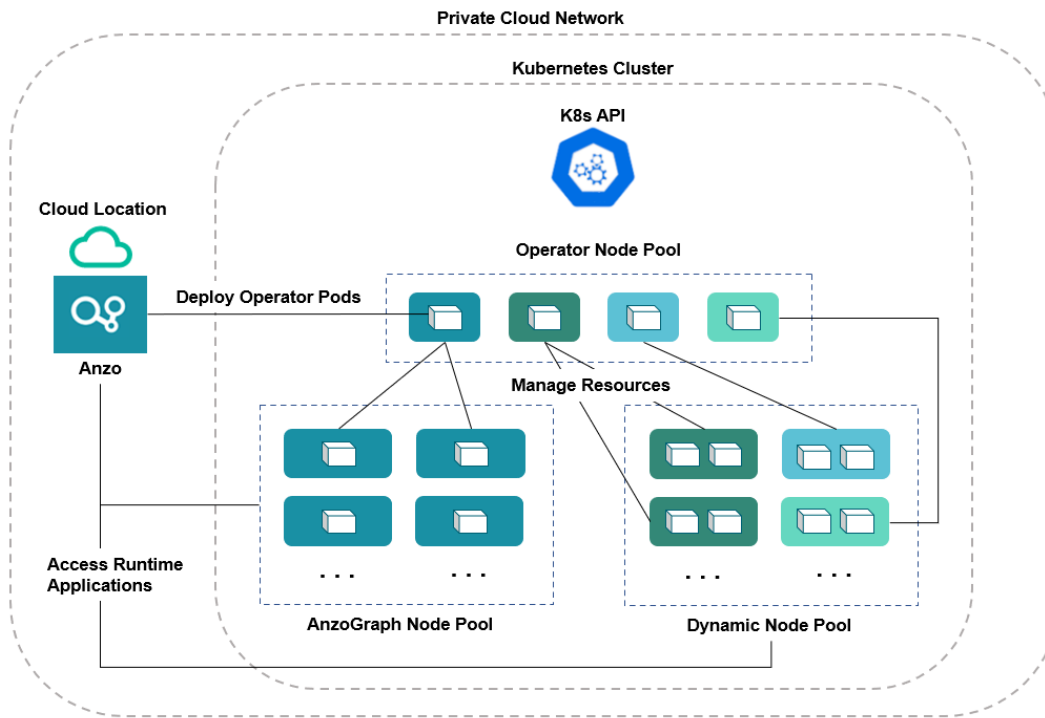
AnzoGraph Node Pool

An AnzoGraph node pool is tuned to run AnzoGraph pods. AnzoGraph node pools are typically configured to auto-scale so that nodes are not deployed unless a user requests an AnzoGraph environment for loading a graphmart or running queries against the data in a graphmart.

Dynamic Node Pool

The Dynamic node pool is tuned to run Elasticsearch, Spark, Anzo Agent, and Anzo Unstructured (AU) pods. Dynamic node pools are also typically configured to auto-scale so that nodes are not deployed unless a user requests an environment for running a structured or unstructured pipeline.

The diagram below shows the K8s cluster architecture with the required node pools.



Note

For Amazon EKS deployments, there is a fourth type of required node group. The additional type, called a Common node group, is tuned to run K8s service pods, such as Cluster Autoscalers and Load Balancers.

For guidance on choosing the instance types and sizes for the nodes in the required node pools, see [Compute Resource Planning](#).

Container Registry Requirements

You are not required to set up an internal container registry for Anzo and K8s integration. However, if your K8s cluster will not have outbound internet access for retrieving container images from the Cambridge Semantics repository, you will need to create a container registry through your Cloud Service Provider.

Related Topics

[Kubernetes Concepts](#)

[Compute Resource Planning](#)

[Deploying the K8s Infrastructure](#)

Compute Resource Planning

This section provides guidance on choosing the instance types for the nodes in your node pools.

- [Operator Nodes](#)
- [AnzoGraph Nodes](#)
- [Dynamic Nodes](#)

Operator Nodes

The operator pods are very small. Each operator requires 0.5 CPU. The table below lists the recommended instance types and sizes for a single operator. If you plan to co-locate operators on a single instance, increase CPU accordingly. For example, an instance with 4 CPU can run up to 7 operators (3.5 CPU for operator pods and 0.5 CPU for the auxiliary service).

CSP	Suggested Instance Type	vCPU	RAM	Disk
AWS	m5.large	2	8 GiB	50 GB
GCP	n1-standard-1	1	3.75 GiB	50 GB
Azure	Standard_DS2_v2	2	7 GiB	50 GB

Note

For Amazon EKS deployments, the Suggested Instance Type for Operator nodes is also recommended for nodes in the Common node group. The Common group runs K8s service pods, such as Cluster Autoscalers and Load Balancers, which are very small and require few resources.

AnzoGraph Nodes

Since AnzoGraph is a high-performance, in-memory database, RAM is generally the most critical resource to consider when determining the overall size and number of nodes to use for AnzoGraph environments. Consider the size of the data that you plan to load and then multiply that size by 3 or 4 to determine the total memory requirement. Query processing and intermediate results can temporarily consume a very large amount of memory. For more information about AnzoGraph sizing guidelines, see [Sizing Guidelines for In-Memory Storage](#).

Also, unlike Anzo Unstructured, for example, where leader and worker pods can be colocated on the same node, Cambridge Semantics recommends that only one AnzoGraph pod is run per node. The table below shows a range of cloud instances to choose from that are ideal for running AnzoGraph pods.

CSP	Suggested Instance Range	vCPU Range	RAM Range	Disk
AWS	m5.4xlarge – m5.16xlarge	8 – 64	32 GiB – 256 GiB	100 GB

CSP	Suggested Instance Range	vCPU Range	RAM Range	Disk
GCP	n1-standard-8 – n1-standard-64	8 – 64	30 GiB – 240 GiB	100 GB
Azure	DSv2 and Dsv3 series	8 – 64	28 GiB – 256 GiB	100 GB

Dynamic Nodes

Nodes in the Dynamic node pool need to be sized to run Anzo Agent pods. An Anzo Agent is a scaled down version of the Anzo server that coordinates the sending of documents to the Anzo Unstructured (AU) worker nodes. Anzo Agent pods require more resources than AU leader and worker, Elasticsearch, and Spark pods. Each unstructured pipeline deploys a single Anzo Agent pod, and the pod needs to have enough resources to coordinate the pipeline. Anzo Agent pods are typically deployed as one pod per node, while the AU worker, Elasticsearch, and Spark nodes run multiple pods per node. The table below lists the recommended instance types and sizes for running the Anzo Agent pods. The recommended instances are also sufficient for running multiple AU, Elasticsearch, and Spark pods.

CSP	Suggested Instance Type	vCPU	RAM	Disk
AWS	m5.2xlarge	8	32 GiB	100 GB
GCP	n1-standard-8	8	30 GiB	100 GB
Azure	Standard_D8s_v3	8	32 GiB	100 GB

For instructions on setting up the K8s infrastructure, see [Deploying the K8s Infrastructure](#).

Related Topics

[Kubernetes Concepts](#)

[Anzo K8s Requirements](#)

[Deploying the K8s Infrastructure](#)

Deploying the K8s Infrastructure

To get started on setting up the K8s infrastructure to support dynamic deployments of Anzo components, see the deployment instructions for your cloud service provider:

- For **Amazon Web Services**, see [Amazon EKS Deployments](#).
- For **Google Cloud Platform**, see [Google Kubernetes Engine Deployments](#).
- For **Microsoft Azure Cloud**, see [Azure Kubernetes Service Deployments](#).



Related Topics

- [Kubernetes Concepts](#)
- [Anzo K8s Requirements](#)
- [Compute Resource Planning](#)

Amazon EKS Deployments

The topics in this section guide you through the process of deploying all of the Amazon Elastic Kubernetes Service (EKS) infrastructure that is required to support dynamic deployments of Anzo components. The topics provide instructions for setting up a workstation to use for deploying the K8s infrastructure, performing the prerequisite tasks before deploying the EKS cluster, creating the EKS cluster, and creating the required node groups.

- [Setting Up a Workstation](#)
- [Planning the Anzo and EKS Network Architecture](#)
- [Creating and Assigning IAM Policies](#)
- [Creating the EKS Cluster](#)
- [Creating the Required Node Groups](#)

Setting Up a Workstation

This topic provides the requirements and instructions to follow for configuring a workstation to use for creating and managing the EKS infrastructure. The workstation needs to be able to connect to the AWS API. It also needs to have the required AWS and Kubernetes (K8s) software packages as well as the deployment scripts and configuration files supplied by Cambridge Semantics. This workstation will be used to connect to the AWS API and provision the K8s cluster and node groups.

Note

You can use the Anzo server as the workstation if the network routing and security policies permit the Anzo server to access the AWS and K8s APIs. When deciding whether to use the Anzo server as the K8s workstation, consider whether Anzo may be migrated to a different server or VPC in the future.

- [Review the Requirements and Install the Software](#)
- [Download the Cluster Creation Scripts and Configuration Files](#)

Review the Requirements and Install the Software

Component	Requirement
Operating System	The operating system for the workstation must be RHEL/CentOS 7.8 or later .

Component	Requirement
Networking	The workstation should be in the same VPC as the EKS cluster. If it is not in the same VPC, make sure that it is on a network that is routable from the cluster's VPC.
Software	<ul style="list-style-type: none"> • AWS-CLI Version 2 is recommended. Version 1.16.156 or later is supported. For instructions, see Install AWS-CLI below. • EKSCTL Version 0.40.0 or later is required. For instructions, see Install EKSCTL below. • Kubectl Versions 1.17 – 1.19 are supported. Cambridge Semantics recommends that you use the same kubectl version as the EKS cluster version. For instructions, see Install Kubectl below.
CSI EKSCTL Package	Cambridge Semantics provides eksctl scripts and configuration files to use for provisioning the EKS cluster and node groups. Download the files to the workstation. See Download the Cluster Creation Scripts and Configuration Files for more information about the eksctl package.

Install AWS-CLI

AWS CLI is the AWS command line interface. Version 2 is recommended. Follow the instructions below to install the latest aws-cli version 2 package. For more information, see [Installing, Updating, and Uninstalling the AWS CLI Version 2 on Linux](#) in the AWS CLI documentation.

1. Run the following command to download the latest aws-cli package to the current directory:

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

2. Run the following command to unzip the package:

```
unzip awscliv2.zip
```

3. Then run the following command to run the install program. By default, the files are all installed to `/usr/local/aws-cli`, and a symbolic link is created in `/usr/local/bin`.

```
sudo ./aws/install
```

Install EKSCTL

EKSCTL is the AWS EKS command line interface. Version 0.40.0 or later is required. Follow the instructions below to download the eksctl package and place it in the `/usr/local/bin` directory. For more information, see [Installing eksctl](#) in the Amazon EKS documentation.

1. Run the following command to download the eksctl package to the /tmp directory:

```
curl --silent --location
"https://github.com/weaveworks/eksctl/releases/download/<tag>/eksctl_$(uname -s)_
amd64.tar.gz" | tar xz -C /tmp
```

Where <tag> is the release that you want to download. For example:

```
curl --silent --location
"https://github.com/weaveworks/eksctl/releases/download/0.40.0/eksctl_$(uname -s)_
amd64.tar.gz" | tar xz -C /tmp
```

2. Then run the following command to move eksctl to the /usr/local/bin directory:

```
sudo mv /tmp/eksctl /usr/local/bin
```

Install Kubectl

Follow the instructions below to install kubectl on your workstation. Cambridge Semantics recommends that you install the same version of kubectl as the K8s cluster API. For more information, see [Install and Set Up kubectl on Linux](#) in the Kubernetes documentation.

1. Run the following cURL command to download the kubectl binary:

```
curl -LO https://dl.k8s.io/release/<version>/bin/linux/amd64/kubectl
```

Where <version> is the version of kubectl to install. For example, the following command downloads version 1.19.12:

```
curl -LO https://dl.k8s.io/release/v1.19.12/bin/linux/amd64/kubectl
```

2. Run the following command to make the binary executable:

```
chmod +x ./kubectl
```

3. Run the following command to move the binary to your PATH:

```
sudo mv ./kubectl /usr/local/bin/kubectl
```

4. To confirm that the binary is installed and that you can run kubectl commands, run the following command to display the client version:

```
kubectl version --client
```

The command returns the following type of information. For example:

```
Client Version: version.Info{Major:"1", Minor:"19", GitVersion:"v1.19.12",
GitCommit:"f3abc15296f3a3f54e4ee42e830c61047b13895f",
GitTreeState:"clean", BuildDate:"2021-06-16T13:21:12Z", GoVersion:"go1.13.15",
Compiler:"gc", Platform:"linux/amd64"}
```

Download the Cluster Creation Scripts and Configuration Files

The Cambridge Semantics GitHub repository, [k8s-genesis](https://github.com/cambridgesemantics/k8s-genesis) (<https://github.com/cambridgesemantics/k8s-genesis.git>), includes all of the files that are needed to manage the configuration, creation, and deletion of the EKS cluster and node groups.

You can clone the repository to any location on the workstation or download the k8s-genesis package as a ZIP file, copy the file to the workstation, and extract the contents. The k8s-genesis directory includes three subdirectories (one for each supported Cloud Service Provider), the license information, and a readme file:

```
k8s-genesis
├─ aws
├─ azure
├─ gcp
├─ LICENSE
└─ README.md
```

Navigate to `/aws/k8s/eksctl`. The **eksctl** directory contains all of the EKS cluster and node group configuration files. You can remove all other directories from the workstation. The eksctl files and subdirectories are shown below:

```
eksctl
├─ aws_cli_common.sh
├─ common.sh
├─ conf.d
│   ├── iam_serviceaccounts.yaml
│   ├── k8s_cluster.conf
│   ├── nodepool_anzograph.yaml
│   ├── nodepool_common.yaml
│   ├── nodepool_dynamic.yaml
│   ├── nodepool_operator.yaml
│   └─ nodepool.yaml
├─ create_k8s.sh
├─ create_nodepools.sh
├─ delete_k8s.sh
├─ delete_nodepools.sh
├─ README.md
├─ reference
│   ├── ca_autodiscover-patch-file.yaml
│   ├── ca_autodiscover.yaml
│   ├── cluster-autoscaler-policy.json
│   └─ nodepool_anzograph_tuner.yaml
```

```

|   ├── nodepool_dynamic_tuner.yaml
|   ├── versions
|   └── warm_ip_target.yaml
└── sample_use_cases
    ├── 1_existing_vpc_private_cluster
    │   └── k8s_cluster.conf
    ├── 2_new_vpc_public_cluster
    │   └── k8s_cluster.conf
    └── 3_nat_ha_private_cluster
        └── k8s_cluster.conf

```

The following list gives an overview of the files. Subsequent topics describe the files in more detail.

- The **aws-cli-common.sh** and **common.sh** scripts are used by the **create*.sh** and **delete*.sh** scripts during EKS cluster and node group creation and deletion.
- The **conf.d** directory contains the configuration files that supply the specifications to follow when creating the EKS cluster and node groups.
 - **iam_serviceaccounts.yaml**: Supplies optional IAM roles for Service Account specifications for use as part of cluster creation if you would like to assign permissions for the applications that run on EKS.
 - **k8s_cluster.conf**: Supplies the specifications for the EKS cluster.
 - **nodepool_anzograph.yaml**: Supplies the specifications for the AnzoGraph node group.
 - **nodepool_common.yaml**: Supplies the specifications for the Common node group.
 - **nodepool_dynamic.yaml**: Supplies the specifications for the Dynamic node group.
 - **nodepool_operator.yaml**: Supplies the specifications for the Operator node group.
 - **nodepool.yaml**: This file is supplied as a reference. It contains the superset of node group parameters and includes comments that provide additional information.
- The **create_k8s.sh** script is used to deploy the EKS cluster.
- The **create_nodepools.sh** script is used to deploy node groups in the EKS cluster.
- The **delete_k8s.sh** script is used to delete the EKS cluster.
- The **delete_nodepools.sh** script is used to remove node groups from the EKS cluster.
- The **reference** directory contains crucial files that are referenced by the cluster and node group creation scripts. The files in the directory should not be edited, and the **reference** directory must exist on the workstation at the same level as the **create*.sh** and **delete*.sh** scripts.
- The **sample_use_cases** directory contains sample EKS cluster configuration files that you can refer to or use as a template for configuring your EKS cluster depending on your use case:
 - The **k8s_cluster.conf** file in the **1_existing_vpc_private_cluster** directory is a sample file for a use case where you want to deploy the EKS cluster in an existing VPC that does not have public internet access.

- The `k8s_cluster.conf` file in the `2_new_vpc_public_cluster` directory is a sample file for a use case where you want to deploy the EKS cluster into a new VPC with public internet access that is restricted to specific IP ranges.
- The `k8s_cluster.conf` file in the `3_nat_ha_private_cluster` directory is a sample file for a use case where you want to create a private EKS cluster in an existing VPC and deploy highly available NAT gateways.

Once the workstation is configured, see [Planning the Anzo and EKS Network Architecture](#) to review information about the network architecture that the `eksctl` scripts create. And see [Creating and Assigning IAM Policies](#) for instructions on creating the IAM policies that are needed for assigning permissions to create and use the EKS cluster.

Related Topics

[Planning the Anzo and EKS Network Architecture](#)

[Creating and Assigning IAM Policies](#)

[Creating the EKS Cluster](#)

[Creating the Required Node Groups](#)

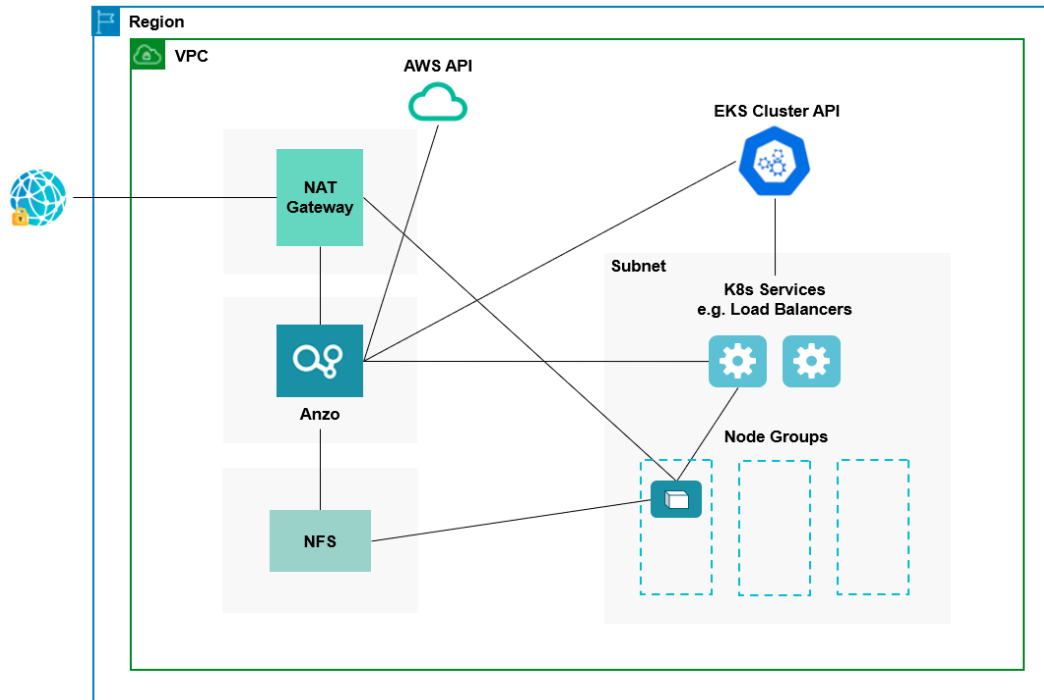
Planning the Anzo and EKS Network Architecture

This topic describes the network architecture that supports the Anzo and EKS integration.

Note

When you deploy the K8s infrastructure, Cambridge Semantics strongly recommends that you create the EKS cluster in the same VPC as Anzo. If you create the cluster in a new VPC, you must configure the new VPC to be routable from the Anzo VPC.

The diagram below shows the typical network components that are employed when an EKS cluster is integrated with Anzo. Most of the network resources shown in the diagram are automatically deployed (and the appropriate routing is configured) according to the values that you supply in the cluster and node group `.conf` files in the `eksctl` package on the workstation.



In the diagram, there are two components that you deploy before configuring and creating the K8s resources:

- **Anzo:** Since the Anzo server is typically deployed before the K8s components, you specify the Anzo VPC ID when creating the EKS cluster, ensuring that Anzo and all of the EKS cluster components are in the same network and can talk to each other. Also, make sure that Anzo has access to the AWS and EKS APIs.
- **NFS:** You are required to create a network file system (NFS). However, Anzo automatically mounts the NFS to the nodes when AnzoGraph, Anzo Unstructured, Spark, and Elasticsearch pods are deployed so that all of the applications can share files. See [Deploying the Shared File System](#) for more information. The NFS does not need to have its own subnet but it can.

The rest of the components in the diagram are automatically provisioned, depending on your specifications, when the EKS cluster and node groups are created. The eksctl scripts can be used to create NAT gateways and subnets for outbound internet access, such as for pulling container images from the Cambridge Semantics repository. In addition, the scripts create a subnet for the K8s services and node groups and configure the routing so that Anzo can communicate with the K8s services and the services can talk to the pods that are deployed in the node groups.

Tip

When considering the network requirements of your organization and planning how to integrate the new K8s infrastructure in accordance with those requirements, it may help to consider the following types of use cases. Cambridge Semantics supplies sample cluster configuration files in the `eksctl/sample_use_cases` directory that are tailored for each of these use cases:

- **Deploy a private EKS cluster in an existing VPC (i.e., the same VPC as Anzo)**

In this use case, the EKS cluster is deployed in a private subnet in your existing VPC. And a new (or existing, if you have one) NAT gateway is used to enable access to external services that are outside of the VPC. The control plane security group is configured to allow access only from certain CIDRs, and communication through VPN can be enabled to allow a virtual private gateway to automatically propagate routes to the route tables.

- **Deploy a public EKS cluster in a new VPC**

In this use case, a new VPC is created with the specified CIDR. A new NAT gateway is deployed to provide outbound connectivity for the cluster nodes. Public and private subnets are also created, and public access is restricted to specific IP ranges. The new VPC will need to be configured so that it is routable from Anzo.

- **Deploy a private, highly available EKS cluster in an existing VPC**

In this use case (like the first case listed above) a private EKS cluster is deployed in an existing VPC. In addition, NAT gateways are created in each of the cluster's Availability Zones, making the cluster highly available.

For a summary of the files in the eksctl directory, see [Download the Cluster Creation Scripts and Configuration Files](#). Specifics about the parameters in the sample files are included in [Creating the EKS Cluster](#).

To get started on creating the EKS infrastructure, see [Creating and Assigning IAM Policies](#) for instructions on creating the IAM policies that are needed for assigning permissions to create and use the EKS cluster.

Related Topics

[Setting Up a Workstation](#)

[Creating and Assigning IAM Policies](#)

[Creating the EKS Cluster](#)

[Creating the Required Node Groups](#)

Creating and Assigning IAM Policies

There are two custom Identity and Access Management (IAM) policies that need to be created in AWS to grant the necessary permissions to the following two types of EKS users:

1. The first type of user is the user who accesses AWS services to set up the K8s infrastructure, i.e., the user who configures, creates, and maintains the EKS cluster and node groups. This policy is called the **EKS Cluster Admin**.

- The second type of user is the user who connects to the EKS cluster and deploys the dynamic Anzo applications. Typically this user is Anzo. Since Anzo communicates to the K8s services that provision the applications, the Anzo service account needs to be granted certain privileges. This user role is called the **EKS Cluster Developer**.

Note

The enterprise-level Anzo service account is a requirement for the Anzo installation and is typically in place before Anzo is installed. For more information, see [Anzo Service Account Requirements](#).

This topic provides instructions for creating the two policies and gives guidance on attaching the policies to the appropriate users or roles.

- [Create and Assign the EKS Cluster Admin Policy](#)
- [Create and Assign the EKS Cluster Developer Policy](#)

Create and Assign the EKS Cluster Admin Policy

The following IAM policy applies the minimum permissions needed for an EKS cluster administrator who will create and manage the cluster and node groups. Follow the steps below to create the policy and attach it to the appropriate principal.

- Refer to [Creating IAM Policies](#) in the AWS documentation to create the following policy using your preferred method. You can save the contents below as a JSON file on your workstation and use the AWS CLI to create the policy, or you can paste the contents on the JSON tab if you use the IAM console.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPermissions",
      "Effect": "Allow",
      "Action": [
        "iam:GetInstanceProfile",
        "iam:CreateInstanceProfile",
        "iam:AddRoleToInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile",
        "iam>DeleteInstanceProfile",
        "iam:GetRole",
        "iam:CreateRole",
        "iam:TagRole",
        "iam:PassRole",
        "iam:GetRolePolicy",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy",

```

```

        "iam:DetachRolePolicy",
        "iam>DeleteRolePolicy",
        "iam:UntagRole",
        "iam>DeleteRole"
    ],
    "Resource": "*"
},
{
    "Sid": "ComputeAndEKS",
    "Effect": "Allow",
    "Action": [
        "autoscaling:*",
        "cloudformation:*",
        "elasticloadbalancing:*",
        "ec2:*",
        "eks:*"
    ],
    "Resource": "*"
},
{
    "Sid": "ECRPushPull",
    "Effect": "Allow",
    "Action": [
        "ecr:CompleteLayerUpload",
        "ecr:DescribeImages",
        "ecr:GetAuthorizationToken",
        "ecr:DescribeRepositories",
        "ecr:UploadLayerPart",
        "ecr:InitiateLayerUpload",
        "ecr:BatchCheckLayerAvailability",
        "ecr:PutImage"
    ],
    "Resource": "*"
}
]
}

```

2. Once the policy has been created, attach the policy to any principal that will be used to configure, create, and maintain the EKS cluster and node groups. For instructions on attaching policies, see [Adding and removing IAM identity permissions](#) in the AWS Identity and Access Management User Guide.

Create and Assign the EKS Cluster Developer Policy

The following IAM policy applies the minimum permissions needed for an EKS cluster developer. Follow the steps below to create the policy and attach it to the Anzo service account.

1. Refer to [Creating IAM Policies](#) in the AWS Identity and Access Management User Guide to create the following policy using your preferred method. You can save the contents below as a JSON file on your workstation and use the AWS CLI to create the policy, or you can paste the contents on the JSON tab if you use the IAM console.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Compute",
      "Effect": "Allow",
      "Action": [
        "ec2:*",
        "elasticloadbalancing:*",
        "autoscaling:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Pricing",
      "Effect": "Allow",
      "Action": [
        "pricing:GetProducts"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EKSListAndDescribe",
      "Effect": "Allow",
      "Action": [
        "eks:ListUpdates",
        "eks:DescribeCluster",
        "eks:ListClusters"
      ],
      "Resource": "arn:aws:eks:*:*:cluster/*"
    },
    {
      "Sid": "ECRPull",
      "Effect": "Allow",
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:GetAuthorizationToken",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ],
      "Resource": "*"
    }
  ]
}
```

```

    }
  ]
}
```

2. Once the policy has been created, attach the policy to the Anzo service user so that Anzo has permission to connect to the EKS services and deploy application pods. For instructions on attaching policies, see [Adding and removing IAM identity permissions](#) in the AWS Identity and Access Management User Guide.

Once the IAM policies are in place and attached to principals, proceed to [Creating the EKS Cluster](#) for instructions on configuring and creating the cluster.

Related Topics

[Creating the EKS Cluster](#)

[Creating the Required Node Groups](#)

Creating the EKS Cluster

Follow the instructions below to define the EKS cluster resource requirements and then create the cluster based on your specifications.

- [Define the EKS Cluster Requirements](#)
- [\(Optional\) Define the IAM Role for K8s Service Accounts Requirements](#)
- [Create the EKS Cluster](#)

Define the EKS Cluster Requirements

The first step in creating the K8s cluster is to define the infrastructure specifications. The configuration file to use for defining the specifications is called **k8s_cluster.conf**. Multiple sample **k8s_cluster.conf** files are included in the **eksctl** directory. Any of them can be copied and used as templates, or the files can be edited directly.

Sample k8s_cluster.conf Files

To help guide you in choosing the appropriate template for your use case, this section describes each of the sample files. Details about the parameters in the sample files are included in [Cluster Parameters](#) below.

eksctl/conf.d/k8s_cluster.conf

This file is a non-specific use case. It includes sample values for all of the available cluster parameters.

eksctl/sample_use_cases/1_existing_vpc_private_cluster/k8s_cluster.conf

This file includes sample values for a use case where:

- The EKS cluster will be deployed in a new private subnet in an existing VPC. You specify the existing VPC ID in the **VPC_ID** parameter.
- A NAT gateway is deployed to enable access to external services. If your VPC has an existing NAT gateway that you want to use, you can specify the CIDR for the existing gateway in the **NAT_SUBNET_CIDRS** parameter.

- The control plane security group is configured to allow access only from certain CIDRs. Those CIDRs are specified in the `ALLOW_NETWORK_CIDRS` parameter.
- Communication through your VPN can be enabled and routes can automatically be propagated to the route tables by including `ENABLE_ROUTE_PROPAGATION=true`.

eksctl/sample_use_cases/2_new_vpc_public_cluster/k8s_cluster.conf

This file includes sample values for a use case where:

- A new VPC will be created and the EKS cluster will be deployed into it. You specify the CIDR for the new VPC in the `VPC_CIDR` parameter.
- A NAT gateway is deployed to enable outbound connectivity for the cluster nodes.
- Public and private subnets will be created in the new VPC based on the CIDRs specified in the `PUBLIC_SUBNET_CIDRS` and `PRIVATE_SUBNET_CIDRS` parameters.
- Public access can be restricted to certain IP ranges by specifying the allowed CIDRs in the `ALLOW_NETWORK_CIDRS` parameter.
- Since a new VPC is created (rather than creating the cluster in the same VPC as Anzo) the new VPC must be configured to allow access from Anzo.

eksctl/sample_use_cases/3_nat_ha_private_cluster/k8s_cluster.conf

Like the `1_existing_vpc_private_cluster` sample file described above, this file includes sample values for a use case where:

- The EKS cluster will be deployed in a new private subnet in an existing VPC. You specify the existing VPC ID in the `VPC_ID` parameter.
- Multiple NAT gateways will be created, making the cluster highly available. One NAT gateway is deployed in each Availability Zone specified in the `AvailabilityZones` parameter. And a CIDR for each gateway needs to be specified in the `NAT_SUBNET_CIDRS` parameter. In addition, `VPC_NAT_MODE="HighlyAvailable"`.
- The control plane security group is configured to allow access only from certain CIDRs. Those CIDRs are specified in the `ALLOW_NETWORK_CIDRS` parameter.
- Communication through VPN is enabled to automatically propagate routes to the route tables by including `ENABLE_ROUTE_PROPAGATION=true`.

Cluster Parameters

The contents of `k8s_cluster.conf` are shown below. Descriptions of the cluster parameters follow the contents.

```
# AWS Configuration parameters
REGION="<region>"
AvailabilityZones="<zones>"
TAGS="<tags>"

# Networking configuration
```

```

VPC_ID="<vpc-id>"
VPC_CIDR="<vpc-cidr>"
NAT_SUBNET_CIDRS="<nat-subnet-cidr>"
PUBLIC_SUBNET_CIDRS="<public-subnet-cidr>"
PRIVATE_SUBNET_CIDRS="<private-subnet-cidr>"
VPC_NAT_MODE="<nat-mode>"
WARM_IP_TARGET="<warm-ip-target>"
PUBLIC_ACCESS_CIDRS="<public-access-cidrs>"
ALLOW_NETWORK_CIDRS="<allow-network-cidrs>"
ENABLE_ROUTE_PROPAGATION=<enable-route-propagation>

# EKS control plane configuration
CLUSTER_NAME="<name>"
CLUSTER_VERSION="<version>"
ENABLE_PRIVATE_ACCESS=<resources-vpc-config endpointPrivateAccess>
ENABLE_PUBLIC_ACCESS=<resources-vpc-config endpointPublicAccess>
CNI_VERSION="<cni-version>"

# Logging types: ["api","audit","authenticator","controllerManager","scheduler"]
ENABLE_LOGGING_TYPES="<logging-types>"
DISABLE_LOGGING_TYPES="<logging-types>"

# Common parameters
WAIT_DURATION=<wait-duration>
WAIT_INTERVAL=<wait-interval>
STACK_CREATION_TIMEOUT="<timeout>"

```

REGION

The AWS region for the EKS cluster. For example, **us-east-1**.

AvailabilityZones

A space-separated list of each of the Availability Zones in which you want to make the EKS cluster highly available. To ensure that the AWS EKS service can maintain high availability, you can list up to three Availability Zones. For example, **us-east-1a us-east-1b**.

TAGS

A comma-separated list of any labels that you want to add to the EKS cluster resources. Tags are optional key/value pairs that you define for categorizing resources.

VPC_ID

The ID of the VPC to provision the cluster into. Typically this value is the ID for the VPC that Anzo is deployed in. For example, **vpc-0dd06b24c819ec3e5**.

Note

If you want eksctl to create a new VPC, you can leave this value blank. However, after deploying the EKS cluster, you must configure the new VPC to make it routable from the Anzo VPC.

VPC_CIDR

The CIDR block to use for the VPC. For example, **10.107.0.0/16**.

Note

Supply this value even if VPC_ID is not set and a new VPC will be created.

NAT_SUBNET_CIDRS

A space-separated list of the CIDR blocks for the public subnets that will be used by the NAT gateway. For example, **10.107.0.0/24 10.107.5.0/24**.

Note

The number of CIDR blocks should equal the number of specified [AvailabilityZones](#) if you want the NAT gateway to be highly available.

PUBLIC_SUBNET_CIDRS

A space-separated list of the CIDR blocks for the public subnets. For example, **10.107.1.0/24 10.107.2.0/24**. For a private cluster, leave this value blank.

PRIVATE_SUBNET_CIDRS

A space-separated list of the CIDR blocks for the private subnets. For example, **10.107.3.0/24 10.107.4.0/24**.

VPC_NAT_MODE

The NAT mode for the VPC. Valid values are "HighlyAvailable," "Single," or "Disable." When this value is **HighlyAvailable** and multiple Availability Zones are specified in [AvailabilityZones](#), a NAT gateway is deployed in each zone.

WARM_IP_TARGET

Specifies the "warm pool" or number of free IP addresses to keep available for pod assignment on each node so that there is less time spent waiting for IP addresses to be assigned when a pod is scheduled. Cambridge Semantics recommends that you set this value to **8**.

PUBLIC_ACCESS_CIDRS

A comma-separated list of the CIDR blocks that can access the K8s API server over the public endpoint.

ALLOW_NETWORK_CIDRS

A comma-separated list of the CIDR blocks that can access the K8s API over port 443.

ENABLE_ROUTE_PROPAGATION

Indicates whether to allow the virtual private gateway to automatically propagate routes to the route tables. This feature is useful when the cluster subnets need access to intranet/VPN routes.

CLUSTER_NAME

Name to give the EKS cluster. For example, **csi-k8s-cluster**.

CLUSTER_VERSION

Kubernetes version of the EKS cluster.

ENABLE_PRIVATE_ACCESS

Indicates whether to enable private (VPC-only) access to the EKS cluster endpoint. This parameter accepts a "true" or "false" value and maps to the EKS `--resources-vpc-config endpointPrivateAccess` option. The default value in `k8s_cluster.conf` is **true**.

ENABLE_PUBLIC_ACCESS

Whether to enable public access to the EKS cluster endpoint. This parameter accepts a "true" or "false" value and maps to the EKS `--resources-vpc-config endpointPublicAccess` option. The default value in `k8s_cluster.conf` is **false**.

CNI_VERSION

An optional property that specifies the version of the VPC CNI plugin to use for pod networking.

ENABLE_LOGGING_TYPES

A comma-separated list of the logging types to enable for the cluster. Valid values are **api**, **audit**, **authenticator**, **controllerManager**, and **scheduler**. For information about the types, see [Amazon EKS Control Plane Logging](#) in the EKS documentation. The default value in `k8s_cluster.conf` is **api,audit** for Kubernetes API logging and Audit logs, which provide a record of the users, administrators, or system components that have affected the cluster.

DISABLE_LOGGING_TYPES

A comma-separated list of the logging types to disable for the cluster. Valid values are **api**, **audit**, **authenticator**, **controllerManager**, and **scheduler**. The default value in `k8s_cluster.conf` is **controllerManager,scheduler**, which disables the Kubernetes Controller Manager daemon as well as the Kubernetes Scheduler.

WAIT_DURATION

The number of seconds to wait before timing out during cluster resource creation. For example, **1200** means the creation of a resource will time out if it is not finished in 20 minutes.

WAIT_INTERVAL

The number of seconds to wait before polling for resource state information. The default value in `k8s_cluster.conf` is **10** seconds.

STACK_CREATION_TIMEOUT

The number of minutes to wait for EKS cluster state changes before timing out. For example, the time to wait for creation or update to complete. For example, **30m**.

Example Cluster Configuration File

An example completed `k8s_cluster.conf` file is shown below.

```
# AWS Configuration parameters
REGION="us-east-1"
AvailabilityZones="us-east-1a us-east-1b"
TAGS="Description=EKS Cluster"

# Networking configuration
VPC_ID="vpc-0dd06b24c819ec3e5"
VPC_CIDR="10.107.0.0/16"
NAT_SUBNET_CIDRS="10.107.0.0/24 10.107.5.0/24"
PUBLIC_SUBNET_CIDRS="10.107.1.0/24 10.107.2.0/24"
PRIVATE_SUBNET_CIDRS="10.107.3.0/24 10.107.4.0/24"
VPC_NAT_MODE="HighlyAvailable"
WARM_IP_TARGET="8"
PUBLIC_ACCESS_CIDRS="1.2.3.4/32,1.1.1.1/32"
ALLOW_NETWORK_CIDRS="10.108.0.0/16 10.109.0.0/16"
ENABLE_ROUTE_PROPAGATION=true

# EKS control plane configuration
CLUSTER_NAME="csi-k8s-cluster"
CLUSTER_VERSION="1.19"
ENABLE_PRIVATE_ACCESS=True
ENABLE_PUBLIC_ACCESS=False
CNI_VERSION="1.7.5"
# Logging types: ["api","audit","authenticator","controllerManager","scheduler"]
ENABLE_LOGGING_TYPES="api,audit"
DISABLE_LOGGING_TYPES="controllerManager,scheduler"

# Common parameters
WAIT_DURATION=1200
WAIT_INTERVAL=10
STACK_CREATION_TIMEOUT="30m"
```

(Optional) Define the IAM Role for K8s Service Accounts Requirements

For fine-grained permission management of the applications that run in the EKS cluster, you can associate an IAM role with a Kubernetes (K8s) Service Account. The Service Account can then be used to grant permissions to the pods in the cluster so that the container applications can use an AWS SDK or AWS CLI to make API requests to AWS services like S3 or Amazon RDS. For details, see [IAM Roles for Service Accounts](#) in the Amazon EKS documentation.

If you want to create a new IAM role with associated K8s Service Accounts during EKS cluster creation, you can define the Service Account requirements in the `iam_serviceaccounts.yaml` file in the `conf.d` directory. When you create the cluster, there is a prompt that asks if you want to update IAM properties for the cluster. Responding `y` (yes) creates the account based on the specifications in `iam_serviceaccounts.yaml`. The contents of the file are shown below. Descriptions of the parameters follow the contents.

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: <eks-cluster-name>
  region: <cluster-region>
iam:
  withOIDC: true
  serviceAccounts:
  - metadata:
      name: <service-account-name>
      namespace: <namespace>
      labels: {<label-name>: "<value>"}
      attachPolicyARNs:
      - "<arn>"
      tags:
        <tag-name>: "<value>"
  - metadata:
      name: <service-account-name>
      namespace: <namespace>
      labels: {<label-name>: "<value>"}
      attachPolicyARNs:
      - "<arn>"
      tags:
        <tag-name>: "<value>"
      wellKnownPolicies:
        <policy>: <enable-policy>
      roleName: <role-name>
      roleOnly: <role-only>
```

apiVersion

The version of the schema for this object.

kind

The schema for this object.

name

The name of the EKS cluster (**CLUSTER_NAME**) to create the Service Accounts for. For example, **csi-k8s-cluster**.

region

The region that the EKS cluster is deployed in (**REGION**). For example, **us-east-1**.

withOIDC

Indicates whether to enable the IAM OpenID Connect Provider (OIDC) as well as IRSA for the Amazon CNI plugin. This value must be **true**. Amazon requires OIDC to use IAM roles for Service Accounts.

serviceAccounts

There are multiple `- metadata` sequences under `serviceAccounts`:

```
- metadata:
  name: <service-account-name>
  namespace: <namespace>
  labels: {<label-name>: "<value>"}
  attachPolicyARNs:
  - "<arn>"
  tags:
    <tag-name>: "<value>"
```

Each sequence supplies the metadata for one Service Account. You can include any number of metadata sequences to create multiple Service Accounts.

name

The name to use for the Service Account.

namespace

The namespace to create the Service Account in. If the namespace you specify does not exist, a new namespace is created. If namespace is not specified, **default** is used.

labels

An optional list of labels to add to the Service Account.

attachPolicyARNs

A list of the Amazon Resource Names (ARN) for the IAM policies to attach to the Service Account.

tags

An optional list of tags to add to the Service Account.

wellKnownPolicies

A list of any common AWS IAM policies that you want to attach to the Service Accounts, such as `imageBuilder`, `autoScaler`, `awsLoadBalancerController`, or `certManager`. For a complete list of the supported well-known policies, see the [eksctl Config File Schema](#).

roleName

The name for the new Service Account IAM Role.

roleOnly

Indicates whether to annotate the Service Accounts with the ARN of the new IAM Role (`eks.amazonaws.com/role-arn`). Cambridge Semantics recommends that you set this value to **true**.

Example IAM Role for Service Accounts Configuration File

An example completed `iam_serviceaccounts.yaml` file is shown below. This example creates a role called `S3ReadRole` with one Service Account that gives AnzoGraph containers read-only access to Amazon S3.

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: csi-k8s-cluster
  region: us-east-1
iam:
  withOIDC: true
  serviceAccounts:
  - metadata:
      name: s3-reader
      namespace: anzograph
      labels: {app: "database"}
    attachPolicyARNs:
    - "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess"
  tags:
    Team: "AnzoGraph Deployment"
  wellKnownPolicies:
    autoScaler: true
  roleName: S3ReadRole
  roleOnly: true
```

Create the EKS Cluster

After defining the cluster requirements, run the `create_k8s.sh` script in the `eksctl` directory to create the cluster.

Note

The `create_k8s.sh` script references the files in the `eksctl/reference` directory. If you customized the directory structure on the workstation, ensure that the **reference** directory is available at the same level as `create_k8s.sh` before creating the cluster.

Run the script with the following command. The arguments are described below.

```
./create_k8s.sh -c <config_file_name> [ -d <config_file_directory> ] [ -f | --force ] [ -h | --help ]
```

-c <config_file_name>

This is a **required** argument that specifies the name of the configuration file that supplies the cluster requirements. For example, `-c k8s_cluster.conf`.

-d <config_file_directory>

This is an **optional** argument that specifies the path and directory name for the configuration file specified for the `-c` argument. If you are using the original `eksctl` directory file structure and the configuration file is in the `conf.d` directory, you do not need to specify the `-d` argument. If you created a separate directory structure for different Anzo environments, include the `-d` option. For example, `-d /eksctl/env1/conf`.

-f | --force

This is an **optional** argument that controls whether the script prompts for confirmation before proceeding with each stage involved in creating the cluster. If `-f (--force)` is specified, the script assumes the answer is "yes" to all prompts and does not display them.

-h | --help

This argument is an **optional** flag that you can specify to display the help from the `create_k8s.sh` script.

For example, the following command runs the `create_k8s` script, using `k8s_cluster.conf` as input to the script. Since `k8s_cluster.conf` is in the `conf.d` directory, the `-d` argument is excluded:

```
./create_k8s.sh -c k8s_cluster.conf
```

The script validates that the required software packages, such as the `aws-cli`, `eksctl`, and `kubectl`, are installed and that the versions are compatible with the script. It also displays an overview of the deployment details based on the values in the configuration file.

The script then prompts you to proceed with deploying each component of the EKS cluster infrastructure. Type **y** (yes) and press **Enter** to proceed with each step in creating the specified network, cluster, Internet gateway, NAT gateway, route table, and security group resources. All resources are created according to the specifications in the configuration file. Once the cluster resources are deployed, the script asks whether you would like to update IAM

properties for the cluster. Continue to [Configuring Cluster IAM Properties](#) below for background information and details on configuring IAM properties.

Configuring Cluster IAM Properties

At the final stage of EKS cluster creation, the last few prompts are related to IAM properties.

First, you are asked about IAM roles for K8s Service Accounts. If you want to create Service Accounts, as described in [\(Optional\) Define the IAM Role for K8s Service Accounts Requirements](#), answer **y** (yes) to the prompt **Do you want to update IAM properties for cluster?** Service Accounts will be created according to the specifications in `iam_serviceaccounts.yaml`. If you do not want to create Service Accounts, answer **n** (no).

The last prompt is related to IAM identity mapping for the EKS cluster. Only the IAM entity that created the cluster has **system:masters** permission for the cluster and its K8s services. To grant additional AWS users or roles the ability to interact with the cluster, IAM identity mapping must be performed by adding the **aws-auth** ConfigMap to the EKS cluster configuration (see [Managing Users or IAM Roles for your Cluster](#) in the Amazon EKS documentation).

To aid you in updating the ConfigMap so that additional users can access the cluster, the `create_k8s.sh` script includes prompts that ask for the required ConfigMap information. If you want to update the ConfigMap, answer **y** (yes) to the **Do you want to add IAM users to control access to cluster** prompt. The script prompts for the following values, which will be used to update `mapRoles` and/or `mapUsers` in **aws-auth** ConfigMap:

- **Account ID:** The AWS account ID where the EKS cluster is deployed.
- **User Name:** The username within Kubernetes to map to the IAM role. For example, **admin**.
- **RBAC Group:** The Kubernetes group to map the IAM role to. For example, **system:masters**.
- **Service Name:** This value must be **emr-containers**.
- **Namespace:** The namespace to create RBAC resources in.
- **User or Role ARN:** The Amazon Resource Name for the IAM role or user to create. For example, **arn:aws:iam::105333188789:role/admin**.

When cluster creation is complete, proceed to [Creating the Required Node Groups](#) to add the required node groups to the cluster.

Related Topics

[Creating and Assigning IAM Policies](#)

[Creating the Required Node Groups](#)

Creating the Required Node Groups

This topic provides instructions for creating the four types of required node groups:

- The **Common** node group for running K8s services such as the Cluster Autoscaler and Load Balancers.
- The **Operator** node group for running the AnzoGraph, Anzo Agent with Anzo Unstructured (AU), Elasticsearch, and Spark operator pods.

- The **AnzoGraph** node group for running AnzoGraph application pods.
- The **Dynamic** node group for running Anzo Agent with AU, Elasticsearch, and Spark application pods.

Tip For more information about the node groups, see [Node Pool Requirements](#).

- [Define the Node Group Requirements](#)
- [Create the Node Groups](#)

Define the Node Group Requirements

Before creating the node groups, configure the infrastructure requirements for each type of group. The `nodepool_*.yaml` object files in the `eksctl/conf.d` directory are sample configuration files that you can use as templates, or you can edit the files directly:

- `nodepool_common.yaml` defines the requirements for the Common node group.
- `nodepool_operator.yaml` defines the requirements for the Operator node group.
- `nodepool_anzograph.yaml` defines the requirements for the AnzoGraph node group.
- `nodepool_dynamic.yaml` defines the requirements for the Dynamic node group.

Each type of node group configuration file contains the following parameters. Descriptions of the parameters and guidance on specifying the appropriate values for each type of node group are provided below.

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: <eks-cluster-name>
  region: <cluster-region>
  tags:
    <metadata-tag-name>: "<value>"
nodeGroups:
- name: <node-prefix>
  amiFamily: <ami-type>
  labels:
    <label-name>: '<value>'
  instanceType: <instance-type>
  desiredCapacity: <desired-capacity>
  availabilityZones:
  - <zones>
  minSize: <min-size>
  maxSize: <max-size>
  volumeSize: <volume-size>
  maxPodsPerNode: <max-pods>
  iam:
    attachPolicyARNs:
    - <arns>
```

```

withAddonPolicies:
  autoScaler: <auto-scaler>
  imageBuilder: <image-builder>
  efs: <efs>
  CloudWatch: <cloud-watch>
volumeType: <volume-type>
privateNetworking: <private-networking>
securityGroups:
  withShared: <shared-security-group>
  withLocal: <local-security-group>
ssh:
  allow: <allow-ssh>
  publicKeyName: <public-key-name>
taints:
  '<taint-name>': '<taint-value>'
tags:
  '<tag-name>': '<tag-value>'
asgMetricsCollection:
  - granularity: <granularity>
    metrics:
      - <metric-name>

```

apiVersion

The version of the schema for this object.

kind

The schema for this object.

name

The name of the EKS cluster that hosts the node group. For example, **csi-k8s-cluster**.

region

The region that the EKS cluster is deployed in. For example, **us-east-1**.

tags

A list of any custom tags to add to the AWS resources that are created by eksctl.

name

The prefix to add to the names of the nodes that are deployed in this node group.

Node Group Type	Sample nodeGroups name Value
Common	common
Operator	operator
AnzoGraph	anzograph
Dynamic	dynamic

amiFamily

The EKS-optimized Amazon Machine Image (AMI) type to use when deploying nodes in the node group. Cambridge Semantics recommends that you specify **AmazonLinux2**.

labels

A space-separated list of key/value pairs that define the type of pods that can be placed on the nodes in this node group. Labels are used to attract pods to nodes, while **taints** (described in [taints](#) below) are used to repel other types of pods from being placed in this node group. For example, the following labels specify that the purpose of the nodes in the groups are to host **operator**, **anzograph**, **dynamic**, or **common** pods.

Node Group Type	Recommended nodeGroups labels Value
Common	cambridgesemantics.com/node-purpose: 'common' deploy-ca: 'true' cluster-autoscaler-version: '<version>'
Operator	cambridgesemantics.com/node-purpose: 'operator'
AnzoGraph	cambridgesemantics.com/node-purpose: 'anzograph'
Dynamic	cambridgesemantics.com/node-purpose: 'dynamic'

Note

The additional Common node group label **deploy-ca: 'true'** identifies this group as the node group to host the Cluster Autoscaler (CA) service. The related **cluster-autoscaler-version** label identifies the CA version. The version that you specify must have the same major and minor version as the Kubernetes version for the EKS cluster ([CLUSTER_VERSION](#)). For example, if the cluster version is 1.19, the CA version must be 1.19.*n*, where *n* is a valid CA patch release number, such as 1.19.1. To view the CA releases for your Kubernetes version, see [Cluster Autoscaler Releases](#) on GitHub.

instanceType

The EC2 instance type to use for the nodes in the node group.

Node Group Type	Sample instanceType Value
Common	m5.large
Operator	m5.large
AnzoGraph	m5.8xlarge
Dynamic	m5.2xlarge

Tip

For more guidance on determining the instance types to use for nodes in the required node groups, see [Compute Resource Planning](#).

desiredCapacity

The number of nodes to deploy when this node group is created. This value must be set to at least **1**. When you create the node group, at least one node in the group needs to be deployed as well. However, if **minSize** is **0** and the **autoScaler** addon is enabled, the autoscaler will deprovision this node because it is not in use.

availabilityZones

A list of the Availability Zones to make this node group available to.

minSize

The minimum number of nodes for the node group. If you set the minimum size to **0**, nodes will not be provisioned unless a pod is scheduled for deployment in that group.

maxSize

The maximum number of nodes that can be deployed in the node group.

volumeSize

The size (in GB) of the EBS volume to add to the nodes in this node group.

maxPodsPerNode

The maximum number of pods that can be hosted on a node in this node group. In addition to Anzo application pods, this limit also needs to account for K8s service pods and helper pods. Cambridge Semantics recommends that you set this value to at least **16** for all node group types.

attachPolicyARNs

A list of the Amazon Resource Names (ARN) for the IAM policies to attach to the node group. These policies apply at the node level. Include the default node policies as well as any other policies that you want to add. For example:

```
attachPolicyARNs:  
- arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy  
- arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy  
- arn:aws:iam::aws:policy/AmazonS3FullAccess
```

autoScaler

Indicates whether to add an autoscaler to this node group. Cambridge Semantics recommends that you set this value to **true**.

imageBuilder

Indicates whether to allow this node group to access the full Elastic Container Registry (ECR). Cambridge Semantics recommends that you set this value to **true**.

efs

Indicates whether to enable access to the persistent volume, Elastic File System (EFS).

CloudWatch

Indicates whether to enable the CloudWatch service, which performs control plane logging when the node group is created.

volumeType

The type of EBS volume to use for the nodes in this node group.

privateNetworking

Indicates whether to isolate the node group from the public internet. Cambridge Semantics recommends that you set this value to **true**.

withShared

Indicates whether to create a shared security group for this node group to allow communication between the other node groups. Setting this value to **true** ensures that there is cluster-wide connectivity between all nodes in all node groups.

withLocal

Indicates whether to create a local security group for this node group. This security group controls access to the EKS cluster API. In addition, if SSH is allowed, port 22 will be opened in this security group. Cambridge Semantics recommends that you set this value to **true**.

allow

Indicates whether to allow SSH access to the nodes in this node group.

publicKeyName

The public key name in EC2 to add to the nodes in this node group. If **allow** is false, this value is ignored.

taints

This parameter defines the type of pods that are allowed to be placed in this node group. When a pod is scheduled for deployment, the scheduler relies on this value to determine whether the pod belongs in this group. If a pod has a **toleration** that is not compatible with this **taint**, the pod is rejected from the group. The following recommended values specify that pods must be operator pods to be deployed in the Operator node group; they must be anzograph pods to be deployed in the AnzoGraph node group; and they must be dynamic pods to be deployed in the Dynamic node group. The **NoSchedule** value means a toleration is required and pods without a toleration will not be allowed in the group.

Node Group Type	Recommended taints Value
Operator	'cambridgesemantics.com/dedicated': 'operator:NoSchedule'
AnzoGraph	'cambridgesemantics.com/dedicated': 'anzograph:NoSchedule'
Dynamic	'cambridgesemantics.com/dedicated': 'dynamic:NoSchedule'

tags

The list of key:value pairs to add to the nodes in this node group. For autoscaling to work, the list of tags must include the namespaced version of the label and taint definitions.

Node Group	Recommended tags Value
Common	'k8s.io/cluster-autoscaler/node-template/label/cambridgesemantics.com/node-purpose': 'common'
Operator	'k8s.io/cluster-autoscaler/node-template/label/cambridgesemantics.com/node-purpose': 'operator' 'k8s.io/cluster-autoscaler/node-template/taint/cambridgesemantics.com/dedicated': 'operator:NoSchedule' 'cambridgesemantics.com/node-purpose': 'operator'
AnzoGraph	'k8s.io/cluster-autoscaler/node-template/label/cambridgesemantics.com/node-purpose': 'anzograph' 'k8s.io/cluster-autoscaler/node-template/taint/cambridgesemantics.com/dedicated': 'anzograph:NoSchedule' 'cambridgesemantics.com/node-purpose': 'anzograph'

Node Group	Recommended tags Value
Dynamic	'k8s.io/cluster-autoscaler/node-template/label/cambridgesemantics.com/node-purpose': 'dynamic' 'k8s.io/cluster-autoscaler/node-template/taint/cambridgesemantics.com/dedicated': 'dynamic:NoSchedule' 'cambridgesemantics.com/node-purpose': 'dynamic'

Tip

You can also augment the required tags with any custom tags that you want to include. For information about tagging, see [Tagging your Amazon EKS Resources](#) in the Amazon EKS documentation.

asgMetricsCollection

If [CloudWatch](#) is enabled, this parameter configures the specific Auto Scaling Group (ASG) metrics to capture as well as the frequency with which to capture the metrics.

granularity

This property is a required property that specifies the frequency with which Amazon EC2 Auto Scaling sends aggregated data to CloudWatch. The only valid value is **1Minute**.

metrics

This property lists the specific group-level metrics to collect. If **granularity** is specified but **metrics** is omitted, all of the metrics are enabled. For more information and a list of valid values, see [AutoScalingGroup MetricsCollection](#) in the AWS CloudFormation documentation.

Example Configuration Files

Example completed configuration files for each type of node group are shown below.

Common Node Group

The example below shows a completed nodepool_common.yaml file.

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: csi-k8s-cluster
  region: us-east-1
  tags:
    description: "K8s cluster Common node group"
nodeGroups:
- name: common
  amiFamily: AmazonLinux2
  labels:
    cambridgesemantics.com/node-purpose: 'common'
```

```

    deploy-ca: 'true'
    cluster-autoscaler-version: '1.19.1'
instanceType: m5.large
desiredCapacity: 1
availabilityZones:
- us-east-1a
minSize: 0
maxSize: 4
volumeSize: 50
maxPodsPerNode: 16
iam:
  attachPolicyARNs:
  - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
  - arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
  - arn:aws:iam::aws:policy/AmazonS3FullAccess
  withAddonPolicies:
    autoScaler: true
    imageBuilder: true
    efs: true
    CloudWatch: true
volumeType: gp2
privateNetworking: true
securityGroups:
  withShared: true
  withLocal: true
ssh:
  allow: true
  publicKeyName: common-keypair
tags:
  'k8s.io/cluster-autoscaler/node-template/label/cambridgesemantics.com/node-purpose':
'common'
  asgMetricsCollection:
    - granularity: 1Minute
      metrics:
        - GroupPendingInstances
        - GroupInServiceInstances
        - GroupTerminatingInstances
        - GroupInServiceCapacity

```

Operator Node Group

The example below shows a completed `nodepool_operator.yaml` file.

```

apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: csi-k8s-cluster

```



```

region: us-east-1
tags:
  description: "K8s cluster Operator node group"
nodeGroups:
- name: operator
  amiFamily: AmazonLinux2
  labels:
    cambridgesemantics.com/node-purpose: 'operator'
  instanceType: m5.large
  desiredCapacity: 1
  availabilityZones:
    - us-east-1a
  minSize: 0
  maxSize: 5
  volumeSize: 50
  maxPodsPerNode: 16
  iam:
    attachPolicyARNs:
      - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
      - arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
      - arn:aws:iam::aws:policy/AmazonS3FullAccess
    withAddonPolicies:
      autoScaler: true
      imageBuilder: true
      efs: true
      cloudWatch: true
  volumeType: gp2
  privateNetworking: true
  securityGroups:
    withShared: true
    withLocal: true
  ssh:
    allow: true
    publicKeyName: operator-keypair
  taints:
    'cambridgesemantics.com/dedicated': 'operator:NoSchedule'
  tags:
    'k8s.io/cluster-autoscaler/node-template/label/cambridgesemantics.com/node-purpose':
'operator'
    'k8s.io/cluster-autoscaler/node-template/taint/cambridgesemantics.com/dedicated':
'operator:NoSchedule'
    'cambridgesemantics.com/node-purpose': 'operator'
  asgMetricsCollection:
    - granularity: 1Minute
      metrics:
        - GroupPendingInstances

```

- GroupInServiceInstances
- GroupTerminatingInstances
- GroupInServiceCapacity

AnzoGraph Node Group

The example below shows a completed `nodepool_anzograph.yaml` file.

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: csi-k8s-cluster
  region: us-east-1
  tags:
    description: "K8s cluster AnzoGraph node group"
nodeGroups:
- name: anzograph
  amiFamily: AmazonLinux2
  labels:
    cambridgesemantics.com/node-purpose: 'anzograph'
  instanceType: m5.8xlarge
  desiredCapacity: 1
  availabilityZones:
  - us-east-1a
  minSize: 0
  maxSize: 12
  volumeSize: 100
  maxPodsPerNode: 16
  iam:
    attachPolicyARNs:
    - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
    - arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
    - arn:aws:iam::aws:policy/AmazonS3FullAccess
    withAddonPolicies:
      autoScaler: true
      imageBuilder: true
      efs: true
      CloudWatch: true
  volumeType: gp2
  privateNetworking: true
  securityGroups:
    withShared: true
    withLocal: true
  ssh:
    allow: true
    publicKeyName: anzograph-keypair
  taints:
```

```

    'cambridgesemantics.com/dedicated': 'anzograph:NoSchedule'
  tags:
    'k8s.io/cluster-autoscaler/node-template/label/cambridgesemantics.com/node-purpose':
'anzograph'
    'k8s.io/cluster-autoscaler/node-template/taint/cambridgesemantics.com/dedicated':
'anzograph:NoSchedule'
    'cambridgesemantics.com/node-purpose': 'anzograph'
  asgMetricsCollection:
    - granularity: 1Minute
      metrics:
        - GroupPendingInstances
        - GroupInServiceInstances
        - GroupTerminatingInstances
        - GroupInServiceCapacity

```

Dynamic Node Group

The example below shows a completed `nodepool_dynamic.yaml` file.

```

apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: csi-k8s-cluster
  region: us-east-1
  tags:
    description: "K8s cluster Dynamic node group"
nodeGroups:
  - name: dynamic
    amiFamily: AmazonLinux2
    labels:
      cambridgesemantics.com/node-purpose: 'dynamic'
    instanceType: m5.2xlarge
    desiredCapacity: 1
    availabilityZones:
      - us-east-1a
    minSize: 0
    maxSize: 12
    volumeSize: 100
    maxPodsPerNode: 16
    iam:
      attachPolicyARNs:
        - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
        - arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
        - arn:aws:iam::aws:policy/AmazonS3FullAccess
      withAddonPolicies:
        autoScaler: true
        imageBuilder: true

```

```

    efs: true
    CloudWatch: true
  volumeType: gp2
  privateNetworking: true
  securityGroups:
    withShared: true
    withLocal: true
  ssh:
    allow: true
    publicKeyName: dynamic-keypair
  taints:
    'cambridgesemantics.com/dedicated': 'dynamic:NoSchedule'
  tags:
    'k8s.io/cluster-autoscaler/node-template/label/cambridgesemantics.com/node-purpose':
'dynamic'
    'k8s.io/cluster-autoscaler/node-template/taint/cambridgesemantics.com/dedicated':
'dynamic:NoSchedule'
    'cambridgesemantics.com/node-purpose': 'dynamic'
  asgMetricsCollection:
    - granularity: 1Minute
      metrics:
        - GroupPendingInstances
        - GroupInServiceInstances
        - GroupTerminatingInstances
        - GroupInServiceCapacity

```

Create the Node Groups

After defining the requirements for the node groups, run the **create_nodepools.sh** script in the `eksctl` directory to create each type of node group. Run the script once for each type of group.

Note

The `create_nodepools.sh` script references the files in the `eksctl/reference` directory. If you customized the directory structure on the workstation, ensure that the **reference** directory is available at the same level as `create_nodepools.sh` before creating the node groups.

Run the script with the following command. The arguments are described below.

```

./create_nodepools.sh -c <config_file_name> [ -d <config_file_directory> ] [ -f | --force
] [ -h | --help ]

```

Important

It is important to create the Common node group first. The Cluster Autoscaler and other core cluster services are dependent on the Common node group.

-c <config_file_name>

This is a **required** argument that specifies the name of the configuration file (i.e., `nodepool_common.yaml`, `nodepool_operator.yaml`, `nodepool_anzograph.yaml`, or `nodepool_dynamic.yaml`) that supplies the node group requirements. For example, **-c `nodepool_dynamic.yaml`**.

-d <config_file_directory>

This is an **optional** argument that specifies the path and directory name for the configuration file specified for the **-c** argument. If you are using the original `eksctl` directory file structure and the configuration file is in the `conf.d` directory, you do not need to specify the **-d** argument. If you created a separate directory structure for different Anzo environments, include the **-d** option. For example, **-d `/eksctl/env1/conf`**.

-f | --force

This is an **optional** argument that controls whether the script prompts for confirmation before proceeding with each stage involved in creating the node group. If **-f** (**--force**) is specified, the script assumes the answer is "yes" to all prompts and does not display them.

-h | --help

This argument is an **optional** flag that you can specify to display the help from the `create_nodepools.sh` script.

For example, the following command runs the `create_nodepools` script, using `nodepool_common.yaml` as input to the script. Since `nodepool_common.yaml` is in the `conf.d` directory, the **-d** argument is excluded:

```
./create_nodepools.sh -c nodepool_common.yaml
```

The script validates that the required software packages, such as `aws-cli`, `eksctl`, and `kubectl`, are installed and that the versions are compatible with the script. It also displays an overview of the deployment details based on the values in the specified configuration file.

The script then prompts you to proceed with deploying each component of the node group. Type **y** and press **Enter** to proceed with the configuration.

Once the Common, Operator, AnzoGraph, and Dynamic node groups are created, the next step is to create a Cloud Location in Anzo so that Anzo can connect to the EKS cluster and deploy applications. See [Connecting to a Cloud Location](#).

Related Topics

[Creating the EKS Cluster](#)

[Connecting to a Cloud Location](#)

Google Kubernetes Engine Deployments

The topics in this section guide you through the process of deploying all of the Google Kubernetes Engine (GKE) infrastructure that is required to support dynamic deployments of Anzo components. The topics provide instructions

for setting up a workstation to use for deploying the K8s infrastructure, performing the prerequisite tasks before deploying the GKE cluster, creating the GKE cluster, and creating the required node pools.

- [Setting Up a Workstation](#)
- [Planning the Anzo and GKE Network Architecture](#)
- [Creating and Assigning IAM Roles](#)
- [Creating the GKE Cluster](#)
- [Creating the Required Node Pools](#)

Setting Up a Workstation

This topic provides the requirements and instructions to follow for configuring a workstation to use for creating and managing the GKE infrastructure. The workstation needs to be able to connect to the Google Cloud API. It also needs to have the required Google Cloud and Kubernetes (K8s) software packages as well as the deployment scripts and configuration files supplied by Cambridge Semantics. This workstation will be used to connect to the Google Cloud API and provision the K8s cluster and node pools.

Note

You can use the Anzo server as the workstation if the network routing and security policies permit the Anzo server to access the Google Cloud and K8s APIs. When deciding whether to use the Anzo server as the K8s workstation, consider whether Anzo may be migrated to a different server or VPC in the future.

- [Review the Requirements and Install the Software](#)
- [Download the Cluster Creation Scripts and Configuration Files](#)

Review the Requirements and Install the Software

The table below lists the requirements for the K8s workstation.

Component	Requirement
Operating System	The operating system for the workstation must be RHEL/CentOS 7.8 or higher .
Networking	The workstation should be in the same VPC network as the GKE cluster. If it is not in the same VPC, make sure that it is on a network that is routable from the cluster's VPC.

Component	Requirement
Software	<ul style="list-style-type: none"> • Kubectl Versions 1.17 – 1.19 are supported. Cambridge Semantics recommends that you use the same kubectl version as the GKE cluster version. For instructions, see Install Kubectl below. • Google Cloud SDK is required. For installation instructions, see Install the Google Cloud SDK below.
CSI GCLOUD Package	Cambridge Semantics provides gcloud scripts and configuration files to use for provisioning the GKE cluster and node pools. Download the files to the workstation. See Download the Cluster Creation Scripts and Configuration Files for more information about the gcloud package.

Install Kubectl

Follow the instructions below to install kubectl on your workstation. Cambridge Semantics recommends that you install the same version of kubectl as the K8s cluster API. For more information, see [Install and Set Up kubectl on Linux](#) in the Kubernetes documentation.

1. Run the following cURL command to download the kubectl binary:

```
curl -LO https://dl.k8s.io/release/<version>/bin/linux/amd64/kubectl
```

Where <version> is the version of kubectl to install. For example, the following command downloads version 1.19.12:

```
curl -LO https://dl.k8s.io/release/v1.19.12/bin/linux/amd64/kubectl
```

2. Run the following command to make the binary executable:

```
chmod +x ./kubectl
```

3. Run the following command to move the binary to your PATH:

```
sudo mv ./kubectl /usr/local/bin/kubectl
```

4. To confirm that the binary is installed and that you can run kubectl commands, run the following command to display the client version:

```
kubectl version --client
```

The command returns the following type of information. For example:

```
Client Version: version.Info{Major:"1", Minor:"19", GitVersion:"v1.19.12",
GitCommit:"f3abc15296f3a3f54e4ee42e830c61047b13895f",
GitTreeState:"clean", BuildDate:"2021-06-16T13:21:12Z", GoVersion:"go1.13.15",
Compiler:"gc", Platform:"linux/amd64"}
```

Install the Google Cloud SDK

Follow the instructions below to install the Google Cloud SDK on your workstation.

1. Run the following command to configure access to the Google Cloud repository:

```
sudo tee -a /etc/yum.repos.d/google-cloud-sdk.repo << EOM
[google-cloud-sdk]
name=Google Cloud SDK
baseurl=https://packages.cloud.google.com/yum/repos/cloud-sdk-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOM
```

2. Run the following command to install google-cloud-sdk:

```
sudo yum install google-cloud-sdk
```

The following packages are installed:

```
google-cloud-sdk-app-engine-grpc
google-cloud-sdk-pubsub-emulator
google-cloud-sdk-app-engine-go
google-cloud-sdk-cloud-build-local
google-cloud-sdk-datastore-emulator
google-cloud-sdk-app-engine-python
google-cloud-sdk-cbt
google-cloud-sdk-bigtable-emulator
google-cloud-sdk-datalab
google-cloud-sdk-app-engine-java
```

3. Next, configure the default project and region settings for the Cloud SDK:
 - a. Run the following command to set the default project for the GKE cluster:

```
gcloud config set project <project_ID>
```

Where <project_ID> is the Project ID for the project in which the GKE cluster will be provisioned.

- b. If you work with zonal clusters, run the following command to set the default compute zone for the GKE cluster:


```
gcloud config set compute/zone <compute_zone>
```

Where <compute_zone> is the default compute zone for the GKE cluster. For example:

```
gcloud config set compute/zone us-central1-a
```

- c. If you work with regional clusters, run the following command to set the default region for the GKE cluster:

```
gcloud config set compute/region <compute_region>
```

Where <compute_region> is the default region for the GKE cluster. For example:

```
gcloud config set compute/region us-east1
```

- d. To make sure that you are using the latest version of the Cloud SDK, run the following command to check for updates:

```
gcloud components update
```

Download the Cluster Creation Scripts and Configuration Files

The Cambridge Semantics GitHub repository, [k8s-genesis](https://github.com/cambridgesemantics/k8s-genesis) (<https://github.com/cambridgesemantics/k8s-genesis.git>), includes all of the files that are needed to manage the configuration, creation, and deletion of the GKE cluster and node pools.

You can clone the repository to any location on the workstation or download the k8s-genesis package as a ZIP file, copy the file to the workstation, and extract the contents. The k8s-genesis directory includes three subdirectories (one for each supported Cloud Service Provider), the license information, and a readme file:

```
k8s-genesis
├─ aws
├─ azure
├─ gcp
├─ LICENSE
└─ README.md
```

Navigate to `/gcp/k8s/gcloud`. The **gcloud** directory contains all of the GKE cluster and node pool configuration files. You can remove all other directories from the workstation. The gcloud files and subdirectories are shown below:

```
gcloud
├─ common.sh
├─ conf.d
│   └─ k8s_cluster.conf
│   └─ nodepool_anzograph.conf
│   └─ nodepool_anzograph_tuner.yaml
│   └─ nodepool_common.conf
```

```

|   ├── nodepool.conf
|   ├── nodepool_dynamic.conf
|   ├── nodepool_dynamic_tuner.yaml
|   └── nodepool_operator.conf
├── create_k8s.sh
├── create_nodepools.sh
├── delete_k8s.sh
├── delete_nodepools.sh
├── gcloud_cli_common.sh
├── README.md
└── sample_use_cases
    ├── 1_usePrivateEndpoint_private_cluster
    │   └── k8s_cluster.conf
    ├── 2_public_cluster
    │   └── k8s_cluster.conf
    ├── 3_useAuthorizedNetworks
    │   └── k8s_cluster.conf
    └── 4_providePublicEndpointAccess
        └── k8s_cluster.conf

```

The following list gives an overview of the files. Subsequent topics describe the files in more detail.

- The **common.sh** and **gcloud_cli_common.sh** scripts are used by the **create*.sh** and **delete*.sh** scripts when the GKE cluster and node pools are created or deleted.
- The **conf.d** directory contains the configuration files that supply the specifications to follow when creating the K8s cluster and node pools.
 - **k8s_cluster.conf**: Supplies the specifications for the GKE cluster.
 - **nodepool_anzograph.conf**: Supplies the specifications for the AnzoGraph node pool.
 - **nodepool_anzograph_tuner.conf**: Supplies the kernel-level tuning and security policies to apply to AnzoGraph runtime environments.
 - **nodepool_common.conf**: Supplies the specifications for a Common node pool. The Common node pool is not required for GKE deployments, and this configuration file is typically not used.
 - **nodepool.conf**: This file is supplied as a reference. It contains the superset of node pool parameters.
 - **nodepool_dynamic.conf**: Supplies the specifications for the Dynamic node pool.
 - **nodepool_dynamic_tuner.conf**: Supplies the kernel-level tuning and security policies to apply to Dynamic runtime environments.
 - **nodepool_operator.conf**: Supplies the specifications for the Operator node pool.
- The **create_k8s.sh** script is used to deploy the GKE cluster.
- The **create_nodepools.sh** script is used to deploy node pools in the GKE cluster.
- The **delete_k8s.sh** script is used to delete the GKE cluster.
- The **delete_nodepools.sh** script is used to remove node pools from the GKE cluster.

- The **sample_use_cases** directory contains sample GKE cluster configuration files that you can refer to or use as a template for configuring your GKE cluster depending on your use case:
 - The **k8s_cluster.conf** file in the **1_usePrivateEndpoint_private_cluster** directory is a sample file for a use case where you want to deploy the GKE cluster in an existing network that does not have public internet access.
 - The **k8s_cluster.conf** file in the **2_public_cluster** directory is a sample file for a use case where you want to deploy the GKE cluster into a new network with public internet access.
 - The **k8s_cluster.conf** file in the **3_useAuthorizedNetworks** directory is a sample file for a use case where you want to deploy the GKE cluster into a private network with master authorized networks.
 - The **k8s_cluster.conf** file in the **4_providePublicEndpointAccess** directory is a sample file for a use case where you want to deploy the GKE cluster into a private network that has public endpoint access enabled.

Once the workstation is configured, see [Planning the Anzo and GKE Network Architecture](#) to review information about the network architecture that the gcloud scripts create. And see [Creating and Assigning IAM Roles](#) for instructions on creating the IAM roles that are needed for assigning permissions to create and use the GKE cluster.

Related Topics

[Planning the Anzo and GKE Network Architecture](#)

[Creating and Assigning IAM Roles](#)

[Creating the GKE Cluster](#)

[Creating the Required Node Pools](#)

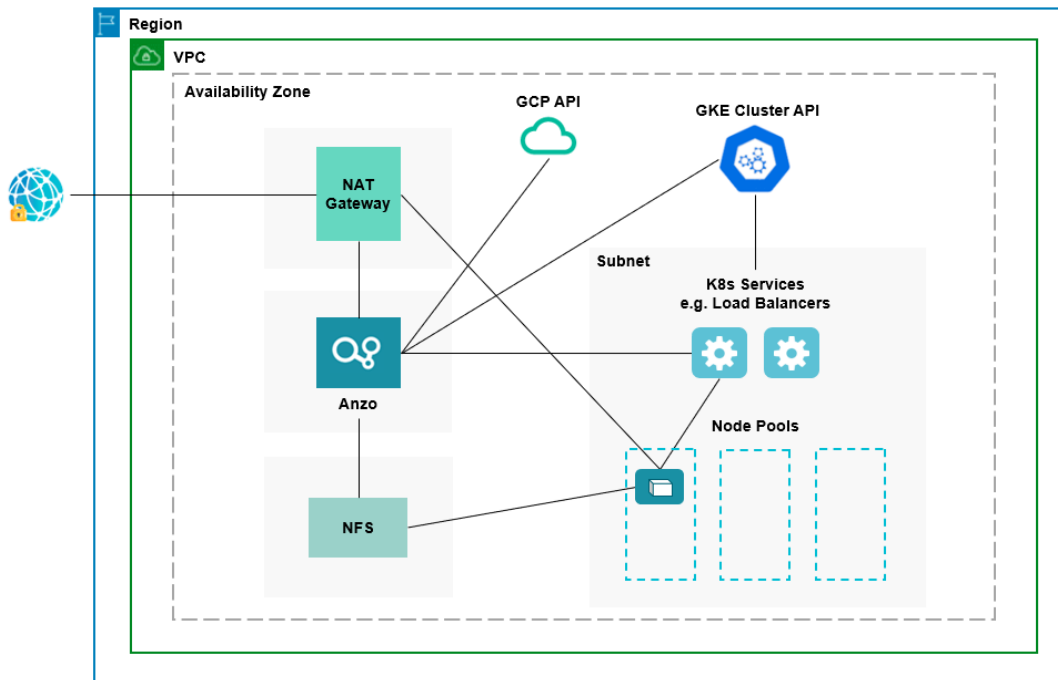
Planning the Anzo and GKE Network Architecture

This topic describes the network architecture that supports the Anzo and GKE integration.

Note

When you deploy the K8s infrastructure, Cambridge Semantics strongly recommends that you create the GKE cluster in the same VPC network as Anzo. If you create the GKE cluster in a new VPC, you must configure the new VPC to be routable from the Anzo VPC.

The diagram below shows the typical network components that are employed when a GKE cluster is integrated with Anzo. Most of the network resources shown in the diagram are automatically deployed (and the appropriate routing is configured) according to the values that you supply in the cluster and node pool .conf files in the **gcloud** package on the workstation.



In the diagram, there are two components that you deploy before configuring and creating the K8s resources:

- **Anzo:** Since the Anzo server is typically deployed before the K8s components, you specify the Anzo network when creating the GKE cluster, ensuring that Anzo and all of the GKE cluster components are in the same network and can talk to each other. Also, make sure that Anzo has access to the GCP and GKE APIs.
- **NFS:** You are required to create a network file system (NFS). However, Anzo automatically mounts the NFS to the nodes when AnzoGraph, Anzo Unstructured, Spark, and Elasticsearch pods are deployed so that all of the applications can share files. See [Deploying the Shared File System](#) for more information. The NFS does not need to have its own subnet but it can.

The rest of the components in the diagram are automatically provisioned, depending on your specifications, when the GKE cluster and node pools are created. The `gcloud` scripts can be used to create a NAT gateway and subnet for outbound internet access, such as for pulling container images from the Cambridge Semantics repository. In addition, the scripts create a subnet for the K8s services and node pools and configure the routing so that Anzo can communicate with the K8s services and the services can talk to the pods that are deployed in the node pools.

Tip

When considering the network requirements of your organization and planning how to integrate the new K8s infrastructure in accordance with those requirements, it may help to consider the following types of use cases. Cambridge Semantics supplies sample cluster configuration files in the `gcloud/sample_use_cases` directory that are tailored for each of these use cases:

- **Deploy a private GKE cluster in an existing network (i.e., the same network as Anzo)**

In this use case, the GKE cluster is deployed in a private subnet in your existing network. And a new (or existing, if you have one) NAT gateway is used to enable outbound access to services that are outside of the network. The control plane (master) is configured to allow access only from certain CIDRs.

- **Deploy a public GKE cluster in a new network**

In this use case, a new network is created with the specified CIDR. A new NAT gateway is deployed to provide outbound connectivity for the cluster nodes. Public and private subnets are also created, and public access is restricted to specific IP ranges. The new network will need to be configured so that it is routable from Anzo.

- **Deploy a private GKE cluster with master authorized networks**

In this use case (like the first case listed above), a private GKE cluster is deployed in an existing network. Master authorized network IP ranges are specified to limit the access to the public endpoint.

- **Deploy a private GKE cluster with public endpoint access enabled**

In this use case, a private GKE cluster is deployed but public endpoint access is enabled and not restricted to specific IP ranges.

For a summary of the files in the gcloud directory, see [Download the Cluster Creation Scripts and Configuration Files](#). Specifics about the parameters in the sample files are included in [Creating the GKE Cluster](#).

To get started on creating the GKE infrastructure, see [Creating and Assigning IAM Roles](#) for instructions on creating the IAM roles that are needed for assigning permissions to create and use the GKE cluster.

Related Topics

[Setting Up a Workstation](#)

[Creating and Assigning IAM Roles](#)

[Creating the GKE Cluster](#)

[Creating the Required Node Pools](#)

Creating and Assigning IAM Roles

There are two custom Identity and Access Management (IAM) roles that need to be created in Google Cloud to grant the necessary permissions to the following two types of GKE users:

1. The first type of user is the user who sets up the K8s infrastructure, i.e., the user who configures, creates, and maintains the GKE cluster and node pools. This user role is called the **GKE Cluster Admin**.
2. The second type of user is the user who connects to the GKE cluster and deploys the dynamic Anzo applications. Typically this user is Anzo. Since Anzo communicates to the K8s services that provision the applications, the

Anzo service account needs to be granted certain privileges. This user role is called the **GKE Cluster Developer**.

Note

The enterprise-level Anzo service account is a requirement for the Anzo installation and is typically in place before Anzo is installed. For more information, see [Anzo Service Account Requirements](#).

This topic provides instructions for creating the two roles and gives guidance on assigning the roles to the appropriate members or service accounts.

- [Create and Assign the GKE Cluster Admin Role](#)
- [Create and Assign the GKE Cluster Developer Role](#)

Create and Assign the GKE Cluster Admin Role

To ensure that the GKE cluster creator has all of the permissions needed for creating and managing K8s resources, there are four predefined Google roles in addition to the GKE Cluster Admin custom role that must be applied to the member or service account that will be used when creating the K8s infrastructure. Follow the instructions below to create the custom role and assign all necessary roles to the appropriate member or service account.

Note

Google Cloud IAM administrator privileges are required to create and assign IAM roles. The steps below give instructions for creating the custom GKE Cluster Admin role from the workstation. For more information about creating roles, including instructions on creating roles from the Cloud Console, see [Creating and Managing Custom Roles](#) in the Google Cloud documentation.

1. Create a JSON file on your workstation and copy the following contents to the file. For example, `vi /tmp/gke-cluster-admin.json`. The contents apply the minimum permissions needed for the GKE Cluster Admin.

```
{
  "name": "customClusterAdminRole",
  "title": "Custom Role for GKE Cluster Admin",
  "includedPermissions": [
    "compute.addresses.create",
    "compute.addresses.delete",
    "compute.addresses.get",
    "compute.addresses.use",
    "compute.firewallPolicies.get",
    "compute.firewalls.get",
    "compute.instanceGroups.get",
    "compute.instanceGroups.list",
    "compute.instances.get",
    "compute.instances.list",
```

```
"compute.networks.create",
"compute.networks.delete",
"compute.networks.get",
"compute.networks.listPeeringRoutes",
"compute.networks.updatePolicy",
"compute.networks.use",
"compute.nodeGroups.get",
"compute.regionOperations.get",
"compute.regionOperations.list",
"compute.regions.get",
"compute.routers.create",
"compute.routers.delete",
"compute.routers.get",
"compute.routers.update",
"compute.routers.use",
"compute.subnetworks.create",
"compute.subnetworks.delete",
"compute.subnetworks.get",
"compute.subnetworks.use",
"compute.vpnTunnels.get",
"container.clusters.create",
"container.clusters.delete",
"container.clusters.update",
"container.daemonSets.create",
"container.daemonSets.delete",
"container.daemonSets.get",
"container.daemonSets.getStatus",
"container.daemonSets.list",
"container.nodes.list",
"container.operations.get",
"container.operations.list",
"container.podSecurityPolicies.create",
"container.podSecurityPolicies.delete",
"container.podSecurityPolicies.get",
"container.podSecurityPolicies.list",
"container.podSecurityPolicies.update",
"container.roleBindings.create",
"container.roleBindings.delete",
"container.roleBindings.get",
"container.roles.bind",
"container.roles.create",
"container.roles.delete",
"container.roles.get",
"container.serviceAccounts.create",
"container.serviceAccounts.delete",
```

```

    "container.serviceAccounts.get"
  ],
  "stage": "GA"
}

```

2. Once the file is created, run the following command to create the GKE Cluster Admin role, named **customClusterAdminRole**:

```

gcloud iam roles create <role_name> --project <project_name> --file=/

```

Where `<project_name>` is the project ID that the GKE cluster will be deployed in. For example:

```

gcloud iam roles create customClusterAdminRole --project cloud-project-1592 --
file=/tmp/gke-cluster-admin.json

```

3. Next, grant the new **customClusterAdminRole** and the following four predefined Compute Engine, Kubernetes Engine, Service Account, and Logging roles to the member or service account that will be used to create the GKE cluster:

- **roles/compute.networkViewer**
- **roles/container.clusterViewer**
- **roles/iam.serviceAccountUser**
- **roles/logging.viewer**

For information about granting roles to a member, see [Granting, changing, and revoking access to resources](#).

For information about applying a role to a service account, see [Creating and managing service accounts](#). And for details about the predefined roles, see [Predefined Roles](#) in the Google Cloud documentation.

Create and Assign the GKE Cluster Developer Role

The following IAM role applies the minimum permissions needed for the GKE Cluster Developer role. Follow the instructions below to create the role and assign it to the Anzo service account.

Note

Google Cloud IAM administrator privileges are required to create and assign IAM roles. The steps below give instructions for creating the custom GKE Cluster Developer role from the workstation. For more information about creating roles, including instructions on creating roles from the Cloud Console, see [Creating and Managing Custom Roles](#) in the Google Cloud documentation.

1. Create a JSON file on your workstation and copy the following contents to the file. For example, `vi /tmp/gke-cluster-developer.json`.


```
{
  "name": "customClusterDevAnzoRole",
  "title": "Custom Role with Additional permissions required to deploy resources
through Anzo",
  "includedPermissions": [
    "compute.machineTypes.list",
    "storage.buckets.get",
    "storage.buckets.list"
  ],
  "stage": "GA"
}
```

2. Once the file is created, run the following command to create the GKE Cluster Developer role, named **customClusterDevAnzoRole**:

```
gcloud iam roles create <role_name> --project <project_name> --file=/

```

Where **<role_ID>** is the unique ID to use for the role and **<project_name>** is the project ID that the GKE cluster will be deployed in. For example:

```
gcloud iam roles create customClusterDevAnzoRole --project cloud-project-1592 --
file=/tmp/gke-cluster-developer.json
```

3. Next, grant the new **customClusterDevAnzoRole** and the following three predefined Kubernetes Engine Developer, Kubernetes Engine Service Agent, and Storage Object Viewer roles to the Anzo service account:
 - **roles/container.developer**
 - **roles/container.serviceAgent**
 - **roles/storage.objectViewer**

For information about applying a role to a service account, see [Creating and managing service accounts](#) in the Google Cloud documentation. For details about the predefined roles, see [Predefined Roles](#) in the Google Cloud documentation.

Once the IAM roles are in place and users are granted access, proceed to [Creating the GKE Cluster](#) for instructions on configuring and creating the cluster.

Related Topics

[Setting Up a Workstation](#)

[Planning the Anzo and GKE Network Architecture](#)

[Creating the GKE Cluster](#)

[Creating the Required Node Pools](#)

Creating the GKE Cluster

Follow the instructions below to define the GKE cluster resource requirements and then create the cluster based on your specifications.

- [Define the GKE Cluster Requirements](#)
- [Create the GKE Cluster](#)

Define the GKE Cluster Requirements

The first step in creating the K8s cluster is to define the infrastructure specifications. The configuration file to use for defining the specifications is called **k8s_cluster.conf**. Multiple sample k8s_cluster.conf files are included in the **gcloud** directory. Any of them can be copied and used as templates, or the files can be edited directly.

Sample k8s_cluster.conf Files

To help guide you in choosing the appropriate template for your use case, this section describes each of the sample files. Details about the parameters in the sample files are included in [Cluster Parameters](#) below.

gcloud/conf.d/k8s_cluster.conf

This file is a non-specific use case. It includes sample values for all of the available cluster parameters.

gcloud/sample_use_cases/1_usePrivateEndpoint_private_cluster/k8s_cluster.conf

This file includes sample values for a use case where:

- The GKE cluster will be deployed in a new private subnet in an existing network. You specify the existing network name in the **G_CLOUD_NETWORK** parameter.
- A NAT gateway is deployed with a private endpoint (**GKE_ENABLE_PRIVATE_ENDPOINT=true**, **GKE_ENABLE_PRIVATE_ENDPOINT=true**, **GKE_PRIVATE_ACCESS=true**). There is no client access to the public endpoint.
- Secondary IP ranges are added to the NAT mapping along with the primary IP when **NETWORK_NAT_ALLOW_SUBNET_SECONDARY_IPS=true**. Outbound connectivity is allowed through the NAT gateway but restricted to the IP ranges specified in the **GKE_MASTER_ACCESS_CIDRS** parameter.

gcloud/sample_use_cases/2_public_cluster/k8s_cluster.conf

This file includes sample values for a use case where:

- A new network with public and private subnetworks will be created and the GKE cluster will be deployed into it.
- The cluster is public (**GKE_PRIVATE_ACCESS=false**).

gcloud/sample_use_cases/3_useAuthorizedNetworks/k8s_cluster.conf

This file includes sample values for a use case where:

- The GKE cluster will be deployed in a new or existing network with public and private subnets.
- The **GKE_MASTER_ACCESS_CIDRS** parameter is used to limit the access to the public endpoint.

gcloud/sample_use_cases/4_providePublicEndpointAccess/k8s_cluster.conf

This file includes sample values for a use case where:

- The GKE cluster will be deployed as a private cluster with public endpoint access enabled (`GKE_ENABLE_PRIVATE_ENDPOINT=false`).

Cluster Parameters

The contents of `k8s_cluster.conf` are shown below. Descriptions of the cluster parameters follow the contents.

```

NETWORK_BGP_ROUTING="<bgp-routing-mode>"
NETWORK_SUBNET_MODE="<subnet-mode>"
NETWORK_ROUTER_NAME="<router>"
NETWORK_ROUTER_MODE="<advertisement-mode>"
NETWORK_ROUTER_ASN=<asn>
NETWORK_ROUTER_DESC="<description>"
NETWORK_NAT_NAME="<nat-name>"
NETWORK_NAT_UDP_IDLE_TIMEOUT="<udp-idle-timeout>"
NETWORK_NAT_ICMP_IDLE_TIMEOUT="<icmp-idle-timeout>"
NETWORK_NAT_TCP_ESTABLISHED_IDLE_TIMEOUT="<tcp-established-idle-timeout>"
NETWORK_NAT_TCP_TRANSITORY_IDLE_TIMEOUT="<tcp-transitory-idle-timeout>"
NETWORK_NAT_ALLOW_SUBNET_SECONDARY_IPS=<allow-subnet-secondary-ips>
K8S_CLUSTER_NAME=${K8S_CLUSTER_NAME:-"<cluster-name>"}
K8S_CLUSTER_PODS_PER_NODE="<default-max-pods-per-node>"
K8S_CLUSTER_ADDONS="<addons>"
GKE_MASTER_VERSION="<cluster-version>"
GKE_PRIVATE_ACCESS=<enable-private-nodes>
GKE_MASTER_NODE_COUNT_PER_LOCATION=<num-nodes>
GKE_NODE_VERSION="<node-version>"
GKE_IMAGE_TYPE="<image-type>"
GKE_MAINTENANCE_WINDOW='<maintenance-window>'
GKE_ENABLE_PRIVATE_ENDPOINT=<enable-private-endpoint>
GKE_MASTER_ACCESS_CIDRS="<master-authorized-networks>"
K8S_PRIVATE_CIDR="<cluster-ipv4-cidr>"
K8S_SERVICES_CIDR="<services-ipv4-cidr>"
GCP_NODES_CIDR="<create-subnetwork>"
K8S_API_CIDR="<master-ipv4-cidr>"
K8S_HOST_DISK_SIZE='<disk-size>'
K8S_HOST_DISK_TYPE="<disk-type>"
K8S_HOST_MIN_CPU_PLATFORM="<min-cpu-platform>"
K8S_POOL_HOSTS_MAX=<max-nodes-per-pool>
K8S_METADATA="<metadata>"
K8S_MIN_NODES=<min-nodes>
K8S_MAX_NODES=<max-nodes>
GCP_RESOURCE_LABELS='<labels>'
GCP_VM_LABELS=<node-labels>
GCP_VM_TAGS="<tags>"

```

```
GCLOUD_VM_MACHINE_TYPE="<machine-type>"
GCLOUD_VM_SSD_COUNT=<local-ssd-count>
GCLOUD_PROJECT_ID=${GCLOUD_PROJECT_ID:-"<project>"}
GCLOUD_NETWORK=${GCLOUD_NETWORK:-"<network>"}
GCLOUD_NODES_SUBNET_SUFFIX="<suffix>"
GCLOUD_CLUSTER_REGION=${GCLOUD_CLUSTER_REGION:-"<region>"}
GCLOUD_NODE_LOCATIONS="<node-locations>"
GCLOUD_NODE_TAINTS='<node-taints>'
GCLOUD_NODE_SCOPE='<scopes>'
```

NETWORK_BGP_ROUTING

The mode the Cloud Router will use to advertise BGP routes when the network is created, i.e, whether the cluster is global or regional. This parameter maps to the gcloud Cloud Router `--bgp-routing-mode` option. The default value is **regional**.

NETWORK_SUBNET_MODE

The method to use when subnets are created. Valid values are "auto" or "custom." This parameter maps to the gcloud VPC `--subnet-mode` option. The recommended value is **custom**.

NETWORK_ROUTER_NAME

The name to assign to the Cloud Router. For example, **csi-cloudrouter**.

NETWORK_ROUTER_MODE

The route advertisement mode for the Cloud Router. This parameter maps to the gcloud Cloud Router `--advertisement-mode` option. The recommended value is **custom**.

NETWORK_ROUTER_ASN

The Border Gateway Protocol (BGP) autonomous system number (ASN). When a router is created, it is assigned an ASN. This parameter maps to the gcloud Cloud Router `--asn` option. Coordinate with your network administrator to determine the number to specify.

NETWORK_ROUTER_DESC

A description of the Cloud Router. This parameter maps to the gcloud Cloud Router `--description` option. For example, **Cloud router for K8S NAT**.

NETWORK_NAT_NAME

The name to assign to the NAT gateway. For example, **csi-natgw**.

NETWORK_NAT_UDP_IDLE_TIMEOUT

The timeout value for UDP connections to the NAT gateway. This parameter maps to the gcloud NAT router `--udp-idle-timeout` option. The default value in `k8s_cluster.conf` is **60s** (60 seconds). For information about duration formats, refer to [gcloud topic datetimes](#) in the Cloud SDK documentation.

NETWORK_NAT_ICMP_IDLE_TIMEOUT

The timeout value for ICMP connections to the NAT gateway. This parameter maps to the gcloud NAT router `--icmp-idle-timeout` option. The default value in `k8s_cluster.conf` is **60s** (60 seconds).

NETWORK_NAT_TCP_ESTABLISHED_IDLE_TIMEOUT

The timeout value for TCP established connections to the NAT gateway. This parameter maps to the gcloud NAT router `--tcp-established-idle-timeout` option. The default value in `k8s_cluster.conf` is **60s** (60 seconds).

NETWORK_NAT_TCP_TRANSITORY_IDLE_TIMEOUT

The timeout value to use for TCP transitory connections to the NAT gateway. This parameter maps to the gcloud NAT router `--tcp-transitory-idle-timeout` option. The default value in `k8s_cluster.conf` is **60s** (60 seconds).

NETWORK_NAT_ALLOW_SUBNET_SECONDARY_IPS

Indicates whether to allow all secondary IP ranges for the GKE cluster to use the NAT gateway. If **true**, the secondary IP ranges for the subnets will have NAT gateway access.

K8S_CLUSTER_NAME

The name to give to the cluster. For example, **csi-k8s-cluster**.

K8S_CLUSTER_PODS_PER_NODE

The maximum number of pods that can be hosted on each compute instance. This parameter maps to the gcloud container cluster `--default-max-pods-per-node` option. This value also applies to the node pools in the cluster if the node pool configuration does not specify the maximum number of pods per node. Cambridge Semantics recommends that you set this value to **16**.

K8S_CLUSTER_ADDONS

A comma-separated list of any additional Kubernetes cluster components to enable for the cluster. This parameter maps to the gcloud container cluster `--addons` option. By default, the `k8s_cluster.conf` file lists **HttpLoadBalancing** and **HorizontalPodAutoscaling**. Cambridge Semantics recommends that you include both of these components as a best practice.

GKE_MASTER_VERSION

The Kubernetes version to use for the GKE cluster. This parameter maps to the gcloud container cluster `--cluster-version` option.

GKE_PRIVATE_ACCESS

Indicates whether the cluster's nodes should have external IP addresses. When **GKE_PRIVATE_ACCESS=true**, the cluster remains private and nodes are not assigned external IP addresses. This parameter maps to the GKE `--enable-private-nodes` option.

GKE_MASTER_NODE_COUNT_PER_LOCATION

The number of nodes to create for running the K8s services in the default node pool in each of the cluster's zones. This value must be at least 1. For high availability, Cambridge Semantics recommends setting this value to 3. This parameter maps to the gcloud container cluster `--num-nodes` option.

GKE_NODE_VERSION

The Kubernetes version to use for nodes in the node pools. This parameter maps to the gcloud container cluster `--node-version` option.

Note

Cambridge Semantics recommends that you specify the same version as the GKE_MASTER_VERSION.

GKE_IMAGE_TYPE

The base operating system that the nodes in the cluster will run on. This parameter maps to the gcloud container cluster `--image-type` option. This value must be **COS**.

GKE_MAINTENANCE_WINDOW

The time of day to start maintenance on this cluster. This parameter maps to the gcloud container cluster `--maintenance-window` option. The time corresponds to the UTC time zone and must be in HH:MM format. The default value in `k8s_cluster.conf` is **06:00** (6:00 am).

GKE_ENABLE_PRIVATE_ENDPOINT

Indicates whether to use a private or public IP address for the master API endpoint. When GKE_ENABLE_PRIVATE_ENDPOINT=true, the IP address for the API endpoint is private. This parameter maps to the GKE `--enable-private-endpoint` option.

GKE_MASTER_ACCESS_CIDRS

The list of CIDR blocks (up to 50) that are allowed to connect to the GKE cluster over HTTPS. This value should include the Anzo subnet CIDR so that Anzo has access to the GKE cluster. This parameter maps to the gcloud container cluster `--master-authorized-networks` option. For example, **10.128.0.0/9**.

K8S_PRIVATE_CIDR

The IP address range (in CIDR notation) for the pods in this cluster. This parameter maps to the gcloud container cluster `--cluster-ipv4-cidr` option. For example, **172.16.0.0/20**.

K8S_SERVICES_CIDR

The IP address range for the cluster services. This parameter maps to the gcloud container cluster `--services-ipv4-cidr` option. For example: **172.17.0.0/20**.

GCLOUD_NODES_CIDR

The CIDR for the new subnet that will be created for the K8s cluster. This parameter maps to the `--create-subnetwork` option. For example, **192.168.0.0/20**.

K8S_API_CIDR

The IPv4 CIDR range to use for the master network. The range should have a subnet mask of /28. This parameter maps to the gcloud container cluster `--master-ipv4-cidr` option. For example, **192.171.0.0/28**.

K8S_HOST_DISK_SIZE

The size of the boot disks on the cluster compute instances. This parameter maps to the gcloud container cluster `--disk-size` option. For example, **50GB**.

K8S_HOST_DISK_TYPE

The type of boot disk to use. This parameter maps to the gcloud container cluster `--disk-type` option. For example, **pd-standard**.

K8S_HOST_MIN_CPU_PLATFORM

The minimum CPU platform to use. This parameter maps to the gcloud container cluster `--min-cpu-platform` option. This value is left blank in the `k8s_cluster.conf` file.

K8S_POOL_HOSTS_MAX

The maximum number of nodes to allocate for the default initial node pool. This parameter maps to the gcloud container cluster `--max-nodes-per-pool` option. The default value is **1000**, but it can be set as low as 100 for the initial creation.

K8S_METADATA

The compute engine metadata (in the format `key=val,key=val`) to make available to the guest operating system running on nodes in the node pools. This parameter maps to the gcloud container cluster `--metadata` option.

Important

Including **disable-legacy-endpoints=true** is required to ensure that legacy metadata APIs are disabled. For more information about the option, see [Protecting Cluster Metadata](#) in the GKE documentation.

K8S_MIN_NODES

The minimum number of nodes in the default node pool. This parameter maps to the gcloud container cluster `--min-nodes` option. For example, **1**.

K8S_MAX_NODES

The maximum number of nodes in the default node pool. This parameter maps to the gcloud container cluster `--max-nodes` option. For example, **3**.

GKLOUD_RESOURCE_LABELS

A comma-separated list of any labels that you want to apply to the Google Cloud resources in use by the GKE cluster (unrelated to Kubernetes labels).

GKLOUD_VM_LABELS

A comma-separated list of any Kubernetes labels to apply to nodes in the default node pool. This parameter maps to the gcloud container cluster `--node-labels` option.

GKLOUD_VM_TAGS

A comma-separated list of strings to add to the instances in the cluster to classify the VMs. This parameter maps to the gcloud container cluster `--tags` option.

GKLOUD_VM_MACHINE_TYPE

The machine type to use for the GKE cluster nodes. This parameter maps to the gcloud container cluster `--machine-type` option. For example, **n1-standard-1**.

GKLOUD_VM_SSD_COUNT

The number of local SSD disks to add to each node. This parameter maps to the gcloud container cluster `--local-ssd-count` option. For example, specify **0** if you do not want to add SSDs to the nodes.

GKLOUD_PROJECT_ID

The Project ID for the GKE cluster. This parameter maps to the gcloud-wide `--project` option. For example, **cloud-project-1592**.

GKLOUD_NETWORK

The network to provision the GKE cluster in. This value should match the name of the network that Anzo is deployed in. This parameter maps to the gcloud container cluster `--network` option. For example, **dev-network**.

Note

If you want gcloud to create a new network, you can leave this value blank. However, after deploying the GKE cluster, you must configure the new network so that it is routable from the Anzo network.

GKLOUD_NODES_SUBNET_SUFFIX

The suffix to add to the subnetworks. For example, **nodes**.

GCLOUD_CLUSTER_REGION

The compute region for the GKE cluster. This value should match the name of the region that Anzo is deployed in.

This parameter maps to the gcloud container cluster `--region` option. For example, **us-central1**.

GCLOUD_NODE_LOCATIONS

A comma-separated list of any zones to replicate the nodes in. This parameter maps to the gcloud container cluster `--node-locations` option. For example, **us-central1-f**.

GCLOUD_NODE_TAINTS

A comma-separated list of the Kubernetes taints for the nodes in the default node pool. When a pod is scheduled for deployment, the scheduler relies on this information to find the node pool that the pod belongs in. A pod has a **toleration** that identifies whether it is compatible with a node taint. This parameter maps to the gcloud container cluster `--node-taints` option. For more information, see [Controlling Scheduling with Node Taints](#) in the GKE documentation.

GCLOUD_NODE_SCOPE

A comma-separated list of the access scopes the nodes should have. This parameter maps to the gcloud container cluster `--scopes` option. For example, **gke-default**.

Example Configuration File

An example completed `k8s_cluster.conf` file is shown below.

```

NETWORK_BGP_ROUTING="regional"
NETWORK_SUBNET_MODE="custom"
NETWORK_ROUTER_NAME="csi-cloudrouter"
NETWORK_ROUTER_MODE="custom"
NETWORK_ROUTER_ASN=64512
NETWORK_ROUTER_DESC="Cloud router for K8S NAT."
NETWORK_NAT_NAME="csi-natgw"
NETWORK_NAT_UDP_IDLE_TIMEOUT="60s"
NETWORK_NAT_ICMP_IDLE_TIMEOUT="60s"
NETWORK_NAT_TCP_ESTABLISHED_IDLE_TIMEOUT="60s"
NETWORK_NAT_TCP_TRANSITORY_IDLE_TIMEOUT="60s"
NETWORK_NAT_ALLOW_SUBNET_SECONDARY_IPS=false
K8S_CLUSTER_NAME=${K8S_CLUSTER_NAME:-"csi-k8s-cluster"}
K8S_CLUSTER_PODS_PER_NODE="16"
K8S_CLUSTER_ADDONS="HttpLoadBalancing,HorizontalPodAutoscaling"
GKE_MASTER_VERSION="1.19.9-gke.1900"
GKE_PRIVATE_ACCESS=true
GKE_MASTER_NODE_COUNT_PER_LOCATION=1
GKE_NODE_VERSION="1.19.9-gke.1900"
GKE_IMAGE_TYPE="COS"
GKE_MAINTENANCE_WINDOW='06:00'

```

```
GKE_ENABLE_PRIVATE_ENDPOINT=true
GKE_MASTER_ACCESS_CIDRS="10.128.0.0/9"
K8S_PRIVATE_CIDR="172.16.0.0/20"
K8S_SERVICES_CIDR="172.17.0.0/20"
GCLOUD_NODES_CIDR="192.168.0.0/20"
K8S_API_CIDR="192.171.0.0/28"
K8S_HOST_DISK_SIZE='50GB'
K8S_HOST_DISK_TYPE="pd-standard"
K8S_HOST_MIN_CPU_PLATFORM=""
K8S_POOL_HOSTS_MAX=1000
K8S_METADATA="disable-legacy-endpoints=true"
K8S_MIN_NODES=1
K8S_MAX_NODES=3
GCLOUD_RESOURCE_LABELS='deleteafter=false,owner=user'
GCLOUD_VM_LABELS=description=k8s_cluster
GCLOUD_VM_TAGS="cluster-vm"
GCLOUD_VM_MACHINE_TYPE="n1-standard-1"
GCLOUD_VM_SSD_COUNT=0
GCLOUD_PROJECT_ID=${GCLOUD_PROJECT_ID:-"cloud-project-1592"}
GCLOUD_NETWORK=${GCLOUD_NETWORK:-"devel-network"}
GCLOUD_NODES_SUBNET_SUFFIX="nodes"
GCLOUD_CLUSTER_REGION=${GCLOUD_CLUSTER_REGION:-"us-central1"}
GCLOUD_NODE_LOCATIONS="us-central1-f"
GCLOUD_NODE_TAINTS='key1=val1:NoSchedule,key2=val2:PreferNoSchedule'
GCLOUD_NODE_SCOPE='gke-default'
```

Create the GKE Cluster

After defining the cluster requirements, run the `create_k8s.sh` script in the `gcloud` directory to create the cluster. Run the script with the following command. The arguments are described below.

```
./create_k8s.sh -c <config_file_name> [ -d <config_file_directory> ] [ -f | --force ] [ -h | --help ]
```

-c <config_file_name>

This is a **required** argument that specifies the name of the configuration file that supplies the cluster requirements. For example, `-c k8s_cluster.conf`.

-d <config_file_directory>

This is an **optional** argument that specifies the path and directory name for the configuration file specified for the `-c` argument. If you are using the original `gcloud` directory file structure and the configuration file is in the `conf.d` directory, you do not need to specify the `-d` argument. If you created a separate directory structure for different Anzo environments, include the `-d` option. For example, `-d /gcloud/env1/conf`.

-f | --force

This is an **optional** argument that controls whether the script prompts for confirmation before proceeding with each stage involved in creating the cluster. If **-f (--force)** is specified, the script assumes the answer is "yes" to all prompts and does not display them.

-h | --help

This argument is an **optional** flag that you can specify to display the help from the `create_k8s.sh` script.

For example, the following command runs the `create_k8s` script, using `k8s_cluster.conf` as input to the script. Since `k8s_cluster.conf` is in the `conf.d` directory, the `-d` argument is excluded:

```
./create_k8s.sh -c k8s_cluster.conf
```

The script validates that the required software packages, such as the `gcloud sdk` and `kubectl`, are installed and that the versions are compatible with the deployment. It also displays an overview of the deployment details based on the values in the specified configuration file. For example:

```
Operating System   : CentOS Linux
- Google Cloud SDK: 322.0.0
  alpha: 2021.01.05
  beta: 2021.01.05
  bq: 2.0.64
  core: 2021.01.05
  gsutil: 4.57
  kubectl cli version: Client Version: v1.19.12
  valid

Deployment details:
  Project           : cloud-project-1592
  Region            : us-centrall
  GKE Cluster       : cloud-k8s-cluster
  GKE Master version : 1.19.9-gke.1900
```

The script then prompts you to proceed with deploying each component of the GKE cluster infrastructure. Type **y** and press **Enter** to proceed with creating the specified network, cluster, cloud router, and NAT gateway components. All components are created according to the specifications in the configuration file.

When cluster creation is complete, proceed to [Creating the Required Node Pools](#) to add the required node pools to the cluster.

Related Topics

[Creating and Assigning IAM Roles](#)

[Creating the Required Node Pools](#)

Creating the Required Node Pools

This topic provides instructions for creating the three types of required node pools:

- The **Operator** node pool for running the AnzoGraph, Anzo Agent with Anzo Unstructured (AU), Elasticsearch, and Spark operator pods.
- The **AnzoGraph** node pool for running AnzoGraph application pods.
- The **Dynamic** node pool for running Anzo Agent with AU, Elasticsearch, and Spark application pods.

Tip For more information about the node pools, see [Node Pool Requirements](#).

- [Define the Node Pool Requirements](#)
- [Create the Node Pools](#)

Define the Node Pool Requirements

Before creating the node pools, configure the infrastructure requirements for each type of pool. The `nodepool_*.conf` files in the `gcloud/conf.d` directory are sample configuration files that you can use as templates, or you can edit the files directly:

- `nodepool_operator.conf` defines the requirements for the Operator node pool.
- `nodepool_anzograph.conf` defines the requirements for the AnzoGraph node pool.
- `nodepool_dynamic.conf` defines the requirements for the Dynamic node pool.

Important

The additional AnzoGraph and Dynamic node pool configuration files, `nodepool_anzograph_tuner.yaml` and `nodepool_dynamic_tuner.yaml`, configure the kernel-level tuning and security policies to apply to AnzoGraph and Dynamic runtime environments. Do not make changes to the files. There is a stage during node pool creation when the script prompts, **Do you want to tune the nodepools?**. It is important to answer **y** (yes) so that the kernel tuning and security policies are applied.

Each type of node pool configuration file contains the following parameters. Descriptions of the parameters and guidance on specifying the appropriate values for each type of node pool are provided below.

```
DOMAIN="<domain>"
KIND="<kind>"
GLOUD_CLUSTER_REGION=${GLOUD_CLUSTER_REGION:-"<region>"}
GLOUD_NODE_TAINTS="<node-taints>"
GLOUD_PROJECT_ID=${GLOUD_PROJECT_ID:-"<project>"}
GKE_IMAGE_TYPE="<image-type>"
GKE_NODE_VERSION="<version>"
K8S_CLUSTER_NAME=${K8S_CLUSTER_NAME:-"<cluster>"}
NODE_LABELS="<node-labels>"
MACHINE_TYPES="<machine-type>"
```

```

TAGS="<tags>"
METADATA="<metadata>"
MAX_PODS_PER_NODE=<max-pods-per-node>
MAX_NODES=<max-nodes>
MIN_NODES=<min-nodes>
NUM_NODES=<num-nodes>
DISK_SIZE="<disk-size>"
DISK_TYPE="<disk-type>"

```

DOMAIN

The name of the domain that hosts the node pool. This is typically prefaced with the name of the organization.

Node Pool Type	Sample DOMAIN Value
Operator	csi-operator
AnzoGraph	csi-anzograph
Dynamic	csi-dynamic

KIND

This parameter classifies the node pool in terms of kernel tuning and the type of pods that the node pool will host.

Node Pool Type	Required KIND Value
Operator	operator
AnzoGraph	anzograph
Dynamic	dynamic

GCPLOUD_CLUSTER_REGION

The compute region for the GKE cluster. This parameter maps to the gcloud container cluster `--region` option. For example, `us-central1`.

GCPLOUD_NODE_TAINTS

This parameter defines the type of pods that are allowed to be placed in this node pool. When a pod is scheduled for deployment, the scheduler relies on this value to determine whether the pod belongs in this pool. If a pod has a **toleration** that is not compatible with this **taint**, the pod is rejected from the pool. The recommended values below specify that operator pods are allowed in the Operator node pool, AnzoGraph pods are allowed in the AnzoGraph node pool, and dynamic pods are allowed in the Dynamic node pool. The **NoSchedule** value means a toleration is

required and pods without the appropriate toleration will not be allowed in the pool. In addition, the values specify that pods should not be placed on preemptible nodes.

Node Pool Type	Recommended GCPLOUD_NODE_TAINTS Value
Operator	cambridgesemantics.com/dedicated=operator:NoSchedule, cloud.google.com/gke-preemptible="false":NoSchedule
AnzoGraph	cambridgesemantics.com/dedicated=anzograph:NoSchedule, cloud.google.com/gke-preemptible="false":PreferNoSchedule
Dynamic	cambridgesemantics.com/dedicated=dynamic:NoSchedule, cloud.google.com/gke-preemptible="false":NoSchedule

GCPLOUD_PROJECT_ID

The Project ID for the node pool. This parameter maps to the gcloud-wide `--project` option. The value should match the Project ID for the GKE cluster. For example, **cloud-project-1592**.

GKE_IMAGE_TYPE

The base operating system that the nodes in the node pool will run on. This parameter maps to the gcloud container cluster `--image-type` option. This value must be **cos_containerd**.

GKE_NODE_VERSION

The Kubernetes version to use for nodes in the node pool. This parameter maps to the gcloud container cluster `--node-version` option.

Note

Cambridge Semantics recommends that you specify the same version as the GKE_MASTER_VERSION.

K8S_CLUSTER_NAME

The name of the GKE cluster to add the node pool to. For example, **csi-k8s-cluster**.

NODE_LABELS

A comma-separated list of key/value pairs that define the type of pods that can be placed on the nodes in this node pool. Labels are used to attract pods to nodes, while **taints** ([GCPLOUD_NODE_TAINTS](#)) are used to repel other types of pods from being placed in this node pool.

For example, the following labels specify that the purpose of the nodes in each pool is to host **operator**, **anzograph**, or **dynamic** pods.

Node Pool Type	Recommended NODE_LABELS Value
Operator	cambridgesemantics.com/node-purpose=operator
AnzoGraph	cambridgesemantics.com/node-purpose=anzograph
Dynamic	cambridgesemantics.com/node-purpose=dynamic

MACHINE_TYPES

A space-separated list of the machine types that can be used for the nodes in this node pool. This parameter maps to the gcloud container cluster `--machine-type` option. If you list multiple machine types, the node pool creation script prompts you to create multiple node pools of the same [KIND](#), one pool for each machine type.

Node Pool Type	Sample MACHINE_TYPES Value
Operator	n1-standard-1
AnzoGraph	n1-standard-16 n1-standard-32 n1-standard-64
Dynamic	n1-standard-4

Tip

For more guidance on determining the instance types to use for nodes in the required node pools, see [Compute Resource Planning](#).

TAGS

A comma-separated list of strings to add to the instances in the node pool to classify the VMs. This parameter maps to the gcloud container cluster `--tags` option. For example, **csi-anzo**.

METADATA

The compute engine metadata (in the format `key=val,key=val`) to make available to the guest operating system running on nodes in the node pool. This parameter maps to the gcloud container cluster `--metadata` option.

Important

Including **disable-legacy-endpoints=true** is required to ensure that legacy metadata APIs are disabled. For more information about the option, see [Protecting Cluster Metadata](#) in the GKE documentation.

MAX_PODS_PER_NODE

The maximum number of pods that can be hosted on a node in this node pool. This parameter maps to the gcloud container cluster `--max-pods-per-node` option. In addition to Anzo application pods, this limit also needs to account for K8s service pods and helper pods. Cambridge Semantics recommends that you set this value to at least **16** for all node pool types.

MAX_NODES

The maximum number of nodes in the node pool. This parameter maps to the gcloud container cluster `--max-nodes` option.

Node Pool Type	Sample MAX_NODES Value
Operator	8
AnzoGraph	64
Dynamic	64

MIN_NODES

The minimum number of nodes in the node pool. This parameter maps to the gcloud container cluster `--min-nodes` option. If you set the minimum nodes to **0** for each node pool type, nodes will not be provisioned unless the relevant type of pod is scheduled for deployment.

NUM_NODES

The number of nodes to deploy when the node pool is created. This value must be set to at least **1**. When you create the node pool, at least one node in the pool needs to be deployed as well. However, if the GKE cluster autoscaler addon is enabled, the autoscaler will deprovision this node because it is not in use.

Note

Depending on the version of gcloud that you are using, you may be able to set NUM_NODES to **0**. Recent versions of gcloud added support for creating node pools without deploying any nodes.

DISK_SIZE

The size of the boot disks on the nodes. This parameter maps to the gcloud container cluster `--disk-size` option.

Node Pool Type	Sample DISK_SIZE Value
Operator	50GB

Node Pool Type	Sample DISK_SIZE Value
AnzoGraph	200GB
Dynamic	100GB

DISK_TYPE

The type of boot disk to use. This parameter maps to the gcloud container cluster `--disk-type` option.

Node Pool Type	Sample DISK_TYPE Value
Operator	pd-standard
AnzoGraph	pd-ssd
Dynamic	pd-ssd

Example Configuration Files

Example completed configuration files for each type of node pool are shown below.

Operator Node Pool

The example below shows a configured `nodepool_operator.conf` file.

```
DOMAIN="csi-operator"
KIND="operator"
GCLOUD_NODE_
TAINTS="cambridgesemantics.com/dedicated=operator:NoSchedule,cloud.google.com/gke-
preemptible="false":NoSchedule"
GCLOUD_CLUSTER_REGION=${GCLOUD_CLUSTER_REGION:-"us-central1"}
GKE_IMAGE_TYPE="cos_containerd"
GKE_NODE_VERSION="1.19.9-gke.1900"
GCLOUD_PROJECT_ID=${GCLOUD_PROJECT_ID:-"cloud-project-1592"}
K8S_CLUSTER_NAME=${K8S_CLUSTER_NAME:-"csi-k8s-cluster"}
NODE_LABELS="cambridgesemantics.com/node-
purpose=operator,cambridgesemantics.com/description=k8snode"
MACHINE_TYPES="n1-standard-1"
TAGS="csi-anzo"
METADATA="disable-legacy-endpoints=true"
MAX_PODS_PER_NODE=16
MAX_NODES=8
MIN_NODES=0
NUM_NODES=1
DISK_SIZE="50Gb"
DISK_TYPE="pd-standard"
```

AnzoGraph Node Pool

The example below shows a configured `nodepool_anzograph.conf` file.

```
DOMAIN="csi-anzograph"
KIND="anzograph"
GCLOUD_CLUSTER_REGION=${GCLOUD_CLUSTER_REGION:-"us-central1"}
GCLOUD_NODE_
TAINTS="cambridgesemantics.com/dedicated=anzograph:NoSchedule,cloud.google.com/gke-
preemptible="false":PreferNoSchedule"
GCLOUD_PROJECT_ID=${GCLOUD_PROJECT_ID:-"cloud-project-1592"}
GKE_IMAGE_TYPE="cos_containerd"
GKE_NODE_VERSION="1.19.9-gke.1900"
K8S_CLUSTER_NAME=${K8S_CLUSTER_NAME:-"csi-k8s-cluster"}
NODE_LABELS="cambridgesemantics.com/node-
purpose=anzograph,cambridgesemantics.com/description=k8snode"
MACHINE_TYPES="n1-standard-16 n1-standard-32 n1-standard-64"
TAGS="csi-anzo"
METADATA="disable-legacy-endpoints=true"
MAX_PODS_PER_NODE=16
MAX_NODES=64
MIN_NODES=0
NUM_NODES=1
DISK_SIZE="200Gb"
DISK_TYPE="pd-ssd"
```

Dynamic Node Pool

The example below shows a configured `nodepool_dynamic.conf` file.

```
DOMAIN="csi-dynamic"
KIND="dynamic"
GCLOUD_CLUSTER_REGION=${GCLOUD_CLUSTER_REGION:-"us-central1"}
GCLOUD_NODE_
TAINTS="cambridgesemantics.com/dedicated=dynamic:NoSchedule,cloud.google.com/gke-
preemptible="false":NoSchedule"
GCLOUD_PROJECT_ID=${GCLOUD_PROJECT_ID:-"cloud-project-1592"}
GKE_IMAGE_TYPE="cos_containerd"
GKE_NODE_VERSION="1.19.9-gke.1900"
K8S_CLUSTER_NAME=${K8S_CLUSTER_NAME:-"csi-k8s-cluster"}
NODE_LABELS="cambridgesemantics.com/node-
purpose=anzograph,cambridgesemantics.com/description=k8snode"
MACHINE_TYPES="n1-standard-4"
TAGS="csi-anzo"
METADATA="disable-legacy-endpoints=true"
MAX_PODS_PER_NODE=16
MAX_NODES=64
```

```
MIN_NODES=0
NUM_NODES=1
DISK_SIZE="100Gb"
DISK_TYPE="pd-ssd"
```

Create the Node Pools

After defining the requirements for the node pools, run the **create_nodepools.sh** script in the `gcloud` directory to create each type of node pool. Run the script with the following command. Run it once for each type of pool. The arguments are described below.

```
./create_nodepools.sh -c <config_file_name> [ -d <config_file_directory> ] [ -f | --force ] [ -h | --help ]
```

-c <config_file_name>

This is a **required** argument that specifies the name of the configuration file (i.e., `nodepool_operator.conf`, `nodepool_anzograph.conf`, or `nodepool_dynamic.conf`) that supplies the node pool requirements. For example, **-c nodepool_dynamic.conf**.

-d <config_file_directory>

This is an **optional** argument that specifies the path and directory name for the configuration file specified for the **-c** argument. If you are using the original `gcloud` directory file structure and the configuration file is in the `conf.d` directory, you do not need to specify the **-d** argument. If you created a separate directory structure for different Anzo environments, include the **-d** option. For example, **-d /gcloud/env1/conf**.

-f | --force

This is an **optional** argument that controls whether the script prompts for confirmation before proceeding with each stage involved in creating the node pool. If **-f (--force)** is specified, the script assumes the answer is "yes" to all prompts and does not display them.

-h | --help

This argument is an **optional** flag that you can specify to display the help from the `create_nodepools.sh` script. For example, the following command runs the `create_nodepools` script, using `nodepool_operator.conf` as input to the script. Since `nodepool_operator.conf` is in the `conf.d` directory, the **-d** argument is excluded:

```
./create_nodepools.sh -c nodepool_operator.conf
```

The script validates that the required software packages are installed and that the versions are compatible with the deployment. It also displays an overview of the deployment details based on the values in the specified configuration file. For example:

```

Operating System   : CentOS Linux
- Google Cloud SDK: 322.0.0
  alpha: 2021.01.05
  beta: 2021.01.05
  bq: 2.0.64
  core: 2021.01.05
  gsutil: 4.57
  kubectl cli version: Client Version: v1.19.12
  valid

```

Deployment details:

```

Project           : cloud-project-1592
Region            : us-centrall
GKE Cluster       : csi-k8s-cluster

```

The script then prompts you to proceed with deploying each component of the node pool. Type **y** and press **Enter** to proceed with the configuration.

Important

When creating the AnzoGraph and Dynamic node pools, there is a stage when the script prompts, **Do you want to tune the nodepools?**. It is important to answer **y** (yes) so that the kernel tuning and security policies from the related nodepool_*_tuner.yaml file are applied to the node pool configuration.

Once the Operator, AnzoGraph, and Dynamic node pools are created, the next step is to create a Cloud Location in Anzo so that Anzo can connect to the GKE cluster and deploy applications. See [Connecting to a Cloud Location](#).

Related Topics

[Creating the GKE Cluster](#)

[Connecting to a Cloud Location](#)

Azure Kubernetes Service Deployments

The topics in this section guide you through the process of deploying all of the Azure Kubernetes Service (AKS) infrastructure that is required to support dynamic deployments of Anzo components. The topics provide instructions for setting up a workstation to use for deploying the K8s infrastructure, performing the prerequisite tasks before deploying the AKS cluster, creating the AKS cluster, and creating the required node pools.

- [Setting Up a Workstation](#)
- [Planning the Anzo and AKS Network Architecture](#)
- [Creating and Assigning IAM Roles](#)
- [Creating the AKS Cluster](#)
- [Creating the Required Node Pools](#)

Setting Up a Workstation

This topic provides the requirements and instructions to follow for configuring a workstation to use for creating and managing the AKS infrastructure. The workstation needs to be able to connect to the Azure API. It also needs to have the required Azure and Kubernetes (K8s) software packages as well as the deployment scripts and configuration files supplied by Cambridge Semantics. This workstation will be used to connect to the Azure API and provision the K8s cluster and node pools.

Note

You can use the Anzo server as the workstation if the network routing and security policies permit the Anzo server to access the Azure and K8s APIs. When deciding whether to use the Anzo server as the K8s workstation, consider whether Anzo may be migrated to a different server or VPC in the future.

- [Review the Requirements and Install the Software](#)
- [Download the Cluster Creation Scripts and Configuration Files](#)

Review the Requirements and Install the Software

Component	Requirement
Operating System	The operating system for the workstation must be RHEL/CentOS 7.8 or higher .
Networking	The workstation should be in the same VPC network as the AKS cluster. If it is not in the same VPC, make sure that it is on a network that is routable from the cluster's VPC.
Software	<ul style="list-style-type: none">• Python 3 is required.• Kubectl Versions 1.18 and 1.19 are supported. Cambridge Semantics recommends that you use the same kubectl version as the AKS cluster version. For instructions, see Install Kubectl below.• Azure CLI Version 2.5.1 or later is required. For installation instructions, see Install Azure CLI below.
CSI AZ Package	Cambridge Semantics provides az scripts and configuration files to use for provisioning the AKS cluster and node pools. Download the files to the workstation. See Download the Cluster Creation Scripts and Configuration Files for more information about the az package.

Install Kubectl

Follow the instructions below to install kubectl on your workstation. Cambridge Semantics recommends that you install the same version of kubectl as the K8s cluster API. For more information, see [Install and Set Up kubectl on Linux](#) in the Kubernetes documentation.

1. Run the following cURL command to download the kubectl binary:

```
curl -LO https://dl.k8s.io/release/<version>/bin/linux/amd64/kubectl
```

Where <version> is the version of kubectl to install. For example, the following command downloads version 1.19.12:

```
curl -LO https://dl.k8s.io/release/v1.19.12/bin/linux/amd64/kubectl
```

2. Run the following command to make the binary executable:

```
chmod +x ./kubectl
```

3. Run the following command to move the binary to your PATH:

```
sudo mv ./kubectl /usr/local/bin/kubectl
```

4. To confirm that the binary is installed and that you can run kubectl commands, run the following command to display the client version:

```
kubectl version --client
```

The command returns the following type of information. For example:

```
Client Version: version.Info{Major:"1", Minor:"19", GitVersion:"v1.19.12",  
GitCommit:"f3abc15296f3a3f54e4ee42e830c61047b13895f",  
GitTreeState:"clean", BuildDate:"2021-06-16T13:21:12Z", GoVersion:"go1.13.15",  
Compiler:"gc", Platform:"linux/amd64"}
```

Install Azure CLI

Follow the instructions below to install the Azure CLI on your workstation. These instructions follow the steps in [Install the Azure CLI on Linux](#) in the Microsoft Azure CLI documentation.

1. Run the following command to import the Microsoft repository key:

```
sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
```

2. Run the following command to create the local azure-cli repository information:

```
echo -e "[azure-cli]
name=Azure CLI
baseurl=https://packages.microsoft.com/yumrepos/azure-cli
enabled=1
gpgcheck=1
gpgkey=https://packages.microsoft.com/keys/microsoft.asc" | sudo tee
/etc/yum.repos.d/azure-cli.repo
```

3. Run the following command to install the CLI:

```
sudo yum install azure-cli
```

4. To ensure that the CLI was installed, run the following command to display the CLI version:

```
az version
```

5. Next, run the following command to run the Azure CLI. Follow the prompts to log in to Azure:

```
az login --use-device-code
```

Download the Cluster Creation Scripts and Configuration Files

The Cambridge Semantics GitHub repository, [k8s-genesis](https://github.com/cambridgesemantics/k8s-genesis.git) (<https://github.com/cambridgesemantics/k8s-genesis.git>), includes all of the files that are needed to manage the configuration, creation, and deletion of the AKS cluster and node pools.

You can clone the repository to any location on the workstation or download the k8s-genesis package as a ZIP file, copy the file to the workstation, and extract the contents. The k8s-genesis directory includes three subdirectories (one for each supported Cloud Service Provider), the license information, and a readme file:

```
k8s-genesis
├─ aws
├─ azure
├─ gcp
├─ LICENSE
└─ README.md
```

Navigate to `/azure/k8s/az`. The **az** directory contains all of the AKS cluster and node pool configuration files. You can remove all other directories from the workstation. The az files and subdirectories are shown below:

```
az
├─ common.sh
├─ conf.d
│   └─ k8s_cluster.conf
│   └─ nodepool_anzograph.conf
```

```

|   ├── nodepool_common.conf
|   ├── nodepool.conf
|   ├── nodepool_dynamic.conf
|   └── nodepool_operator.conf
├── create_k8s.sh
├── create_nodepools.sh
├── delete_k8s.sh
├── delete_nodepools.sh
├── exec_samples
|   ├── rbac_aad_group.yaml
|   └── rbac_aad_user.yaml
├── permissions
|   ├── aks_admin_role.json
|   └── cluster_developer_role.json
├── README.md
├── reference
|   ├── nodepool_anzograph_tuner.yaml
|   └── nodepool_dynamic_tuner.yaml
└── sample_use_cases
    ├── 10_useExistingResources
    |   └── k8s_cluster.conf
    ├── 11_useProximityPlacementGroups
    |   └── k8s_cluster.conf
    ├── 1_azureManagedIdentity_private_cluster
    |   └── k8s_cluster.conf
    ├── 2_createServicePrincipal_public_cluster
    |   └── k8s_cluster.conf
    ├── 3_useServicePrincipal
    |   └── k8s_cluster.conf
    ├── 4_userManagedAAD
    |   └── k8s_cluster.conf
    ├── 5_azureManagedAAD
    |   └── k8s_cluster.conf
    ├── 6_attachACR
    |   └── k8s_cluster.conf
    ├── 7_clusterAutoscalerSupport
    |   └── k8s_cluster.conf
    ├── 8_MonitoringEnabled
    |   └── k8s_cluster.conf
    └── 9_RBACSupport
        └── k8s_cluster.conf

```

The following list gives an overview of the files. Subsequent topics describe the files in more detail.

- The **common.sh** script is used by the create and delete cluster and node pool scripts.
- The **conf.d** directory contains the configuration files that are used to supply the specifications to follow when creating the K8s cluster and node pools:
 - **k8s_cluster.conf**: Supplies the specifications for the AKS cluster.
 - **nodepool_anzograph.conf**: Supplies the specifications for the AnzoGraph node pool.
 - **nodepool_common.conf**: Supplies the specifications for a Common node pool. The Common node pool is not required for AKS deployments, and this configuration file is typically not used.
 - **nodepool.conf**: This file is supplied as a reference. It contains the super set of node pool parameters.
 - **nodepool_dynamic.conf**: Supplies the specifications for the Dynamic node pool.
 - **nodepool_operator.conf**: Supplies the specifications for the Operator node pool.
- The **create_k8s.sh** script is used to deploy the AKS cluster, and the **k8s_cluster.conf** file in the **conf.d** directory is the configuration file that is input to the **create_k8s.sh** script.
- The **create_nodepools.sh** script is used to deploy the required node pools in the AKS cluster. The **nodepool_*.conf** files in the **conf.d** directory are the configuration files that are input to the **create_nodepools.sh** script.
- The **delete_k8s.sh** script is used to delete the AKS cluster.
- The **delete_nodepools.sh** script is used to remove node pools from the AKS cluster.
- The **exec_samples** and **permissions** directories contain role definitions and scripts for creating the custom roles that are needed to grant access to the Azure users and groups who will create or use the AKS cluster.
- The **reference** directory contains crucial files that are referenced by the cluster and node pool creation scripts. The files in the directory should not be edited, and the **reference** directory must exist on the workstation at the same level as the **create*.sh** and **delete*.sh** scripts.
- The **sample_use_cases** directory contains sample AKS cluster configuration files that you can refer to or use as a template for configuring your AKS cluster depending on your use case. There are several files in the directory because there is an example for each type of AKS-supported identity and authentication management option. You can use a combination of settings from different sample files to configure your cluster, but you can only choose one type of authentication mode. For example, you cannot enable Service Principals with Azure Active Directory.
 - The **k8s_cluster.conf** file in the **1_azureManagedIdentity_private_cluster** directory is a sample file for a use case where you want to deploy the AKS cluster into a private Virtual Network and let Azure handle identity creation and management. Using an Azure managed identity is recommended.
 - The **k8s_cluster.conf** file in the **2_createServicePrincipal_public_cluster** directory is a sample file for a use case where you want to create a new Service Principal to deploy a public AKS cluster. Access to the cluster is limited to certain IP ranges. Managing Service Principals adds more complexity than using an Azure managed identity.

- The **k8s_cluster.conf** file in the **3_useServicePrincipal** directory is a sample file for a use case that is similar to the **2_createServicePrincipal_public_cluster** use case above but uses an existing Service Principal.
- The **k8s_cluster.conf** file in the **4_userManagedAAD** directory is a sample file for a use case where you want to deploy an AKS cluster that connects to your user-managed Azure Active Directory (AAD) server for identity management. You supply the AAD client and server applications and the AAD tenant.
- The **k8s_cluster.conf** file in the **5_azureManagedAAD** directory is a sample file for a use case where you want to deploy an AKS cluster that connects to an Azure-managed Azure Active Directory (AAD) server for identity management. In this case, the AKS resource provider manages the client and server AAD applications.
- The **k8s_cluster.conf** file in the **6_attachACR** directory is a sample file for a use case where you want to deploy an AKS cluster that retrieves images from a private Azure Container Registry.
- The **k8s_cluster.conf** file in the **7_clusterAutoscalerSupport** directory is a sample file for a use case where you want to deploy an AKS cluster that employs the Cluster Autoscaler service. The autoscaler automatically adds nodes to the node pool when demand increases and then deprovisions the nodes when demand decreases.
- The **k8s_cluster.conf** file in the **8_MonitoringEnabled** directory is a sample file for a use case where you want to deploy an AKS cluster with cluster monitoring enabled.
- The **k8s_cluster.conf** file in the **9_RBACSupport** directory is a sample file for a use case where you want to deploy an AKS cluster with Azure Role-Based Access Control (RBAC). Enabling RBAC allows you to use Azure AD users, groups, or service principals as subjects in Kubernetes RBAC.
- The **k8s_cluster.conf** file in the **10_useExistingResources** directory is a sample file for a use case where you want to deploy the AKS cluster into existing resources, such as an existing Virtual Network with existing resource groups and subnetworks.
- The **k8s_cluster.conf** file in the **11_useProximityPlacementGroups** directory is a sample file for a use case where you want to use proximity placement groups for reduced latency. A proximity placement group is a logical grouping used to make sure Azure compute resources are physically located close to each other.

Once the workstation is configured, see [Planning the Anzo and AKS Network Architecture](#) to review information about the network architecture that the az scripts create. And see [Creating and Assigning IAM Roles](#) for instructions on creating the IAM roles that are needed for assigning permissions to create and use the AKS cluster.

Related Topics

[Planning the Anzo and AKS Network Architecture](#)

[Creating and Assigning IAM Roles](#)

[Creating the AKS Cluster](#)

[Creating the Required Node Pools](#)

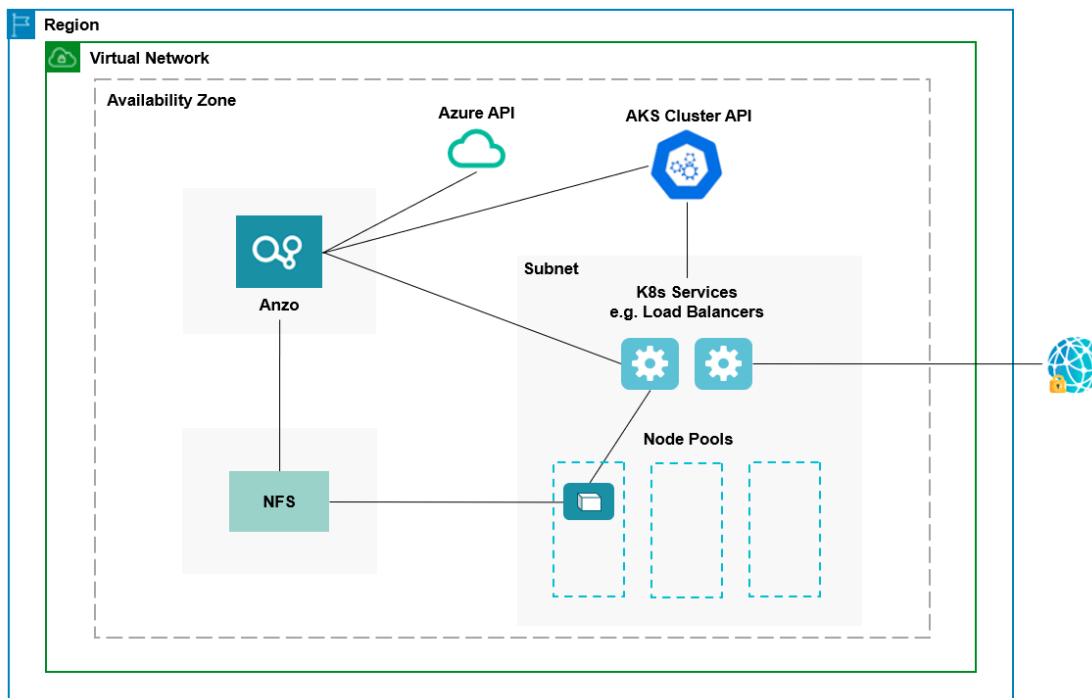
Planning the Anzo and AKS Network Architecture

This topic describes the network architecture that supports the Anzo and AKS integration.

Note

When you deploy the K8s infrastructure, Cambridge Semantics strongly recommends that you create the AKS cluster in the same Virtual Network as Anzo. If you create the AKS cluster in a new Virtual Network, you must configure the new network to be routable from the Anzo Virtual Network.

The diagram below shows the typical network components that are employed when an AKS cluster is integrated with Anzo. Most of the network resources shown in the diagram are automatically deployed (and the appropriate routing is configured) according to the values that you supply in the cluster and node group .conf files in the `az` package on the workstation.



In the diagram, there are two components that you deploy before configuring and creating the K8s resources:

- **Anzo:** Since the Anzo server is typically deployed before the K8s components, you specify the Anzo network when creating the AKS cluster, ensuring that Anzo and all of the AKS cluster components are in the same network and can talk to each other. Also, make sure that Anzo has access to the Azure and AKS APIs.
- **NFS:** You are required to create a network file system (NFS). However, Anzo automatically mounts the NFS to the nodes when AnzoGraph, Anzo Unstructured, Spark, and Elasticsearch pods are deployed so that all of the applications can share files. See [Deploying the Shared File System](#) for more information. The NFS does not need to have its own subnet but it can.

The rest of the components in the diagram are automatically provisioned, depending on your specifications, when the AKS cluster and node pools are created. The az scripts can be used to create a subnet for the K8s services and node pools and configure the routing so that Anzo can communicate with the K8s services and the services can talk to the pods that are deployed in the node pools. In addition, a Standard Load Balancer can be used to provide outbound internet access, such as for pulling container images from the Cambridge Semantics repository.

Tip

When considering the network requirements of your organization and planning how to integrate the new K8s infrastructure in accordance with those requirements, it may help to consider the following types of use cases. Cambridge Semantics supplies sample cluster configuration files in the `az/sample_use_cases` directory that are tailored for each of these use cases:

- **Deploy a private AKS cluster with Azure Managed Identity**

In this use case, the AKS cluster is deployed as a private cluster with no public access, and the Azure Managed Identity service is enabled for identity and authorization management. Using Azure Managed Identity is the recommended method to choose for AKS access control.

- **Deploy a public AKS cluster with a new Service Principal**

In this use case, the AKS cluster is deployed as a public cluster, and a Service Principal is created for managing access control. You are responsible for maintaining the Service Principal to keep the cluster functional.

- **Deploy a public AKS cluster with an existing new Service Principal**

This use case is similar to the use case described above but uses an existing Service Principal instead of creating a new one.

- **Deploy a private AKS cluster with a user-managed Azure Active Directory server**

In this use case, the AKS cluster is deployed as a private cluster and a user-managed Azure Active Directory (AAD) server is used for identity and authorization management. In this case, you supply the AAD client and server applications and the AAD tenant.

- **Deploy a private AKS cluster with an Azure-managed AAD server**

In this use case, the AKS cluster is deployed as a private cluster and an Azure-managed AAD server is used for identity and authorization management. In this case, the AKS resource manager manages the AAD client and server applications.

- **Deploy an AKS cluster and access a private Azure Container Registry**

In this use case, an AKS cluster is deployed and accesses images that are maintained in a private Azure Container Registry.

- **Deploy an AKS cluster that Auto Scales on Demand**

In this use case, an AKS cluster is deployed and the Cluster Autoscaler service is enabled. The Cluster Autoscaler automatically adds nodes to the node pool when demand increases and deprovisions nodes when demand decreases.

- **Deploy an AKS with the Monitoring Addon**

In this use case, an AKS cluster is deployed and the Log Analytics monitoring service is enabled.

- **Deploy an AKS cluster with RBAC enabled**

In this use case, an AKS cluster is deployed and Role-Based Access Control (RBAC) is enabled. RBAC manages Kubernetes user identities and credentials. RBAC can be enabled in conjunction with other authorization modes, such as Azure Managed Identity or AAD.

- **Deploy an AKS cluster using existing resources**

In this use case, an AKS cluster is deployed without creating new network components. The cluster is deployed into an existing Virtual Network and uses existing resource groups and subnetworks.

- **Deploy an AKS cluster with Proximity Placement Groups**

In this case, an AKS cluster is deployed with specified Proximity Placement Groups to ensure that compute resources are deployed physically close to each other to reduce latency.

For a summary of the files in the az directory, see [Download the Cluster Creation Scripts and Configuration Files](#). Specifics about the parameters in the sample files are included in [Creating the AKS Cluster](#).

To get started on creating the AKS infrastructure, see [Creating and Assigning IAM Roles](#) for instructions on creating the IAM roles that are needed for assigning permissions to create and use the AKS cluster.

Related Topics

[Setting Up a Workstation](#)

[Creating and Assigning IAM Roles](#)

[Creating the AKS Cluster](#)

[Creating the Required Node Pools](#)

Creating and Assigning IAM Roles

This topic provides instructions for creating the Identity and Access Management (IAM) roles that are needed to supply the necessary permissions for creating and managing the AKS cluster and using the cluster to deploy applications.

Note

AKS is typically configured to use Azure Active Directory (AD) for user authentication. AKS integration with Azure AD is optional but highly recommended. For more information, see [Azure Active Directory Integration](#) in the AKS documentation.

There are two custom roles that need to be created in Azure to grant the necessary permissions to the following two types of AKS users:

1. The first type of user is the user who sets up the K8s infrastructure, i.e., the user who configures, creates, and maintains the AKS cluster and node pools. This policy is called the **AKS Cluster Admin**.
2. The second type of user is the user who connects to the AKS cluster and deploys the dynamic Anzo applications. Typically this user is Anzo. Since Anzo communicates with the K8s services that provision the applications, the Anzo service principal needs to be granted certain privileges. This user role is called the **AKS Cluster Developer**.

Note

The enterprise-level Anzo service principal is a requirement for the Anzo installation and is typically in place before Anzo is installed. For more information, see [Anzo Service Account Requirements](#).

This topic provides instructions for creating the two roles and gives guidance on assigning the roles to the appropriate users, groups, or service principals.

- [Create and Assign the AKS Cluster Admin Role](#)
- [Create and Assign the AKS Cluster Developer Role](#)

Create and Assign the AKS Cluster Admin Role

The following IAM role applies the minimum permissions needed for an AKS Cluster Admin who will create and manage the AKS cluster and node pools. Follow the instructions below to create the role and assign it to the user, group, or service principal that will be used when creating the K8s infrastructure.

Note

The `az` file package on the workstation includes the configuration file that defines the AKS Cluster Admin role: `az/permissions/aks_admin_role.json`.

1. Open the `az/permissions/aks_admin_role.json` file for editing. At the bottom of the file, replace `<subscription_id>` with the ID for the subscription to attach the new AKS Cluster Admin role to. Then save and close the file. The contents of `aks_admin_role.json` are shown below:

```
{
  "Name": "AKS Cluster Admin",
```

```

    "IsCustom": true,
    "Description": "AKS cluster admin role.",
    "Actions": [
        "Microsoft.Resources/subscriptions/resourcegroups/read",
        "Microsoft.Resources/subscriptions/resourcegroups/write",
        "Microsoft.Resources/subscriptions/resourcegroups/delete",
        "Microsoft.Network/virtualNetworks/read",
        "Microsoft.Network/virtualNetworks/write",
        "Microsoft.Network/virtualNetworks/delete",
        "Microsoft.Network/virtualNetworks/subnets/read",
        "Microsoft.Network/virtualNetworks/subnets/write",
        "Microsoft.Network/virtualNetworks/subnets/delete",
        "Microsoft.Network/virtualNetworks/subnets/join/action",
        "Microsoft.Network/publicIPPrefixes/read",
        "Microsoft.Network/publicIPPrefixes/write",
        "Microsoft.Network/publicIPPrefixes/delete",
        "Microsoft.Network/publicIPPrefixes/join/action",
        "Microsoft.Authorization/roleAssignments/read",
        "Microsoft.Authorization/roleAssignments/write",
        "Microsoft.Authorization/roleAssignments/delete",
        "Microsoft.Resources/deployments/write",
        "Microsoft.ContainerService/managedClusters/read",
        "Microsoft.ContainerService/managedClusters/write",
        "Microsoft.ContainerService/managedClusters/delete",
        "Microsoft.ContainerService/managedClusters/agentPools/read",
        "Microsoft.ContainerService/managedClusters/agentPools/write",
        "Microsoft.ContainerService/managedClusters/agentPools/delete",
        "Microsoft.ContainerService/managedClusters/listClusterAdminCredential/action",
        "Microsoft.OperationsManagement/solutions/read",
        "Microsoft.OperationsManagement/solutions/write",
        "Microsoft.OperationalInsights/workspaces/read",
        "Microsoft.OperationalInsights/workspaces/sharedkeys/read",
        "Microsoft.ContainerRegistry/registries/read"
    ],
    "NotActions": [

    ],
    "AssignableScopes": [
        "/subscriptions/<subscription_id>"
    ]
}

```

2. Next, run the following Azure CLI command to create a custom role definition based on `aks_admin_role.json`. For information about managing role definitions, see [az role definition](#) in the Azure CLI documentation.

```
az role definition create --role-definition cluster-admin-role.json
```

- Once the role is defined in Azure, run the following command to assign the role to the user, group, or service principal who will create and manage the AKS cluster. For information about managing role assignments, see [az role assignment](#) in the Azure CLI documentation.

```
az role assignment create --assignee "<user_group_or_sp_name_or_id>" --role "<role_name_or_id>"
```

Create and Assign the AKS Cluster Developer Role

The following IAM role applies the minimum permissions needed for the AKS Cluster Developer role. Follow the instructions below to create the role and assign it to the Anzo service account.

Note

The `az` file package on the workstation includes the configuration file that defines the AKS Cluster Developer role: `az/permissions/cluster_developer_role.json`.

- Open the `az/permissions/cluster_developer_role.json` file for editing. At the bottom of the file, replace `<subscription_id>` with the ID for the subscription to attach the new AKS Cluster Developer role to. Then save and close the file. The contents of `cluster_developer_role.json` are shown below:

```
{
  "Name": "AKS Cluster Developer",
  "IsCustom": true,
  "Description": "AKS cluster developer role.",
  "Actions": [
    "Microsoft.ContainerService/managedClusters/listClusterUserCredential/action"
  ],
  "NotActions": [
  ],
  "AssignableScopes": [
    "/subscriptions/<subscription_id>"
  ]
}
```

- Next, run the following Azure CLI command to create a custom role definition based on `cluster_developer_role.json`.

```
az role definition create --role-definition cluster_developer_role.json
```

For more information about managing role definitions in Azure, see [az role definition](#) in the Azure CLI documentation.

3. Once the role is defined in Azure, run the following command to assign the role to the Anzo service principal.

```
az role assignment create --assignee "<anzo_sp>" --role "<role_name_or_id>"
```

For more information about managing role assignments in Azure, see [az role assignment](#) in the Azure CLI documentation.

Once the IAM roles are in place and users are granted access, proceed to [Creating the AKS Cluster](#) for instructions on configuring and creating the cluster.

Related Topics

[Setting Up a Workstation](#)

[Planning the Anzo and AKS Network Architecture](#)

[Creating the AKS Cluster](#)

[Creating the Required Node Pools](#)

Creating the AKS Cluster

Follow the instructions below to define the AKS cluster resource requirements and then create the cluster based on your specifications.

Note

For integration with Anzo, Kubernetes versions 1.18 and 1.19 are supported. See the [AKS Engine Release Notes](#) for details about the available versions.

- [Define the AKS Cluster Requirements](#)
- [Create the AKS Cluster](#)

Define the AKS Cluster Requirements

The first step in creating the K8s cluster is to define the infrastructure specifications. The configuration file to use for defining the specifications is called **k8s_cluster.conf**. Multiple sample **k8s_cluster.conf** files are included in the **az** directory. Any of them can be copied and used as templates, or the files can be edited directly.

Sample k8s_cluster.conf Files

To help guide you in choosing the appropriate template for your use case, this section describes each of the sample files. Details about the parameters in the sample files are included in [Cluster Parameters](#) below.

Note

There are several sample use case files because there is an example for each type of AKS-supported identity and authentication management option. You can use a combination of settings from different sample files to configure your cluster, but you can only choose one type of authentication. For example, you

cannot configure Service Principals and enable Azure Active Directory.

az/conf.d/k8s_cluster.conf

This file is a non-specific use case. It includes sample values for all of the available cluster parameters.

az/sample_use_cases/1_azureManagedIdentity_private_cluster/k8s_cluster.conf

This file includes sample values for a use case where:

- A private AKS cluster is deployed (`PRIVATE_CLUSTER="true"`) so that the cluster is only accessible from within the Virtual Network or a connected network.
- Azure Managed Identity is enabled (`ENABLE_MANAGED_IDENTITY="true"`) so that Azure manages identity creation and management. Using Azure Managed Identity is highly recommended.

az/sample_use_cases/2_createServicePrincipal_public_cluster/k8s_cluster.conf

This file includes sample values for a use case where:

- A public AKS cluster is deployed (`PRIVATE_CLUSTER="false"`).
- A Service Principal is created (`SP=${SP:-"<service-principal>"}`) that must be renewed and managed by you.
- Public access to the cluster can be limited to certain IP ranges by specifying the approved ranges in the `API_SERVER_AUTHORIZED_IP_RANGES` parameter.

az/sample_use_cases/3_useServicePrincipal/k8s_cluster.conf

This file includes sample values for a use case where:

- A public AKS cluster is deployed (`PRIVATE_CLUSTER="false"`).
- An existing Service Principal is used for identity and access management. The `SP_ID` and `SP_SECRET` parameters are used to specify the ID and secret for the existing Service Principal.
- Public access to the cluster can be limited to certain IP ranges by specifying the approved ranges in the `API_SERVER_AUTHORIZED_IP_RANGES` parameter.

az/sample_use_cases/4_userManagedAAD/k8s_cluster.conf

This file includes sample values for a use case where:

- A private AKS cluster is deployed (`PRIVATE_CLUSTER="true"`) so that the cluster is only accessible from within the Virtual Network or a connected network.
- An existing Azure Active Directory (AAD) server is used for identity and authorization management. Details about the existing AAD client and server applications as well as the tenet ID need to be specified in the `AAD_CLIENT_APP_ID`, `AAD_SERVER_APP_ID`, `AAD_SERVER_APP_SECRET`, and `AAD_TENANT_ID` parameters.

az/sample_use_cases/5_azureManagedAAD/k8s_cluster.conf

This file includes sample values for a use case where:

- A private AKS cluster is deployed (`PRIVATE_CLUSTER="true"`) so that the cluster is only accessible from within the Virtual Network or a connected network.
- An Azure-managed Active Directory (AAD) server is enabled (`ENABLE_AAD="true"`).
- The AKS resource provider manages the AAD client and server applications.

az/sample_use_cases/6_attachACR/k8s_cluster.conf

This file includes sample values for a use case where:

- A private AKS cluster is deployed (`PRIVATE_CLUSTER="true"`) so that the cluster is only accessible from within the Virtual Network or a connected network.
- The cluster is configured to retrieve images from an existing private Azure Container Registry (ACR) by specifying the name of the ACR in the `ATTACH_ACR` parameter.

az/sample_use_cases/7_clusterAutoscalerSupport/k8s_cluster.conf

This file includes sample values for a use case where:

- A private AKS cluster is deployed (`PRIVATE_CLUSTER="true"`) so that the cluster is only accessible from within the Virtual Network or a connected network.
- The Cluster Autoscaler service is enabled (`ENABLE_CLUSTER_AUTOSCALER="true"`) so that nodes are automatically added to the node pool when demand increases and removed from the node pool when demand decreases.
- The parameter `CLUSTER_AUTOSCALER_PROFILE` parameter is used to configure the autoscaler.

az/sample_use_cases/8_MonitoringEnabled/k8s_cluster.conf

This file includes sample values for a use case where:

- A private AKS cluster is deployed (`PRIVATE_CLUSTER="true"`) so that the cluster is only accessible from within the Virtual Network or a connected network.
- The Monitoring service is enabled (`AKS_ENABLE_ADDONS="monitoring"`) for the cluster.

az/sample_use_cases/9_RBACSupport/k8s_cluster.conf

This file includes sample values for a use case where:

- A private AKS cluster is deployed (`PRIVATE_CLUSTER="true"`) so that the cluster is only accessible from within the Virtual Network or a connected network.
- Azure Role-Based Access Control (RBAC) is enabled (`DISABLE_RBAC="false"`).

az/sample_use_cases/10_useExistingResources/k8s_cluster.conf

This file includes sample values for a use case where:

- A private AKS cluster is deployed (`PRIVATE_CLUSTER="true"`) so that the cluster is only accessible from within the Virtual Network or a connected network.
- You deploy the cluster into your existing Resource Group, Virtual Network, and Subnetwork by specifying the values for those resources in the `RESOURCE_GROUP=${RESOURCE_GROUP:-"<resource-group>"}`, `VNET_NAME=${VNET_NAME:-"<name>"}`, and `SUBNET_NAME="<subnet-name>"` parameters.

az/sample_use_cases/11_useProximityPlacementGroups/k8s_cluster.conf

This file includes sample values for a use case where:

- A private AKS cluster is deployed (`PRIVATE_CLUSTER="true"`) so that the cluster is only accessible from within the Virtual Network or a connected network.
- You define a Proximity Placement Group (PPG) so that Azure deploys compute resources into a logical grouping where they are physically located close to each other to reduce latency. You specify the PPG name and type of group in the `PPG=${PPG:-"<name>"}` and `PPG_TYPE=${PPG_TYPE:-"<type>"}` parameters.

Cluster Parameters

The contents of `k8s_cluster.conf` are shown below. Descriptions of the cluster parameters follow the contents.

```
ENABLE_MANAGED_IDENTITY="<enable-managed-identity>"
SP=${SP:-"<service-principal>"}
SP_VALIDITY_YEARS="<years>"
SP_ID="<id>"
SP_SECRET="<client-secret>"
RESOURCE_GROUP=${RESOURCE_GROUP:-"<resource-group>"}
RESOURCE_GROUP_TAGS="<tags>"
LOCATION=${LOCATION:-"<location>"}
SUBSCRIPTION_ID="<subscription-id>"
VNET_NAME=${VNET_NAME:-"<name>"}
VNET_CIDR="<vnet-cidr>"
VNET_TAGS="<tags>"
VNET_VM_PROTECTION="<vm-protection>"
SUBNET_NAME="<subnet-name>"
SUBNET_CIDR="<subnet-cidr>"
NODE_ZONES="<zones>"
NODEPOOL_NAME="<name>"
NODEPOOL_TAGS="<tags>"
MACHINE_TYPE="<machine-type>"
K8S_CLUSTER_NAME=${K8S_CLUSTER_NAME:-"<name>"}
K8S_CLUSTER_VERSION=${K8S_CLUSTER_VERSION:-"<kubernetes-version>"}
K8S_CLUSTER_NODE_COUNT="<node-count>"
K8S_NODE_ADMIN_USER="<admin-username>"
AKS_TAGS="<tags>"
AKS_ENABLE_ADDONS="<addons>"
PRIVATE_CLUSTER="<enable-private-cluster>"
LOAD_BALANCER_SKU="<load-balancer-sku>"
```

```

LB_BALANCER_IDLE_TIMEOUT=<load-balancer-idle-timeout>
LB_OUTBOUND_IP_PREFIXES="<load-balancer-outbound-ip-prefixes>"
LB_OUTBOUND_IPS="<load-balancer-outbound-ips>"
LB_OUTBOUND_PORTS=<load-balancer-outbound-ports>
LB_MANAGED_OUTBOUND_IP_COUNT=<load-balancer-managed-outbound-ip-count>
VM_SET_TYPE="<vm-set-type>"
NETWORK_PLUGIN="<network-plugin>"
NETWORK_POLICY="<network-policy>"
DOCKER_BRIDGE_ADDRESS="<docker-bridge-address>"
DNS_SERVICE_IP="<dns-service-ip>"
DNS_NAME_PREFIX="<dns-name-prefix>"
SERVICE_CIDR="<service-cidr>"
MIN_NODES="<min-count>"
MAX_NODES="<max-count>"
MAX_PODS_PER_NODE="<max-pods>"
DISK_SIZE="<node-osdisk-size>"
AZURE_CLI_VERSION="<azure-cli-version>"
NODE OSDISK_TYPE="<node-osdisk-type>"
OS_DISK_ENCRYPTIONSET_ID="<node-osdisk-diskencryptionset-id>"
ENABLE_CLUSTER_AUTOSCALER="<enable-cluster-autoscaler>"
CLUSTER_AUTOSCALER_PROFILE="<cluster-autoscaler-profile>"
ATTACH_ACR="<attach-acr>"
ENABLE_AAD="<enable-aad>"
AAD_ADMIN_GROUP_OBJECT_IDS="<aad-admin-group-object-ids>"
AAD_CLIENT_APP_ID="<aad-client-app-id>"
AAD_SERVER_APP_ID="<aad-server-app-id>"
AAD_SERVER_APP_SECRET="<aad-server-app-secret>"
AAD_TENANT_ID="<tenant-id>"
ENABLE_POD_SECURITY_POLICY="<enable-pod-security-policy>"
DISABLE_RBAC="<disable-rbac>"
ENABLE_NODE_PUBLIC_IP="<enable-node-public-ip>"
SSH_PUB_KEY_VALUE="<ssh-key-value>"
API_SERVER_AUTHORIZED_IP_RANGES="<api-server-authorized-ip-ranges>"
NODEPOOL_LABELS="<nodepool-labels>"
PPG=${PPG:-"<name>"}
PPG_TYPE=${PPG_TYPE:-"<type>"}
UPTIME_SLA="<uptime-sla>"
OUTBOUND_TYPE="<outbound-type>"

```

ENABLE_MANAGED_IDENTITY

Indicates whether to use a system-assigned managed identity for cluster resource management. When enabled, this identity is used to create the K8s cluster resources. In addition, if Managed Identity is enabled, the Service Principal parameters (**SP**, **SP_VALIDITY_YEARS**, **SP_ID**, and **SP_SECRET**) are not required.

SP

The Service Principal to use for the AKS cluster. If you want to use an existing Service Principal, specify the name for that principal. If you want to create a new Service Principal, specify a new name, and the new Service Principal will be created when the cluster is created. For example, **aks-service-principal**.

SP_VALIDITY_YEARS

The number of years for which the Service Principal credentials should be valid. For example, **2**.

SP_ID

The ID for the existing Service Principal. Leave this value blank if you chose to create a new principal.

SP_SECRET

The secret for the existing Service Principal. Leave this value blank if you chose to create a new principal.

RESOURCE_GROUP

The name of the Azure Resource Group to allocate the AKS cluster resources to. You can specify the name of an existing group, or you can specify a new name if you want the K8s scripts to create a new Resource Group.

RESOURCE_GROUP_TAGS

A space-separated list of any tags (key=value pairs) to add to the Resource Group.

LOCATION

The Region code for the location where the AKS cluster will be deployed. For example, **eastus**.

SUBSCRIPTION_ID

The ID for your Azure subscription.

VNET_NAME

The name of the Virtual Network to provision the AKS cluster in. This value should match the name of the network that Anzo is deployed in.

VNET_CIDR

The IP address prefix in CIDR format to use for the Virtual Network.

Note

Supply this value even if VNET_NAME is not set and a new Virtual Network will be created.

VNET_TAGS

A space-separated list of any tags (in key=value format) to add to the Virtual Network.

VNET_VM_PROTECTION

A true or false value that indicates whether to enable VM protection for the subnets in the Virtual Network.

SUBNET_NAME

The name of the new subnetwork to create in the Virtual Network.

SUBNET_CIDR

The IP address prefix in CIDR format for the new subnetwork.

NODE_ZONES

The number of Availability Zones to place the agent nodes in. Valid values are **1**, **2**, or **3**.

NODEPOOL_NAME

The name to give the default node pool that is created in the AKS cluster.

NODEPOOL_TAGS

A space-separated list of any tags (in key=value format) to add to resources in the default node pool.

MACHINE_TYPE

The Virtual Machine Type to use for the nodes in the AKS cluster.

K8S_CLUSTER_NAME

The name to give the AKS cluster.

K8S_CLUSTER_VERSION

The version of Kubernetes to use for creating the cluster.

Note

Kubernetes versions 1.18 and 1.19 are supported. See the [AKS Engine Release Notes](#) for details about the available versions.

K8S_CLUSTER_NODE_COUNT

The number of nodes to deploy in the default node pool.

K8S_NODE_ADMIN_USER

The user account to create on the K8s cluster nodes for SSH access.

AKS_TAGS

A space-separated list of any tags (in key=value format) to add to the cluster.

AKS_ENABLE_ADDONS

A comma-separated list of addons to enable for the AKS cluster. Cambridge Semantics recommends that you include the **monitoring** addon.

PRIVATE_CLUSTER

Indicates whether to make the AKS cluster a private cluster. If the cluster is private, network traffic between the K8s API server and node pools remains on the private network.

Tip

When deciding whether to configure the cluster as a private cluster, you may want to review the [Limitations](#) described in "Create a private Azure Kubernetes Service cluster" in the Azure AKS documentation.

LOAD_BALANCER_SKU

The Azure Load Balancer SKU selection for your cluster. The options are **basic** or **standard**. The standard SKU is recommended for AKS clusters. For information about the SKUs, see [Azure Load Balancer SKUs](#) in the Azure documentation.

LB_BALANCER_IDLE_TIMEOUT

This optional parameter specifies the number of minutes to wait before dropping idle connections to the Load Balancer. For example, a value of **5** means that idle connections are dropped after 5 minutes. If this parameter is not specified, the default value is 30 minutes.

Tip

For more information about configuring the Load Balancer, including details about the idle timeout parameter as well as the outbound IP address and port parameters, see [Configure the Public Standard Load Balancer](#) in the Azure AKS documentation.

LB_OUTBOUND_IP_PREFIXES

This optional parameter specifies a comma-separated list of outbound IP prefix resource IDs.

LB_OUTBOUND_IPS

This optional parameter specifies a comma-separated list of outbound IP resource IDs.

LB_OUTBOUND_PORTS

This optional parameter specifies the number of outbound ports to allocate for the Load Balancer. For example, **8000**.

LB_MANAGED_OUTBOUND_IP_COUNT

This optional parameter specifies the number of AKS-managed outbound IP addresses to allocate for the Load Balancer. For example, **10**.

VM_SET_TYPE

The Agent pool VM set type. Valid values are **VirtualMachineScaleSets** or **AvailabilitySet**. Cambridge Semantics recommends that you set this value to **VirtualMachineScaleSets**.

NETWORK_PLUGIN

The type of Kubernetes network plugin to use, i.e. whether to use basic (kubenet) networking or advanced CNI (azure) networking. Valid values are **kubenet** or **azure**.

NETWORK_POLICY

The type of the network policy (Azure Network Policies or Calico Network Policies) to apply to the pods in the AKS cluster. The network policy defines the rules for ingress and egress traffic between pods in the cluster. Valid values are **azure** or **calico**. For information about the policies, see [Network Policy Options in AKS](#) in the Azure AKS documentation.

DOCKER_BRIDGE_ADDRESS

The CIDR block to use for the Docker bridge. The Docker bridge is not used by the AKS cluster or pods but does need to be set up since Docker is configured as part of the Kubernetes setup. Choose an address space that does not collide with any other CIDRs on your networks, including the cluster's service CIDR and pod CIDR. For example, **172.17.0.1/16**.

DNS_SERVICE_IP

The IP address to assign to the Kubernetes DNS service.

DNS_NAME_PREFIX

This optional parameter specifies the prefix to use for hostnames that are created for the DNS service. If not specified, a hostname is generated using the managed cluster and resource group names.

SERVICE_CIDR

The IP address range in CIDR notation from which to assign the Kubernetes DNS service IP addresses.

MIN_NODES

The minimum number of nodes in the default node pool.

MAX_NODES

The maximum number of nodes in the default node pool.

MAX_PODS_PER_NODE

The maximum number of pods deployable to a node in the default node pool.

DISK_SIZE

The size in GB of the OS disk for each node in the default node pool.

AZURE_CLI_VERSION

The version of the Azure CLI on the workstation. For example, **2.25.0**.

NODE_OSDISK_TYPE

The type of OS disk to use for machines in the cluster. The options are **Ephemeral** or **Managed**.

OS_DISK_ENCRYPTIONSET_ID

Specifies the Resource ID of the disk encryption set to use for encryption at rest on the agent node OS disk.

ENABLE_CLUSTER_AUTOSCALER

Indicates whether to enable the cluster autoscaler for the default node pool.

CLUSTER_AUTOSCALER_PROFILE

A space-separated list of any key=value pairs to use for configuring the Cluster Autoscaler. For example, **scan-interval=10s scale-down-delay-after-delete=10s**. For information about all of the configuration options, see [Using the Autoscaler Profile](#) in the Azure AKS documentation.

ATTACH_ACR

The name or resource ID of the Azure Container Registry to grant the `acrpull` role assignment to.

ENABLE_AAD

Indicates whether to enable managed Azure Active Directory (AAD) for the cluster. When AAD is enabled, the Admin Group Object IDs, AAD Client ID, Server ID, Server Secret, and Tenant ID parameters ([AAD_ADMIN_GROUP_OBJECT_IDS](#), [AAD_CLIENT_APP_ID](#), [AAD_SERVER_APP_ID](#), [AAD_SERVER_APP_SECRET](#), and [AAD_TENANT_ID](#)) are not required.

AAD_ADMIN_GROUP_OBJECT_IDS

This parameter specifies the comma-separated list of AAD group object IDs to set as cluster admin.

AAD_CLIENT_APP_ID

The ID of a "Native" type Azure Active Directory client application. This application is for user logins via kubectl.

AAD_SERVER_APP_ID

The ID of a "Web app/API" Azure Active Directory server application. This application represents the managed cluster's API Server (apiserver application).

AAD_SERVER_APP_SECRET

The secret for the Azure Active Directory server application.

AAD_TENANT_ID

The ID of the Azure Active Directory tenant.

ENABLE_POD_SECURITY_POLICY

Indicates whether to enable the pod security policy for the AKS cluster.

Note

Azure will deprecate this feature in June 2021. For information, see [Secure your cluster using pod security policies in Azure Kubernetes Service \(AKS\)](#) in the Azure AKS documentation.

DISABLE_RBAC

Indicates whether to disable Kubernetes Role-Based Access Control (RBAC).

ENABLE_NODE_PUBLIC_IP

Indicates whether to enable a public IP address for the Virtual Machine Scale Set (VMSS) node.

SSH_PUB_KEY_VALUE

The public key path or key contents to install on the K8s cluster nodes for SSH access. If not specified, the default value is `~\.ssh\id_rsa.pub`.

API_SERVER_AUTHORIZED_IP_RANGES

The list of IP address ranges in CIDR notation that are authorized to access the AKS cluster.

NODEPOOL_LABELS

A space-separated list (in key=value format) of labels to add to the nodes in the default node pool. For information about using labels in Kubernetes clusters, see [Labels and Selectors](#) in the Kubernetes documentation.

PPG

This optional parameter specifies the name of the Proximity Placement Group (PPG) to use for the cluster. For information about using proximity placement groups, see [Use Proximity Placement Groups](#) in the Azure AKS documentation.

PPG_TYPE

If using a Proximity Placement Group (PPG), this parameter specifies the type of PPG to use. The only valid value is **Standard**.

UPTIME_SLA

Indicates whether to enable a paid managed cluster service with a financially backed SLA.

OUTBOUND_TYPE

Specifies how to configure outbound traffic for the cluster. Valid values are **loadBalancer** and **userDefinedRouting**.

Example Configuration File

An example completed `k8s_cluster.conf` file is shown below.

```
ENABLE_MANAGED_IDENTITY="true"
#SP=${SP:-"aks-service-principal"}
#SP_VALIDITY_YEARS="2"
#SP_ID="291bba3f-e0a5-47bc-a099-3bdc2a50a05"
#SP_SECRET="ValidServicePrincipalSecretIfPresent"
RESOURCE_GROUP=${RESOURCE_GROUP:-"aks-resource-group"}
RESOURCE_GROUP_TAGS="description=aks-cluster"
LOCATION=${LOCATION:-"eastus"}
SUBSCRIPTION_ID="ValidSubscriptionId"
VNET_NAME=${VNET_NAME:-"anzo-vnet"}
VNET_CIDR="20.20.0.0/16"
VNET_TAGS="description=aks-virtual-network"
VNET_VM_PROTECTION="true"
SUBNET_NAME="k8s-subnet"
SUBNET_CIDR="20.20.0.0/19"
#NODE_ZONES=""
NODEPOOL_NAME="defaultpool"
NODEPOOL_TAGS="description=default-nodepool"
MACHINE_TYPE="Standard_DS1_v2"
K8S_CLUSTER_NAME=${K8S_CLUSTER_NAME:-"k8s-cluster"}
K8S_CLUSTER_VERSION=${K8S_CLUSTER_VERSION:-"1.18"}
K8S_CLUSTER_NODE_COUNT="2"
K8S_NODE_ADMIN_USER="azureuser"
AKS_TAGS="description=aks-cluster"
AKS_ENABLE_ADDONS="monitoring"
PRIVATE_CLUSTER="false"
LOAD_BALANCER_SKU="standard"
#LB_BALANCER_IDLE_TIMEOUT=5
#LB_OUTBOUND_IP_PREFIXES="<ip-prefix-resource-id-1,ip-prefix-resource-id-2>"
#LB_OUTBOUND_IPS="<ip-resource-id-1,ip-resource-id-2>"
#LB_OUTBOUND_PORTS=8000
#LB_MANAGED_OUTBOUND_IP_COUNT=10
VM_SET_TYPE="VirtualMachineScaleSets"
NETWORK_PLUGIN="azure"
NETWORK_POLICY="azure"
DOCKER_BRIDGE_ADDRESS="172.17.0.1/16"
DNS_SERVICE_IP="10.0.0.10"
#DNS_NAME_PREFIX="k8stest"
```

```

SERVICE_CIDR="10.0.0.0/16"
MIN_NODES="1"
MAX_NODES="8"
MAX_PODS_PER_NODE="16"
DISK_SIZE="100"
AZURE_CLI_VERSION="2.19.1"
NODE_OSDISK_TYPE="Ephemeral"
#OS_DISK_ENCRYPTIONSET_ID=""
ENABLE_CLUSTER_AUTOSCALER="true"
CLUSTER_AUTOSCALER_PROFILE="scan-interval=10s scale-down-delay-after-delete=10s"
ATTACH_ACR="ContainerRegistry"
ENABLE_AAD="true"
AAD_ADMIN_GROUP_OBJECT_IDS="5d24455a-1111-3333-4444-5dv77afa27aed"
#AAD_CLIENT_APP_ID="ValidAADClientAppId"
#AAD_SERVER_APP_ID="ValidAADServerAppId"
#AAD_SERVER_APP_SECRET="ValidAADServerAppSecret"
#AAD_TENANT_ID="8f70baf1-1f6e-46a2-a1ff-238dac1ebfb7"
ENABLE_POD_SECURITY_POLICY="true"
ENABLE_MANAGED_IDENTITY="false"
DISABLE_RBAC="false"
SSH_PUB_KEY_VALUE=""
API_SERVER_AUTHORIZED_IP_RANGES="10.107.1.0/24"
NODEPOOL_LABELS="description=k8scluster"
#PPG=${PPG:-"csippg"}
#PPG_TYPE=${PPG_TYPE:-"Standard"}
UPTIME_SLA="false"
OUTBOUND_TYPE="loadBalancer"

```

Create the AKS Cluster

After defining the cluster requirements, run the **create_k8s.sh** script in the **az** directory to create the cluster. Run the script with the following command. The arguments are described below.

```

./create_k8s.sh -c <config_file_name> [ -d <config_file_directory> ] [ -f | --force ] [ -h
| --help ]

```

-c <config_file_name>

This is a **required** argument that specifies the name of the configuration file that supplies the cluster requirements. For example, **-c k8s_cluster.conf**.

-d <config_file_directory>

This is an **optional** argument that specifies the path and directory name for the configuration file specified for the **-c** argument. If you are using the original **az** directory file structure and the configuration file is in the **conf.d** directory, you do not need to specify the **-d** argument. If you created a separate directory structure for different Anzo environments, include the **-d** option. For example, **-d /az/env1/conf**.

-f | --force

This is an **optional** argument that controls whether the script prompts for confirmation before proceeding with each stage involved in creating the cluster. If **-f (--force)** is specified, the script assumes the answer is "yes" to all prompts and does not display them.

-h | --help

This argument is an **optional** flag that you can specify to display the help from the `create_k8s.sh` script.

For example, the following command runs the `create_k8s` script, using `k8s_cluster.conf` as input to the script. Since `k8s_cluster.conf` is in the `conf.d` directory, the `-d` argument is excluded:

```
./create_k8s.sh -c k8s_cluster.conf
```

The script validates that the required software packages, such as the Azure CLI and `kubectl`, are installed and that the versions are compatible with the script. It also displays an overview of the deployment details based on the values in the specified configuration file.

The script then prompts you to proceed with deploying each component of the AKS cluster infrastructure. Type `y` and press **Enter** to proceed with each step in creating the specified Service Principal, Virtual Network, subnet, and Load Balancer components. All components are created according to the specifications in the configuration file.

When cluster creation is complete, proceed to [Creating the Required Node Pools](#) to add the required node pools to the cluster.

Related Topics

[Creating and Assigning IAM Roles](#)

[Creating the Required Node Pools](#)

Creating the Required Node Pools

This topic provides instructions for creating the three types of required node pools:

- The **Operator** node pool for running the AnzoGraph, Anzo Agent with Anzo Unstructured (AU), Elasticsearch, and Spark operator pods.
- The **AnzoGraph** node pool for running AnzoGraph application pods.
- The **Dynamic** node pool for running Anzo Agent with AU, Elasticsearch, and Spark application pods.

Tip For more information about the node pools, see [Node Pool Requirements](#).

- [Define the Node Pool Requirements](#)
- [Create the Node Pools](#)

Define the Node Pool Requirements

Before creating the node pools, configure the infrastructure requirements for each type of pool. The `nodepool_*.conf` files in the `az/conf.d` directory are sample configuration files that you can use as templates, or you can edit the files directly:

- `nodepool_operator.conf` defines the requirements for the Operator node pool.
- `nodepool_anzograph.conf` defines the requirements for the AnzoGraph node pool.
- `nodepool_dynamic.conf` defines the requirements for the Dynamic node pool.

Each type of node pool configuration file contains the following parameters. Descriptions of the parameters and guidance on specifying the appropriate values for each type of node pool are provided below.

```
NODEPOOL_NAME="<name>"
KUBERNETES_VERSION="<kubernetes-version>"
DOMAIN="<domain>"
KIND="<kind>"
MACHINE_TYPE="<node-vm-size>"
LOCATION=${LOCATION:-"<location>"}
RESOURCE_GROUP=${RESOURCE_GROUP:-"<resource-group>"}
VNET_NAME=${VNET_NAME:-"<vnet-name>"}
SUBNET_NAME="<name>"
SUBNET_CIDR="<address-prefix>"
K8S_CLUSTER_NAME=${K8S_CLUSTER_NAME:-"<cluster-name>"}
NODE_TAINTS="<node-taints>"
MAX_PODS_PER_NODE=<max-pods>
MAX_NODES=<max-count>
MIN_NODES=<min-count>
NUM_NODES=<node-count>
DISK_SIZE="<node-osdisk-size>"
OS_TYPE="<os-type>"
PRIORITY="<priority>"
ENABLE_CLUSTER_AUTOSCALER=<enable-cluster-autoscaler>
LABELS="<nodepool-labels>"
MODE="<mode>"
NODE OSDISK_TYPE="<node-osdisk-type>"
PPG="<name>"
PPG_TYPE=${PPG_TYPE:-"<type>"}

```

NODEPOOL_NAME

The name to give the node pool.

Node Pool Type	Sample NODEPOOL_NAME Value
Operator	csi-operator

Node Pool Type	Sample NODEPOOL_NAME Value
AnzoGraph	csi-anzograph
Dynamic	csi-dynamic

KUBERNETES_VERSION

The version of Kubernetes to use for creating the node pool. This value must match the AKS cluster version ([K8S_CLUSTER_VERSION](#)). For example, **1.18**.

DOMAIN

The name of the domain that hosts the node pool. This is typically the name or acronym for the organization, such as **csi**.

KIND

This parameter classifies the node pool in terms of kernel tuning and the type of pods that the node pool will host.

Node Pool Type	Required KIND Value
Operator	operator
AnzoGraph	anzograph
Dynamic	dynamic

MACHINE_TYPE

The Virtual Machine Type to use for the nodes in the node pool.

Node Pool Type	Sample MACHINE_TYPE Value
Operator	Standard_DS2_v2
AnzoGraph	Standard_D16s_v3
Dynamic	Standard_D8s_v3

Tip

For more guidance on determining the instance types to use for nodes in the required node pools, see [Compute Resource Planning](#).

LOCATION

The Region code for the location of the AKS cluster. For example, **eastus**.

RESOURCE_GROUP

The name of the Azure Resource Group to allocate the node pool's resources to. You can specify the name of an existing group, or you can specify a new name if you want the K8s scripts to create a new Resource Group for the node pool.

VNET_NAME

The name of the Virtual Network that the AKS cluster was deployed in.

SUBNET_NAME

The name of the subnetwork to create.

SUBNET_CIDR

The IP address prefix to use when creating the subnetwork.

K8S_CLUSTER_NAME

The name of the AKS cluster.

NODE_TAINTS

This parameter defines the type of pods that are allowed to be placed in this node pool. When a pod is scheduled for deployment, the scheduler relies on this value to determine whether the pod belongs in this pool. If a pod has a **toleration** that is not compatible with this **taint**, the pod is rejected from the pool. The recommended values below specify that operator pods are allowed in the Operator node pool, AnzoGraph pods are allowed in the AnzoGraph node pool, and dynamic pods are allowed in the Dynamic node pool. The **NoSchedule** value means a toleration is required and pods without the appropriate toleration will not be allowed in the pool.

Node Pool Type	Recommended NODE_TAINTS Value
Operator	cambridgesemantics.com/dedicated=operator:NoSchedule
AnzoGraph	cambridgesemantics.com/dedicated=anzograph:NoSchedule
Dynamic	cambridgesemantics.com/dedicated=dynamic:NoSchedule

MAX_PODS_PER_NODE

The maximum number of pods that can be hosted on a node in the node pool. In addition to Anzo application pods, this limit also needs to account for K8s service pods and helper pods. Cambridge Semantics recommends that you set this value to at least **16** for all node pool types.

MAX_NODES

The maximum number of nodes that can be deployed in the node pool.

Node Pool Type	Sample MAX_NODES Value
Operator	8
AnzoGraph	16
Dynamic	32

MIN_NODES

The minimum number of nodes to remain deployed in the node pool at all times. If the cluster autoscaler is enabled for the node pool, you can set this value to **1** (the lowest value allowed by AKS). The autoscaler will automatically provision additional nodes if multiple pods are scheduled for deployment.

NUM_NODES

The number of nodes to deploy when this node pool is created. This value must be set to at least **1**. When you create the node pool, at least one node in the pool needs to be deployed as well.

DISK_SIZE

The size in GB of the OS disk for each node in the node pool.

Node Pool Type	Sample DISK_SIZE Value
Operator	50
AnzoGraph	100
Dynamic	100

OS_TYPE

The operating system to use for the nodes in the node pool. Specify **Linux** for each type of node pool.

PRIORITY

Specifies the priority level of the VMs for the nodes in the node pool. Valid values are **Regular** (dedicated) or **Spot** (low-priority or preemptible).

ENABLE_CLUSTER_AUTOSCALER

Indicates whether to enable the cluster autoscaler for the node pool.

LABELS

A space-separated list (in key=value format) of labels to add to the nodes in the node pool. For information about using labels in Kubernetes clusters, see [Labels and Selectors](#) in the Kubernetes documentation.

MODE

The mode for the node pool. The mode defines the node pool's primary function, i.e., whether it is a **System** node pool or a **User** pool. System node pools serve the primary purpose of hosting critical system pods. User node pools serve the primary purpose of hosting application pods. For the Operator, AnzoGraph, and Dynamic node pools, the mode should be set to **User**. For more information, see [System and User Node Pools](#) in the Azure AKS documentation.

NODE_OSDISK_TYPE

The type of OS disk to use for machines in the node pool. The options are **Ephemeral** or **Managed**.

PPG

This optional parameter specifies the name of the Proximity Placement Group (PPG) to use for the node pool. For information about using proximity placement groups, see [Use Proximity Placement Groups](#) in the Azure AKS documentation.

PPG_TYPE

If using a Proximity Placement Group (PPG), this parameter specifies the type of PPG to use. The only valid value is **Standard**.

Example Configuration Files

Example completed configuration files for each type of node pool are shown below.

Operator Node Pool

The example below shows a configured nodepool_operator.conf file.

```
NODEPOOL_NAME="csi-operator"
KUBERNETES_VERSION="1.18"
DOMAIN="csi"
KIND="operator"
MACHINE_TYPE="Standard_DS2_v2"
LOCATION=${LOCATION:-"eastus"}
RESOURCE_GROUP=${RESOURCE_GROUP:-"aks-resource-group"}
VNET_NAME=${VNET_NAME:-"anzo-vnet"}
SUBNET_NAME="k8s-subnet"
SUBNET_CIDR="20.20.2.0/19"
K8S_CLUSTER_NAME=${K8S_CLUSTER_NAME:-"k8s-cluster"}
NODE_TAINTS="cambridgesemantics.com/dedicated=operator:NoSchedule"
MAX_PODS_PER_NODE=16
```

```

MAX_NODES=8
MIN_NODES=1
NUM_NODES=1
DISK_SIZE="50"
OS_TYPE="Linux"
PRIORITY="Regular"
ENABLE_CLUSTER_AUTOSCALER=true
LABELS="description=k8s-operator-nodepool"
MODE="User"
NODE_OSDISK_TYPE="Managed"
#PPG="testppg"
#PPG_TYPE=${PPG_TYPE:-"standard"}

```

AnzoGraph Node Pool

The example below shows a configured `nodepool_anzograph.conf` file.

```

NODEPOOL_NAME="csi-anzograph"
KUBERNETES_VERSION="1.18"
DOMAIN="csi"
KIND="anzograph"
MACHINE_TYPE="Standard_D16s_v3"
LOCATION=${LOCATION:-"eastus"}
RESOURCE_GROUP=${RESOURCE_GROUP:-"aks-resource-group"}
VNET_NAME=${VNET_NAME:-"anzo-vnet"}
SUBNET_NAME="k8s-subnet"
SUBNET_CIDR="20.20.2.0/19"
K8S_CLUSTER_NAME=${K8S_CLUSTER_NAME:-"k8s-cluster"}
NODE_TAINTS="cambridgesemantics.com/dedicated=anzograph:NoSchedule"
MAX_PODS_PER_NODE=16
MAX_NODES=16
MIN_NODES=1
NUM_NODES=1
DISK_SIZE="100"
OS_TYPE="Linux"
PRIORITY="Regular"
ENABLE_CLUSTER_AUTOSCALER=true
LABELS="description=k8s-anzograph-nodepool"
MODE="User"
NODE_OSDISK_TYPE="Managed"
#PPG="testppg"
#PPG_TYPE=${PPG_TYPE:-"standard"}

```

Dynamic Node Pool

The example below shows a configured `nodepool_dynamic.conf` file.

```

NODEPOOL_NAME="csi-dynamic"
KUBERNETES_VERSION="1.18"
DOMAIN="csi"
KIND="dynamic"
MACHINE_TYPE="Standard_D8s_v3"
LOCATION=${LOCATION:-"eastus"}
RESOURCE_GROUP=${RESOURCE_GROUP:-"aks-resource-group"}
VNET_NAME=${VNET_NAME:-"anzo-vnet"}
SUBNET_NAME="k8s-subnet"
SUBNET_CIDR="20.20.2.0/19"
K8S_CLUSTER_NAME=${K8S_CLUSTER_NAME:-"k8s-cluster"}
NODE_TAINTS="cambridgesemantics.com/dedicated=dynamic:NoSchedule"
MAX_PODS_PER_NODE=16
MAX_NODES=32
MIN_NODES=1
NUM_NODES=1
DISK_SIZE="100"
OS_TYPE="Linux"
PRIORITY="Regular"
ENABLE_CLUSTER_AUTOSCALER=true
LABELS="description=k8s-dynamic-nodepool"
MODE="User"
NODE_OSDISK_TYPE="Managed"
#PPG="testppg"
#PPG_TYPE=${PPG_TYPE:-"standard"}

```

Create the Node Pools

After defining the requirements for the node pools, run the **create_nodepools.sh** script in the `az` directory to create each type of node pool. Run the script once for each type of pool.

Note

The `create_nodepools.sh` script references the files in the `az/reference` directory. If you customized the directory structure on the workstation, ensure that the **reference** directory is available at the same level as `create_nodepools.sh` before creating the node pools.

Run the script with the following command. The arguments are described below.

```

./create_nodepools.sh -c <config_file_name> [ -d <config_file_directory> ] [ -f | --force ] [ -h | --help ]

```

-c <config_file_name>

This is a **required** argument that specifies the name of the configuration file (i.e., `nodepool_operator.conf`, `nodepool_anzograph.conf`, or `nodepool_dynamic.conf`) that supplies the node pool requirements. For example, **-c nodepool_dynamic.conf**.

-d <config_file_directory>

This is an **optional** argument that specifies the path and directory name for the configuration file specified for the **-c** argument. If you are using the original **az** directory file structure and the configuration file is in the **conf.d** directory, you do not need to specify the **-d** argument. If you created a separate directory structure for different Anzo environments, include the **-d** option. For example, **-d /az/env1/conf**.

-f | --force

This is an **optional** argument that controls whether the script prompts for confirmation before proceeding with each stage involved in creating the node pool. If **-f (--force)** is specified, the script assumes the answer is "yes" to all prompts and does not display them.

-h | --help

This argument is an **optional** flag that you can specify to display the help from the **create_nodepools.sh** script. For example, the following command runs the **create_nodepools** script, using **nodepool_operator.conf** as input to the script. Since **nodepool_operator.conf** is in the **conf.d** directory, the **-d** argument is excluded:

```
./create_nodepools.sh -c nodepool_operator.conf
```

The script validates that the required software packages, such as the Azure CLI and **kubectl**, are installed and that the versions are compatible with the script. It also displays an overview of the node pool deployment details based on the values in the specified configuration file.

The script then prompts you to proceed with deploying each component of the node pool. Type **y** and press **Enter** to proceed with the configuration.

Once the Operator, AnzoGraph, and Dynamic node pools are created, the next step is to create a Cloud Location in Anzo so that Anzo can connect to the AKS cluster and deploy applications. See [Connecting to a Cloud Location](#).

Related Topics

[Creating the AKS Cluster](#)

[Connecting to a Cloud Location](#)

User Guide

The User Guide provides usage information for all of the Anzo components.

Tip

For an introduction to Anzo concepts, an overview of the user interface, basic setup steps, and instructions for building a sample solution from scratch, see the [Getting Started Guide](#).

- [Onboarding Structured Data](#)
- [Onboarding Unstructured Data](#)
- [Modeling Data](#)
- [Blending Data](#)
- [Sharing Access to Artifacts](#)
- [Accessing and Analyzing Data](#)
- [Versioning and Migrating Artifacts](#)
- [Graph Data Storage Reference](#)
- [Exploring Data Provenance](#)

Onboarding Structured Data

Structured data sources such as relational databases or flat files are onboarded to Anzo using Anzo's built-in pipelines. These pipelines natively support CSV, JSON, XML, SAS, and Parquet files, along with all common database connections, including SQL, Oracle, MySQL, HIVE, and others.

The topics in this section provide instructions for connecting to and importing data from structured data sources, ingesting the data, and working with schemas, mappings, and pipelines.

Note

For instructions on importing files that are in RDF format, see [Adding a Dataset to the Dataset Catalog](#).

- [Adding Data Sources and Schemas](#)
- [Managing Data Source Metadata](#)
- [Ingesting Data](#)
- [Working with Mappings](#)
- [Working with Pipelines](#)

Adding Data Sources and Schemas

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

- [Creating a Database Data Source](#)
- [Creating a CSV Data Source](#)
- [Creating a JSON Data Source](#)
- [Creating an XML Data Source](#)
- [Creating a SAS Data Source](#)
- [Creating a Parquet Data Source and Ingesting the Data](#)
- [Generating a Source Data Profile](#)
- [Assigning Primary Keys in an Onboarded Schema](#)
- [Creating or Changing Foreign Keys](#)

Related Topics

[Setting a Base File Store Path for File Uploads](#)

Creating a Database Data Source

The topics in this section provide instructions for connecting to a structured data source, such as a Microsoft, Oracle, Hadoop, Teradata, PostgreSQL, or Google database, and defining the schema to use for onboarding the data.

- [Performance Considerations for Database Pipelines](#)
- [Connecting to a Database](#)
- [Defining a Database Schema](#)
- [Partitioning a Database Table for Parallel Ingestion](#)

Performance Considerations for Database 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](#)

Connecting to a Database

This topic provides instructions for connecting to a structured 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 Data Sources. For example:

Data Sources Schemas Mappings Pipelines								
	<input type="text" value="Search"/>	Sort By: Title ▼	View:	<button>Add Data Source ▼</button>				
<input type="checkbox"/>	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021		
	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021		
	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021		
	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021		
Rows per page: 25 ▼ 1-4 of 4 < >								

2. 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

3. At the top of the screen, specify a **Title** for the source, and enter an optional **Description**.
4. 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.

5. 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.

After connecting to the data source, the next step is to define the Schema that Anzo will use to determine the data's structure and import the data. See [Defining a Database Schema](#) for instructions.

Related Topics

[Defining a Database Schema](#)

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

Defining a Database Schema

This topic provides information about creating the schema to use when importing data from a database. The schema defines the source data to onboard. Anzo supports multiple options for defining the schema. You can import a predefined schema from the database, you can write a static SQL query that defines the data, or, if you want to import data incrementally, you can write an incremental SQL query.

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.

Select 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](#)

Importing a Predefined Schema

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

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](#).

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

Data Sources Schemas Mappings Pipelines								
<div><div></div><div>Search</div></div>		Sort By: Title		View: <div><div></div><div></div></div>		Add Data Source		
	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021	<div></div>	<div></div>
	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021	<div></div>	<div></div>
	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021	<div></div>	<div></div>
	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021	<div></div>	<div></div>
Rows per page: 25 1-4 of 4 < >								

2. Click the data source for which you want to import a schema. Anzo displays the Tables tab for the source. For example, the image below shows the Tables tab before any Schemas have been added for this data source:


Overview

Tables

Versions

Discussion

Sharing

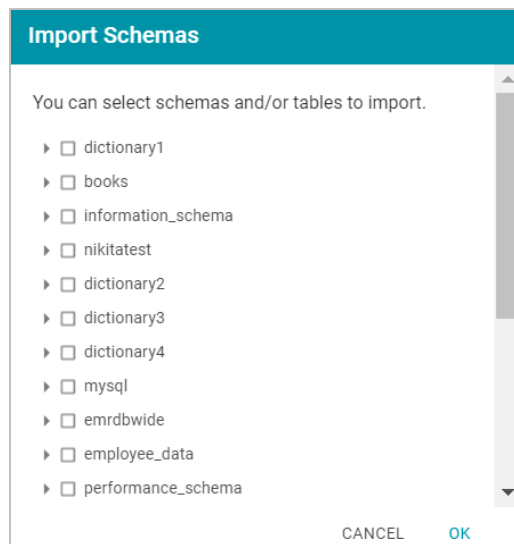


Please import a schema, or create a schema from a custom query.

Create Schemas from Query

Import Schemas

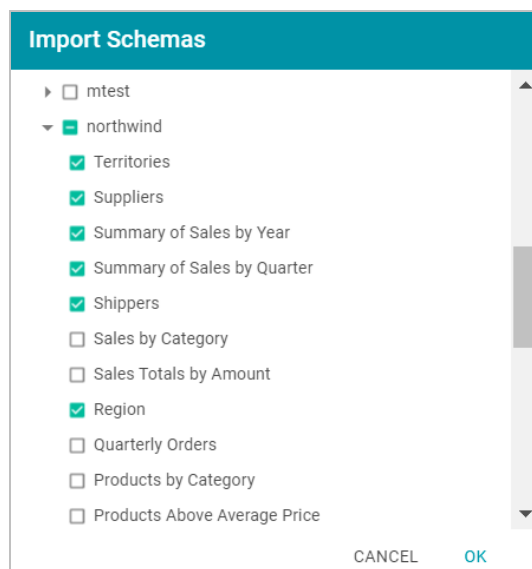
3. Click the **Import Schemas** button. Anzo displays the Import Schemas dialog box, which lists any predefined Schemas in the database. For example:



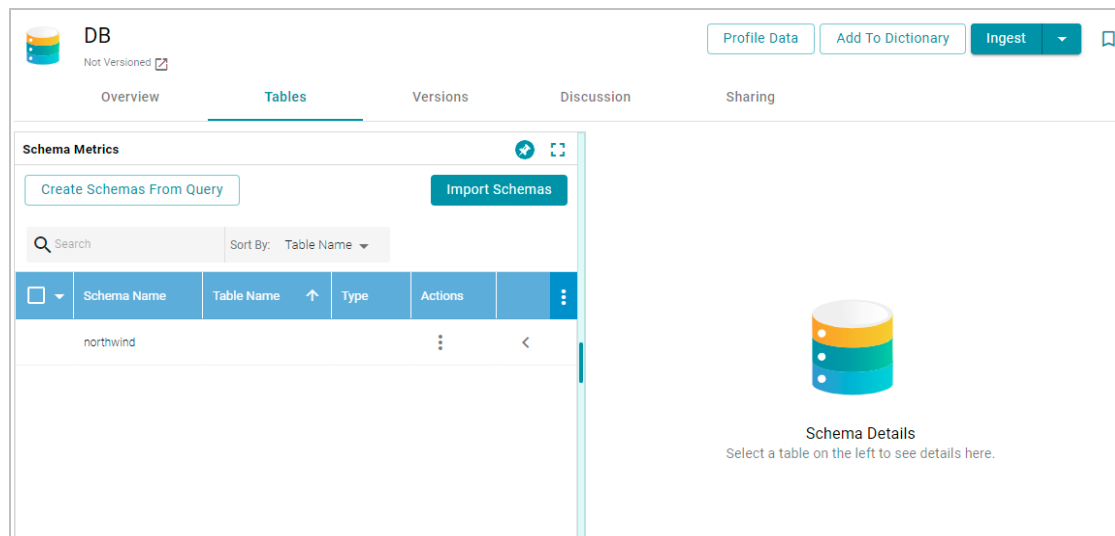
Note

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 or schemas, 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 schema or schemas and lists the imported schemas on the Tables screen. For example:



Once the Schema or 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. For information about creating a Metadata Dictionary for this Data Source, see [Creating a Metadata Dictionary](#). For instructions on onboarding the data by automatically generating the Model, Mappings, and ETL Pipeline, see [Ingesting a New Data Source](#).

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. For information about writing a Schema query that onboards data from a database incrementally, see [Creating an Incremental Schema](#).

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 Pipelines](#).

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

Data Sources								
Schemas								
Mappings								
Pipelines								
<div><div><div></div><div>Q Search</div></div><div>Sort By: Title</div><div>View: <div></div><div></div></div><div>Add Data Source</div></div>								
<div></div>	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
<div></div>	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021	<div></div>	<div></div>
<div></div>	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021	<div></div>	<div></div>
<div></div>	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021	<div></div>	<div></div>
<div></div>	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021	<div></div>	<div></div>
Rows per page: 25 1-4 of 4								

2. Click the data source for which you want to create a schema. Anzo displays the Tables tab for the source. For example, the image below shows the Tables tab before any Schemas have been created for this data source:

Overview

Tables

Versions

Discussion

Sharing

Please import a schema, or create a schema from a custom query.

Create Schemas from Query

Import Schemas

3. 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 Schema Query *

Include increment data

CANCEL

SAVE

4. 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

Schema Query *

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

☐ Include increment data

CANCEL SAVE

7. 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:

Schema Metrics

Create Schemas From Query

Import Schemas

Search

Sort By: Table Name

	Schema Name	Table Name	Type	Actions	
<input type="checkbox"/>	employees				<
	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. For example:

OverviewTablesVersionsDiscussionSharing

Schema Metrics

Create Schemas From Query

Import Schemas

Search

Sort By: Table Name

	Schema Name	Table Name	Type	Actions	
<input type="checkbox"/>	employees				>
		all_employees	Manual		
	northwind				<

all_employees

CreatorSystem AdministratorLast Modified Date05/06/2021Column Count9Row CountNot Calculated

Sample Data

Metrics

Foreign Keys

Mappings

EmployeeID	FirstName	LastName	Title	Salary	BirthDate	HireDate	Region
Int	String	String	String	Float	Date time	Date time	String
6	Michael	Suyama	Sales Represent	2004.07	1963-07-02 00:00	1993-10-17 00:00	
9	Anne	Dodsworth	Sales Represent	2333.33	1966-01-27 00:00	1994-11-15 00:00	
8	Laura	Callahan	Inside Sales Coor	2100.5	1958-01-09 00:00	1994-03-05 00:00	WA
2	Andrew	Fuller	Vice President, Sa	2254.49	1952-02-19 00:00	1992-08-14 00:00	WA
4	Margaret	Peacock	Sales Represent	1861.08	1937-09-19 00:00	1993-05-03 00:00	WA
5	Steven	Buchanan	Sales Manager	1744.21	1955-03-04 00:00	1993-10-17 00:00	

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

	Schema Name	Table Name	Type	Actions	
<input type="checkbox"/>	employees				>
		all_employees	Manual		

Add Table

Reload Schema

Delete

The Create New Table dialog box is displayed.

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

Now that a schema has been defined, the source data can be onboarded to Anzo. For information about creating a metadata dictionary for this data source, see [Creating a Metadata Dictionary](#). For instructions on onboarding the data by automatically generating the model, mappings, and ETL pipeline, see [Ingesting a New Data Source](#).

Creating an Incremental Schema

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

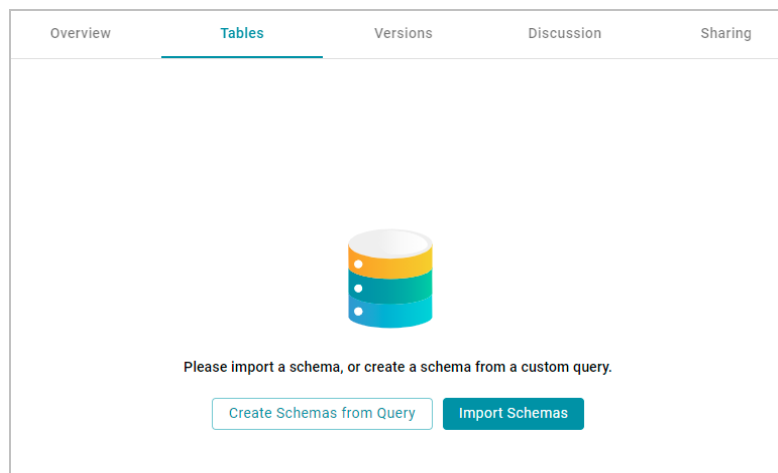
Tip

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 Pipelines](#).

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

<

- Click the data source for which you want to create an incremental schema. Anzo displays the Tables tab for the source. For example, the image below shows the Tables tab before any Schemas have been created for this data source:



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

4. 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:

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 that 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} will be 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

X

CANCEL

SAVE

9. 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:

Schema Metrics

Create Schemas From Query

Import Schemas

Search

Sort By: Table Name

	Schema Name	Table Name	Type	Actions	
<input type="checkbox"/>	employees			<div></div>	<
<input type="checkbox"/>	new-hires			<div></div>	<
	northwind			<div></div>	<

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.

Overview

Tables

Versions

Discussion

Sharing

Schema Metrics

Create Schemas From Query

Import Schemas

Search

Sort By: Table Name

Schema Name	Table Name	Type	Actions
employees			
new-hires			
new-employees	Manual		
northwind			

new-employees

Creator System Administrator Last Modified Date 05/06/2021 Column Count 9 Row Count Not Calculated

Sample Data Metrics Foreign Keys Mappings

EmployeeID	FirstName	LastName	Title	Salary	BirthDate	HireDate	Region
int	String	String	String	Float	Date time	Date time	String
6	Michael	Suyama	Sales Represent	2004.07	1963-07-02 00:00	1993-10-17 00:00	
8	Laura	Callahan	Inside Sales Cool	2100.5	1958-01-09 00:00	1994-03-05 00:00	WA
9	Anne	Dodsworth	Sales Represent	2333.33	1966-01-27 00:00	1994-11-15 00:00	
7	Robert	King	Sales Represent	1991.55	1960-05-29 00:00	1994-01-02 00:00	

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

Schema Name	Table Name	Type	Actions
employees			<div>⋮</div> <div>Add Table</div> <div>Reload Schema</div> <div>Delete</div>
	all_employees	Manual	

The Create New Table dialog box is displayed.

Create New Table

Table Name *

Schema Query *

1 Enter Query Here

+ Incremental Load

CANCEL SAVE

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

Now that a schema has been defined, the source data can be onboarded to Anzo. For information about creating a metadata dictionary for this data source, see [Creating a Metadata Dictionary](#). For instructions on onboarding the data by automatically generating the model, mappings, and ETL pipeline, see [Ingesting a New Data Source](#).

Note

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

Related Topics

[Connecting to a Database](#)

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

[Incremental Pipeline Reference](#)

[Ingesting Data](#)

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](#).

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.

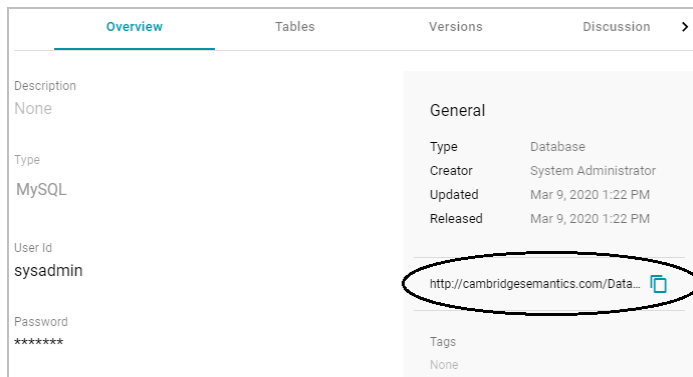
Note

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.



For example:

```
anzo get
http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763dffefaab4
```

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/aff6a2f7a1354140871b763dffefaab4/Sche
ma/northwind>
```

- 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/aff6a2f7a1354140871b763dffefaab4/Sche
ma/northwind
```

Anzo returns the metadata for the schema.

- 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/aff6a2f7a1354140871b763dffefaab4/Sche
ma/northwind/ORDERS>
```

- 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/aff6a2f7a1354140871b763dffefaab4/Sche
ma/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/aff6a2f7a1354140871b763dffefaab4/Sche
ma/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/aff6a2f7a1354140871b763dffefaab4/Sche
ma/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/aff6a2f7a1354140871b763dffefaab4/Sch
ema/northwind> {

<http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763dffefaab4/Sch
ema/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#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:

- a. At the top of the file, replace the schema URI with the following service URI:

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

In the example above,

<http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763dffeef
aab4/Schema/northwind> is replaced by

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

- b. 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#ComputePartitioningPr
edicatesRequest>
  <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#BaseComputePartitioni
ngPredicatesRequest> ,

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

- 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:

Schemas					Orders						
+ Create Schemas From Query Import Schemas					Creator: System Administrator Last Modified Date: 03/09/2020 Column Count: 10 Row Count: 830						
Schema ...	Table Name	Actions			Sample Data		Metrics	Foreign Keys	Mappings		
Employees		⋮			OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia
EmployeeTerritory		⋮			Int	String	Int	Date time	Date time	Date time	Int
Order Details		⋮			10250	HANAR	4	1996-07-08...	1996-08-05...	1996-07-12...	2
Orders		⋮			10253	HANAR	3	1996-07-10...	1996-07-24...	1996-07-16...	2
Products		⋮			10256	WELLI	3	1996-07-15...	1996-08-12...	1996-07-17...	2
Region		⋮			10254	CHOPS	5	1996-07-11...	1996-08-08...	1996-07-23...	2
					10257	HILAA	4	1996-07-16...	1996-08-13...	1996-07-22...	3

- **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#ComputePartitioningPr
edicatesRequest> {

<http://cambridgesemantics.com/DatabaseDataSource/aff6a2f7a1354140871b763dffefaab4
/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#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" .
# PARTITIONING METADATA

<http://cambridgesemantics.com/ontologies/2015/08/SDIService#ComputePartitioningPr
edicatesRequest>
<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#BaseComputePartitioni
ngPredicatesRequest> ,

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

```

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#computeColumnBasedTableParti
tioningPredicates /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#computeColumnBasedTableParti
tioningPredicates 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 ,
_: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 .
```

- 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>
s> _: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#tableURI>

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

```

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#assignTablePartitioningPredicates filename.trig

```

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

```

anzo call
http://cambridgesemantics.com/semanticServices/SDIService#assignTablePartitioningPredicates 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](#).

Related Topics

[Connecting to a Database](#)

[Defining a Database Schema](#)

[Ingesting Data](#)

[Generating a Source Data Profile](#)

Creating a CSV Data Source

Follow the instructions below to create a CSV Data Source and import the data from the files.

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 Data Sources. For example:

Data Sources Schemas Mappings Pipelines								
	<input type="text" value="Search"/>	Sort By: Title ▼	View:	<button>Add Data Source ▼</button>				
<input type="checkbox"/>	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021		
	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021		
	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021		
	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021		
Rows per page: 25 ▼ 1-4 of 4 < >								

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



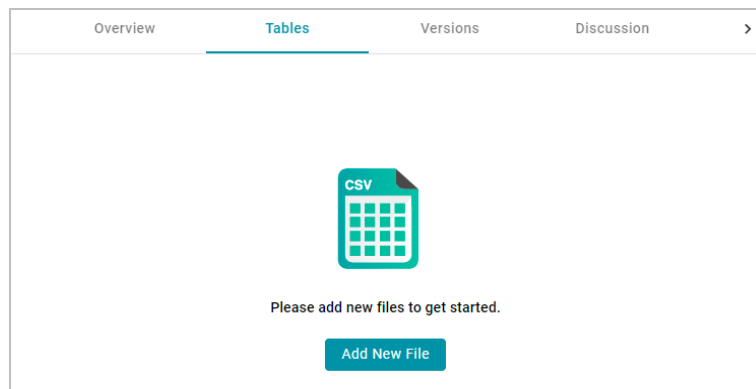
Create CSV Data Source

Title

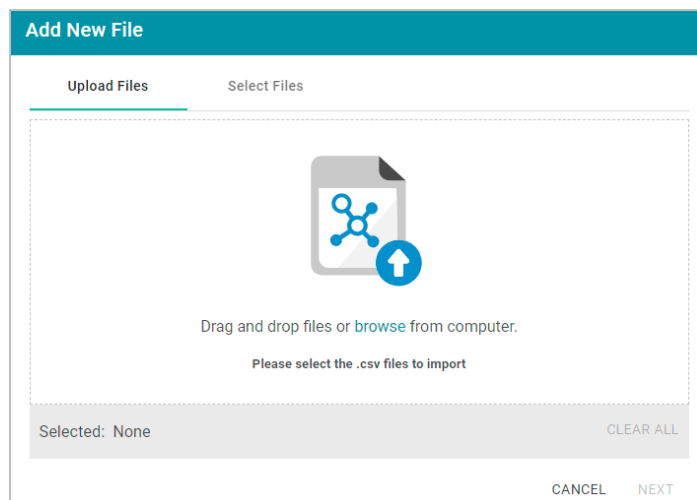
Description

CANCEL SAVE

- Specify a name for the Data 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.



- Click the **Add New File** button. Anzo displays the Add New File dialog box, and the **Upload Files** tab is selected.



Add New File

Upload Files **Select Files**

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

Please select the .csv files to import

Selected: None CLEAR ALL

CANCEL NEXT

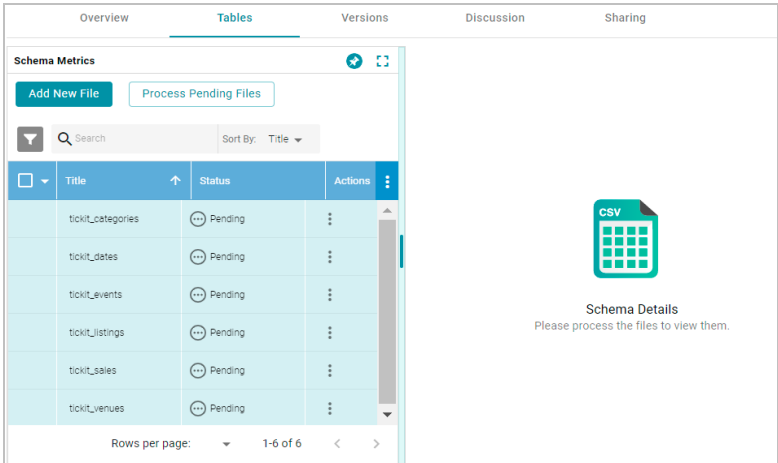
- Follow the appropriate steps below depending on whether the CSV files are on your computer or the shared File Store:

If the files are on your computer:

Note

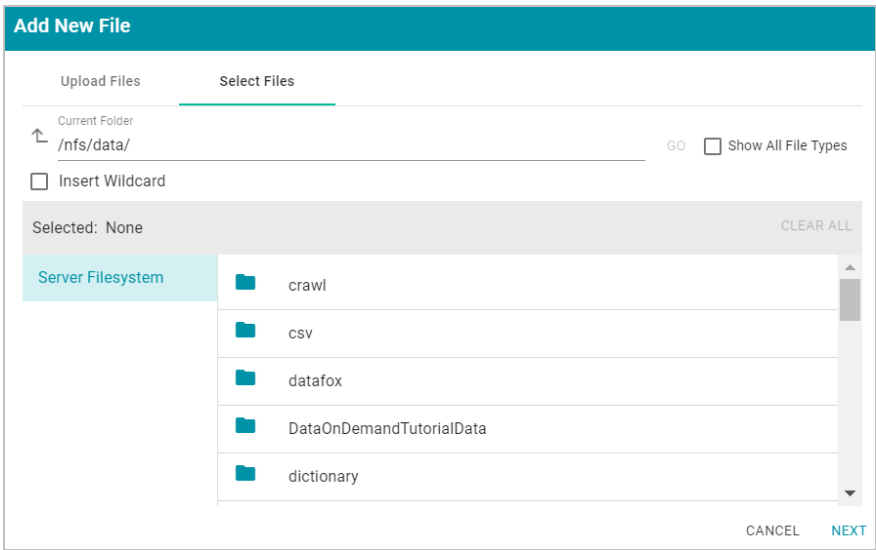
The Upload Files option is a convenient way to do a one-time ingestion so you can quickly get started with your data. It should not be relied upon as part of a regular ingestion workflow unless the server is configured to store uploaded files on the shared file store. For more information, see [Setting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

- a. Drag and drop the files onto the Upload Files tab or click **browse** to navigate to the files and select them. Anzo attaches the files and the **Next** button becomes active.
- b. Click **Next**. Anzo lists the uploaded files on the left side of the screen with a status of Pending. For example:



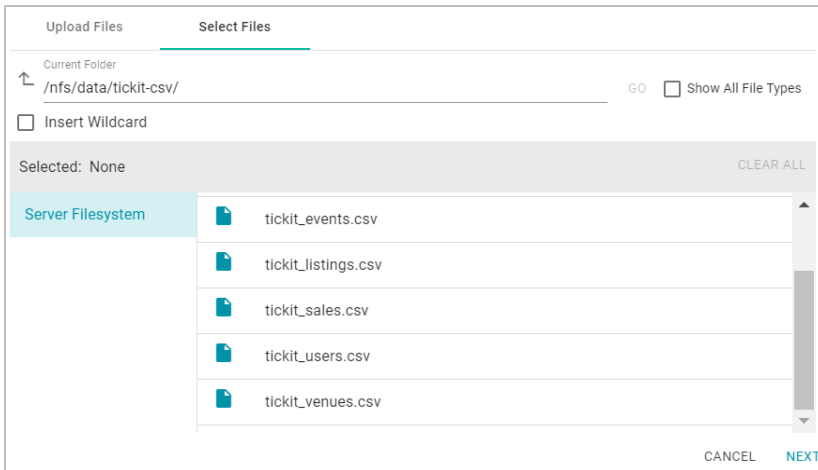
If the files are on the file store:

- a. Click the **Select Files** tab. Anzo displays the file selection dialog box.



- 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.

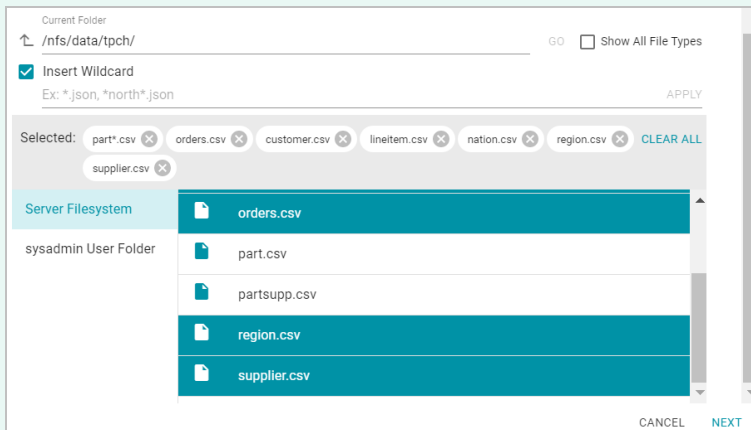
For example:



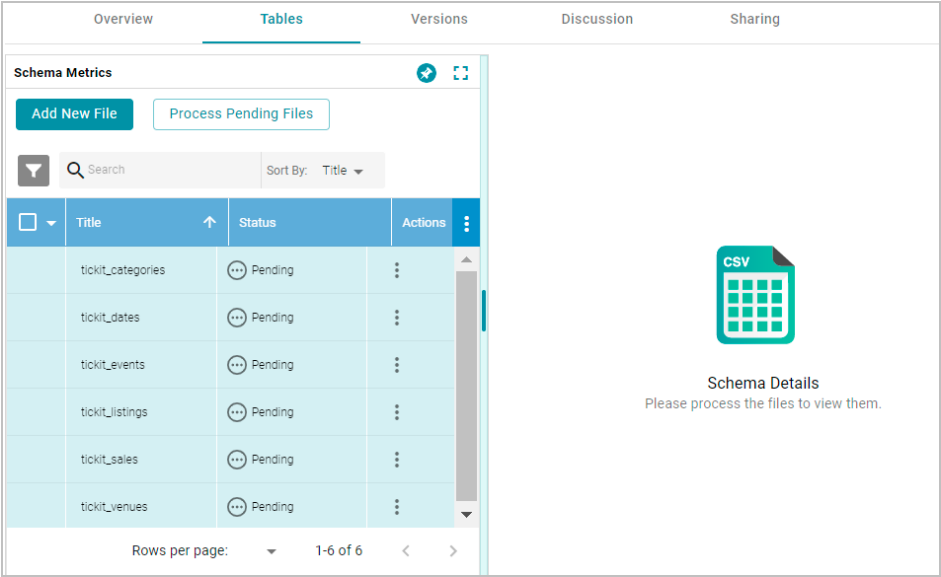
- c. Select each file that you want to import. If you have multiple files with the same schema—the files contain the same columns listed in the same order—and you want the files to be imported as if they are a **single file**, you can select the **Insert Wildcard** option. Then type a string using asterisks 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.



- 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:



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:

The 'Edit CSV File' dialog box is shown. It has a title bar 'Edit CSV File'. Inside, there are fields for 'Title' (ticket_categories) and 'File Path' (/nfs/data/ticketit-csv/ticket_categories.csv) with a 'BROWSE' button. Below these are three checkboxes: 'Contains Header' (checked), 'Use Extended Sample' (unchecked), and 'Infer All Columns as Strings' (unchecked). There are also fields for 'Delimiter', 'Row Delimiter', and 'Csv Escape Character'. At the bottom right, there are 'CANCEL' and 'SAVE & IMPORT' buttons.

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, the **Profile Data**, **Add To Dictionary**, and **Ingest** buttons become available. And 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:

Tickets

Not Versioned

Profile Data

Add To Dictionary

Ingest

Overview

Tables

Versions

Discussion

Sharing

Schema Metrics

Add New File

Process Pending Files

Search

Sort By: Title

Title	Status	Actions
ticket_categories	Processed	
ticket_dates	Processed	
ticket_events	Processed	
ticket_listings	Processed	
ticket_sales	Processed	
ticket_venues	Processed	

Rows per page: 1-6 of 6

ticket_categories

Creator: System Administrator

Last Modified Date: 05/06/2021

Column Count: 4

Row Count: Not Calculated

Sample Data

Metrics

Foreign Keys

catid	catgroup	catname	catdesc
Int	String	String	String
8	Shows	Opera	All opera and light opera
3	Sports	NFL	National Football League
5	Sports	MLS	Major League Soccer
1	Sports	MLB	Major League Baseball
6	Shows	Musicals	Musical theatre
9	Concerts	Pop	All rock and pop music cc

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. For instructions on onboarding the data by letting Anzo automatically generate the Mappings, Model, and ETL Pipeline, see [Ingesting a New Data Source](#). For information about adding a Schema to a Metadata Dictionary, see [Creating a Metadata Dictionary](#).

Related Topics

- [Assigning Primary Keys in an Onboarded Schema](#)
- [Creating or Changing Foreign Keys](#)
- [Ingesting Data](#)
- [Managing Data Source Metadata](#)

Creating a JSON Data Source

This topic provides instructions for creating a JSON data source, scanning a 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 Data Sources. For example:

Data Sources Schemas Mappings Pipelines								
	<input type="text" value="Search"/>	Sort By: Title ▼	View:	<button>Add Data Source ▼</button>				
<input type="checkbox"/>	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021		
	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021		
	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021		
	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021		
Rows per page: 25 ▼ 1-4 of 4 < >								

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

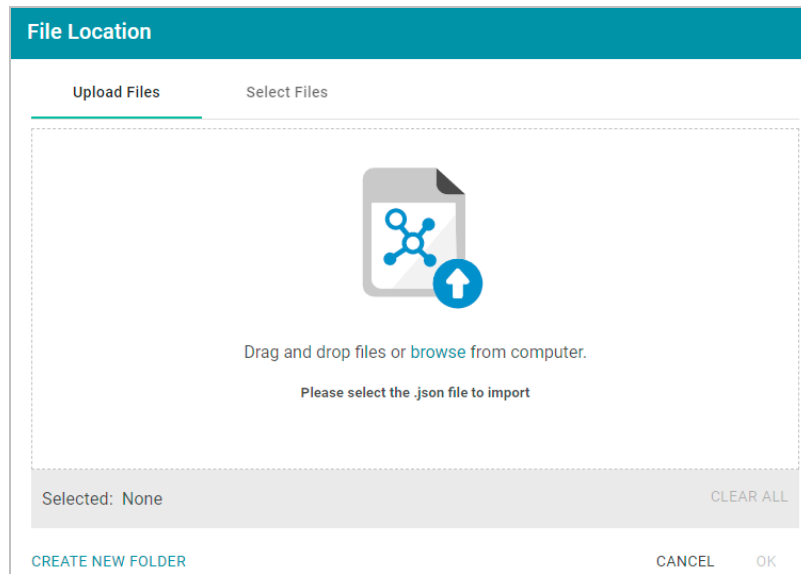
Schema Type
Flat x ▼

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. Anzo displays the File Location dialog box, and the **Upload Files** tab is selected.



5. Follow the appropriate steps below depending on whether you want to import a file for one-time ingestion or you have the files on the shared file store:

If the file is on your computer:

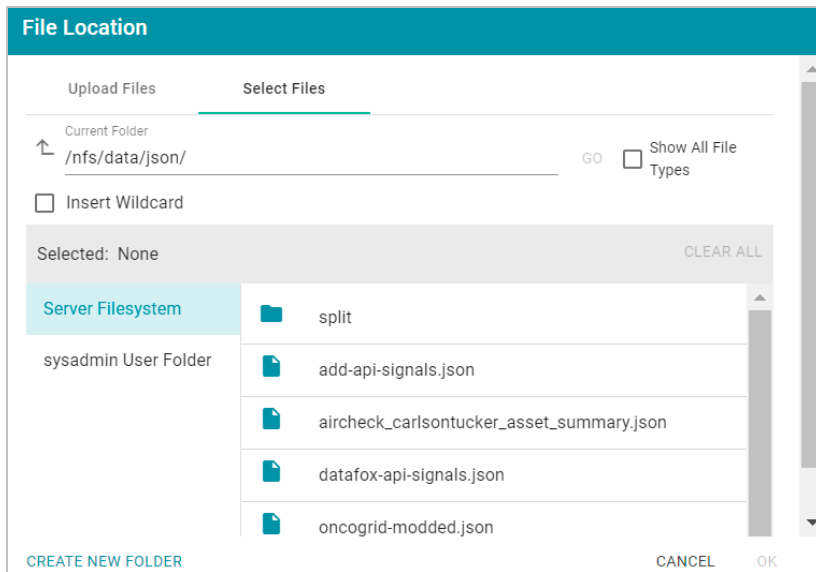
Note

The Upload Files 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 a Base File Store Path for File Uploads](#). Data source files that are routinely updated and re-ingested should be hosted on a configured file store.

- a. Drag and drop the file onto the Upload Files tab or click **browse** to navigate to the file and select it. Anzo attaches the file and the **OK** button becomes active.
- b. 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 **Select Files** tab. Anzo displays the File Location dialog box.
- b. In the File Location dialog box, on the left side of the screen, select the file store for the JSON files. 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:



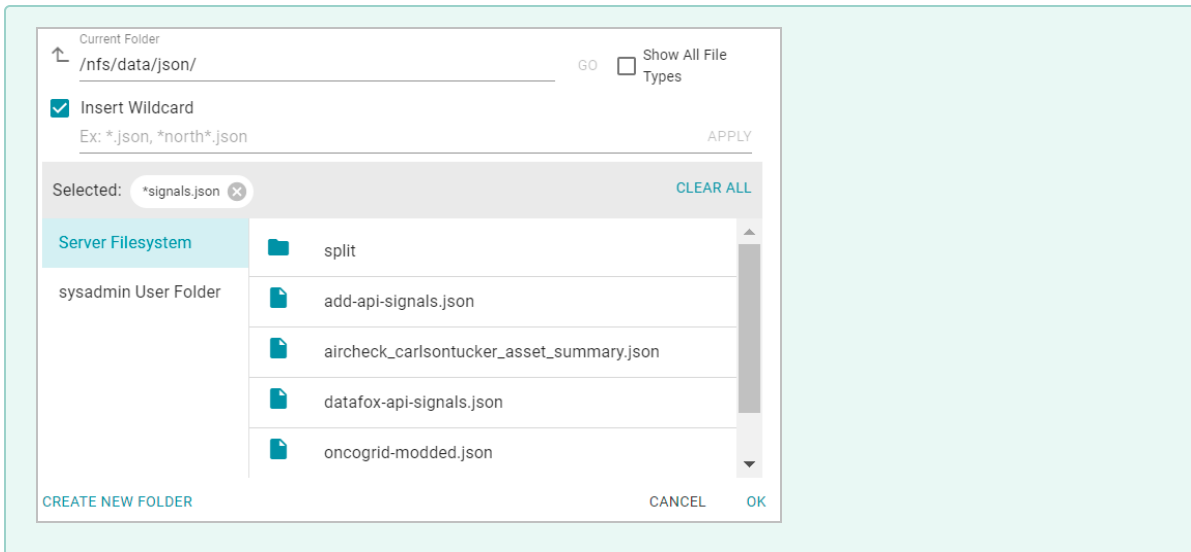
- c. Select the file that you want to import and then click **OK** to close the dialog box. If you have multiple files with the same schema—the files contain the same arrays in the same order—you can select the **Insert Wildcard** option. Then type a string using asterisks as wildcard characters to find 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.



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.

The screenshot shows the 'Advanced' configuration panel. It contains the following fields and controls:

- Schema File Location**: A text input field with a **BROWSE** button to its right.
- Scan Depth (-1 means scan complete file) ***: A text input field containing the value **100**.
- Repeating Element Paths**: A text input field.
- Add Part JSON File locations**: A text input field with a **BROWSE** button to its right.

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:

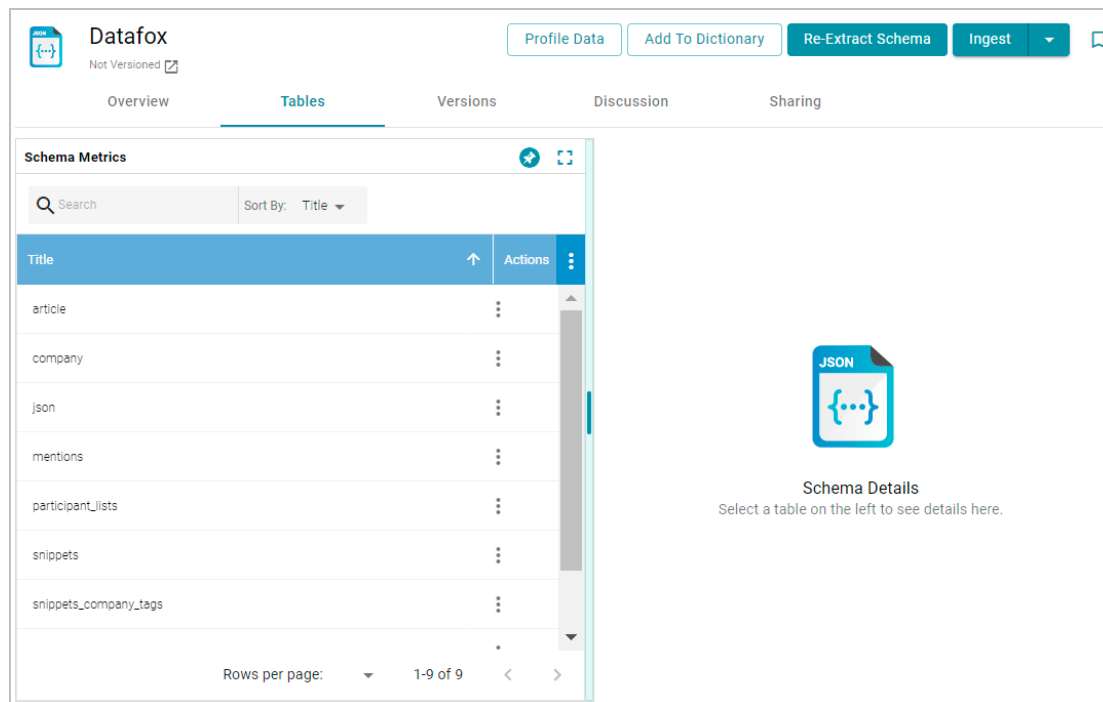
```
{
  "people":
  {
    "personal": {
      "age": 20,
      "gender": "M",
      "name": {
        "first": "John",
        "last": "Doe"
      }
    },
    "vehicles": {
```

```
    "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. For example:



The source data can now be onboarded to Anzo. For instructions on onboarding the data by letting Anzo automatically generate the Mappings, Model, and ETL Pipeline, see [Ingesting a New Data Source](#). For information about adding a Schema to a Metadata Dictionary, see [Creating a Metadata Dictionary](#).

Related Topics

[Ingesting Data](#)

[Managing Data Source Metadata](#)

Creating an XML Data Source

This topic provides instructions for creating an XML data source, scanning a 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 Data Sources. For example:

Data Sources Schemas Mappings Pipelines								
	Search	Sort By: Title	View:	Add Data Source				
<input type="checkbox"/>	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021		
	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021		
	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021		
	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021		
Rows per page: 25 1-4 of 4 < >								

- Click the **Add Data Source** button and select **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

- Specify a name for the data source in the **Title** field, and type an optional description in the **Description** field.
- Click the **XML File Location** field to open the File Location dialog box. The **Upload Files** tab is selected.
- Follow the appropriate steps below depending on whether you want to import a file for one-time ingestion or you have the file on the shared file store:

If the file is on your computer:

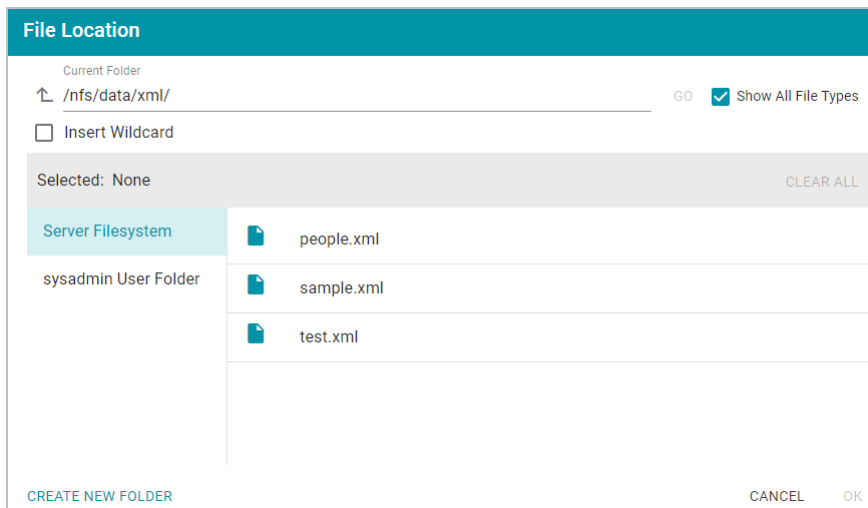
Note

The Upload Files 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 a Base File Store Path for File Uploads](#). Data source files that are routinely updated and re-ingested should be hosted on a configured file store.

- a. Drag and drop the file onto the Upload Files tab or click **browse** to navigate to the file and select it. Anzo attaches the file and the **OK** button becomes active.
- b. 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 **Select Files** tab. Anzo displays the File Location dialog box.
- b. On the left side of the screen, select the file connection for the 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:



- 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
```

- 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:

root								
Creator	System Administrator	Last Modified Date	06/02/2021	Column Count	26	Row Count	Not Calculated	
Sample Data								
root_id	root_pe...	root_pe...	root:peopl...	root:peopl...	root:peopl...	root:pe...	root:peopl...	root:pe...
T String	T String	T String	T String	% Int	T String	T String	% Int	T String
1	1	1	work	12345		1	20	
1	1	2	home	12357		1	20	
1	2	3	home	98765		2	24	
1	1	1	work	12345		1	20	
1	2	3	home	98765		2	24	
1	1	2	home	12357		1	20	

The source data can now be onboarded to Anzo. For instructions on onboarding the data by letting Anzo automatically generate the Mappings, Model, and ETL Pipeline, see [Ingesting a New Data Source](#). For information about adding a Schema to a Metadata Dictionary, see [Creating a Metadata Dictionary](#).

Related Topics

[Ingesting Data](#)

[Managing Data Source Metadata](#)

Creating a SAS Data Source

Follow the instructions below to create 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](#).

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

Data Sources Schemas Mappings Pipelines								
	Search	Sort By: Title	View:	Add Data Source				
<input type="checkbox"/>	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021		
	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021		
	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021		
	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021		
Rows per page: 25 1-4 of 4 < >								

- 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

- Specify a name for the Data 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.

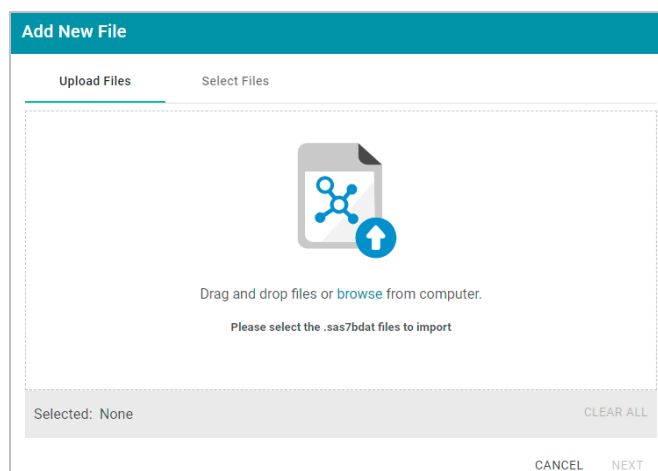
Overview **Tables** Versions Discussion Sharing



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, and the **Upload Files** tab is selected.



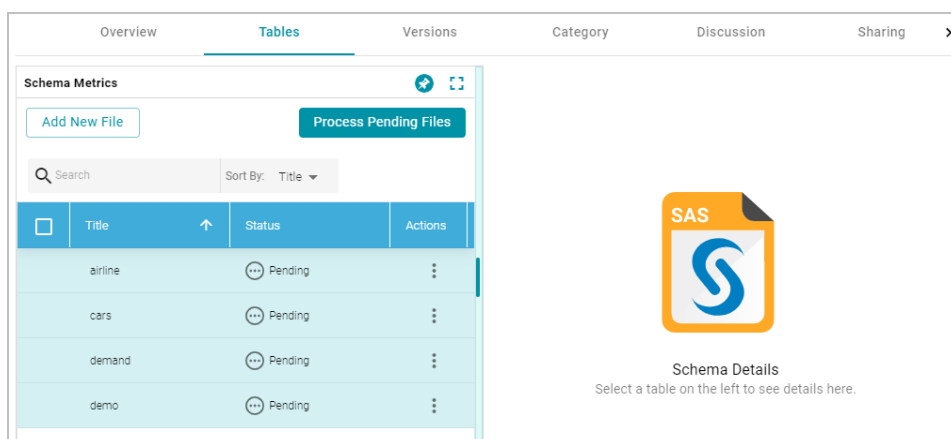
5. Follow the appropriate steps below depending on whether the SAS files are on your computer or the shared File Store:

If the files are on your computer:

Note

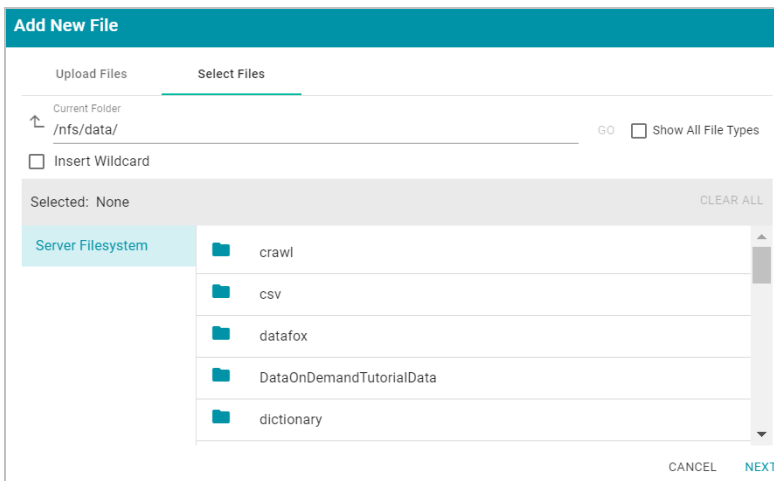
The Upload Files option is a convenient way to do a one-time ingestion so you can quickly get started with your data. It should not be relied upon as part of a regular ingestion workflow unless the server is configured to store uploaded files on the shared file store. For more information, see [Setting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

- Drag and drop the files onto the Upload Files tab or click **browse** to navigate to the files and select them. Anzo attaches the files and the Next button becomes active.
- Click **Next**. Anzo lists the uploaded files on the left side of the screen with a status of Pending. For example:

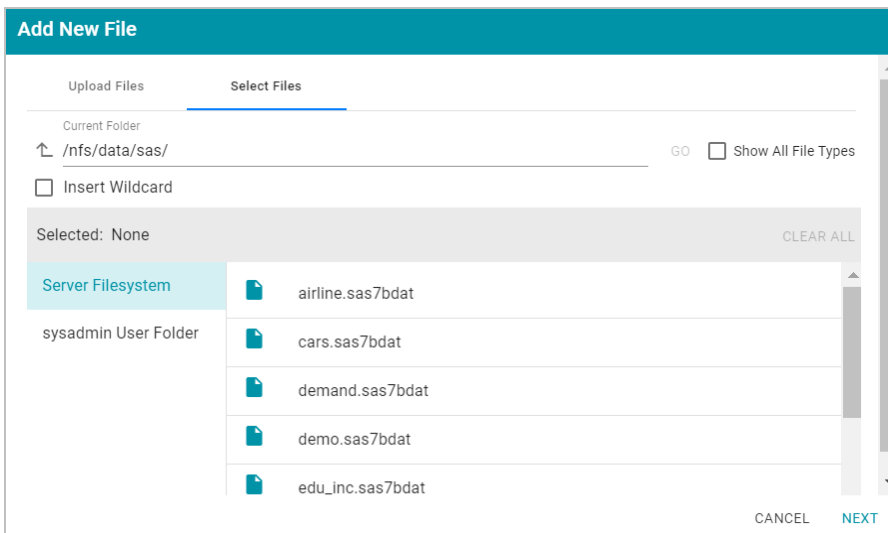


If the files are on the file store:

- a. Click the **Select Files** tab. Anzo displays the file selection dialog box.



- b. On the left side of the screen, select the file store for the SAS 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. For example:

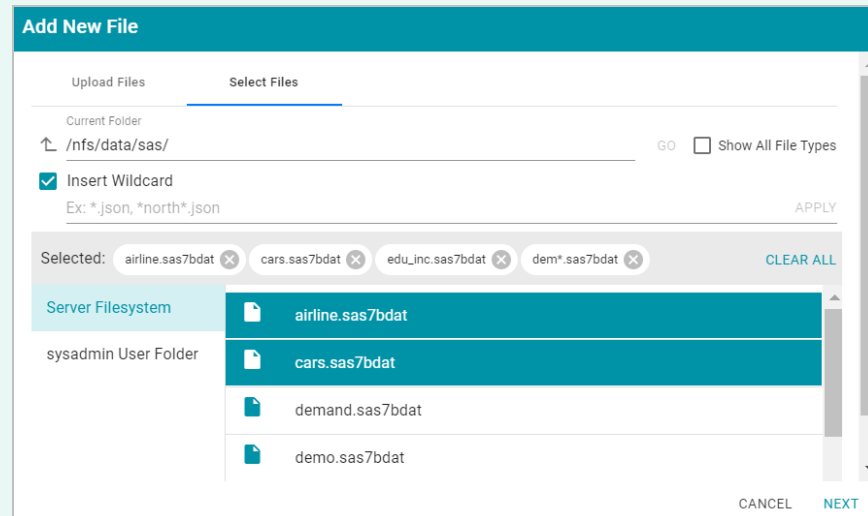


- c. Select each file that you want to import. If you have multiple files with the same schema—the files contain the same columns listed in the same order—and you want the files to be imported as if they are a **single file**, you can select the **Insert Wildcard** option. Then type a string using asterisks as wildcard characters to indicate find the files with similar names. Files that match the specified string will be imported as **one file**. After typing a string, click **Apply** to include that string in the Selected list.

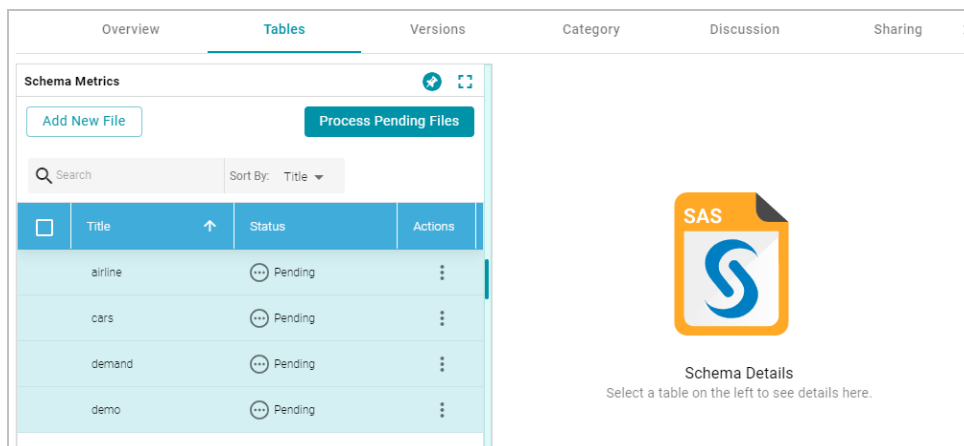
Example

The image below shows a directory with several SAS files. For this example, **demand.sas7bat** and **demo.sas7bat** have the same schema and can be imported as one file. The **Insert Wildcard**

option is selected, and **dem*.sas7bat** is specified to identify the two files.



- 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:



6. 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 File

Title

airline

File Path

/nfs/data/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

Maximum 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.

7. Once the files are processed, the **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:

SAS

Not Versioned

Profile Data

Add To Dictionary

Ingest

Overview

Tables

Versions

Discussion

Sharing

Schema Metrics

Add New File

Process Pending Files

Search

Sort By: Title

Title	Status	Actions
airline	Processed	
cars	Processed	
demo	Processed	

Rows per page:

1-3 of 3

airline

CreatorSystem Administrator

Last Modified Date06/02/2021

Column Count6

Row CountNot Calculated

Sample Data

Metrics

Foreign Keys

Mappings

Pipelines

YEAR	Y	W	R	L	K
Double	Double	Double	Double	Double	Double
1951	1.94799995422363	0.296999990940094	0.393999993801117	1.54999995231628	0.564000010490417
1954	3.02500009536743	0.33500000834465	0.402500003576279	1.96399998664856	0.776000022888184
1949	1.35399997234344	0.259999990463257	0.218099996447563	1.38399994373322	0.5590000115258789
1956	3.97900009155273	0.361000001430511	0.382200002670288	2.43499994277954	0.800000011920929
1952	2.26500010490417	0.310000002384186	0.35589998960495	1.80200004577637	0.574000000953674
1955	3.56200003623962	0.349999994039536	0.396100014448166	2.1159999370575	0.827000021934509

The source data can now be onboarded to Anzo. For instructions on onboarding the data by letting Anzo automatically generate the Mappings, Model, and ETL Pipeline, see [Ingesting a New Data Source](#). For information about adding a Schema to a Metadata Dictionary, see [Creating a Metadata Dictionary](#).

© 2023 Cambridge Semantics, Inc.

Related Topics

[Ingesting Data](#)

[Managing Data Source Metadata](#)















Creating a Parquet Data Source and Ingesting the Data

Follow the instructions below to create a Parquet Data Source and ingest the data.

Note

You can onboard one file or multiple files with the identical format per Parquet 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 Data Sources. For example:

Data Sources Schemas Mappings Pipelines								
<input type="text" value="Search"/>		Sort By: Title		View:  		Add Data Source		
<input type="checkbox"/>	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021	 	
	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021	 	
	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021	 	
	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021	 	
Rows per page: 25 1-4 of 4 < >								

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

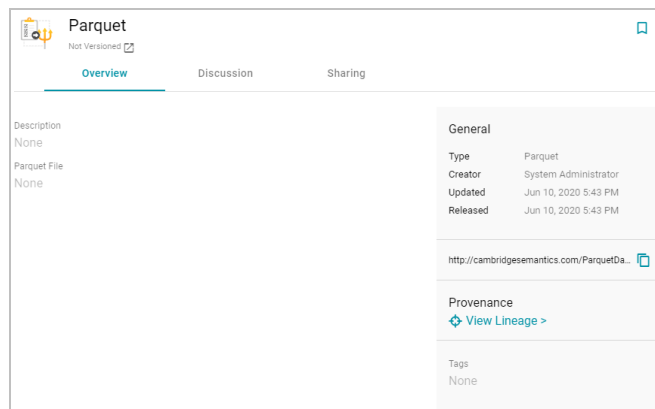
Create Parquet Data Source

Title

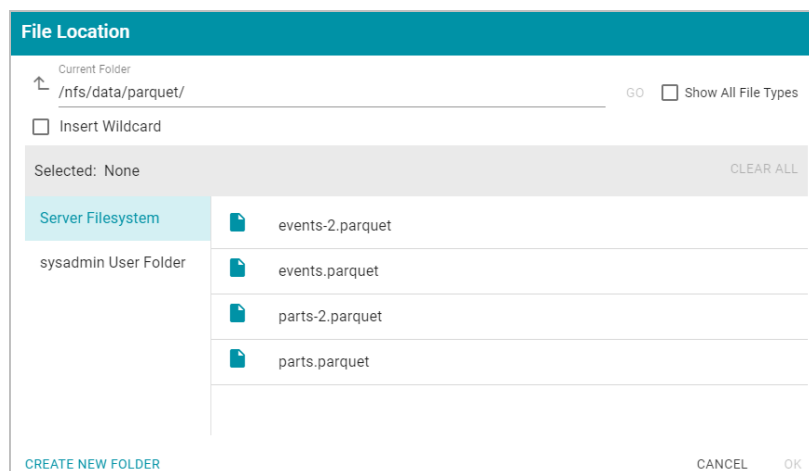
Description

CANCEL SAVE

3. Specify a name for the data 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:



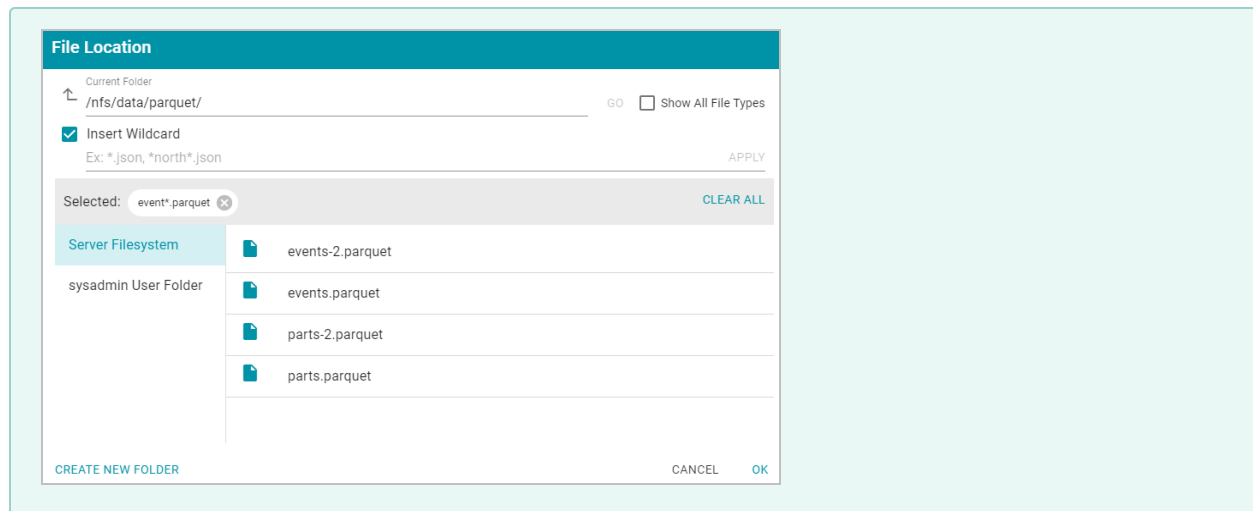
- On the Overview tab, click in the **Parquet File** field to make the value editable. Then click **Browse** to open the File Location dialog box and select the file to import.
- In the File Location dialog box on the left side of the screen, select the file store for the Parquet file. On the right side 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:



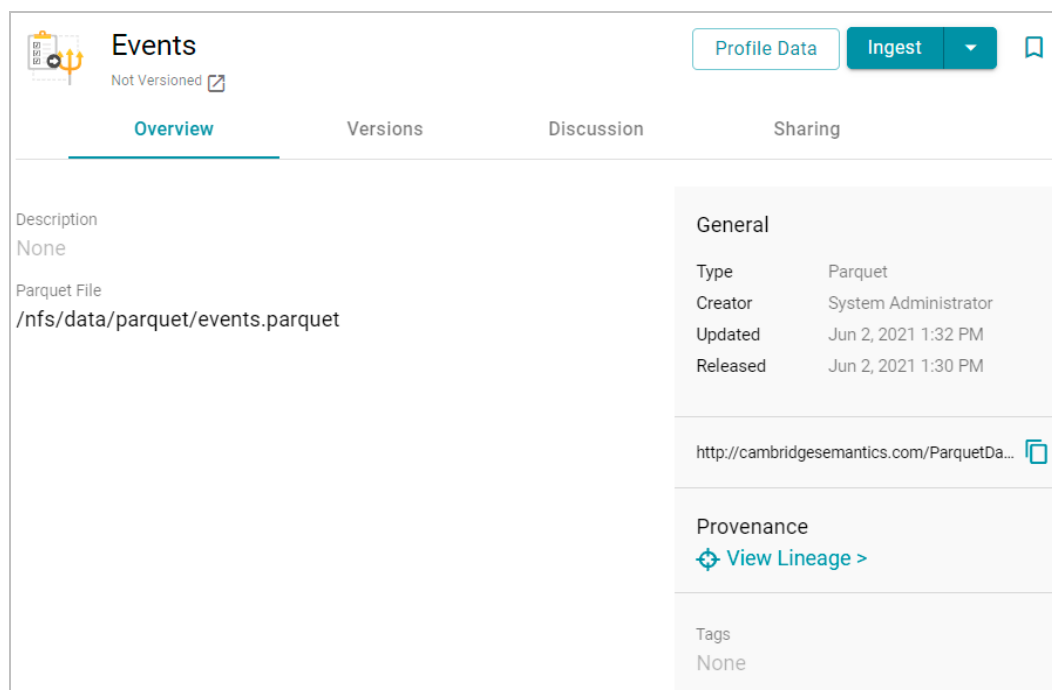
- Select the file that you want to import. If you have multiple files with the identical format you can select the **Insert Wildcard** option. Then type a string using asterisks as wildcard characters to find the files with similar names. Files that match the specified string will be imported as **one 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 to the Parquet File field. Anzo imports the data and automatically generates a Data Model. To view the Model, you can click the Related Model link. For example:



- To onboard the data to Anzo, click the **Ingest** button at the top of the screen. 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](#)). 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:

9. If necessary, click the **Anzo Data Store** field and select the data store for this pipeline.
10. If necessary, click the **Auto Map Engine Config** field and select the ETL engine to use.
11. By default, the **Select all tables** radio button is selected to ingest the data for all tables in the source. If you do not want to add all tables, click the **Custom select** radio button and then select each of the tables to add.
12. 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:

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

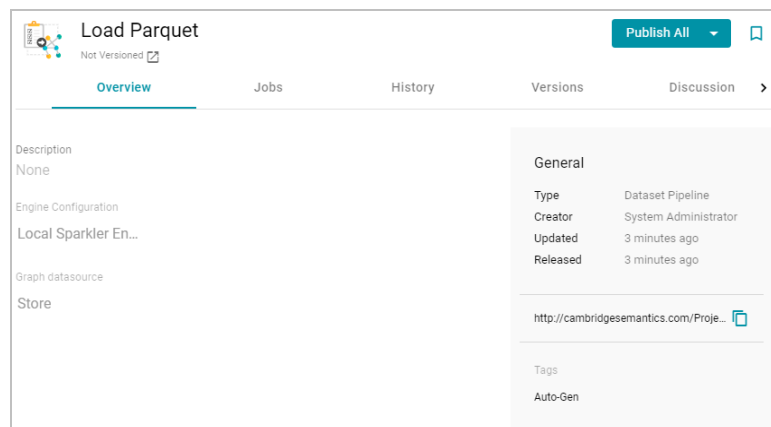
`http://mycompany.com.ontology/property#<class_name>_<property_name>.`

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.

13. Click **Save** if you changed advanced options. Anzo creates a Pipeline and generates the Model and Mappings according to the options you specified.
14. In the main navigation menu under **Onboard**, click **Structured Data**. Then click the **Pipelines** tab.
15. Click the name of the new pipeline. Anzo displays the pipeline overview screen. For example:



16. Click the **Publish All** button to run the ETL jobs in this pipeline.

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 Dataset catalog. From the catalog, you can generate graph Data Profiles and create Graphmarts. See [Blending Data](#) for next steps.

Related Topics

[Ingesting Data](#)

[Creating a New Graphmart](#)

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 Data Sources. For example:

Data Sources Schemas Mappings Pipelines								
	Q Search	Sort By: Title	View:	Add Data Source				
<input type="checkbox"/>	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021		
	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021		
	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021		
	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021		
Rows per page: 25 1-4 of 4 < >								

2. Click the **Schemas** tab to view the list of Schemas. For example:

Data Sources

Schemas

Mappings

Pipelines

▼

Q Search


Sort By: Updated Date ▼

View:

☰

☰

Import Schemas




Parquet

System Administrator | Jun 2, 2021

Data Source: [Events](#)

🔖

⋮




SAS

System Administrator | Jun 2, 2021

Data Source: [SAS](#)

🔖

⋮




People

System Administrator | Jun 2, 2021

Data Source: [People](#)

🔖

⋮



Datafox

System Administrator | Jun 2, 2021

Data Source: [Datafox](#)

🔖

⋮

Rows per page: 25 ▼ 1-9 of 9 < >

3. Click the Schema for which you want to assign Primary Keys. Anzo displays the Tables tab. Click a row on the left side of the screen to display details on the right side of the screen. For example:

Overview

Tables

Versions

Discussion

Sharing

Schema Metrics

Add New File

Process Pending Files

Search

Sort By: Title

Title

Status

Actions

tickit_categories

Processed

tickit_dates

Processed

tickit_events

Processed

tickit_listings

Processed

tickit_sales

Processed

tickit_venues

Processed

Rows per page:

1-6 of 6

tickit_categories

Creator

System Administrator

Last Modified Date

05/06/2021

Column Count

4

Row Count

Not Calculated

Sample Data

Metrics

Foreign Keys

Mappings

catid

Int

catgroup

String

catname

String

catdesc

String

8

Shows

Opera

All opera and light opera

3

Sports

NFL

National Football League

5

Sports

MLS

Major League Soccer

1

Sports

MLB

Major League Baseball

6

Shows

Musicals

Musical theatre

9

Concerts

Pop

All rock and pop music concerts

10

Concerts

Jazz

All jazz singers and bands

4. In the table details, find the column that you want to label as the Primary Key. Hover your pointer over the column name to display additional icons. Edit and Delete icons replace the data type under the column name. For example:

Sample Data		Metrics
catid		catgroup String
1		Sports
4		Sports
3		Sports
2		Sports

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

Edit

Name *

catid

Name of the column

Semantic Type

Int


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
 catid <small>Int</small>		catgroup <small>String</small>
7		Shows
10		Concerts
3		Sports

Repeat the steps above to assign primary keys for additional Schema tables.

Related Topics
















[Creating or Changing Foreign Keys](#)

[Ingesting Data](#)





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 Data Sources. For example:

Data Sources								
<div><div></div><div><input type="text" value="Search"/></div><div>Sort By: Title</div><div>View:  </div><div>Add Data Source</div></div>								
<input type="checkbox"/>	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021	 	
	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021	 	
	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021	 	
	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021	 	
Rows per page: 25 1-4 of 4								

2. Click the **Schemas** tab to view the list of Schemas. For example:

Data Sources	Schemas	Mappings	Pipelines
<div><div><div><div></div><div>Search</div></div><div>Sort By: Updated Date</div><div>View: <div></div><div></div></div></div><div>Import Schemas</div></div>			
<div><div><div><div></div><div>Parquet</div></div><div>System Administrator Jun 2, 2021</div><div>Data Source: Events</div></div><div><div><div><div></div><div>SAS</div></div><div>System Administrator Jun 2, 2021</div><div>Data Source: SAS</div></div><div><div><div><div></div><div>People</div></div><div>System Administrator Jun 2, 2021</div><div>Data Source: People</div></div><div><div><div><div></div><div>Datafox</div></div><div>System Administrator Jun 2, 2021</div><div>Data Source: Datafox</div></div></div><div>Rows per page: 25 1-9 of 9</div></div></div></div>			

3. Click the Schema for which you want to assign Foreign Keys. Anzo displays the Tables tab. Click a row on the left side of the screen to display details on the right side of the screen. For example:

Overview

Tables

Versions

Discussion

Sharing

Schema Metrics

Add New File

Process Pending Files

Search

Sort By: Title

Title

↑

Status

Actions

tickit_categories

Processed

tickit_dates

Processed

tickit_events

Processed

tickit_listings

Processed

tickit_sales

Processed

tickit_venues

Processed

Rows per page: 1-6 of 6

tickit_categories

Creator System Administrator Last Modified Date 05/06/2021 Column Count 4 Row Count Not Calculated

Sample Data

Metrics

Foreign Keys

Mappings

catid	catgroup	catname	catdesc
Int	String	String	String
8	Shows	Opera	All opera and light opera
3	Sports	NFL	National Football League
5	Sports	MLS	Major League Soccer
1	Sports	MLB	Major League Baseball
6	Shows	Musicals	Musical theatre
9	Concerts	Pop	All rock and pop music concerts
10	Concerts	Jazz	All jazz singers and bands

4. Click the **Foreign Keys** tab on the right side of the screen. The tab lists any existing active Foreign Keys. For example:

Active Keys		Suggested Keys						
<div> <div> <div></div> <div>Search</div> </div> <div>Sort By: Title</div> <div>View: <div></div></div> </div> <div>Create</div>								
<input type="checkbox"/>	Title	Source Table	Source Columns	Target Table	Target Columns	Updated Date	Tags	Actions
	categories	ticket_categories	catid	ticket_events	catid	Jun 16, 2020		
	dates	ticket_dates	dateid	ticket_events	dateid	Jun 17, 2020		

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](#).

- 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

- On the Create Foreign Key screen, specify a name for the key in the **Name** field.
- 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.
- 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 **ticket_events** table becomes the foreign key and references the values from the primary key column, **dateid**, in the **ticket_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

ticket_events

dateid

Only View Selected Columns ☒

dateid

2109

1982

Target Table

ticket_dates

dateid

Only View Selected Columns ☒

dateid

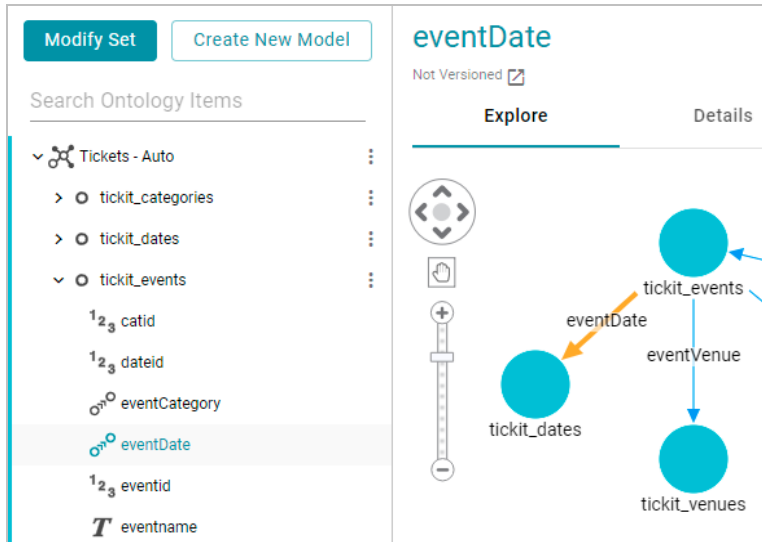
1833

1836

CANCEL SAVE

9. 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:



Related Topics

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

[Ingesting Data](#)

Managing Data Source Metadata

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

- [Creating a Metadata Dictionary](#)
- [Configuring Data Source Categories](#)
- [Generating a Source Data Profile](#)

Creating a Metadata Dictionary

Metadata Dictionaries are similar to Data 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.

This topic provides instructions for creating and managing data 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.

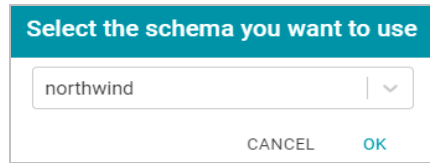
1. In the Anzo application, expand the **Onboard** menu and click **Structured Data**. Anzo displays the Data Sources screen, which lists any existing Data Sources. For example:

Data Sources								
Schemas Mappings Pipelines								
	<input type="text" value="Search"/>	Sort By: Title	View:	<button>Add Data Source</button>				
<input type="checkbox"/>	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021		
	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021		
	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021		
	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021		
Rows per page: 25 1-4 of 4								

2. 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:

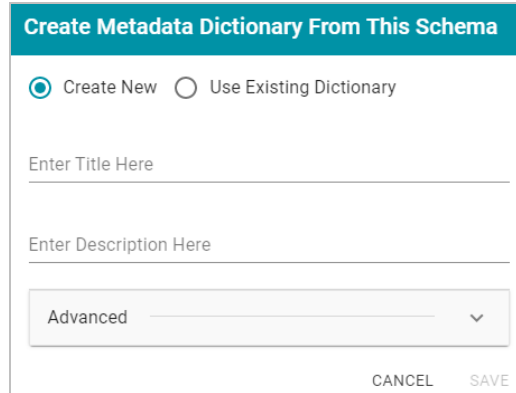
DB Not Versioned		Profile Data	Add To Dictionary	Ingest													
Overview	Tables	Versions	Discussion	Sharing													
<div>Schema Metrics</div> <div>Create Schemas From Query Import Schemas</div> <div><input type="text" value="Search"/> Sort By: Table Name</div> <table> <tr> <th><input type="checkbox"/></th><th>Schema Name</th><th>Table Name</th><th>Type</th><th>Actions</th><th></th></tr> <tr> <td></td><td>northwind</td><td></td><td></td><td></td><td></td></tr> </table>						<input type="checkbox"/>	Schema Name	Table Name	Type	Actions			northwind				
<input type="checkbox"/>	Schema Name	Table Name	Type	Actions													
	northwind																
<div>Schema Details</div> <p>Select a table on the left to see details here.</p>																	

- 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:



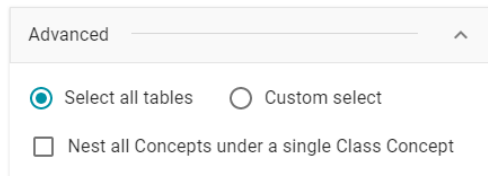
A dialog box titled "Select the schema you want to use". It features a text input field containing "northwind" and a dropdown arrow on the right. Below the input field are two buttons: "CANCEL" and "OK".

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



A dialog box titled "Create Metadata Dictionary From This Schema". It has two radio buttons: "Create New" (selected) and "Use Existing Dictionary". Below these are two text input fields labeled "Enter Title Here" and "Enter Description Here". At the bottom is a dropdown menu currently showing "Advanced". Two buttons, "CANCEL" and "SAVE", are at the bottom right.

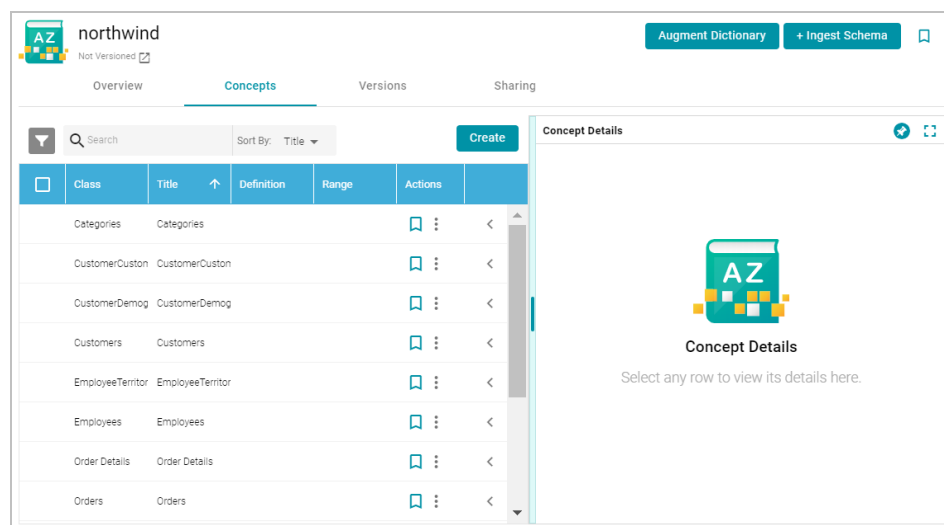
- In the dialog box, leave the **Create New** radio button selected.
- Enter a name for the dictionary in the **Title** field and specify an optional description in the **Description** field.
- To configure additional options, such as limiting the schema tables that are added to the dictionary, click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



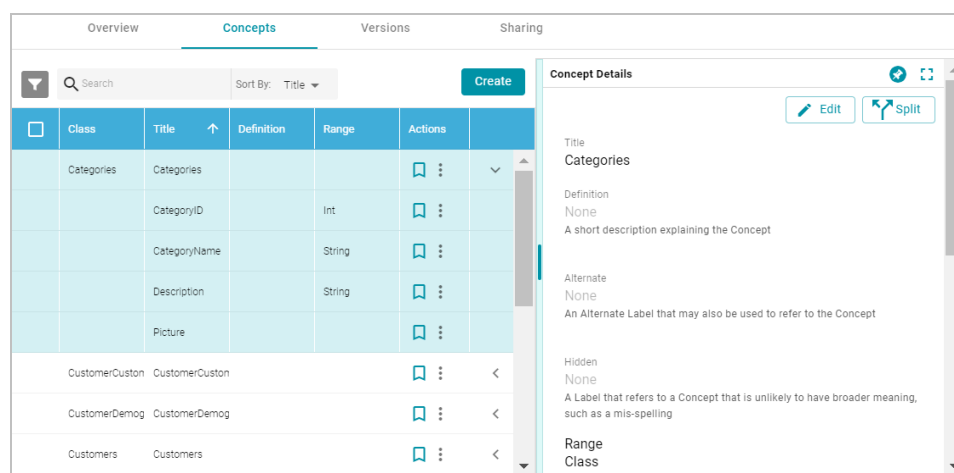
An expanded "Advanced" options section. It shows a dropdown menu with "Advanced" and an upward arrow. Below are two radio buttons: "Select all tables" (selected) and "Custom select". At the bottom is a checkbox labeled "Nest all Concepts under a single Class Concept".

- 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.

- 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:



- 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.

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

<div> <div> <div> <div></div> <div>Search</div> </div> <div>Sort By: Title</div> <div>View: <div></div> <div></div> </div> </div> <div>Import</div> <div>Create</div> </div>		
<input type="checkbox"/>	Title	Description
	Customer Data	Metadata dictionary generated from Customers in Customers
	Flight Data	Metadata dictionary generated from Flights in Flights

- 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

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

AZ

Doc

Not Versioned

Augment Dictionary

Overview

Concepts

Versions

Sharing

Search

Sort By: Title

Create

No concepts found

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:

Search

Sort By: Title

Create

Selected: CustomerDemographics

	Class	Title		Definition	Range	Actions	
<input type="checkbox"/>	Categories	Categories					<
<input checked="" type="checkbox"/>	CustomerCustomerDer	CustomerCustomerDer					<
<input checked="" type="checkbox"/>	CustomerDemographic	CustomerDemographic					<
<input type="checkbox"/>	Customers	Customers					<
<input type="checkbox"/>	EmployeeTerritories	EmployeeTerritories					<

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

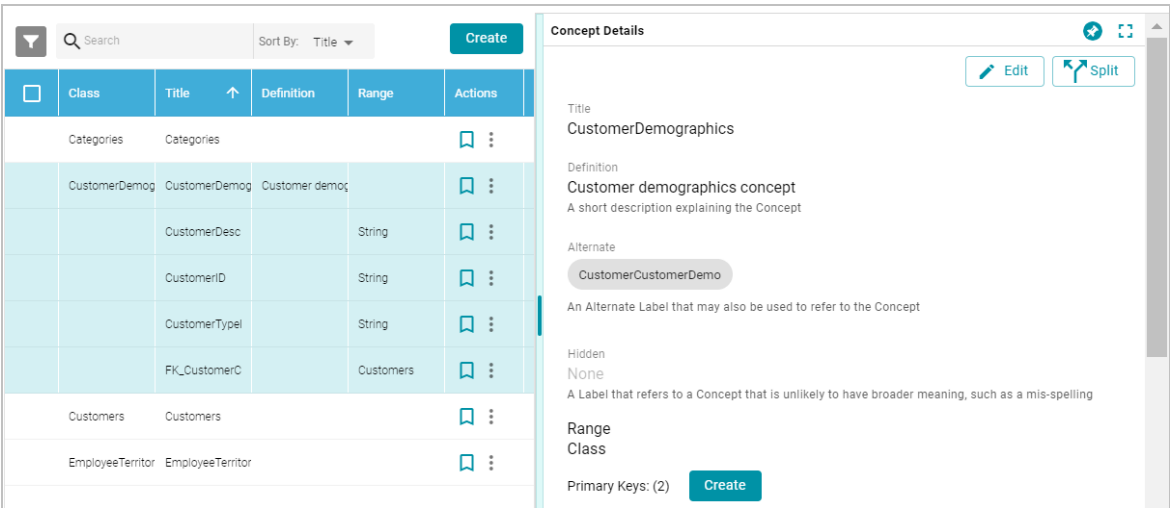
	Concept 1 - CustomerDemographics	Concept 2 - CustomerCustomerDemo	Merged Concepts
Title	<input type="checkbox"/> CustomerDemographics	<input type="checkbox"/> CustomerCustomerDemo	<div>Title</div>
Definition			<div>Definition</div>
			<div></div>
			<div>CANCEL</div> <div>SAVE</div>

2. 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.
3. 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	<div>Title</div> <div>CustomerDemographics</div>
Definition			<div>Definition</div> <div>Customer demographics concept</div>
			<div></div>
			<div>CANCEL</div> <div>SAVE</div>

4. 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.
5. 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.



Creating a Concept

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

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

The 'Create New Concept' form is shown. It has a title bar 'Create New Concept'. Below it, 'New Concept Type:' has three radio buttons: 'Data Property' (selected), 'Object Property', and 'Class'. There are input fields for 'Title *', 'Definition', 'Alternate' (with an 'ADD' button), 'Hidden', 'Range *' (with a dropdown), and 'Class *' (with a dropdown). At the bottom are 'CANCEL' and 'SAVE' buttons.

- 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 ADD

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 border="1"> <thead> <tr> <th>Schema</th> <th>Table</th> </tr> </thead> <tbody> <tr> <td>Flights</td> <td>flights10k</td> </tr> </tbody> </table>	Schema	Table	Flights	flights10k	Schema Locations: (none) <table border="1"> <thead> <tr> <th>Schema</th> <th>Table</th> </tr> </thead> <tbody> <tr> <td colspan="2">You can select and drag locations from the original concept to here</td> </tr> </tbody> </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 border="1"> <thead> <tr> <th>Title</th> <th>Definition</th> </tr> </thead> <tbody> <tr><td>SCHEDULED_ARRIVAL</td><td></td></tr> <tr><td>SECURITY_DELAY</td><td></td></tr> <tr><td>SCHEDULED_TIME</td><td></td></tr> </tbody> </table>	Title	Definition	SCHEDULED_ARRIVAL		SECURITY_DELAY		SCHEDULED_TIME		Property Concepts: (none) <table border="1"> <thead> <tr> <th>Title</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td colspan="2">You can select and drag property concepts from the original class concept to here</td> </tr> </tbody> </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	Enter Title Here delays																
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 border="1"> <thead> <tr> <th>Schema</th> <th>Table</th> </tr> </thead> <tbody> <tr> <td>Flights</td> <td>flights10k</td> </tr> </tbody> </table>	Schema	Table	Flights	flights10k	Schema Locations: (none) <table border="1"> <thead> <tr> <th>Schema</th> <th>Table</th> </tr> </thead> <tbody> <tr> <td colspan="2">You can select and drag locations from the original concept to here</td> </tr> </tbody> </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: (24) <table border="1"> <thead> <tr> <th>Title</th> <th>Definition</th> </tr> </thead> <tbody> <tr><td>SCHEDULED_ARRIVAL</td><td></td></tr> <tr><td>SCHEDULED_TIME</td><td></td></tr> <tr><td>TAXI_OUT</td><td></td></tr> </tbody> </table>	Title	Definition	SCHEDULED_ARRIVAL		SCHEDULED_TIME		TAXI_OUT		Property Concepts: (7) <table border="1"> <thead> <tr> <th>Title</th> <th>Definition</th> </tr> </thead> <tbody> <tr><td>AIRLINE_DELAY</td><td></td></tr> <tr><td>ARRIVAL_DELAY</td><td></td></tr> <tr><td>WEATHER_DELAY</td><td></td></tr> </tbody> </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.

Related Topics

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

Configuring Data Source Categories

Anzo's Category manager provides a way to define metadata about a data source that can be used to classify or catalog data for a customized asset browsing and search experience. Categories describe the properties in a source but are independent of the instance data. When categories are configured for a source, they are displayed as choices in the list of quick filters that are available when sorting data sources. This topic provides instructions for configuring data source categories.

Note















Before you can configure categories for a data source, the **Category** setting must be enabled for the classes in the data model for that source. If necessary, open the model for editing and select the **Category** checkbox for each class that you want to list as a category. For example:

The screenshot shows the 'Manage Working Set' interface. On the left, there is a 'Search Ontology Items' section with a list of items including 'venueid', 'ticket_listings', 'ticket_sales', 'ticket_users', 'card', 'city', 'email', 'firstname', 'lastname', and 'likebroadway'. The 'ticket_users' item is selected. On the right, there are configuration options for the selected item. The 'Storage' section has two radio buttons: 'Allocate storage as needed' (selected) and 'Top level storage container'. Below this, there are three dropdown menus: 'firstname', 'Resource Template', and 'Graph Template'. The 'Category' checkbox is checked and circled. At the bottom, there is a 'Provenance' section with a 'View Lineage >' link. A 'Save' button is at the bottom left.

Make sure that you save the model changes. You do not need to re-ingest the data source. The Category tab for that data source becomes available once the model is saved. For more information about changing a model, see [Editing a Model](#).


Follow the steps below to configure categories.


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

Data Sources								
Search		Sort By: Title	View:  		Add Data Source			
<input type="checkbox"/>	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021		
	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021		
	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021		
	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021		



Rows per page: 25 1-4 of 4 < >

2. Click the data source for which you want to configure categories. Then click the **Category** tab. For example:


Tickets
 Not Versioned

Profile Data
 Add To Dictionary
 + Ingest
 

Overview
 Tables
 Versions
 Category
 Discussion

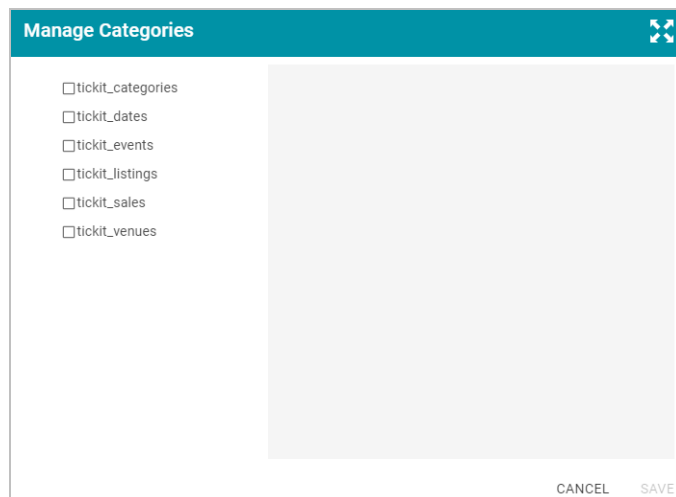
Categories
 


Manage Categories

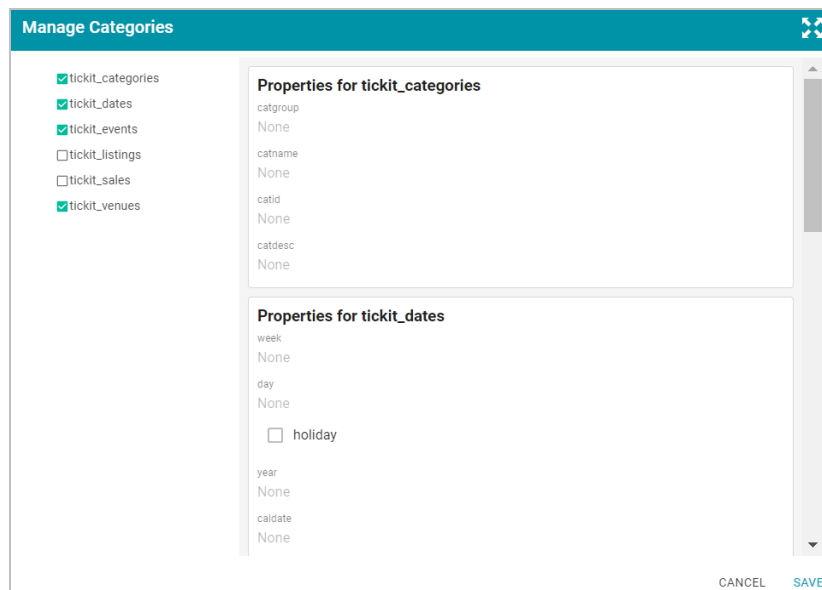
There are no categories applied to this object.

Category/Class Properties
 Select a Category or Class on the left to see details here

3. Click the **Manage Categories** button. The Manage Categories dialog box is displayed, which lists each of the classes that are designated as categories in the data model. For example:

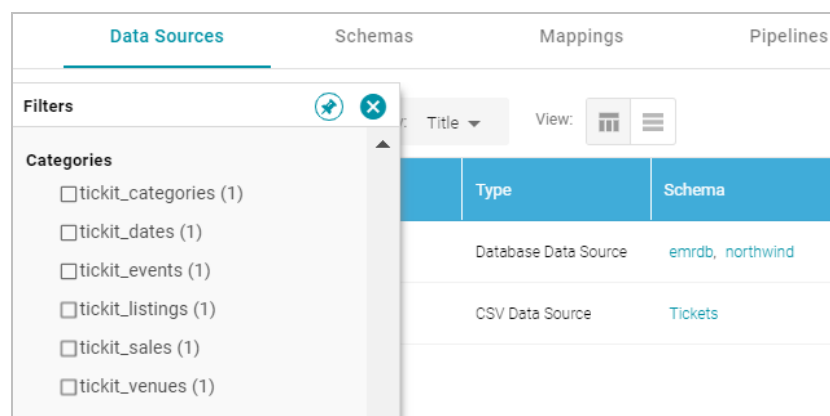


4. On the left side of the screen, select the checkbox next to each class that you want to display as a category. When you select a class, the properties for that class are displayed on the right side of the screen. For example:



5. In the list of properties for each class, you can edit the values to categorize the source data for that property. For example, if you know that the source data has date values that fall in a specific date range, you can specify that range in a date-related property, such as the "year" property in the image above. To add a description for a property, click the value field under the property to make the field editable. The characters that are supported depend on the data type of the property. Click the checkmark icon (✓) to save the change. Repeat this step for any of the properties that you want to describe.
6. When you have finished adding values, click **Save** to save the configuration and close the Manage Categories dialog box. Categories can be modified any time from the Category tab.

Categories are displayed as quick filters in the Filters panel that is available when sorting the data source list on the Data Sources screen. Open the Filters panel by clicking the filter icon (🔍) in the top left corner of the screen. For example:



When a category is selected, the properties for that class are also displayed in the Filters panel.

Related Topics

[Configuring Dataset Categories](#)

Generating a Source Data Profile

To help users assess the quality of the data coming from a Data Source, Anzo provides the ability to calculate metrics for each source. When metrics are generated, Anzo profiles the entire source data set and reports statistics for each table in the Schema, such as the number of populated, null, or empty rows for each column in a table and the number of rows for each column grouped by value. It also reports column-level metrics such as the smallest and largest values in a column, the number of unique values, and the value that appears most often. For Schemas with multiple tables, Anzo also generates a list of Foreign Key suggestions between tables that include the same column.

Note

AnzoGraph performs Data Source Profiling. That means AnzoGraph needs to connect directly to any Data Sources that you profile. For file-based data sources, AnzoGraph uses the Graph Data Interface (GDI) Java plugin to access the sources. The plugin is not installed by default and needs to be deployed to AnzoGraph. For instructions, see [Deploy the Graph Data Interface Java Plugin](#).

For database Data Sources, AnzoGraph requires the GDI plugin plus the same drivers that you have configured to access those sources in Anzo. To configure AnzoGraph to profile relational data sources, copy the necessary driver .jar files to AnzoGraph and restart the database. For instructions, see [Deploy Optional Drivers for Accessing Database Sources](#).

Tip

Dynamic AnzoGraph deployments are pre-configured with JDBC drivers for the following database types:

- Apache Derby, Hive, and Impala
- Google BigQuery
- IBM DB2
- Microsoft SQL Server
- MariaDB/MySQL
- Hyper SQL Database (HSQLDB)
- PostgreSQL
- SAP Sybase (jTDS)

Follow the instructions below to generate and review data quality 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 Data Sources. For example:

Data Sources								
Schemas Mappings Pipelines								
	<input type="text" value="Search"/>	Sort By: Title ▼	View:	<button>Add Data Source ▼</button>				
	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021		
	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021		
	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021		
	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021		
Rows per page: 25 ▼ 1-4 of 4 < >								

2. Click the data source that you want to profile. Anzo displays the Tables tab for the source, which lists the schema and table details. For example:

Tickets
Not Versioned

Profile Data Add To Dictionary + Ingest

Overview **Tables** Versions Category Discussion Sharing

Schema Metrics

Add New File Process Pending Files

Search Sort By: Title

Title	Status	Actions
ticket_categories	Processed	
ticket_dates	Processed	
ticket_events	Processed	
ticket_listings	Processed	
ticket_sales	Processed	
ticket_users	Processed	
ticket_venues	Processed	

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

- Click the **Profile Data** button at the top of the screen. AnzoGraph runs several queries to calculate metrics about the source data. The process may take several minutes. 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:

Search Sort By: Start Time View: [Icons]

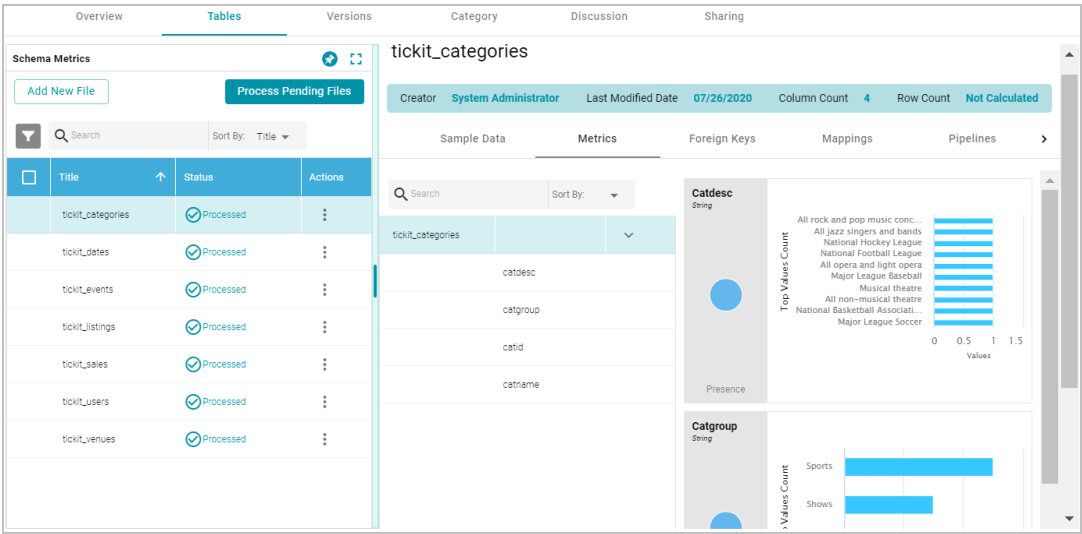
Provisioning Graphmart - Tickets graphmart.
Provisioning Graphmart - Tickets graphmart.
Successfully activated the graphmart on AnzoGraph
Run Time: 26 secs | End time: Aug 1, 2020 - 9:30:54 pm
View Logs

Profiling Metrics DataSource - Tickets
Profiling Metrics DataSource - Tickets
Profiling Metrics
Elapsed Time: 1 min, 34 secs
Computing: Pearson Skew
Cancel
View Logs

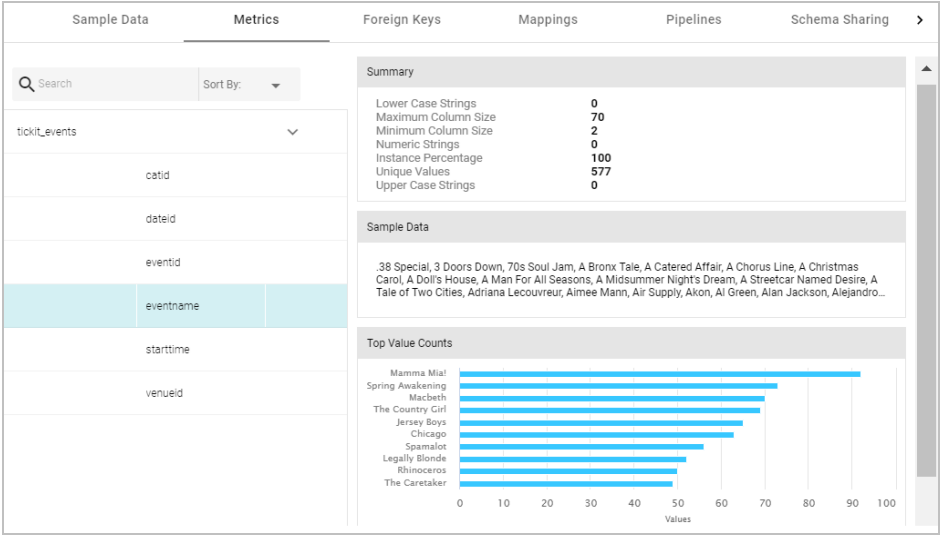
Rows per page: 20 1-6 of 6

View All History

- When the process is complete, select a schema row on the left side of the screen, and then click the **Metrics** tab next to the Sample Data tab in the middle of the screen. Anzo displays the metrics for the selected table. For example:



On the far right of the screen, Anzo shows the metrics for each column in the selected table. To select a different table, click another row on the left side of the screen. To view additional metrics for a column in the table, click a column name in the middle of the screen. For example:



Depending on the data type of the column, one or more of the following metrics are shown:

- Extrema Metric: Shows the smallest and largest values.
- Median Metric: Shows the middle value.
- Mode Metric: Shows the value that appears most often.
- Unique Values Metric: Shows the number of unique values.

For additional metrics based on the type of data quality checks needed, contact Cambridge Semantics. For information about generating metrics for an onboarded Dataset, see [Generating a Dataset Data Profile](#).

Related Topics

[Managing Data Source Metadata](#)

Ingesting Data

The topics in this section provide instructions for ingesting data from structured data sources using the **Ingest** 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](#)

Ingesting a New Data Source

Follow the instructions below to set up the Ingest workflow for a new Data 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.

For information about initial Data Source creation, see [Adding Data Sources and Schemas](#).

Tip

For instructions on ingesting an updated Data 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 Data Sources. For example:

Data Sources								
Search		Sort By: Title		View:		Add Data Source		
	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021		
	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021		
	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021		
	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021		
Rows per page: 25 1-4 of 4								

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

DB

Not Versioned

Overview

Tables

Versions

Discussion

Sharing

Schema Metrics

Create Schemas From Query

Import Schemas

Search

Sort By: Table Name

	Schema Name	Table Name	Type	Actions
	emrdbsmall			
	northwind			

emr_activity

Creator System AdministratorLast Modified Date 05/31/2021Column Count 8Row Count Not Calculated

Sample Data

Metrics

Foreign Keys

Mappings

PATIENTID	ACTIVITY...	ACTIVITY...	ACTIVITY...	ACTIVITY...	ACTIVITY...	ACTIVITY...	ACTIVITY...
Int	Int	Int	Date	Int	Int	String	Int
9	9	90	2012-05-06	77	44	TREADMILL REP	
7	7	82	2012-05-06	139	144	XRAYS	
10	10	50	2012-05-06	139	144	TREADMILL REP	
6	6	98	2011-02-03	129	17	TREADMILL REP	
1	1	68	2010-03-04	24	49	WELLNESS	
4	4	65	2015-08-09	17	144	WRITTEN PROG	
5	5	50	2013-06-07	24	17	WOUND CHECK	

3. 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

northwind

CANCEL 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](#)). 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:

The screenshot shows the 'Ingest' configuration window. It has a teal header with the word 'Ingest'. Below it are three main sections: 'Source Data Connection' with a dropdown showing 'DB', 'Anzo Data Store' with a dropdown showing 'Anzo Data Store', and 'Auto Map Engine Config' with a dropdown showing 'Local Sparkler Engine'. Below these are two radio buttons: 'Select all tables' (which is selected) and 'Custom select'. At the bottom of the main area is an 'Advanced' dropdown menu. At the very bottom right are 'CANCEL' and 'SAVE' buttons.

4. If necessary, click the **Anzo Data Store** field and select the Data Store for this pipeline. For information about creating an Anzo Data Store, see [Creating an Anzo Data Store](#).
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:

The screenshot shows the 'Advanced' configuration window. It has a header with 'Advanced' and an upward arrow. Below are two radio buttons: 'Create New Model' (selected) and 'Use Existing Model'. Below these are three text input fields: 'Schema Ontology URI', 'Schema Class Prefix', and 'Schema Property Prefix'. At the bottom is a checkbox labeled '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:

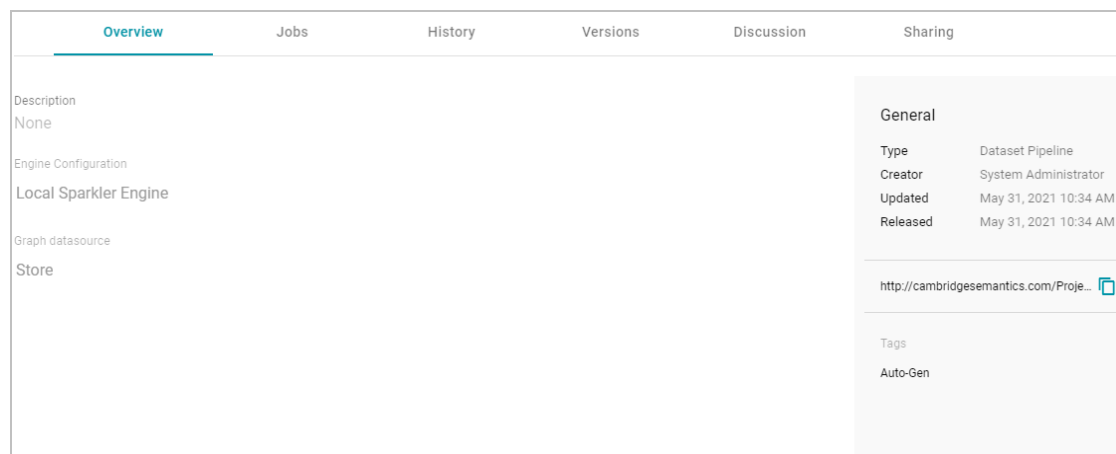
```
http://cambridgesemantics.com/ont/autogen/xx/<schema_name>#<class_name>_<property_name>
```

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>`.

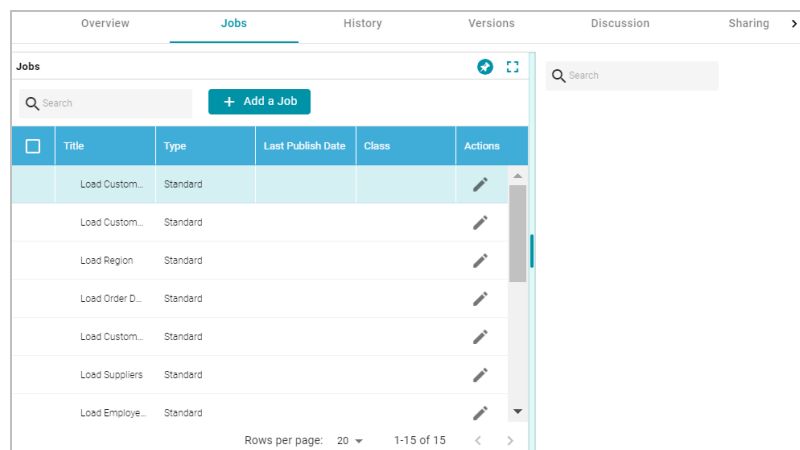
Tip

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.
- Click **Save** if you changed advanced options. Anzo creates a Pipeline and generates the Model and Mappings according to the options you specified.
 - In the main navigation menu under **Onboard**, click **Structured Data**. Then click the **Pipelines** tab.
 - Click the name of the Pipeline that you created. Anzo displays the Pipeline Overview screen. For example:



- 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:



This image shows an example of a pipeline that has been published previously and has job history:

Overview

Jobs

History

Versions

Discussion

Sharing

Jobs

Q Search

+ Add a Job

<input type="checkbox"/>	Title	Type	Last Publish Date	Class	Actions
<input checked="" type="checkbox"/>	Load Customer...	Standard	07/01/2020 11:40AM	CustomerDemograp...	
<input checked="" type="checkbox"/>	Load Customers	Standard	07/01/2020 11:42AM	Customers	
<input checked="" type="checkbox"/>	Load Region	Standard	07/01/2020 11:41AM	Region	
<input checked="" type="checkbox"/>	Load Order Deta...	Standard	07/01/2020 11:41AM	Order Details	
<input checked="" type="checkbox"/>	Load Customer...	Standard	07/01/2020 11:42AM	CustomerDemograp...	
<input checked="" type="checkbox"/>	Load Suppliers	Standard	07/01/2020 11:43AM	Suppliers	
<input checked="" type="checkbox"/>	Load Employees	Standard	07/01/2020 11:41AM	Employees	

Rows per page: 20 1-15 of 15

Q Search

Start Time

End Time

Run Status

Actions

07/01/2020 11:3...

07/01/2020 11:40...

Completed

View Logs

Rows per page: 20 1-1 of 1

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 is updated, 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 Data Sources. For example:

Data Sources Schemas Mappings Pipelines								
	<input type="text" value="Search"/>	Sort By: Title	View:	<button>Add Data Source</button>				
<input type="checkbox"/>	Title	Type	Schema	Creator	Created Date	Updated Date	Actions	
	Datafox	JSON Data Source	Datafox	System Administrator	May 6, 2021	May 6, 2021		
	DB	Database Data Source	employees, new-hir...	System Administrator	May 6, 2021	May 6, 2021		
	Flight Data	CSV Data Source	Flight Data	System Administrator	May 10, 2021	May 10, 2021		
	Tickets	CSV Data Source	Tickets	System Administrator	May 6, 2021	May 6, 2021		
Rows per page: 25 1-4 of 4 < >								

2. On the Data Sources screen, click the name of the data source to re-ingest. Anzo displays the Tables screen. For example:

DB

Not Versioned

Profile Data

Add To Dictionary

Ingest

Overview

Tables

Versions

Discussion

Sharing

Schema Metrics

Create Schemas From Query

Import Schemas

Search

Sort By: Table Name

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

emr_activity

CreatorSystem AdministratorLast Modified Date05/31/2021Column Count8Row CountNot Calculated

Sample Data

Metrics

Foreign Keys

Mappings

PATIENTID	ACTIVITYID	ACTIVITYDATE	ACTIVITYTIME	ACTIVITYTYPE	ACTIVITYLOCATION	ACTIVITYDESCRIPTION
9	9	2012-05-06	77	44	TREADMILL REP	
7	7	2012-05-06	139	144	XRAYS	
10	10	2012-05-06	139	144	TREADMILL REP	
6	6	2011-02-03	129	17	TREADMILL REP	
1	1	2010-03-04	24	49	WELLNESS	
4	4	2015-08-09	17	144	WRITTEN PROG	
5	5	2013-06-07	24	17	WOUND CHECK	

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:

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

Alphabetical list ...

Categories

Category Sales f...

Add Table

Reload Schema

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

northwind

CANCEL

OK

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

© 2023 Cambridge Semantics, Inc.

Ingest

Source Data Connection
DB

Anzo Data Store
Store

Auto Map Engine Config
Local Sparkler Engine

☒ Select all tables ☐ Custom select

Advanced

CANCEL SAVE

5. 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 'Overview' tab of a pipeline configuration. The main content area displays the following information:

- Description:** None
- Engine Configuration:** Local Sparkler Engine
- Graph datasource:** Store
- Dataset:** DB - northwind to Store (with a link to the dataset URL)
- Managed Editions:** A table showing one edition: 'Default Edition' with a description 'Contains the latest successfully publishe...' and a date '05/31/2021 11:00AM'.
- Saved Editions:** A section with a search bar, a 'Sort By: Title' dropdown, and a 'Create New Edition' button. It shows 'No editions found'.

On the right side, there is a 'General' sidebar with the following details:

- Type:** Dataset Pipeline
- Creator:** System Administrator
- Updated:** May 31, 2021 10:34 AM
- Released:** May 31, 2021 10:34 AM
- Tags:** Auto-Gen

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. 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](#).

Related Topics

[Ingesting a New Data Source](#)

[Managing Dataset Editions](#)

[Blending Data](#)

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 the available data sources. For example:

Data Sources					
Schemas					
Mappings					
Pipelines					
<div><div><div></div><div>Search</div></div><div>Sort By: Title <div></div></div><div>View: <div></div></div><div>Import</div><div>Create <div></div></div></div>					
	Title	Description	Type	Schema	Actions
	Datafox		JSON Data Source	Datafox	<div></div> <div></div>
	DB		Database Data Source	emrdb, northwind	<div></div> <div></div>
	Flights		CSV Data Source	Flights	<div></div> <div></div>
	GHIB		CSV Data Source	GHIB	<div></div> <div></div>

2. On the Data Sources screen, click the name of the Data 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 'Tables' tab is selected, displaying a list of schemas and a table named 'emr_activity'. The table has columns: PATIENTID, ACTIVITYID, ACTIVITYDATE, ACTIVITYTYPE, ACTIVITYCOUNT, ACTIVITYCOST, and ACTIVITYDESCRIPTION. The table is sorted by Table Name. The 'Ingest' button is highlighted in the top right corner.

- Click the **Ingest With 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 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 "northwind" and a dropdown arrow. Below the input field 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](#) for more information). For example:

The screenshot shows the "Ingest With Dictionary" dialog box. It contains a "Dictionary" field with "emrdb" selected. Below it are two dropdown fields: "Anzo Data Store" and "ETL Engine". There are two radio buttons: "Select all tables" (selected) and "Custom select". Below these is an "Advanced" dropdown menu. At the bottom are two buttons: "CANCEL" and "SAVE".

- If necessary, click the **Anzo Data Store** field and select the data store for this pipeline. For information about creating an Anzo data store, see [Creating an Anzo Data Store](#).
- If necessary, click the **ETL Engine** field and select the ETL engine to use for this pipeline.

6. 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.
7. 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>
```

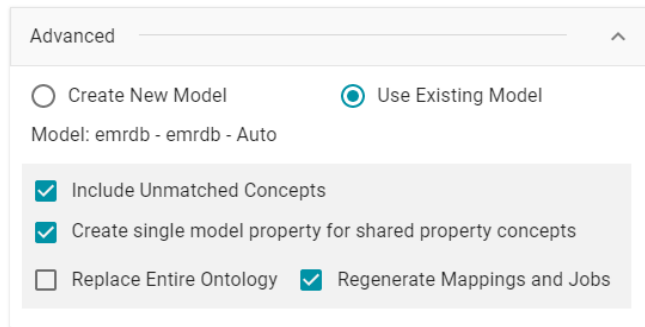
Where xx is a hash snippet based on the model's GUID. If you want to specify a different format for property URIs, type the prefix to use in this field. you can type that URI into the Schema Property Prefix field. For example, a prefix such as `http://mycompany.com.ontology/property` results in property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

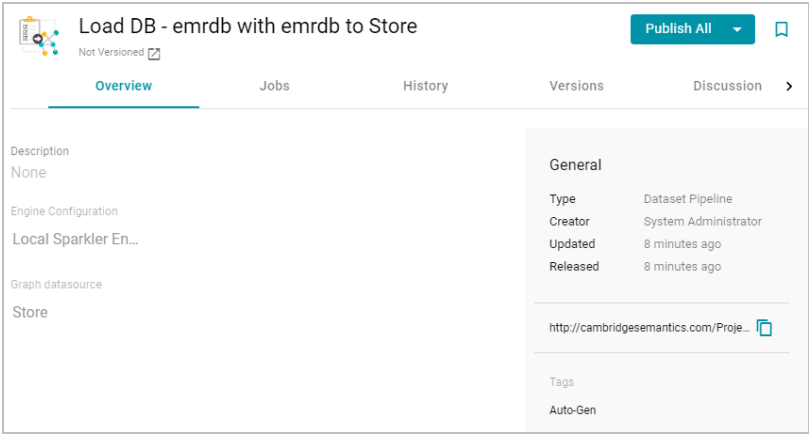
☐ Replace Entire Ontology ☒ Regenerate Mappings and Jobs

The list below describes the options:

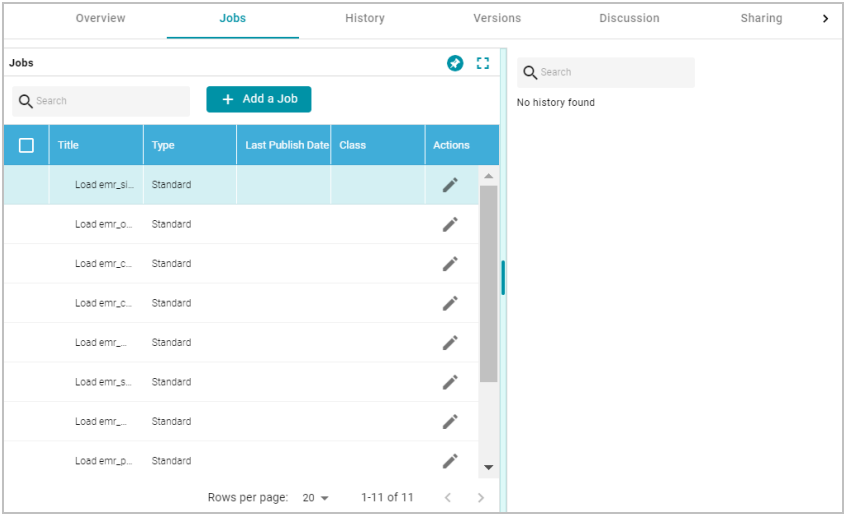
- **Use Existing Model:** Select this option if you want to reuse the existing model. Anzo displays the name of the model that will be used and presents additional model-related options at the bottom of the screen.
- **Include Unmatched Concepts:** This option specifies whether to ingest new data that 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.

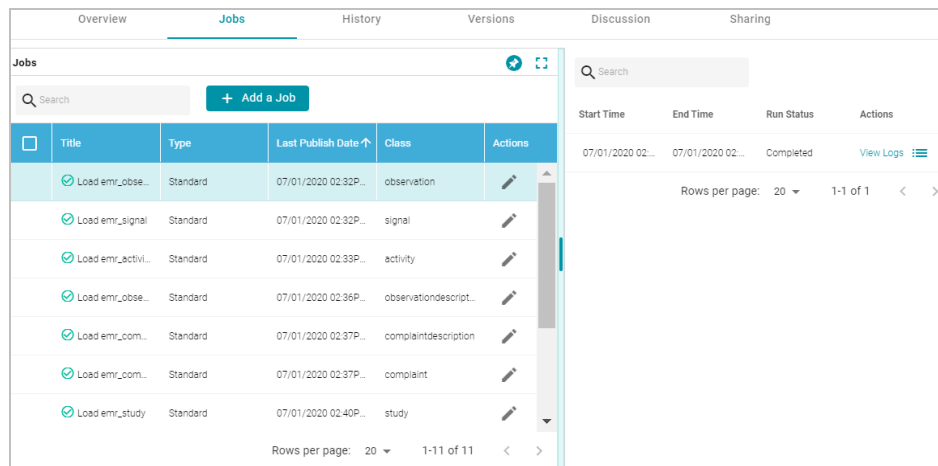
- 9. In the main navigation menu under **Onboard**, click **Structured Data**. Then click the **Pipelines** tab.
- 10. Click the name of the pipeline to run. Anzo displays the pipeline overview screen. For example:



- 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:



This image shows an example of a pipeline that has been published previously and has job history:



The screenshot shows the 'Jobs' tab in the Anzo interface. It features a table with columns: Title, Type, Last Publish Date, Class, and Actions. There are 11 jobs listed, all of type 'Standard' and published on 07/01/2020. The jobs are: Load emr_obse..., Load emr_signal, Load emr_activ..., Load emr_obse..., Load emr_com..., Load emr_com..., and Load emr_study. Each job has an 'Actions' button with a pencil icon. To the right of the table, there is a search bar and a table with columns: Start Time, End Time, Run Status, and Actions. The first row shows a completed run on 07/01/2020. Below this table, there are pagination controls: 'Rows per page: 20' and '1-1 of 1'.

Start Time	End Time	Run Status	Actions
07/01/2020 02:...	07/01/2020 02:...	Completed	View Logs

- 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 Dataset catalog. From the catalog, you can generate graph Data Profiles and create Graphmarts. See [Blending Data](#) for next steps.

Related Topics

[Creating a Metadata Dictionary](#)

[Managing Dataset Editions](#)

[Ingesting a New Data Source](#)

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

[Blending Data](#)

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](#).

- [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

Search

Sort By: Title

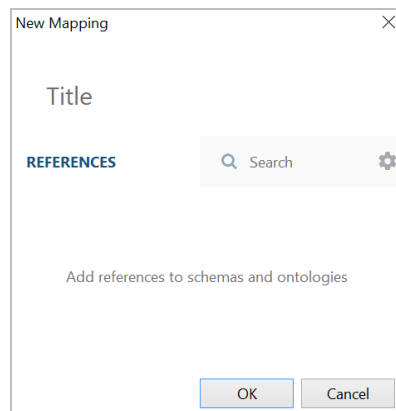
View:

Add Mappings

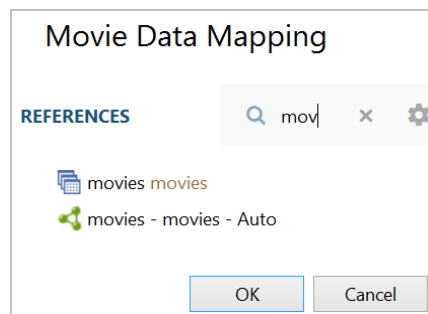
<input type="checkbox"/>	Title	Description	Updated Date	Tags	Actions
	DB - emrdb - emr_acti...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
	DB - emrdb - emr_com...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
	DB - emrdb - emr_com...		Jun 18, 2020	Auto-Gen	<div><div></div><div>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

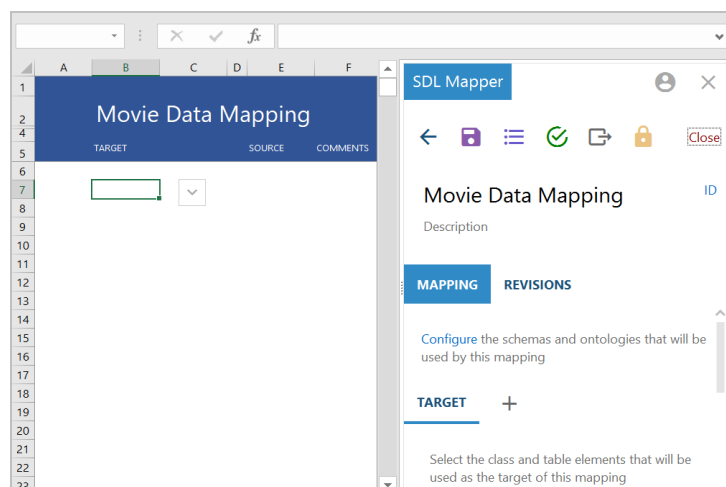
2. Click the **Add Mappings** button at the top of the screen and select **Add Mappings**. Anzo opens the Anzo for Office plugin in Microsoft Excel and prompts you to enter the Anzo server connection information.
3. Provide your server connection and login information and then click the arrow icon (➔)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



4. In the **Title** field, type a name for the mapping.
5. 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.



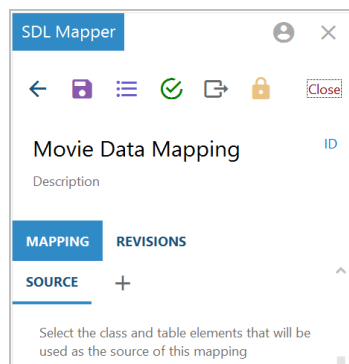
6. Hover over a source in the list of references to display the plus icon (+). Click the icon for each source that you want to add as a reference, and then click **OK**. The new mapping opens. The mapping workbook is in the center pane and the mapping menu and configuration details are displayed in the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



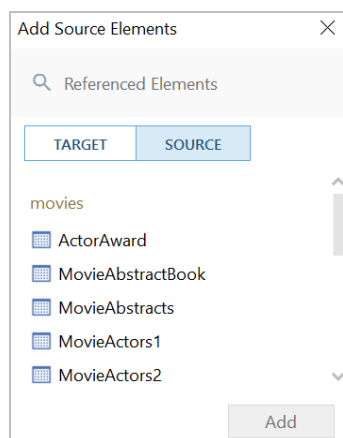
Define the Source for the Mapping

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

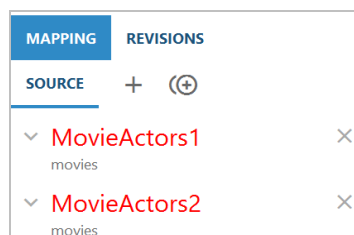
1. In the right pane, scroll to the **Source** section in the **Mapping** tab.



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



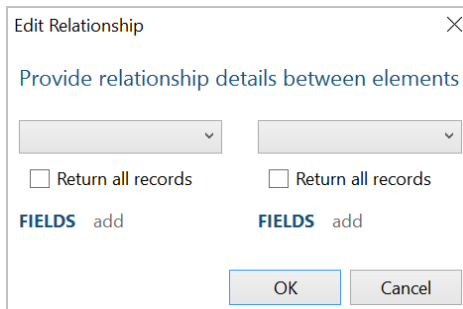
3. Select a source to add to the mapping and click **Add**. If you want to add additional sources, select another source and click **Add** again. Anzo adds the source to the list under Source. You can click the X icon to the right of a source name to delete that source from the list.



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 (two circles with a plus sign) next to the plus icon so that you can create a relationship between the sources by specifying the join criteria.

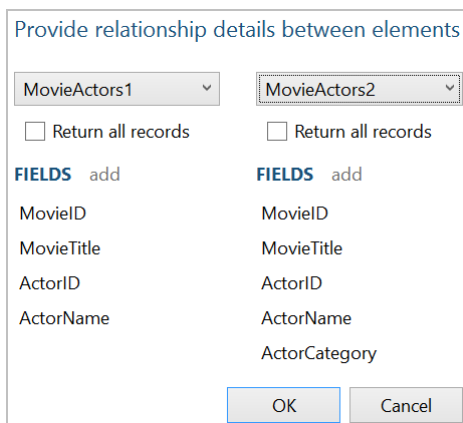
Follow these steps to specify relationships:

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



The 'Edit Relationship' dialog box has a title bar with a close button. Below the title bar is the instruction 'Provide relationship details between elements'. There are two empty dropdown menus at the top. Below each dropdown is a checkbox labeled 'Return all records'. Under each checkbox is a 'FIELDS' label followed by an 'add' button. At the bottom are 'OK' and 'Cancel' buttons.

- 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.

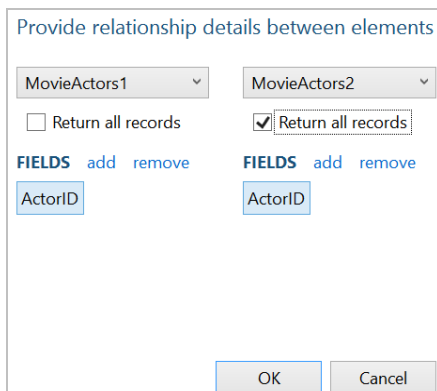


The dialog box now shows 'MovieActors1' and 'MovieActors2' in the dropdown menus. The 'Return all records' checkboxes are still unchecked. The 'FIELDS' lists are populated with the following fields:

- Left side: MovieID, MovieTitle, ActorID, ActorName
- Right side: MovieID, MovieTitle, ActorID, ActorName, ActorCategory

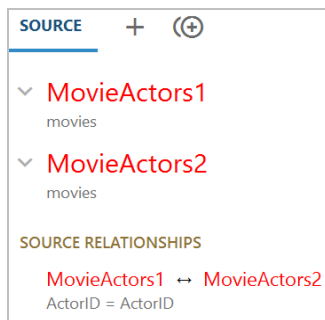
 The 'add' buttons are still present. 'OK' and 'Cancel' buttons are at the bottom.

- 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.



The dialog box shows the 'Return all records' checkbox for 'MovieActors2' checked. In the 'FIELDS' lists, 'ActorID' has been selected on both the left and right sides, indicated by blue highlights. The 'add' and 'remove' buttons are now visible next to the 'FIELDS' labels. 'OK' and 'Cancel' buttons are at the bottom.

- 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.



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

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

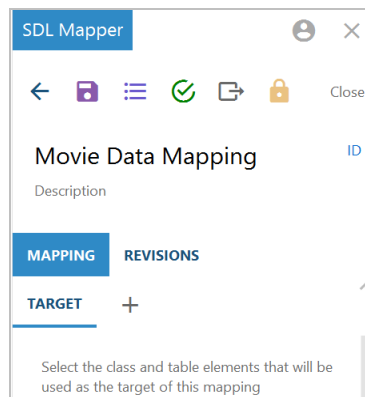
Define the Target for the Mapping

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

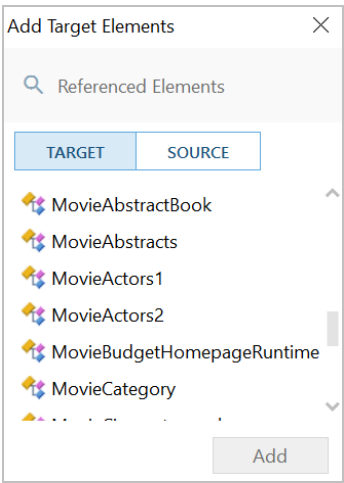
Tip

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](#).

- 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:

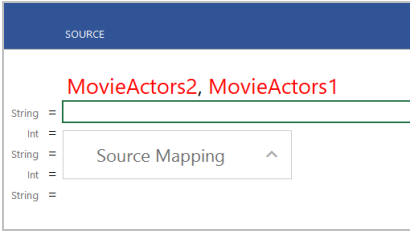
Movie Data Mapping	
TARGET	SOURCE
MovieActors	MovieActors1, MovieActors2
ActorID	Int
ActorName	String
MovieID	Int
MovieTitle	String

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.

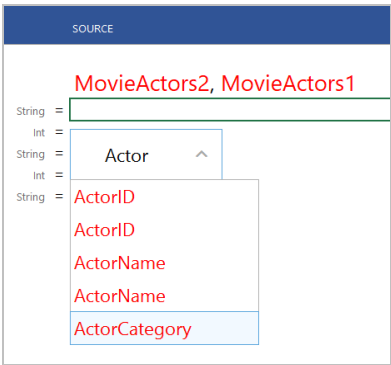
Map the Source Elements to Target Elements

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.



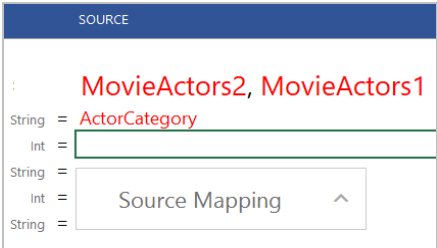
2. In the Source Mapping text box, start typing the source field name. Anzo displays a list of fields that match the text.



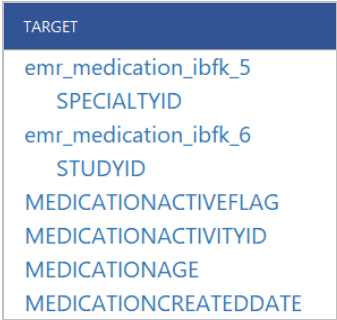
Important

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.



4. 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:



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.

5. 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"> 1. Click the target element above the row where you want to add a new pair. 2. Click the lines icon (≡) that appears to the left of the target name. 3. Click the Add icon (+) that becomes available under the lines icon. Anzo selects the target cell and opens the Target Mapping text box. 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. 5. In the Source column, click the cell that corresponds to the target you added. Anzo opens the Source Mapping text box. 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 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
<input checked="" type="checkbox"/>	DB - emrdb - emr_acti...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
<input checked="" type="checkbox"/>	DB - emrdb - emr_com...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
<input checked="" type="checkbox"/>	DB - emrdb - emr_com...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
<input checked="" type="checkbox"/>	DB - emrdb - emr_med...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
<input checked="" type="checkbox"/>	DB - emrdb - emr_med...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
<input checked="" type="checkbox"/>	DB - emrdb - emr_obs...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
<input checked="" type="checkbox"/>	DB - emrdb - emr_obs...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>
<input checked="" type="checkbox"/>	DB - emrdb - emr_pati...		Jun 18, 2020	Auto-Gen	<div><div></div><div></div></div>

Rows per page: 20

1-20 of 34

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

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

- 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

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

New Parameter

@parameterName

String

DESCRIPTION

-

DEFAULT VALUE

-

OK

Cancel

- Type a name for the new parameter in the **parameterName** field.
- 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.
- Type an optional description for the parameter in the **Description** field.
- 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:

@startDate Date ▾

DESCRIPTION

The start date to use when ingesting observation date values.

DEFAULT VALUE

1/1/2008

- 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:

@endDate Date ▾

DESCRIPTION

The end date to use when ingesting observation date values.

DEFAULT VALUE

1/1/2018

- 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:

PARAMETERS +

@startDate Date X

The start date to use when ingesting observation date values.

@endDate Date

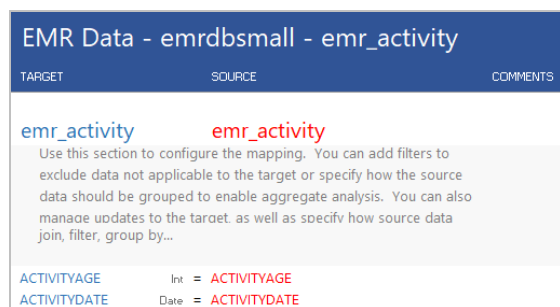
The end date to use when ingesting observation date values.

Add Filters to the Mapping to Apply the Criteria

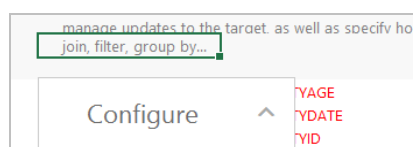
- Click the cell to the left of the target table name to open the menu. For example:


9				
10	☰	emr_activity	emr_activity	
14		ACTIVITYAGE	Int = ACTIVITYAGE	
15	+	ACTIVITYDATE	Date = ACTIVITYDATE	
16	⚙️	ACTIVITYID	Int = ACTIVITYID	
17	🗑️	ACTIVITYPROVIDERSPECIALTY	Int = ACTIVITYPROVIDERSPECIALTY	

- 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.



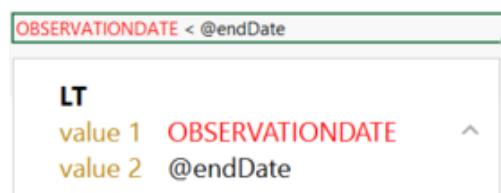
- In the Configure text box, start typing **filter**. When "filter" appears below the text box, click it to add the filter keyword to the cell.
- Click the cell in the Source column that corresponds to the filter you entered in the Target column. Anzo displays the **Source Mapping** text box below the cell. If necessary, click the drop-down arrow () next to the cell to open the Source Mapping text box.
- 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:




- 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 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](#).

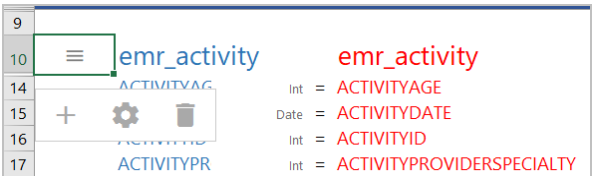
- [Configuring Groups, Filters, Joins, Updates, and Merges](#)
- [Using Functions to Transform Source Data](#)
- [Commonly Used Functions](#)

Configuring Groups, Filters, Joins, Updates, and Merges

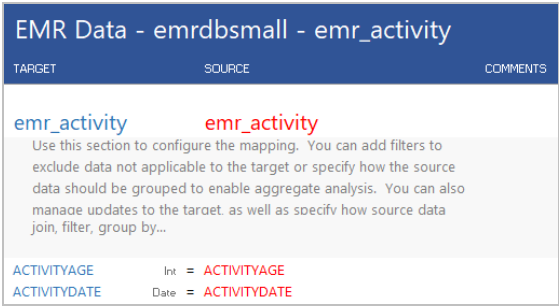
This section provides information about configuring groups, joins, filters, references, and merges at the mapping level so that they can be used by any functions that you use to transform the source data. The table below the steps describes each of the mapping level configuration options.

To implement a mapping level configuration:

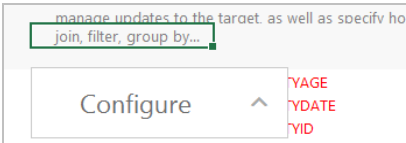
1. Click the cell to the left of the target table name to open the menu. For example:



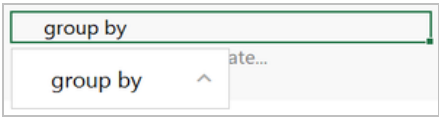
2. Click the Cog icon () in the menu to open the configuration section of the mapping.



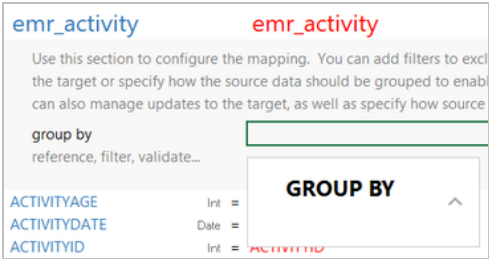
3. In the configuration section, click the cell that contains the **join, filter, group by...** text in the Target column. If necessary, click the drop-down arrow next to the cell to open the Configure text box.



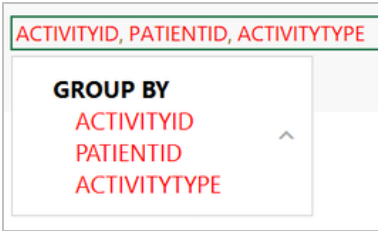
4. In the Configure text box, start typing the option that you want to configure. Anzo displays the options that match the text. Select an option to add it to the cell. For example:



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.
GROUP BY	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.

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.

```
inner join with MovieActors1 where MovieActors1.MovieID, condition, MovieActors2.MovieID

JOIN
join inner join
element MovieActors1
condition MovieActors1.MovieID
=
MovieActors2.MovieID
```

When including joins in mappings, do not create joins that result in multiple 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.

The screenshot shows the configuration page for a report named "NorthwindVirtualOrdersWithCustomerReport". The left sidebar contains navigation links: "REPORTS", "ORDERS", "EMPLOYEES", "CATEGORIES", and "CUSTOMERS". The main area has a title bar "OrdersWithCustomer" followed by a gear icon and the text "customers, orders, employees, categories". Below this is a descriptive paragraph about mapping filters. A code editor displays two SQL snippets. The first snippet defines "viewCustomerName" and "viewOrderId" using REPLACE_NULL and ISNULL functions. The second snippet defines "innerJoinWithCustomers" and "innerJoinWithEmployees" using JOIN clauses. On the right, there are tabs for "MAPPIING" (active) and "REVIEWS". Below the tabs is a section titled "OrdersWithCustomerView" containing a "SOURCE" tab and a "SOURCE RELATIONSHIPS" section.

NorthwindVirtualOrdersWithCustomerReport

REPORTS

OrdersWithCustomer customers, orders, employees, categories

Use this section to configure the mapping. You can add filters to exclude data not applicable to the target or specify how the source data should be grouped to enable aggregate analysis. You can also manage updates to the target, as well as specify how source data elements are joined.

```
join inner join with customers where customers.customerID = orders.customerID
join inner join with employees where employees.firstName = orders.customerID
```

reference, filter, validate...

```
String viewCustomerName = REPLACE_NULL(firstName, contactName)
String viewOrderId = orderID
```

MAPPING REVIEWS

Configure the schema and ontologies that will be used by this model.

TARGET + OrdersWithCustomerView

SOURCE + @

- customers
 - Northwind.CS\101019 - Auto
- orders
 - Northwind.CS\101019 - Auto
- employees
 - Northwind.CS\101019 - Auto
- categories
 - Northwind.CS\101019 - Auto

SOURCE RELATIONSHIPS

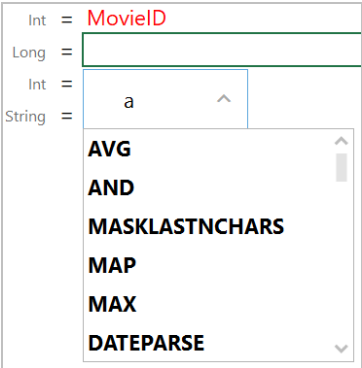
- categories ↔ customers
categoryID → customerID
- orders ↔ customers
customerID → customerID
- employees ↔ customers
customerID → customerID
- orders ↔ employees
customerID → firstName

Option & Arguments	Description
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	

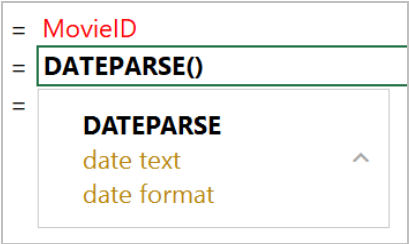
Using Functions to Transform Source Data

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:



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:



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 = DATEPARSE
         date text  MovieReleaseDate
         date format "dd-MM-yyyy"
```

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
IF	This function evaluates the condition in the test argument and assigns the value in value if true or value if false based on the results.
test	
value if true	<ul style="list-style-type: none">test: Use boolean columns or functions that return boolean: LE, LT, GE, GT, EQUALS, NOT_EQUAL, ISNULL, NOT, IN.
value if false	<ul style="list-style-type: none">value if true: The value to output if test returns true.value if false: The value to output if test returns false.
value if error	<ul style="list-style-type: none">value if error: Cambridge Semantics recommends that you leave this argument blank.

Function & Arguments	Description
DATEPARSE date text date format	<p>This function converts a string that contains a date value (date text) to the specified date format.</p> <ul style="list-style-type: none"> • date text: The property that contains the date value in string format. • date format: 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." <p>For example, the source mapping below converts the MovieReleaseDate values from strings to dates in the format "dd-MM-yyyy":</p> <p>DATEPARSE 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>
DATETIMEPARSE date text date format	<p>This function converts a string that contains a datetime value (date text) to the specified date format.</p> <ul style="list-style-type: none"> • date text: The property that contains the datetime value in string format. • date format: 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." <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>DATETIMEPARSE date text PATIENTLASTPMODATE date format "MM-dd-yyyy HH:mm:ss"</p>
UPPER value	<p>This function converts a string value to upper case letters.</p>
LOWER value	<p>This function converts a string value to lower case letters.</p>

Function & Arguments	Description
REPLACEIFNULL expression if null expression	<p>This function evaluates the expression. If the result is null, Anzo replaces the null with the value in if null expression.</p> <ul style="list-style-type: none"> • expression: The source column or expression to evaluate. • if null expression: 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. <p>For example, the source mapping below replaces any null values in the PATIENTID integer column with the integer 999:</p> <p>REPLACEIFNULL expression PATIENTID if null expression 999</p>
REPLACEIFNULLLOREEMPTY string expression if null or empty expression	<p>This function evaluates the string expression. If the result is null or empty (""), Anzo replaces the empty or null with the value in if null or empty expression.</p> <ul style="list-style-type: none"> • string expression: The source column or expression that evaluates to string. • if null or empty expression: The expression to replace null or empty values with. The resulting value must be a string. <p>For example, the source mapping below replaces any null or empty values in the GENDER column with "Not Specified":</p> <p>REPLACEIFNULLLOREEMPTY string expression GENDER if null or empty string expression "Not Specified"</p>
ISNULL expression	<p>This function evaluates the source column values in expression and returns "true" if the value is null and "false" if it is not null. You must choose a column in the expression argument; do not type a literal value or a function.</p>
SPLIT string delimiter	<p>This function splits a string value into multiple values based on the specified delimiter.</p> <ul style="list-style-type: none"> • string: The source column or function that evaluates to a string. • delimiter: The character to use to delimit the string.

Function & Arguments	Description
SPLITARRAY string delimiter index	<p>This function splits a string value into an array based on the delimiter. From the array, the function retrieves only the portion of the value that you specify in the index.</p> <ul style="list-style-type: none"> • string: The source column or function that evaluates to a string. • delimiter: The character to use to delimit the string. • index: 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. <p>For example, the following source mapping retrieves only the last four digits of social security numbers:</p> <p>SPLITARRAY string SSN delimiter "-" index 2</p>
IN value set to check	<p>This function checks whether a given value exists in a set of values (set to check). 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"> • value: The value to look for in the set. • set to check: The set of values to compare the value against. After typing a character, press Enter to submit the value, then press Ctrl + to add the next value. All items in the set must be the same data type. <p>For example, the following source mapping checks to see if PATIENTID falls in the set of 1, 100, 1000:</p> <p>IN value PATIENTID set to check 1 100 1000</p>

Function & Arguments	Description
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"> • expression: The list of columns that you want to map to the target. After adding a source column press Ctrl + to select the next column.
REGEX input regex replace	<p>This function finds all patterns in the input string that match the specified regular expression (regex). It replaces the input patterns with the value in replace and returns the resulting string.</p> <ul style="list-style-type: none"> • input: The source column or expression that evaluates to a string. • regex: The regular expression to use to find matches in the input string. For information about REGEX syntax, see the W3C Regular Expression Syntax specification. • replace: The string that should replace the input patterns that match regex. <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>REGEX</p> <pre>input COMPLAINTSTRING regex "PS" replace "-"</pre>
CONCATENATE text	<p>This function concatenates multiple string values (text) and returns a single string.</p> <ul style="list-style-type: none"> • text: The string values to concatenate, including any delimiters that you want to use. Press Ctrl + to enter multiple values. <p>For example, the source mapping below concatenates PATIENTHOMESTATE and PATIENTHOMEZIP:</p> <p>CONCATENATE</p> <pre>text PATIENTHOMESTATE PATIENTHOMEZIP</pre>

Function & Arguments	Description
EQUAL value1 value2	<p>This function compares numeric values and returns "true" if value1 is equal to value2 and "false" if the values are not equal ($\text{value1} = \text{value2}$).</p> <ul style="list-style-type: none"> • value1: The numeric value to compare to value2. • value2: The numeric value to compare to value1.
NOT_EQUAL value1 value2	<p>This function compares numeric values and returns "true" if value1 does not equal value2 and "false" if the values are equal ($\text{value1} \neq \text{value2}$).</p> <ul style="list-style-type: none"> • value1: The numeric value to compare to value2. • value2: The numeric value to compare to value1.
GE value1 value2	<p>This function compares numeric values and returns "true" if value1 is greater than or equal to value2 and "false" if value1 is less than value2 ($\text{value1} \geq \text{value2}$).</p> <ul style="list-style-type: none"> • value1: The numeric value to compare to value2. • value2: The numeric value to compare to value1.
GT value1 value2	<p>This function compares numeric values and returns "true" if value1 is greater than value2 and "false" if value1 is less than or equal to value2 ($\text{value1} > \text{value2}$).</p> <ul style="list-style-type: none"> • value1: The numeric value to compare to value2. • value2: The numeric value to compare to value1.
LE value1 value2	<p>This function compares numeric values and returns "true" if value1 is less than or equal to value2 and "false" if value1 is greater than value2 ($\text{value1} \leq \text{value2}$).</p> <ul style="list-style-type: none"> • value1: The numeric value to compare to value2. • value2: The numeric value to compare to value1.
LT value1 value2	<p>This function compares numeric values and returns "true" if value1 is less than value2 and "false" if value1 is greater than or equal to value2 ($\text{value1} < \text{value2}$).</p> <ul style="list-style-type: none"> • value1: The numeric value to compare to value2. • value2: The numeric value to compare to value1.

Function & Arguments	Description
AND logical1 logical2	<p>This logical function evaluates two or more logical statements (logical1, logical2) 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"> • logical1: 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. • logical2: The second logical condition to evaluate. This argument also needs to include a logical function that returns a boolean value.
OR logical1 logical2	<p>This logical function evaluates two or more logical statements (logical1, logical2) 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"> • logical1: 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. • logical2: The second logical condition to evaluate. This argument also needs to include a logical function that returns a boolean value.
NOT logical	<p>This logical function evaluates whether data does not meet the condition (logical) that you specify.</p> <ul style="list-style-type: none"> • logical: 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.
NUMERIC_ADD v1 v2	<p>This function adds the values of the numeric expressions that you specify (v1 + v2).</p> <ul style="list-style-type: none"> • v1: The first addend in the equation. • v2: The second addend in the equation.
DIVIDE v1 v2	<p>This function divides the values of the numeric expressions that you specify (v1/v2).</p> <ul style="list-style-type: none"> • v1: The divisor in the equation. • v2: The dividend in the equation.

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

Function & Arguments	Description
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> <div><pre>String = URI("http://csi.com/", ENCODE(eventname)) Date Time = starttime Int =</pre><div><div>URI</div><div>"http://csi.com/"</div><div>ENCODE</div><div>value eventname</div></div></div> <p>This example mapping results in triples such as:</p> <div><pre><ticket_events> <eventname> "http://csi.com/Rolling+Stones"</pre></div> <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> <div><div>FLIGHT_NUMBER uri LATE_AIRCRAFT_DELAY MONTH ORIGIN_AIRPORT SCHEDULED_ARRIVAL SCHEDULED_DEPARTURE</div><div><pre>Int = FLIGHT_NUMBER Expression = URI("http://csi.com/Flight", ENCODE(FLIGHT_NUMBER)) Int = Int = String = Int = Int =</pre><div><div>URI</div><div>"http://csi.com/Flight"</div><div>ENCODE</div><div>value FLIGHT_NUMBER</div></div></div></div> <p>The example URI specification results in triples such as:</p> <div><pre><http://csi.com/Flight1234> <FLIGHT_NUMBER> 1234 <http://csi.com/Flight1234> <ORIGIN_AIRPORT> "BOS"</pre></div>

Related Topics

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

Supported Mapping Functions

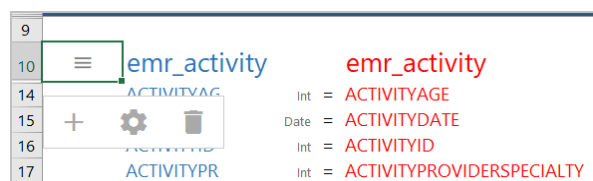
This topic describes the mapping functions that Anzo supports. For information about adding functions to mappings, see [Transforming Data in Mappings](#).

- [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:

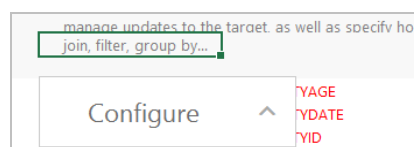
1. Click the cell to the left of the target table name to open the menu. For example:



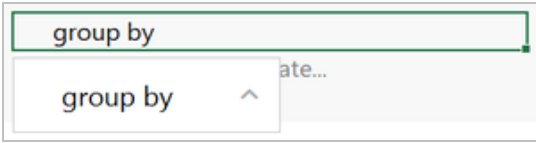
2. Click the Cog icon (⚙️) in the menu to open the configuration section of the mapping.



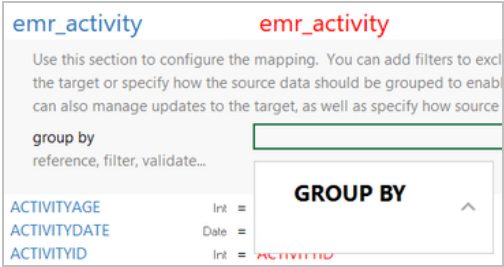
3. In the configuration section, click the cell that contains the **join, filter, group by...** text in the Target column. If necessary, click the drop-down arrow next to the cell to open the Configure text box.



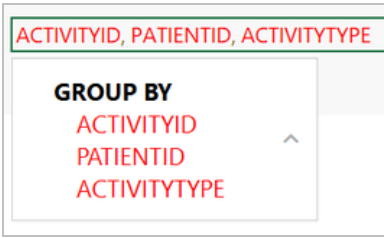
4. In the Configure text box, start typing "group by." Anzo completes the text and displays **group by** in the box. Click the up arrow to enter group by in the cell.



5. 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.



6. 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 number argument.</p> <ul style="list-style-type: none"> number: 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. <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>AVG number NUMBER_OF_BYTES</p>
COUNT value	<p>This function counts the number of instances for a grouped value. 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 value argument.</p>
MIN value	<p>This function calculates the minimum value for the group of numeric values that you specify in the value argument.</p>
SUM number	<p>This function calculates the sum of the group of numeric values that you specify in the number 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
EQUAL value1 value2	<p>This function compares numeric values and returns "true" if value1 is equal to value2 and "false" if the values are not equal (value1 = value2).</p> <ul style="list-style-type: none"> value1: The numeric value to compare to value2. value2: The numeric value to compare to value1.

Function & Arguments	Description
GE value1 value2	<p>This function compares numeric values and returns "true" if value1 is greater than or equal to value2 and "false" if value1 is less than value2 (value1 >= value2).</p> <ul style="list-style-type: none"> • value1: The numeric value to compare to value2. • value2: The numeric value to compare to value1.
GT value1 value2	<p>This function compares numeric values and returns "true" if value1 is greater than value2 and "false" if value1 is less than or equal to value2 (value1 > value2).</p> <ul style="list-style-type: none"> • value1: The numeric value to compare to value2. • value2: The numeric value to compare to value1.
IN value set to check	<p>This function checks whether a given value exists in a set of values (set to check). 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"> • value: The value to look for in the set. • set to check: The set of values to compare the value against. After typing a character, press Enter to submit the value, then press Ctrl + to add the next value. All items in the set must be the same data type. <p>For example, the following source mapping checks to see if PATIENTID falls in the set of 1, 100, 1000:</p> <pre> IN value PATIENTID set to check 1 100 1000 </pre>
ISNULL expression	<p>This function evaluates the source column values in expression and returns "true" if the value is null and "false" if it is not null. You must choose a column in the expression argument; do not type a literal value or a function.</p>
LE value1 value2	<p>This function compares numeric values and returns "true" if value1 is less than or equal to value2 and "false" if value1 is greater than value2 (value1 <= value2).</p> <ul style="list-style-type: none"> • value1: The numeric value to compare to value2. • value2: The numeric value to compare to value1.

Function & Arguments	Description
LT value1 value2	This function compares numeric values and returns "true" if value1 is less than value2 and "false" if value1 is greater than or equal to value2 (value1 < value2). <ul style="list-style-type: none"> • value1: The numeric value to compare to value2. • value2: The numeric value to compare to value1.
NOT_EQUAL value1 value2	This function compares numeric values and returns "true" if value1 does not equal value2 and "false" if the values are equal (value1 != value2). <ul style="list-style-type: none"> • value1: The numeric value to compare to value2. • value2: The numeric value to compare to value1.

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 logical1 logical2	This logical function evaluates two or more logical statements (logical1 , logical2) 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. <ul style="list-style-type: none"> • logical1: 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. • logical2: The second logical condition to evaluate. This argument also needs to include a logical function that returns a boolean value.
IF test value if true value if false value if error	This function evaluates the condition in the test argument and assigns the value in value if true or value if false based on the results. <ul style="list-style-type: none"> • test: Use boolean columns or functions that return boolean: LE, LT, GE, GT, EQUALS, NOT_EQUAL, ISNULL, NOT, IN. • value if true: The value to output if test returns true. • value if false: The value to output if test returns false. • value if error: Cambridge Semantics recommends that you leave this argument blank.

Function & Arguments	Description
OR logical1 logical2	<p>This logical function evaluates two or more logical statements (logical1, logical2) 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"> • logical1: 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. • logical2: The second logical condition to evaluate. This argument also needs to include a logical function that returns a boolean value.
NOT logical	<p>This logical function evaluates whether data does not meet the condition (logical) that you specify.</p> <ul style="list-style-type: none"> • logical: 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.
REPLACEIFNULL expression if null expression	<p>This function evaluates the expression. If the result is null, Anzo replaces the null with the value in if null expression.</p> <ul style="list-style-type: none"> • expression: The source column or expression to evaluate. • if null expression: 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. <p>For example, the source mapping below replaces any null values in the PATIENTID integer column with the integer 999:</p> <p>REPLACEIFNULL expression PATIENTID if null expression 999</p>

Function & Arguments	Description
REPLACEIFNULLOREMPY string expression if null or empty expression	<p>This function evaluates the string expression. If the result is null or empty (""), Anzo replaces the empty or null with the value in if null or empty expression.</p> <ul style="list-style-type: none"> • string expression: The source column or expression that evaluates to string. • if null or empty expression: The expression to replace null or empty values with. The resulting value must be a string. <p>For example, the source mapping below replaces any null or empty values in the GENDER column with "Not Specified":</p> <p>REPLACEIFNULLOREMPY 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
BOOLEANPARSE value	<p>This function converts a string (value) that contains "true" and "false" values to boolean format.</p> <div> <p>Note</p> <p>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 value contain the words "true" or "false."</p> </div>

Function & Arguments	Description
<div>DATEPARSE</div> <div>date text</div> <div>date format</div>	<div>This function converts a string that contains a date value (date text) to the specified date format.</div> <div><div>Note</div><div>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 date text contain a date.</div></div> <div><ul style="list-style-type: none">date text: The property that contains the date value in string format.date format: 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."</div> <div>For example, the source mapping below converts the MovieReleaseDate values from strings to dates in the format "dd-MM-yyyy":</div> <div><div>DATEPARSE</div><div>date text MovieReleaseDate</div><div>date format "dd-MM-yyyy"</div></div> <div>The format that you specify for dates is flexible. For example, typing the format "dd-MMM-yy" displays values such as "01-JAN-19."</div>

Function & Arguments	Description
DATETIMEPARSE date text date format	<p>This function converts a string that contains a datetime value (date text) to the specified date format.</p> <div> <p>Note</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 date text contain a datetime.</p> </div> <ul style="list-style-type: none"> • date text: The property that contains the datetime value in string format. • date format: 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." <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>DATETIMEPARSE date text PATIENTLASTPMODATE date format "MM-dd-yyyy HH:mm:ss"</p>
DECIMALPARSE value	<p>This function converts a string (value) that contains a decimal value to decimal format.</p> <div> <p>Note</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 value contain a decimal.</p> </div>

Function & Arguments	Description
DOUBLEPARSE value	<p>This function converts a string (value) that contains a double value to double format.</p> <div> Note 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 value contain a double. </div>
FLOATPARSE value	<p>This function converts a string (value) that contains float values to float format.</p> <div> Note 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 value contain floats. </div>
INTPARSE value	<p>This function converts a string (value) that contains integer values to integer format.</p> <div> Note 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 value contain integers. </div>
LONGPARSE value	<p>This function converts a string (value) that contains a long integer value (from -2,147,483,648 to 2,147,483,647) to long format.</p> <div> Note 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 value contain long data. </div>

Function & Arguments	Description
SHORTPARSE value	<p>This function converts a string (value) that contains a short integer value (from -32,678 to 32,767) to short format.</p> <div> <p>Note</p> <p>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 value contain short data.</p> </div>
TIMEPARSE time text time format	<p>This function converts a string that contains time text to a time value in the time format that you specify.</p> <div> <p>Note</p> <p>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 time text contain a time value.</p> <ul style="list-style-type: none"> • time text: The property that contains the time value in string format. • time format: 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." </div>
TOSTRING value format	<p>This function converts a value that is a double data type to string format.</p> <ul style="list-style-type: none"> • value: The double type values that you want to convert to string format. • format: The format code for the new string value. For example, "%.0f".
TRYPARSEBOOLEAN value if error	<p>This function attempts to convert a string value to a boolean data type. If an instance cannot be converted, Anzo replaces the string with the value in if error.</p> <ul style="list-style-type: none"> • value: The string value that contains "true" or "false" values. • if error: The boolean value to replace the string with if an error occurs with the conversion.

Function & Arguments	Description
TRYPARSEDATE value date format if error	<p>This function attempts to convert a string value to a date data type in the date format that you specify. If an instance cannot be converted, Anzo replaces the string with the value in if error.</p> <ul style="list-style-type: none"> • value: The string value that contains date data. • date format: 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." • if error: The date value to replace the string with if an error occurs with the conversion.
TRYPARSEDATETIME value date format if error	<p>This function attempts to convert a datetime string value 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 if error.</p> <ul style="list-style-type: none"> • value: The string value that contains datetime data. • date format: Anzo outputs values in SQL date format, yyyy-MM-dd. • if error: The date value to replace the string with if an error occurs with the conversion.
TRYPARSEDECIMAL value if error	<p>This function attempts to convert a string value to a decimal data type. If an instance cannot be converted, Anzo replaces the string with the value in if error.</p> <ul style="list-style-type: none"> • value: The string value that contains decimal data. • if error: The decimal value to replace the string with if an error occurs with the conversion.
TRYPARSEDOUBLE value if error	<p>This function attempts to convert a string value to a double data type. If an instance cannot be converted, Anzo replaces the string with the value in if error.</p> <ul style="list-style-type: none"> • value: The string value that contains double data. • if error: The double value to replace the string with if an error occurs with the conversion.

Function & Arguments	Description
TRYPARSEFLOAT value if error	<p>This function attempts to convert a string value to a float data type. If an instance cannot be converted, Anzo replaces the string with the value in if error.</p> <ul style="list-style-type: none">• value: The string value that contains float values.• if error: The float value to replace the string with if an error occurs with the conversion.
TRYPARSELONG value if error	<p>This function attempts to convert a string value to a long data type. If an instance cannot be converted, Anzo replaces the string with the value in if error.</p> <ul style="list-style-type: none">• value: The string value that contains long data (-2,147,483,648 to 2,147,483,647).• if error: The long value to replace the string with if an error occurs with the conversion.
TRYPARSESHORT value if error	<p>This function attempts to convert a string value to a short data type. If an instance cannot be converted, Anzo replaces the string with the value in if error.</p> <ul style="list-style-type: none">• value: The string value that contains short data (-32,768 to 32,767).• if error: The short value to replace the string with if an error occurs with the conversion.

Function & Arguments	Description
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> <div><div><div>String =</div><div>Date Time =</div><div>Int =</div></div><div><div>URI("http://csi.com/", ENCODE(eventname))</div><div>starttime</div><div></div></div><div><div>URI</div><div>"http://csi.com/"</div><div>ENCODE</div><div>value eventname</div></div></div> <p>This example mapping results in triples such as:</p> <div><ticket_events> <eventname> "http://csi.com/Rolling+Stones"</div> <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> <div><div>FLIGHT_NUMBER</div><div>uri</div><div>LATE_AIRCRAFT_DELAY</div><div>MONTH</div><div>ORIGIN_AIRPORT</div><div>SCHEDULED_ARRIVAL</div><div>SCHEDULED_DEPARTURE</div></div> <div><div>Int =</div><div>Expression =</div><div>Int =</div><div>Int =</div><div>String =</div><div>Int =</div><div>Int =</div></div> <div><div>FLIGHT_NUMBER</div><div>URI("http://csi.com/Flight", ENCODE(FLIGHT_NUMBER))</div><div></div><div>URI</div><div>"http://csi.com/Flight"</div><div>ENCODE</div><div>value FLIGHT_NUMBER</div></div> <p>The example URI specification results in triples such as:</p> <div><http://csi.com/Flight1234> <FLIGHT_NUMBER> 1234 <http://csi.com/Flight1234> <ORIGIN_AIRPORT> "BOS"</div>

Lookup and Mapping Functions

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

Function & Arguments	Description
LOOKUP from get fields values	<p>This function enables you to look up values in a supplemental table. LOOKUP joins the from lookup table to a source table on the columns specified in the fields and values arguments.</p> <p>The function returns the value from get in the lookup table that corresponds to each row's value in the values argument.</p> <ul style="list-style-type: none"> • from: The lookup table to perform the join against. • get: The field or property to retrieve from the lookup table in the from argument. • fields: The field or fields from the lookup table to compare with the values from the primary table. • values: The values from the primary table to compare with the fields from the lookup table.
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"> • expression: The list of columns that you want to map to the target. After adding a source column press Ctrl + to select the next column.
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 value as a key in the map and returns the value associated with the key.</p> <ul style="list-style-type: none"> • value: The key or keys to use to look up the value from the map. • map: The map to use to look up the values. You can click the map argument name to open the Edit Map dialog box and define or change a map.

Numeric Functions

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

Function & Arguments	Description
CEILING value	<p>This function rounds the value up to the next whole number if the value has a fractional part.</p> <ul style="list-style-type: none"> • value: The source values that you want to round up to the next whole number.

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

String Functions

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

Function & Arguments	Description
CONCATENATE text	<p>This function concatenates multiple string values (text) and returns a single string.</p> <ul style="list-style-type: none"> text: The string values to concatenate, including any delimiters that you want to use. Press Ctrl + to enter multiple values. <p>For example, the source mapping below concatenates PATIENTHOMESTATE and PATIENTHOMEZIP:</p> <p>CONCATENATE text PATIENTHOMESTATE PATIENTHOMEZIP</p>
DATEPARSE date text date format	<p>This function converts a string that contains a date value (date text) to the specified date format.</p> <ul style="list-style-type: none"> date text: The property that contains the date value in string format. date format: 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." <p>For example, the source mapping below converts the MovieReleaseDate values from strings to dates in the format "dd-MM-yyyy":</p> <p>DATEPARSE 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>

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

Function & Arguments	Description
RIGHT text num chars	This function starts on the right side of a text string, keeps the number of characters in num chars , and returns the truncated string.
SPLIT string delimiter	This function splits a string value into multiple values based on the specified delimiter . <ul style="list-style-type: none"> • string: The source column or function that evaluates to a string. • delimiter: The character to use to delimit the string.
SPLITARRAY string delimiter index	<p>This function splits a string value into an array based on the delimiter. From the array, the function retrieves only the portion of the value that you specify in the index.</p> <ul style="list-style-type: none"> • string: The source column or function that evaluates to a string. • delimiter: The character to use to delimit the string. • index: 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. <p>For example, the following source mapping retrieves only the last four digits of social security numbers:</p> <p>SPLITARRAY</p> <pre>string SSN delimiter "-" index 2</pre>
STRLEN term	This function returns the number of characters in the specified text string (term).
UPPER value	This function converts a string value 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](#)
- [Incremental Pipeline Reference](#)







Related Topics

[Managing Dataset Editions](#)

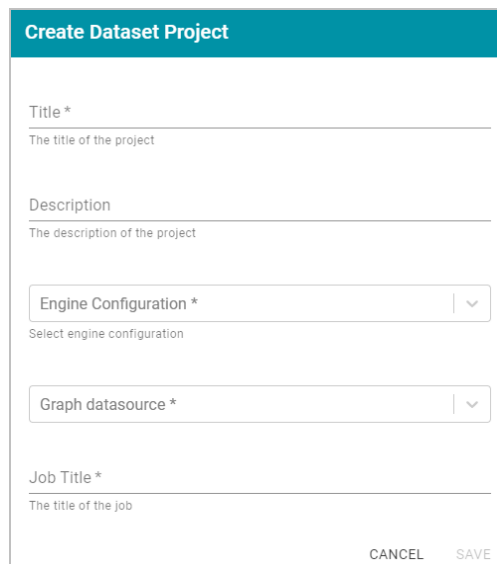
Creating a Dataset Pipeline

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					
<input type="text" value="Search"/>		Sort By: Title ▼	View:  		Add Pipeline ▼			
<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 **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

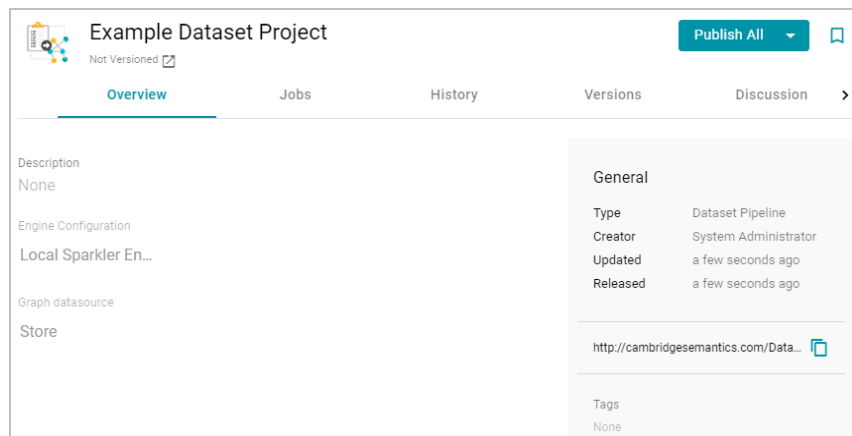
Engine Configuration *
Select engine configuration

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

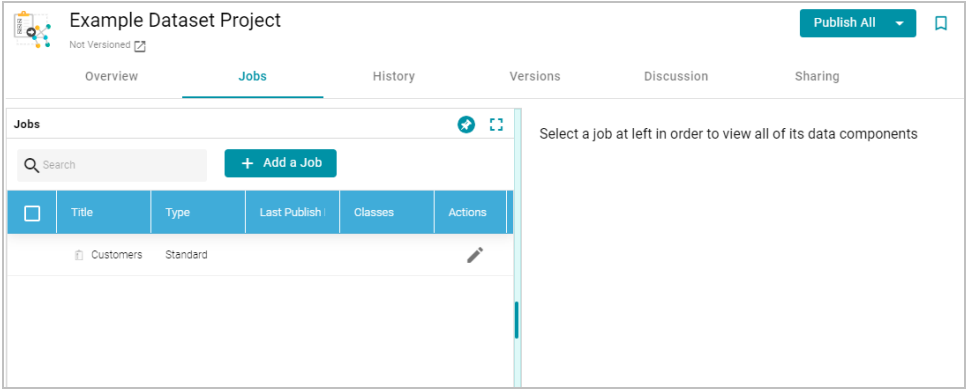
General

Type	Dataset Pipeline
Creator	System Administrator
Updated	a few seconds ago
Released	a few seconds ago

<http://cambridgesemantics.com/Data...>

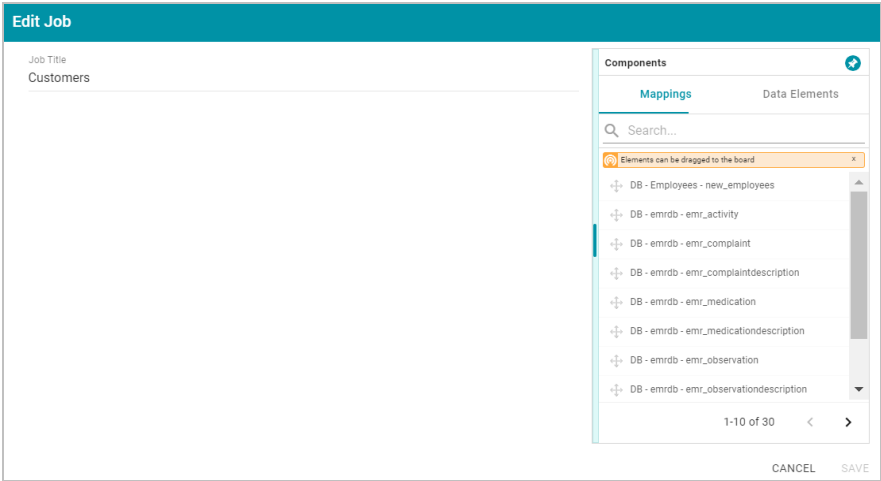
Tags
None

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:

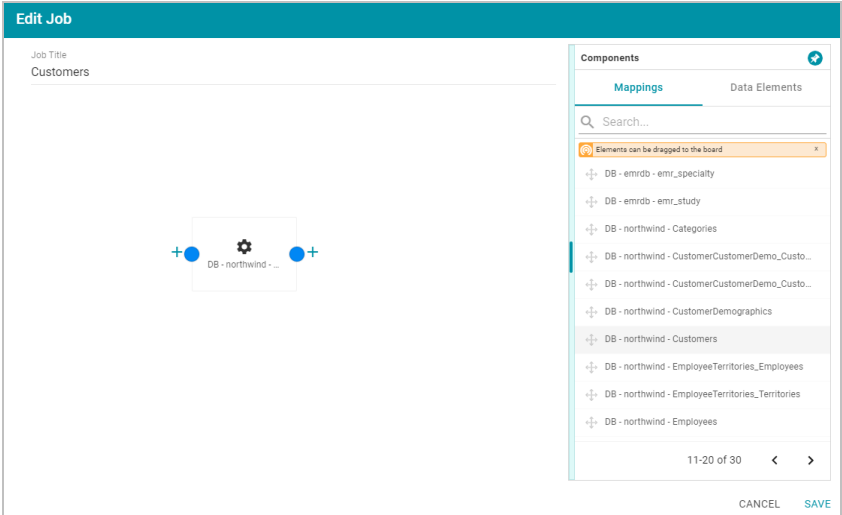


The Jobs pane on the left side of the screen lists the jobs in the project. Since this is a new project, the right side of the screen remains blank. Once this project is published, you can select a job to view the history of data components that the job has created.

- 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:

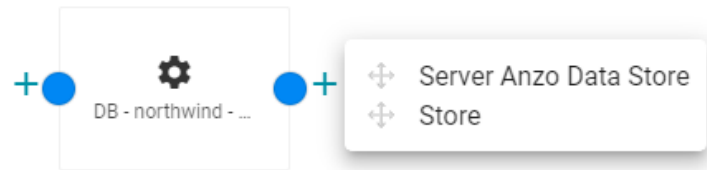


- 10. To configure the job, drag and drop a data source element from the Mappings or Data Elements tab in the Components list onto main part of the screen. For example, in the image below, the DB-northwind-Customers mapping is added to the job canvas:

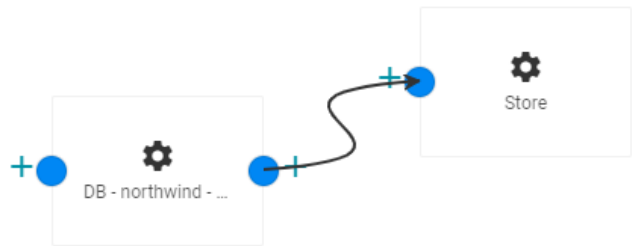


Tip
If you drag a mapping onto the job canvas and there is only one source and one target for the mapping, Anzo automatically adds that source and target to the job.

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:

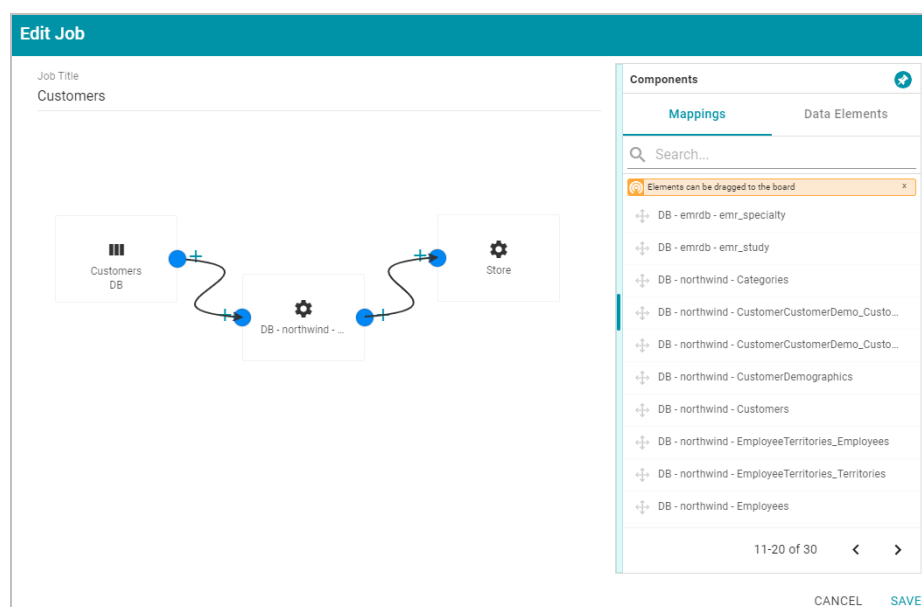


To add one of the options to the job, drag the option from the tooltip onto the canvas. For example:



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**.

	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 New Graphmart](#) for instructions.

Related Topics

[Blending Data](#)

[Accessing and Analyzing Data](#)

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

Q

Search

Sort By:

Title

View:

Add Pipeline

<div></div>	Title	Type	Version	Creator	Created Date	Updated Date	Tags	
<div></div>	<div>Load DB - northwind to Store</div>	Dataset Pipel...	Derived from: Untitled Version	System Admi...	May 31, 2021	May 31, 2021	Auto-Gen	
<div></div>	<div>Load Movies to Store</div>	Dataset Pipel...	Derived from: Untitled Version	System Admi...	Jun 7, 2021	Jun 7, 2021	Auto-Gen	
<div></div>	<div>Load Sample Movie Data to Store</div>	Dataset Pipel...	Derived from: Untitled Version	System Admi...	Jun 2, 2021	Jun 2, 2021	Auto-Gen	
<div></div>	<div>Load Tickets to Store</div>	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:

Load DB northwind
 Not Versioned

Publish All

Overview

Jobs

History

Versions

Discussion

Sharing

Description
None

Engine Configuration
Local Sparkler En...

Graph datasource
Store

Dataset
DB northwind
<http://csi.com/FileBasedLinkedDataSet/387f6d29e96d5a2016f2e6f58c3e4b09>

Managed Editions

Title	Description	Most Recent Published Date	Actions
Default Edition	Contains the latest successfully p...	10/26/2020 01:00PM	

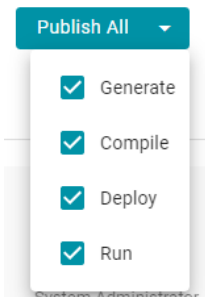
General

Type	Dataset Pipeline
Creator	System Administrator
Updated	21 hours ago
Released	21 hours ago

<http://cambridgesemantics.com/Proje...>

Tags
Auto-Gen

3. 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:



4. 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:

Overview					
Jobs					
History					
Versions					
Discussion					
Sharing					
Jobs					
Search					
+ Add a Job					
<input type="checkbox"/>	Title	Type	Last Publish Date	Class	Actions
<input checked="" 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 checked="" type="checkbox"/>	Load Order Deta...	Standard	10/26/2020 12:58PM	Order Details	
<input checked="" type="checkbox"/>	Load Customer...	Standard	10/26/2020 12:59PM	CustomerDemograp...	
<input checked="" type="checkbox"/>	Load Suppliers	Standard	10/26/2020 12:57PM	Suppliers	
<input checked="" type="checkbox"/>	Load Employees	Standard	10/26/2020 12:55PM	Employees	
Rows per page: 20 1-15 of 15 < >					

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:

Jobs

Search

+ Add a Job

Selected: Load Customers Load Region

DeletePublish

	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. See [Blending Data](#) for next steps.

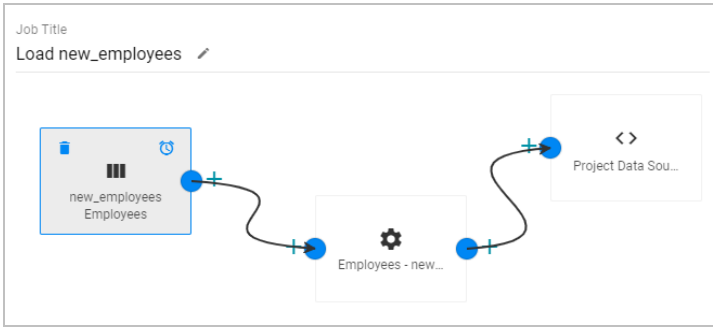
Related Topics

[Managing Dataset Editions](#)

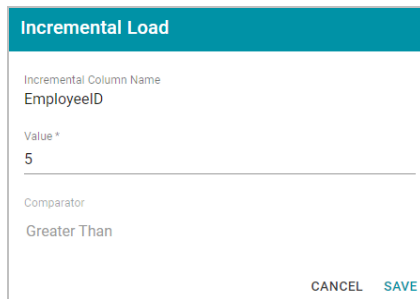
[Blending Data](#)

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:



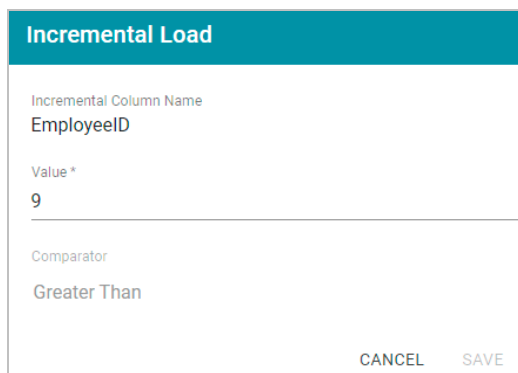
Clicking the clock icon opens the Incremental Load dialog box, which lists the Incremental Column Name, Value, and Comparator from the schema query. For example:



The dialog box is titled "Incremental Load". It contains three input fields: "Incremental Column Name" with the value "EmployeeID", "Value *" with the value "5", and "Comparator" with the value "Greater Than". At the bottom right, there are two buttons: "CANCEL" and "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:



The dialog box is titled "Incremental Load". It contains three input fields: "Incremental Column Name" with the value "EmployeeID", "Value *" with the value "9", and "Comparator" with the value "Greater Than". At the bottom right, there are two buttons: "CANCEL" and "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

Clear

Add Statements

Find

Results (1)

☒ Show native

☒ Subject

☒ Predicate

☒ Object

☐ Named Graph

EDIT

DELETE

<input type="checkbox"/>	Subject ↓	Predicate	Object
<input type="checkbox"/>	« http://cambridgesemantics.com/Project/0a5a4d73-8606-5c6c-6ac1-9c303768c7e1/b7b4f057-e364-fa72-dcd6-324002a671e3/fdb0c388-de01-d545-017c-df9ccf00eb72/Job/1c570a9c-00e4-8020-7b1c-7669d072f670/0 »	« http://cambridgesemantics.com/ontologies/2015/08/SDIService#rowsProcessed »	« 4 » <div></div>

Rows per page: 50

<

>

Related Topics

[Creating an Incremental Schema](#)

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

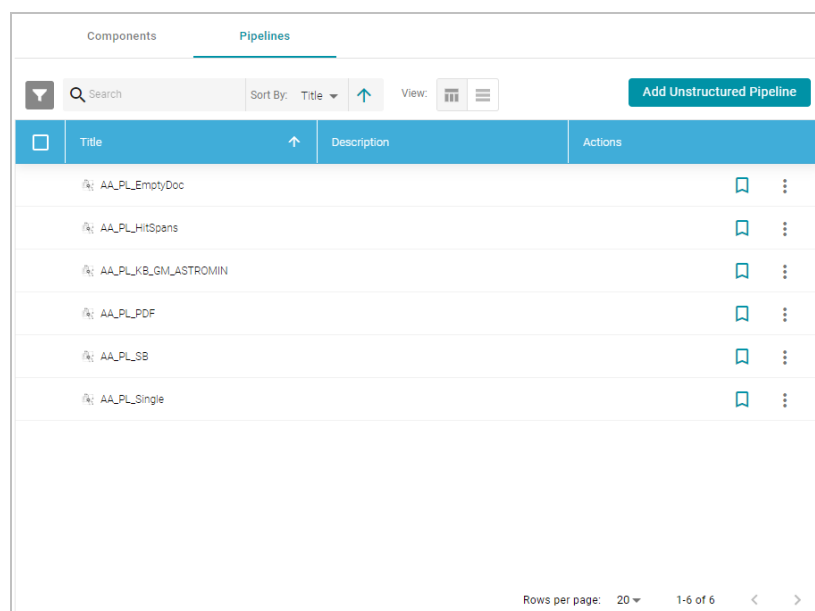
The topics in this section provide instructions for creating pipelines and onboarding unstructured data. For information about setting up the Anzo Unstructured environment, see [Deploying a Static Anzo Unstructured Cluster](#).

- [Creating an Unstructured Pipeline](#)
- [Running an Unstructured Pipeline](#)

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 Pipeline screen, which lists any existing unstructured pipelines. For example:



The screenshot shows the 'Pipelines' tab in the Anzo application. At the top, there is a search bar, a 'Sort By: Title' dropdown, and a 'View' toggle. A blue button labeled 'Add Unstructured Pipeline' is in the top right. Below is a table with columns: Title, Description, and Actions. The table lists six pipelines, each with a blue bookmark icon and a vertical ellipsis in the Actions column.

Title	Description	Actions
AA_PL_EmptyDoc		
AA_PL_HitSpans		
AA_PL_KB_GM_ASTROMIN		
AA_PL_PDF		
AA_PL_SS		
AA_PL_Single		

At the bottom right, it says 'Rows per page: 20' and '1-6 of 6'.

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

3. 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.

4. Type an optional description for the pipeline in the **Description** field.
5. If necessary, click the **Target Anzo Data Store** field and select the Data Store for this Pipeline. For instructions on creating an Anzo Data Store, see [Creating an Anzo Data Store](#).
6. 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.
7. 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. For instructions on creating a static Elasticsearch connection, see [Connecting to Elasticsearch](#).
8. Click **Save** to create the Pipeline. Anzo displays the pipeline Overview screen. For example:

PL_PDF Run Pipeline

Derived from: Untitled Version 1.0.1

Overview Crawlers Annotators History Progress

Description: None

Target Anzo Data Store: global__store

☐ Deploy Unstructured Infrastructure Dynamically

Static Elastic Search Config: ES_10.115.3.25

+ Advanced

General

Type: Unstructured Pipeline

Creator: uid=aa5,ou=users,dc=10,dc=

Updated: 7 hours ago

Released: Jan 16, 2020 8:52 AM

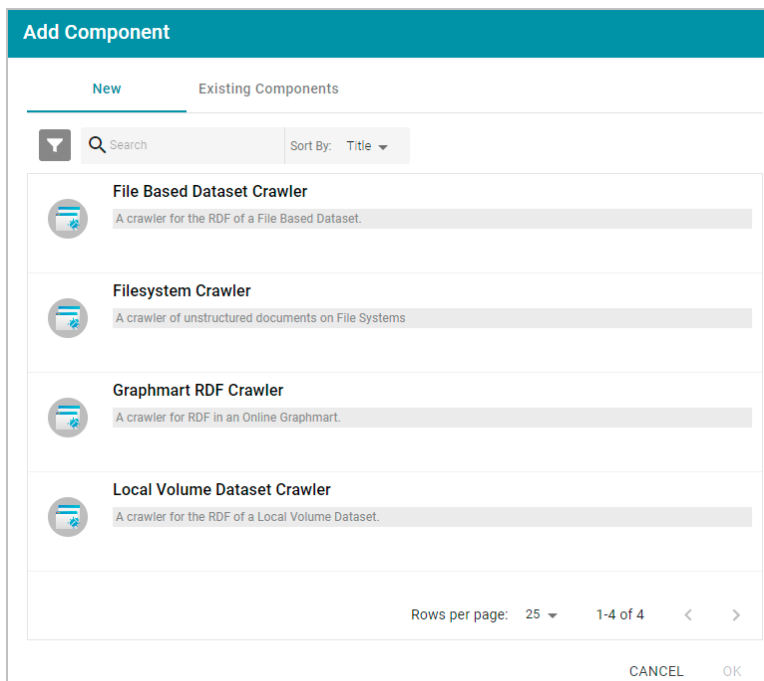
http://cambridgesemantics.com/Distribute...

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.
10. Click the **Crawlers** tab and follow the substeps below to add a Crawler to the Pipeline:
 - a. Click **Add Input**. Anzo opens the Add Component dialog box. The **New** tab lists the default crawlers and 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 to the pipeline, 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:

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 linked content

Content Property

Property(s) to match on for text content

Base Path Connection *

[BROWSE](#)

VFS location base directory for linked documents. For literal content/documents, this path is used to temporarily store their content to be crawled.

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 backing 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

VFS Crawl Location *

A VFS location to crawl

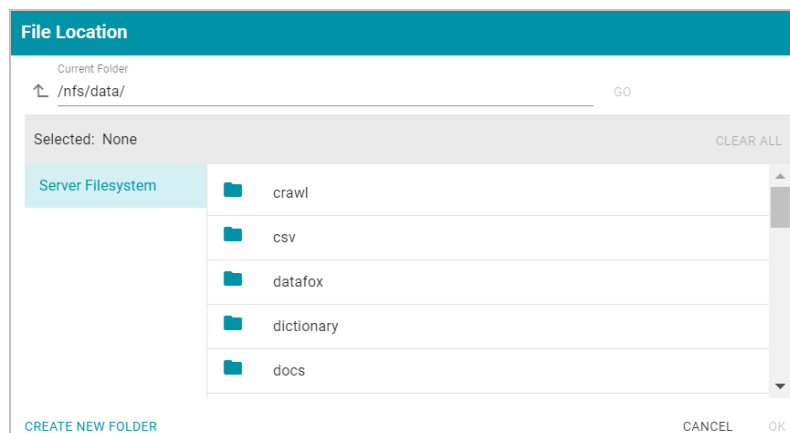
BROWSE

☒ 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.
- **VFS Crawl Location:** Required field that specifies the virtual 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

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 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 Data 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 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:

- **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 backing 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:

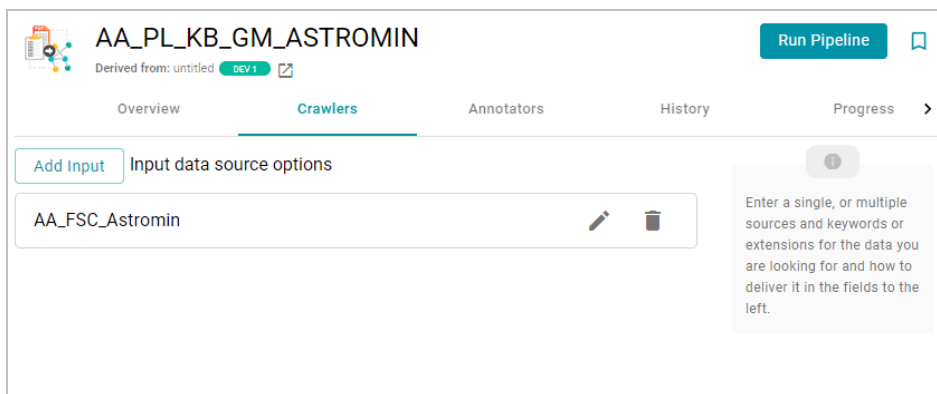
- If a **Link Property** was specified, the Base Path Connection is the base path to use for resolving relative file paths in the Link Property values. For example, using the example triples:

```
<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.

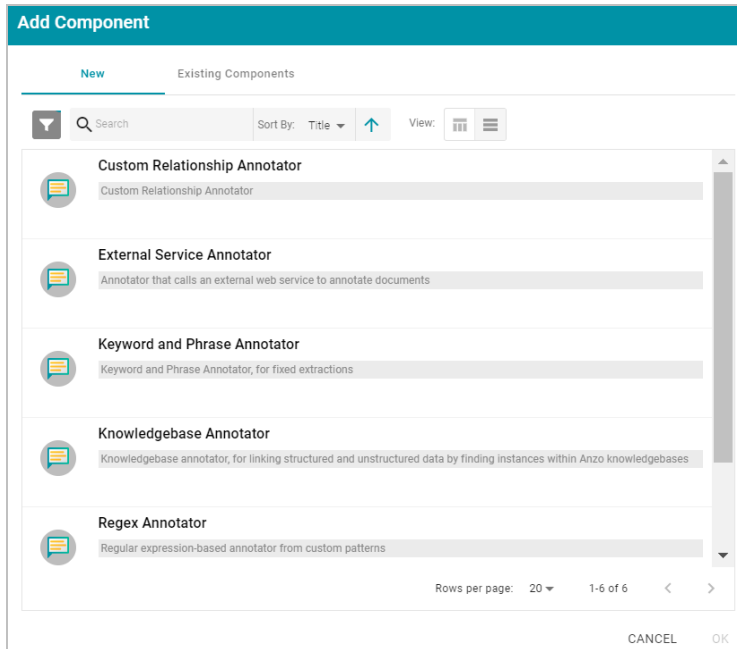
- If a **Content Property** was specified, the Base Path Connection is a directory on the File Store where the Crawler can save a copy of the Content Property strings for the Anzo Unstructured Worker instances. Saving the content to a shared file location avoids the overhead of sending the strings to 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.



In the Add Component dialog box, the **New** tab lists the default 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.
 - **Semantria Annotator:** Include this annotator to use the Semantria web service to find entities, sentiment, and topics in documents. It requires an Semantria API access key from [Lexalytics](#).

- **Significant Phrases Annotator:** Include this annotator to annotate statistically significant words and phrases.
- 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:

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 Class Path *
The entity Class path for entities returned in the service response

Entity Name Path *
The entity Name 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 Class Path:** Required field that specifies the class URI for an annotation.
- **Entity Name Path:** Required field that specifies the annotation object name path.

Knowledgebase Annotator

This section describes the settings that are available on the Create Knowledgebase Annotator screen:

Create Knowledgebase Annotator

Title *

Description

Backing Layer

A backing layer

Backing Graphmart

A backing graphmart

Backing Ontology *

The backing ontology

Term Class

The owl:Class of the knowledge base terms

Term Label Property

Primary name or label property of the resources

Term Identifying Properties

Properties identifying the resources, i.e. name, alias, and any other identifying properties

CANCEL SAVE

- **Title:** Required field that specifies the unique name for this annotator.
- **Description:** Optional field that provides a description of this annotator.
- **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 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 Ontology:** Required field that specifies the model for the backing data layers and/or graph-mart. Click the field and select a model from the drop-down list.
- **Term Class:** Required field that specifies the class of data for the annotation.
- **Term Label Property:** Required field that lists the property for which to find entities.
- **Term Identifying Properties:** Required field that specifies the properties that contain names, aliases, or other identifiers by which you want to find entities.

Regex Annotator

This section describes the settings that are available on the Create Regex Annotator screen:

- **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 the lists the regular expression rules for this annotator. To add a rule, click the plus icon (+) in the field. Anzo opens the Create Regular Expression Rule dialog box where you can define the rule:

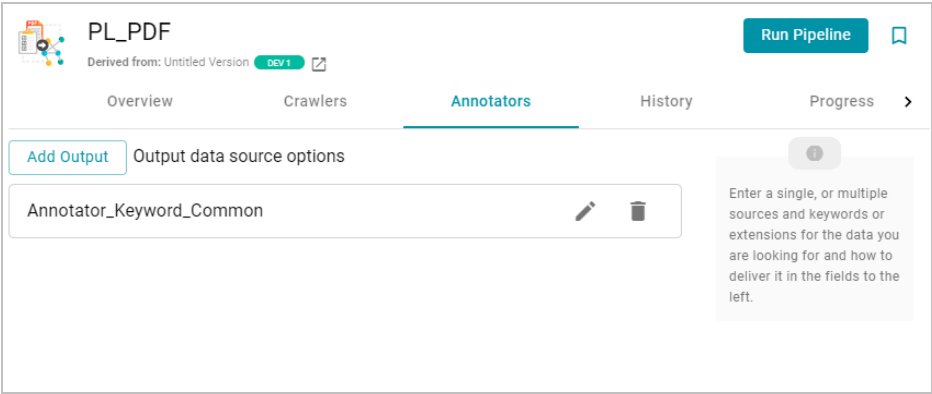
- **Title:** Required field that specifies the name of the rule.
- **Description:** Optional field that describes the rule.
- **Regular Expression:** Required field that specifies the regular expression to use for finding matching entities.

- **Class Structure:** Required field that specifies the class structure for the entities in the format `group_number:class_name`. For example, `0:person,1:Company`.

Tip

For information about the options that are presented when you edit a Regex Annotator, refer to the **Field Summary** section in the [Java Regex Compiler](#) documentation.

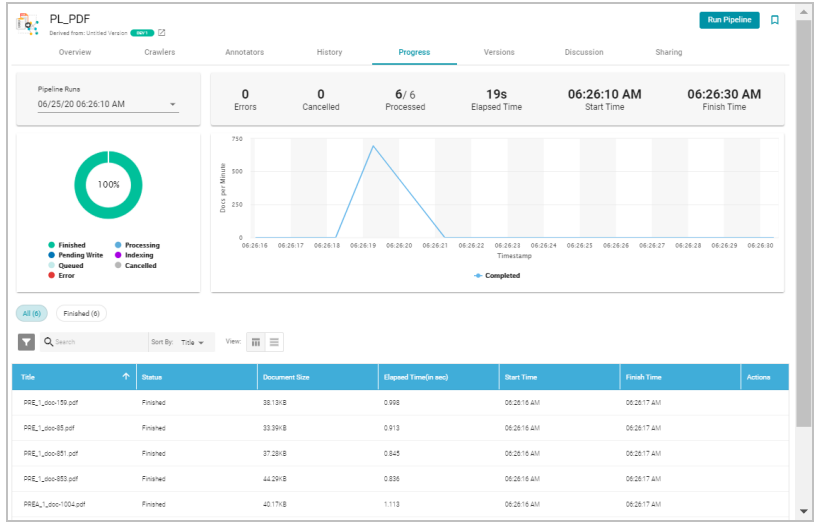
- d. When you have finished configuring the annotator, click **Save**. Anzo adds the annotator to the pipeline and returns to the Annotators screen. For example:



- e. If you want to change the annotator configuration, click the Edit icon (✎) for the annotator and modify the settings as needed. 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, 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 Dataset catalog. From the catalog, you can generate graph Data Profiles and create Graphmarts. See [Blending Data](#) for next steps.

Related Topics

[Running an Unstructured Pipeline](#)

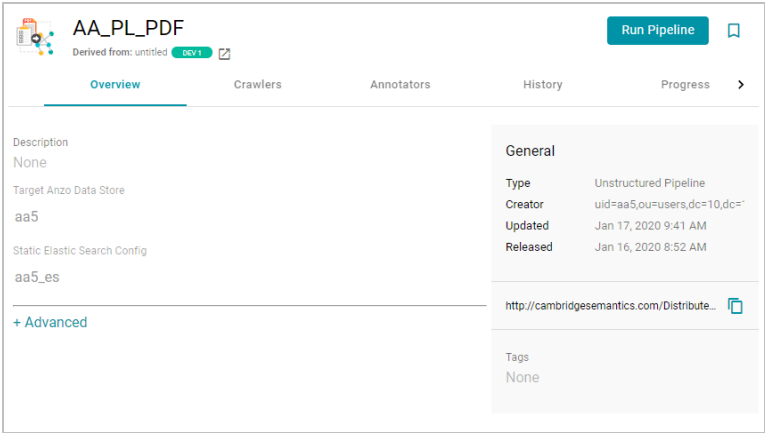
Running an Unstructured Pipeline

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 Pipeline screen, which lists any existing unstructured pipelines. For example:

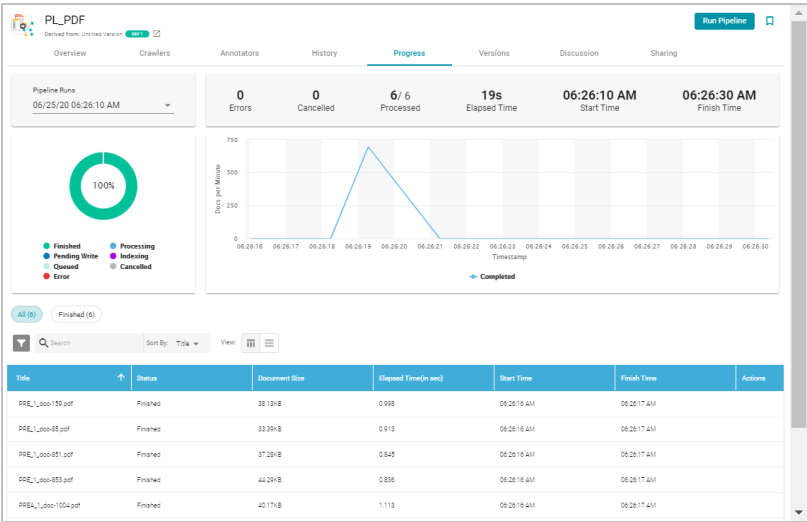
Components		Pipelines	
<div><div></div><div>Search</div></div>		Sort By: Title	View: <div><div></div><div></div></div>
		<div>Add Unstructured Pipeline</div>	
	Title	Description	Actions
	AA_PL_EmptyDoc		<div></div> <div></div>
	AA_PL_Hitspens		<div></div> <div></div>
	AA_PL_KB_GM_ASTROMIN		<div></div> <div></div>
	AA_PL_PDF		<div></div> <div></div>
	AA_PL_SS		<div></div> <div></div>
	AA_PL_Single		<div></div> <div></div>
Rows per page: 20 1-6 of 6			

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



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:



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 Dataset catalog. From the catalog, you can generate graph Data Profiles and create Graphmarts. See [Blending Data](#) for next steps.

Related Topics

[Creating an Unstructured Pipeline](#)

Modeling Data

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.

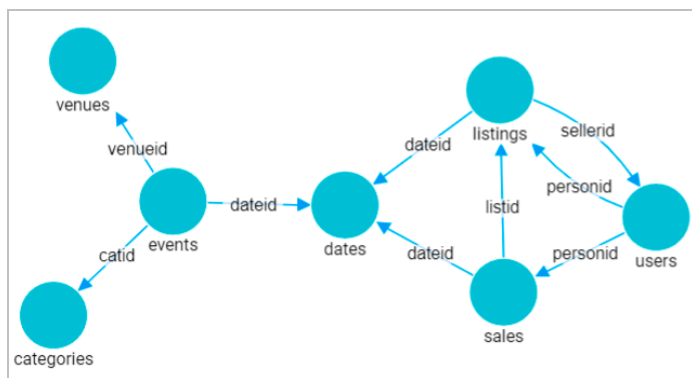
Users can import existing or external models (OWL ontologies) to Anzo, and models can be shared and reused. The topics in this section provide introductory information about data model terminology, describe model requirements and recommendations, and include instructions for creating and editing models.

- [Introduction to Models](#)
- [Model Requirements and Recommendations](#)
- [Uploading a Model to Anzo](#)
- [Creating a Model](#)
- [Editing a Model](#)
- [Setting Class Instance URI Patterns](#)
- [Downloading a Model](#)

Introduction to Models

This topic provides a brief introduction to data models and defines the terminology that is used in Anzo.

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.



Class

Models are made up of classes. Classes describe a concept or a group of related objects. For example, the model above contains events, dates, categories, sales, users, and listings classes.



Property

Properties are attributes that describe the data in a class. For example the users class has properties such as firstName, lastName, and personID. The events class has properties such as eventName, dateID, and startTime.

Anzo uses two kinds of properties:

- **Data property:** Relates a class to a simple value. For example, in the users class, the firstName and lastName properties relate to simple values.
- **Object property:** Relates a class to another class. For example, the listID property relates to the sales and listing classes.

Property Type

The specific type that can be used as the value of a property. Also known as "range."

Instance

Instances are concrete occurrences of a class. For example, an event's name is an instance of the events class.

Simple value

Also known as literals. For example:

- Numbers (for example, 15, -9, 10.35)
- Text strings (for example, "Jane Doe" or "a long description")
- Dates and times (for example, "13-Dec-2008", or "April, 2017")
- Boolean (true or false)

Type

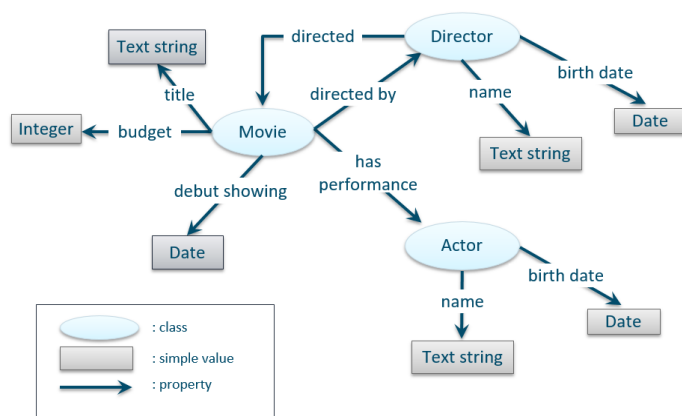
Either a class or a simple value.

Example: A Film Ontology

The example below shows classes, properties, and instances in a worksheet.

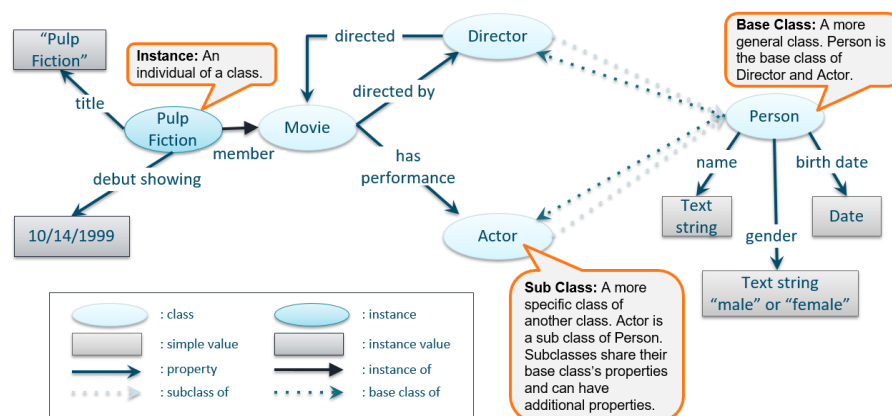
	A	B	C	D	E
1			MOVIES		
2	Title	Genre	Director1	Budget	First Ever Showing
114	Persepolis	Animation	Mariane Satrapi		
115	Pulp Fiction	Black comedy	Quentin Tarantino		10/14/1994 10:00 PM
116	Quiz Show	Drama	Robert Redford		9/16/1994 8:00 PM
117	Raging Bull	Biographical		18000000	11/14/1980 9:00 PM
118	Raiders of the Lost Ark	Action		20000000	6/12/1981 10:00 PM

In a model, you can define relationships between the properties, instances, and classes.



Instances, Subclasses, and Base Classes

In the example below, "Pulp Fiction" is an instance of the Movie class. Person is the more general class, or base class, for Actor and Director. And Actor and Director are subclasses of Person.



Related Topics

[Model Requirements and Recommendations](#)

[Uploading a Model to Anzo](#)

[Creating a Model](#)

[Editing a Model](#)

[Downloading a Model](#)

Model Requirements and Recommendations

Anzo uses models to describe and manage RDF data sets. 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.
- **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 Models](#)

[Uploading a Model to Anzo](#)

[Creating a Model](#)

[Editing a Model](#)

[Downloading a Model](#)

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](#).

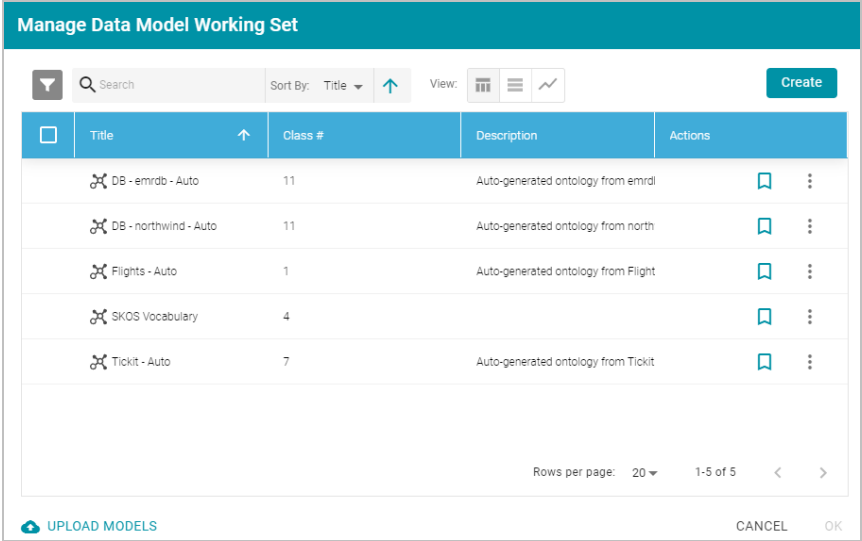
If you want to import a version of a Model that was exported from Anzo (as described in [Exporting an Artifact](#)), follow the instructions in [Importing Exported Versions of Artifacts](#) to import the model.

Note

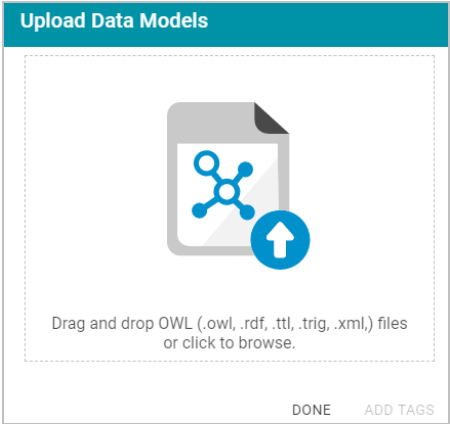
One of the following outcomes will occur if two users upload the same data model:

- 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.

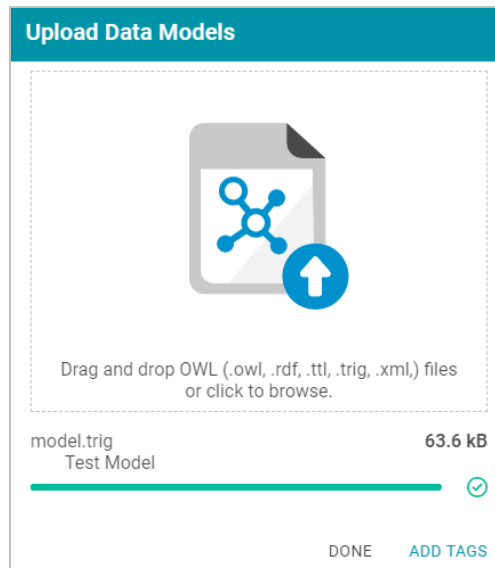
1. In the Anzo application, click **Model**. Anzo displays the Manage Data Model Working Set screen. For example:



2. On the bottom left corner of the screen, click **Upload Models**. The Upload Data Models dialog box opens.



3. 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

[Introduction to Models](#)

[Model Requirements and Recommendations](#)

[Creating a Model](#)

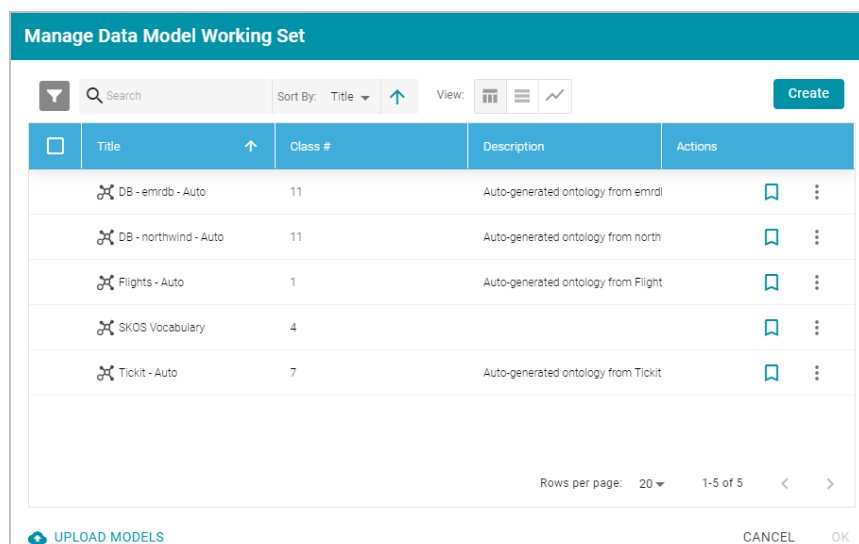
[Editing a Model](#)

[Downloading a Model](#)

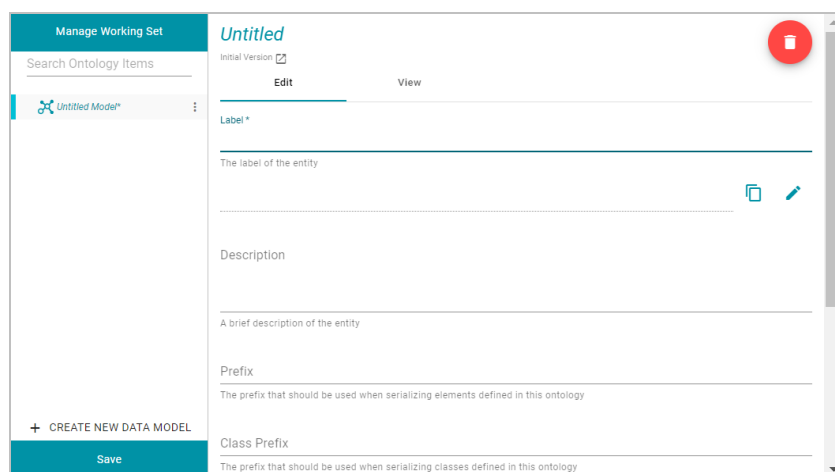
Creating a Model

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:



2. Click the **Create** button on the top right of the screen. Anzo displays the Model editor.



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**.

When Class Prefix is not set, Anzo generates the model's class URIs in the following format:
http://cambridgesemantics.com/ontologies/<model_label>#<class_label>.

- **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**.

When Property Prefix is not set, Anzo generates the model's property URIs in the following format:
http://cambridgesemantics.com/ontologies/<model_label>#p_<property_label>.

- **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](#). You can change or create a mapping to associate the new model with a data set. For information, see [Working with Mappings](#).

Related Topics

[Introduction to Models](#)

[Model Requirements and Recommendations](#)

[Uploading a Model to Anzo](#)

[Editing a Model](#)

[Downloading 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

When editing a data model, follow the requirements and guidelines defined in [Model Requirements and Recommendations](#).

- [Opening Models in the Editor](#)
- [Changing Model Components](#)
- [Class Editor Reference](#)
- [Property Editor Reference](#)

Tip

Before editing a data 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

Sort By: Title

View:

Create

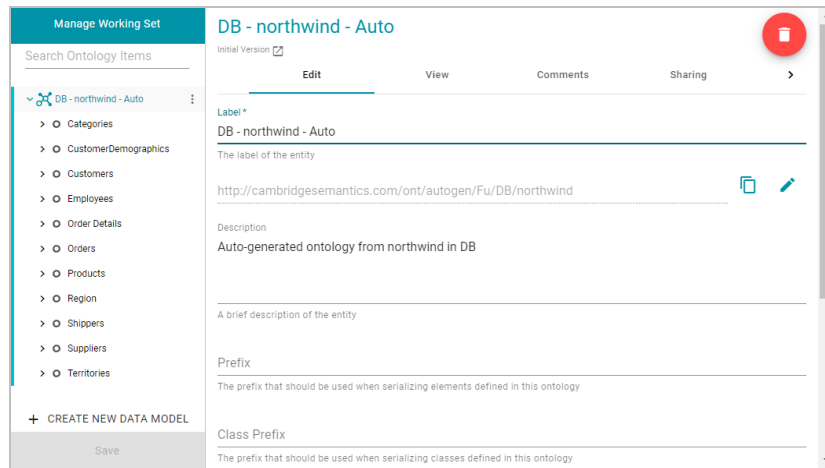
<input type="checkbox"/>	Title	Class #	Description	Actions
<input type="checkbox"/>	DB - emrdb - Auto	11	Auto-generated ontology from emrdb	
<input type="checkbox"/>	DB - northwind - Auto	11	Auto-generated ontology from north	
<input type="checkbox"/>	Flights - Auto	1	Auto-generated ontology from Flight	
<input type="checkbox"/>	SKOS Vocabulary	4		
<input type="checkbox"/>	Tickit - Auto	7	Auto-generated ontology from Tickit	

Rows per page: 20 1-5 of 5

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:



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`.

When Class Prefix is not set, Anzo generates the model's class URIs in the following format:

`http://cambridgesemantics.com/ontologies/<model_label>#<class_label>`.

- **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`.

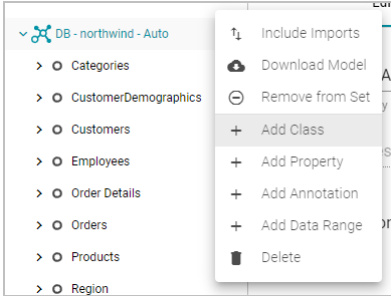
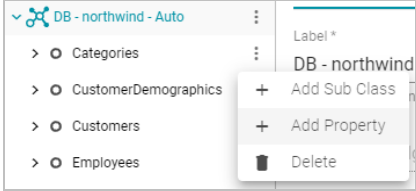
When Property Prefix is not set, Anzo generates the model's property URIs in the following format:

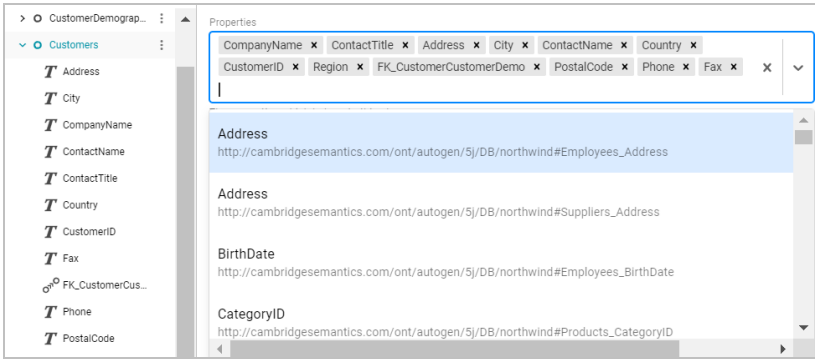
`http://cambridgesemantics.com/ontologies/<model_label>#p_<property_label>`.

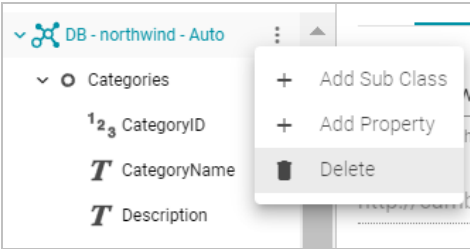
- **Imports:** Lists any definitions that you want to import from another model into this model. To select models to import, click in the **Imports** field and select a model from the drop-down list. Select the field again to select additional models.
- **System Model:** Indicates that the data model 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 Add Class.</p>  <p>Anzo opens the class editor so that you can configure the new class. See Class Editor Reference 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 Add Property.</p>  <p>Anzo opens the property editor so you can configure the new property. See Property Editor Reference below for information about property settings.</p>

What do you want to do?	Instructions
Add an existing property to a class	<p>To add an existing property to a class, click the class in the left pane to display the 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:</p> 
Edit a class	<p>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.</p>
Delete a property from a class	<p>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 trashcan icon (🗑️) on the top right of the screen. Then click Delete in the dialog box to confirm that you want to delete the property.</p>

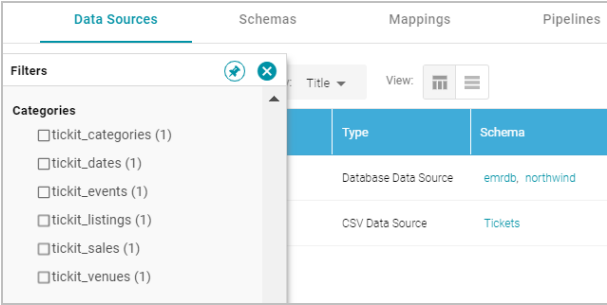
What do you want to do?	Instructions
Delete a class from the model	<p>Click the menu icon (⋮) to the right of the class that you want to remove from the model.</p>  <p>Click Delete. Anzo displays a dialog box that asks if you want to delete only the class or all of the subclasses and properties in the class. Select the appropriate option and then click Delete to confirm that you want to delete the class.</p> <div><p>Important</p><p>This action cannot be undone. Anzo removes the class and saves the model.</p></div>
Add a data range	<p>Click the menu icon (⋮) to the right of the model name. Then select Add Data Range. Anzo opens the data range editor so that you can configure the new range.</p>
Add an annotation	<p>Click the menu icon (⋮) to the right of the model name. Then select Add Annotation. Anzo opens the editor so that you can configure the annotation.</p>

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.

Field	Description
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 Enter or by choosing an available property from the drop-down list. For more information, see Setting Class Instance URI Patterns .
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 Enter 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 Title

Field	Description
Category	<p>Indicates whether the class should be listed as one of the categories that can be managed in the Data Sources and Datasets Category tabs and displayed in the list of quick filters that are available when sorting resource lists. For example:</p> <div></div> <p>For more information about categories, see Configuring Data Source Categories and Configuring Dataset Categories.</p>

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	<p>Indicates whether more than one value can exist for this property.</p> <div><p>Note</p><p>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.</p></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.

Field	Description
Property Range	The data type for the property. See Property Range Guidelines 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.
Value Restriction	Indicates whether to restrict the property's values to certain data types or specific values in a list.

Related Topics

[Introduction to Models](#)

[Model Requirements and Recommendations](#)

[Setting Class Instance URI Patterns](#)

[Downloading a Model](#)

Setting Class Instance URI Patterns

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.

When using the Ingest workflow (with Anzo-generated models, mappings, and pipelines), if a Resource Template is not defined for the classes in a model, Anzo generates class URIs by following this pattern:

```
<uri_prefix>/<class_name>/<primary_key>
```

Anzo uses the URI prefix of **http://csi.com/**, appends the name of the table (class), and adds the primary key value for each instance of the table. For example, the following URI is generated for an instance of a class called **MovieActors2**. The primary key for the **MovieActors2** table is **ActorID**, so the **ActorID** value is appended to the URI.

```
<http://csi.com/MovieActors2/31211756>
```

Note

For property URIs, the default URI prefix is <http://cambridgesemantics.com/>. The value is controlled by the URI Prefix option in system settings. See [Configure URI Prefix and SPARQL Options](#) 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.

Example

If you ingest movie data from multiple sources and each source assigns a movie ID as the primary key, the same movie title will likely be associated with multiple IDs. If the auto-generated URI pattern for the Movies class is `http://csi.com/Movies/<movie_ID>`, then all of the data for the same movie title will not be joined by the same class instance URI. In this case, defining a resource template that uses the movie title as the uniqueness condition rather than the ID would automatically join movie data from different sources.

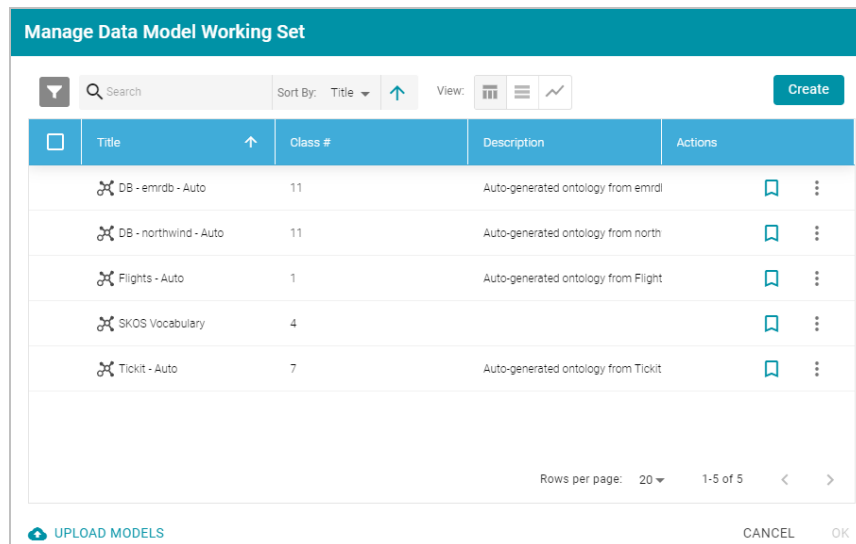
Follow the steps below to configure resource templates for each of the classes in a data model. For automatic ingestion workflows, configure resource templates after Anzo generates the model, mappings, and pipeline and before the pipeline is published. To configure resource templates for pipelines that have been published, edit the model and then re-publish the pipeline to update the instance data.

Important

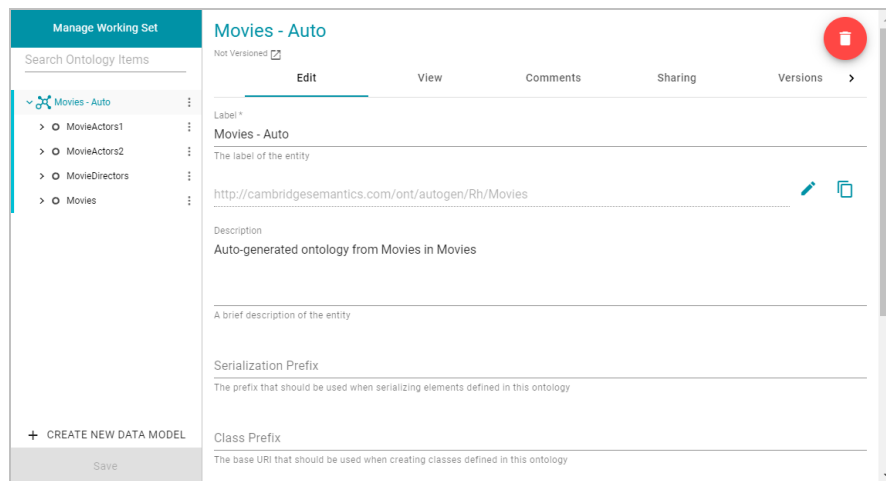
Keep the following points in mind when defining class instance URI patterns:

- 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.

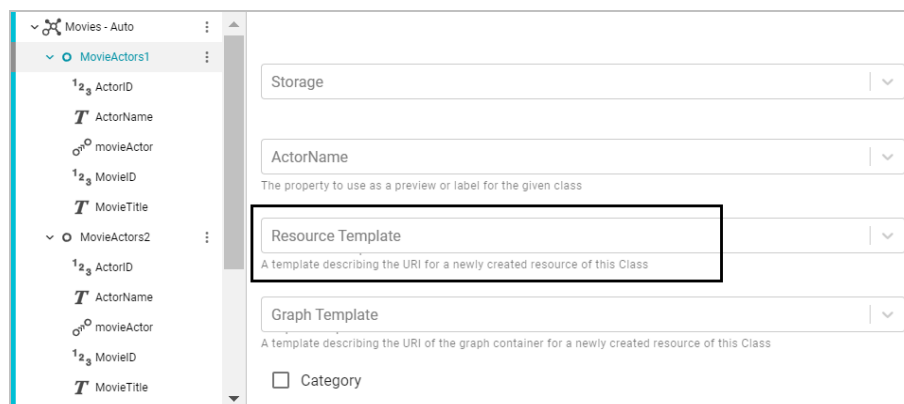
1. In the Anzo application, click **Model**. Anzo displays the Manage Data Model Working Set screen. For example:



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 for editing. Then click **OK**. Anzo opens the selected model in the editor. For example:



3. 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 MovieActors1 class.

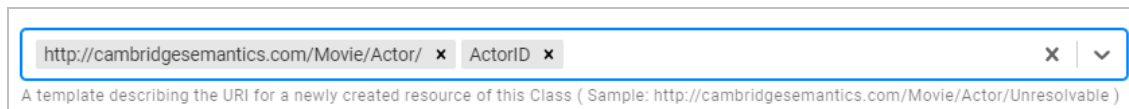


- Click the **Resource Template** field and specify the URI pattern to use for instances of this class. First, type a base value in the field and press **Enter** to add the value to the field. For example, for MovieActors1 in the step above: `http://cambridgesemantics.com/Movie/Actor/`.



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 MovieActors1 class, ActorID provides unique values.



A template describing the URI for a newly created resource of this Class (Sample: `http://cambridgesemantics.com/Movie/Actor/Unresolvable`)

- 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.

Related Topics

[Introduction to Models](#)

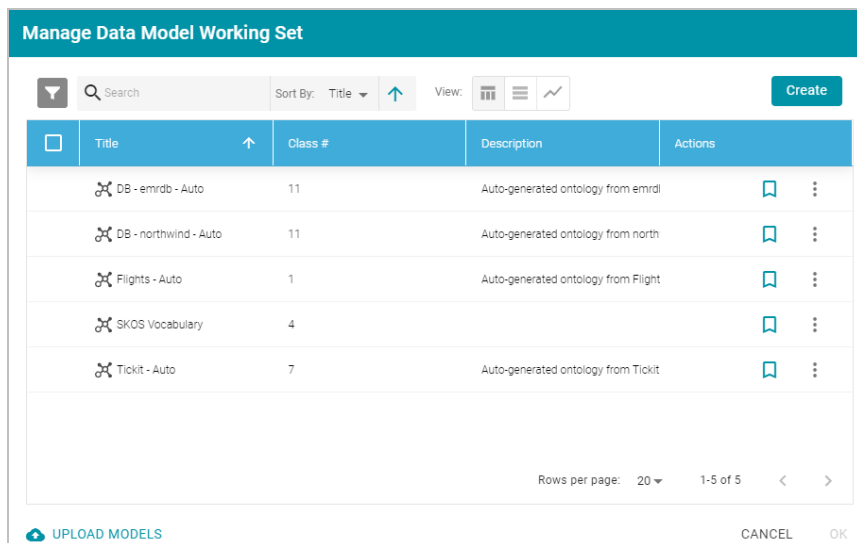
[Model Requirements and Recommendations](#)

[Editing a Model](#)

Downloading a Model

This topic provides instructions for downloading a data model from Anzo.

- In the Anzo application, click **Model**. Anzo displays the Manage Data Model Working Set screen. For example:

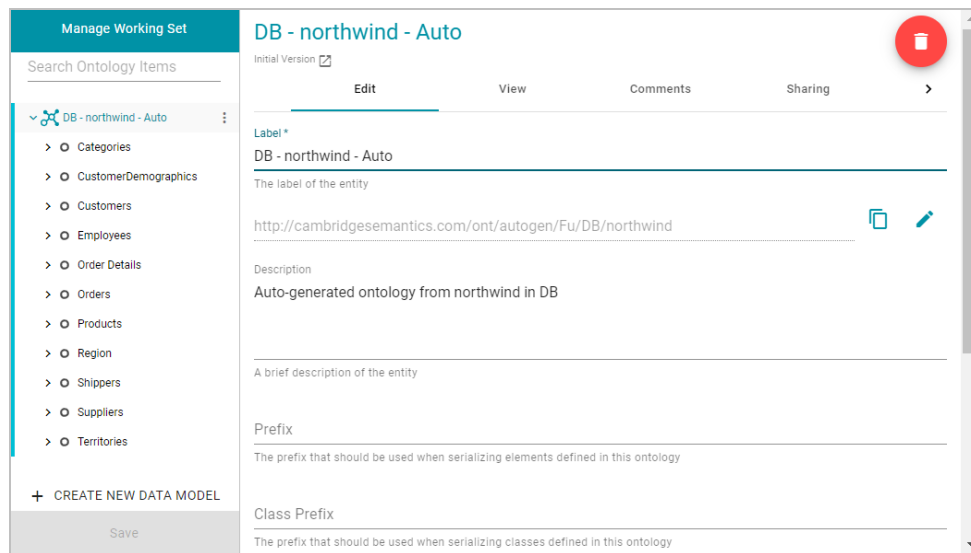


Manage Data Model Working Set				
Search		Sort By: Title	View:	Create
<input type="checkbox"/>	Title	Class #	Description	Actions
<input type="checkbox"/>	DB - emrdb - Auto	11	Auto-generated ontology from emrdb	
<input type="checkbox"/>	DB - northwind - Auto	11	Auto-generated ontology from north	
<input type="checkbox"/>	Flights - Auto	1	Auto-generated ontology from Flight	
<input type="checkbox"/>	SKOS Vocabulary	4		
<input type="checkbox"/>	Tickit - Auto	7	Auto-generated ontology from Tickit	

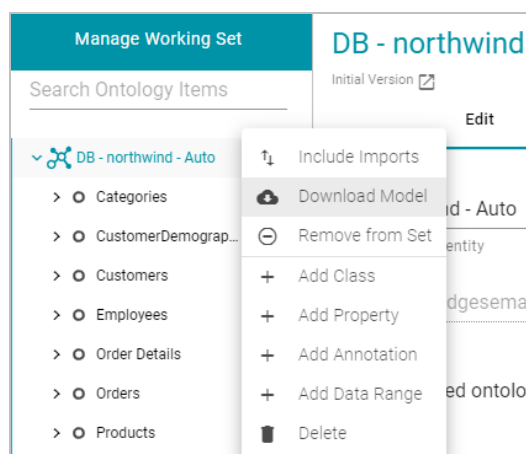
Rows per page: 20 1-5 of 5

UPLOAD MODELS CANCEL OK

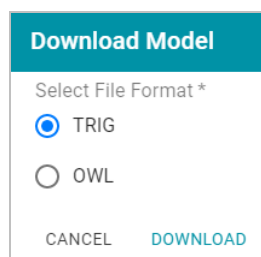
- On the Manage Working Set screen, select the checkbox next to the model that you want to export, and then click **OK**. Anzo opens the selected model in the editor. For example:



- 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 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 data 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

[Introduction to Models](#)

[Model Requirements and Recommendations](#)

[Uploading a Model to Anzo](#)

[Creating a Model](#)

[Editing a Model](#)

Blending Data

Once your data is onboarded to Anzo, the data sets in the Dataset 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 data sets in the Dataset catalog and Graphmarts.

- [Working with Datasets](#)
- [Working with Graphmarts](#)
- [Blending Data from Remote Sources \(Preview\)](#)

Working with Datasets

The topics in this section provide guidance on working with the Datasets catalog.

- [Adding a Dataset to the Dataset Catalog](#)
- [Adding an Empty Dataset for an Export Step](#)
- [Managing Dataset Editions](#)
- [Configuring Dataset Categories](#)
- [Generating a Dataset Data Profile](#)

Related Topics

[Sharing Access to Artifacts](#)

[Versioning and Migrating Artifacts](#)

Adding a Dataset to the Dataset Catalog

Source data that is not in RDF format is onboarded through structured or unstructured pipelines, where the data is imported to Anzo and converted to RDF format before becoming available in the Dataset catalog. Certain RDF file types, however, can be added to the catalog directly, making the data available to add to a Graphmart for loading and analyzing in AnzoGraph.

Users can add to the Dataset catalog any pre-existing file-based linked data set (FLDS), such as when migrating a Dataset from one Anzo server to another. Or they can point Anzo to a directory of Turtle or N-Triple files and Anzo will create the FLDS and add the Dataset to the catalog.

Note

To import data from CSV, JSON, XML, Parquet, or SAS files, follow the processes described in [Adding Data Sources and Schemas](#).

This topic provides instructions for making RDF files available as a Dataset in the catalog.

- [File Requirements](#)
- [Importing RDF Files](#)
- [Importing an Existing Dataset](#)

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 must be staged on a configured file store.
- **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 imported as *filename.filetype.gz* files.

- **Supported Directory Structure:** The directory structure that is required depends on whether you are importing a File-Based Linked Data Set (FLDS)—a data set that was previously created by onboarding data to Anzo—or files that are not yet part of an FLDS:
 - **FLDS Imports:** 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:

```
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
```

Note Models must be in TriG format, regardless of the file type of the data files.

- **RDF File Imports:** 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>` 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.

Importing RDF Files

Follow the instructions below to create a Dataset catalog entry from a directory of Turtle or N-Triple. Make sure that the files and directory meet the requirements in [File Requirements](#).

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

Sort By: Title

View:

Add Dataset

<div></div>	Title		Edition	Creator	Created Date	Updated Date	Actions	
<div></div>	1}. DB - new-hires to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021	<div></div>	<div></div>
<div></div>	1}. DB - northwind to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	<div></div>	<div></div>
<div></div>	1}. Datafox to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	<div></div>	<div></div>
<div></div>	1}. Export Tickets		http://openanzo.org/DatasetEditio...	System Administrator	May 11, 2021	May 11, 2021	<div></div>	<div></div>
<div></div>	1}. Flight Data to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	<div></div>	<div></div>
<div></div>	1}. LeagueData with LeagueDict to		http://csi.com/FileBasedLinkedDat...	System Administrator	May 11, 2021	May 11, 2021	<div></div>	<div></div>
<div></div>	1}. Movie Data to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 12, 2021	May 12, 2021	<div></div>	<div></div>
<div></div>	1}. Tickets to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021	<div></div>	<div></div>

Rows per page: 251-8 of 8<>

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. Find and select the **rdf.<filetype>** directory that you want to import, and then click **OK** to close the dialog box.
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** to create the FLDS, add it to the catalog, and return to the Datasets screen. You can now select the FLDS from the catalog and create a Graphmart. See [Creating a New Graphmart](#) for instructions.



















Note

Anzo generates an flds.trig file at the same level as the rdf.ttl or rdf.ttl.gz directory. The file contains metadata about the load files.

Importing an Existing Dataset

Follow the instructions below to add an existing Dataset, such as an exported Dataset, to the catalog. Make sure that the FLDS meets the requirements in [File Requirements](#).

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:

<div> <div> <div> <div></div> <div>Search</div> </div> <div>Sort By: Title</div> <div>View:  </div> </div> <div>Add Dataset</div> </div>							
<input type="checkbox"/>	Title	Edition	Creator	Created Date	Updated Date	Actions	
<input type="checkbox"/>	DB - new-hires to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021	 	
<input type="checkbox"/>	DB - northwind to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	 	
<input type="checkbox"/>	Datafox to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	 	
<input type="checkbox"/>	Export Tickets	http://openanzo.org/DatasetEditio...	System Administrator	May 11, 2021	May 11, 2021	 	
<input type="checkbox"/>	Flight Data to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	 	
<input type="checkbox"/>	LeagueData with LeagueDict to	http://csi.com/FileBasedLinkedDat...	System Administrator	May 11, 2021	May 11, 2021	 	
<input type="checkbox"/>	Movie Data to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 12, 2021	May 12, 2021	 	
<input type="checkbox"/>	Tickets to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021	 	
<div>Rows per page: 25 1-8 of 8 < ></div>							

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

- Click the **RDF File Location** field to open the File Location dialog box. Select 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. For example:

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 *

[/nfs/data/LoadLeagueDataWithLeagueDicttoLocalDataStore_f15f4/](#) **BROWSE**

File path should be the root of the existing FLDS

CANCEL **SAVE**

- Click **Save** to import the FLDS and return to the Datasets screen. You can now select the Dataset in the catalog and create a Graphmart. See [Creating a New Graphmart](#) for instructions.

Related Topics

[Creating a CSV Data Source](#)

[Creating a JSON Data Source](#)

[Creating an XML Data Source](#)

[Creating a SAS Data Source](#)

[Creating a Parquet Data Source and Ingesting the Data](#)

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 RDF files that are output from Export Steps.

Note

The RDF File Location that you specify when creating an empty Dataset must preexist on the file store. When the Dataset is saved, Anzo generates an **flds.trig** file in the specified directory.

- If necessary, create an empty directory on the File Store where Anzo can save the exported RDF files.
- In the Anzo application, expand the **Blend** menu and click **Datasets**. Anzo displays the Datasets screen, which lists the catalog of Datasets. For example:

<div> <div></div> <div>Search</div> </div>		Sort By: Title	View: <div> <div></div> <div></div> </div>		Add Dataset		
	Title	Edition	Creator	Created Date	Updated Date	Actions	
	1}. DB - new-hires to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021		
	1}. DB - northwind to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021		
	1}. Datafox to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021		
	1}. Export Tickets	http://openanzo.org/DatasetEditio...	System Administrator	May 11, 2021	May 11, 2021		
	1}. Flight Data to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021		
	1}. LeagueData with LeagueDict to	http://csi.com/FileBasedLinkedDat...	System Administrator	May 11, 2021	May 11, 2021		
	1}. Movie Data to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 12, 2021	May 12, 2021		
	1}. Tickets to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021		
<div> <div>Rows per page: 25</div> <div>1-8 of 8</div> <div> <div></div> <div></div> </div> </div>							

- 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

4. 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

Set the dataset's default data format, such as ttl or ttl.gz.

CANCEL SAVE

5. Type a name for the new Dataset in the **Title** field and an optional description in the **Description** field.
6. Click the **RDF File Location** field to open the File Location dialog box. Find and select the empty directory that you created for the exported files, and then click **OK** to close the dialog box.
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 Dataset.
8. Finally, by default, Export steps export the RDF files in uncompressed Turtle (.ttl) format. If you would like to generate compressed Turtle files, specify **ttl.gz** in the **RDF Format** field. When the data is exported, Anzo creates an **rdf.ttl.gz** directory with the exported files. If the **RDF Format** field is blank, the exported files are saved to an **rdf.ttl** directory.
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

[Adding a Dataset to the Dataset Catalog](#)

[Configuring an Export Step](#)

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 three times:

Title *				Description			
Job Data included in this Edition				Data Component for selected job included in this Edition			
<input type="checkbox"/>	Name	Published Date	Job Type	<input type="checkbox"/>	Name	Published Date	
<input checked="" type="checkbox"/>	Load Movies	05/12/2021 11:48AM	Standard	<input checked="" type="checkbox"/>	LoadMovies_Movies	05/12/2021 11:48AM	
<input type="checkbox"/>	Load MovieProducers	05/12/2021 11:47AM	Standard	<input type="checkbox"/>	LoadMovies_Movies	05/12/2021 11:46AM	
<input type="checkbox"/>	Load MovieComposers	05/12/2021 11:42AM	Standard	<input type="checkbox"/>	LoadMovies_Movies	05/12/2021 11:40AM	
<input type="checkbox"/>	Load MovieDirectors	05/12/2021 11:41AM	Standard				
<input type="checkbox"/>	Load MovieActors	05/12/2021 11:41AM	Standard				
Rows per page: 25 1-6 of 6				Rows per page: 25 1-3 of 3			
1 Items Selected				1 Items Selected			

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 three 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.

The screenshot shows the 'Overview' tab of a Dataset in Anzo. The 'Managed Editions' section is highlighted with a black box. It contains a table with the following data:

Title	Description	Most Recent Published Date	Actions
Default Edition	Contains the latest successfully publishe...	05/12/2021 11:48AM	⋮

Below the Managed Editions section, there is a 'Saved Editions' section with a search bar, a 'Sort By: Title' dropdown, and a 'Create New Edition' button. The search results show 'No editions found'.

Tip

Note that the **Title** of the Managed Edition in the image is **Default Edition**. The Title of your Managed Edition may vary, depending on whether the Edition was created by publishing a new structured pipeline (as is the case in the example) or whether it resulted from an unstructured pipeline or an Anzo upgrade where the Dataset from the previous Anzo version was converted to an Edition in the new version. The Title for an Edition that was converted during an upgrade is in the form of **<dataset_name> working edition**.

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](#)

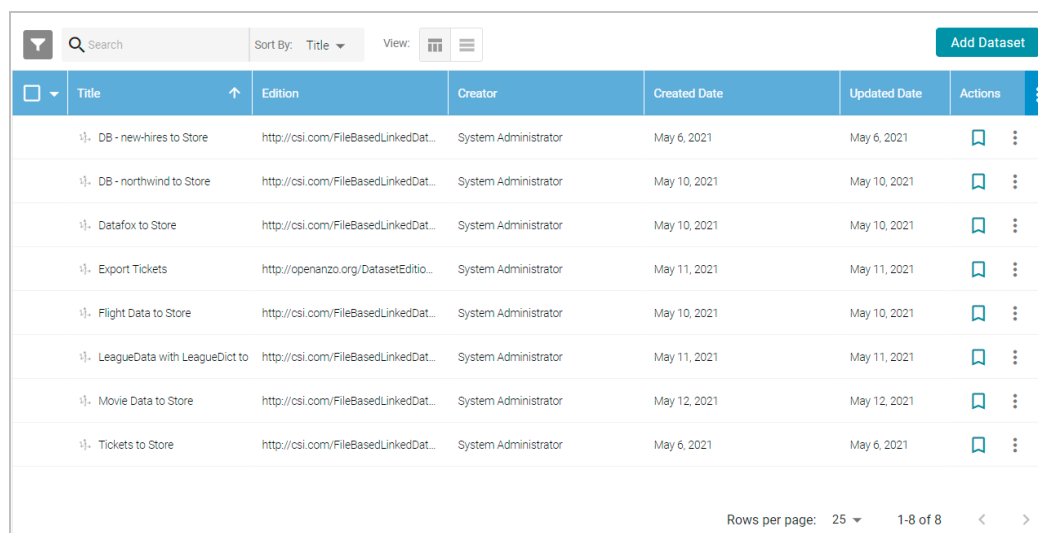
[Modifying an Edition](#)

[Deleting a Saved 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:



	Title	Edition	Creator	Created Date	Updated Date	Actions
1	DB - new-hires to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021	
2	DB - northwind to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	
3	Datafox to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	
4	Export Tickets	http://openanzo.org/DatasetEditio...	System Administrator	May 11, 2021	May 11, 2021	
5	Flight Data to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	
6	LeagueData with LeagueDiot to	http://csi.com/FileBasedLinkedDat...	System Administrator	May 11, 2021	May 11, 2021	
7	Movie Data to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 12, 2021	May 12, 2021	
8	Tickets to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021	

Rows per page: 25 1-8 of 8

2. 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:

The screenshot shows the 'Overview' tab of the Anzo interface. At the top, there are navigation tabs: Overview (selected), Explore, Graphmarts, Pipelines, and Dashboards. Below the tabs, the 'Description' is set to 'None'. A dropdown menu labeled 'Advanced' is visible. The 'Pipeline' section shows 'Load Movie Data to Store' with a long URL and a copy icon. Below this is the 'Managed Editions' section, which contains a table with one row: 'Default Edition'. The table has columns for Title, Description, Most Recent Published Date, and Actions. Below the Managed Editions section is the 'Saved Editions' section, which includes a search bar, a 'Sort By: Title' dropdown, and a 'Create New Edition' button. At the bottom of the Saved Editions section, it says 'No editions found'.

3. 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, and the right side of the screen lists the job data components for the selected job. For example:

Create New Edition

Title *

Description

Job Data included in this Edition

	Name	Published Date	Job Type
<input checked="" type="checkbox"/>	Load Movies	05/12/2021 11:48AM	Standard
<input type="checkbox"/>	Load MovieProducers	05/12/2021 11:47AM	Standard
<input type="checkbox"/>	Load MovieComposers	05/12/2021 11:42AM	Standard
<input type="checkbox"/>	Load MovieDirectors	05/12/2021 11:41AM	Standard
<input type="checkbox"/>	Load MovieActors	05/12/2021 11:41AM	Standard

Rows per page: 25 1-6 of 6

1 Items Selected

Data Component for selected job included in this Edition

	Name	Published Date
<input checked="" type="checkbox"/>	LoadMovies_Movies	05/12/2021 11:48AM
<input type="checkbox"/>	LoadMovies_Movies	05/12/2021 11:46AM
<input type="checkbox"/>	LoadMovies_Movies	05/12/2021 11:40AM

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.
- In the Job Data list, select the checkbox next to a job that you want to add to this Edition. The Data Components for the job are displayed on the right side of the screen. The most recent version of the Data Component for the job is selected by default. For example:

Title *

Description

Original Version of Movies and Producers

Use older versions of the movies and producers 1

Job Data included in this Edition

	Name	Published Date	Job Type
<input type="checkbox"/>	Load Movies	05/12/2021 11:48AM	Standard
<input checked="" type="checkbox"/>	Load MovieProducers	05/12/2021 11:47AM	Standard
<input type="checkbox"/>	Load MovieComposers	05/12/2021 11:42AM	Standard
<input type="checkbox"/>	Load MovieDirectors	05/12/2021 11:41AM	Standard
<input type="checkbox"/>	Load MovieActors	05/12/2021 11:41AM	Standard
<input type="checkbox"/>	Load MovieEditors	05/12/2021 11:40AM	Standard

Rows per page: 25 1-6 of 6

1 Items Selected

Data Component for selected job included in this Edition

	Name	Published Date
<input checked="" type="checkbox"/>	LoadMovieProducers_MovieProducers	05/12/2021 11:47AM
<input type="checkbox"/>	LoadMovieProducers_MovieProducers	05/12/2021 11:45AM
<input type="checkbox"/>	LoadMovieProducers_MovieProducers	05/12/2021 11:42AM

Rows per page: 25 1-3 of 3

1 Items Selected

- If necessary, modify the Data Component list to select the version that you want to include in the new Edition. Repeat steps 5 and 6 for each job that you want to include in the Edition.

© 2023 Cambridge Semantics, Inc.

Note

When multiple jobs are selected in the Job Data list, the Data Component list shows the Data Components for the most recently selected job.

- When you are finished selecting jobs and Data Components, click **Save** to save the Edition. The new edition is added to the list of **Saved Editions** on the Overview screen. For example:

<input type="text" value="Search"/>		Sort By: Title		Create New Edition	
Title	↑	Description	Last Modified Date	Most Recent Published Date	Actions
Use Original Producers and Movies		Older versions of producer and ...	05/12/2021 01:23PM	05/12/2021 11:48AM	⋮

From the **Actions** menu for an Edition, you can create a Graphmart, or you can browse, clone, or delete the Edition.

Related Topics

[Introduction to Editions](#)

[Modifying an Edition](#)











[Deleting a Saved Edition](#)

[Creating a New Graphmart](#)

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.

- In the Anzo application, expand the **Blend** menu and click **Datasets**. Anzo displays the Datasets screen, which lists the catalog of Datasets. For example:

<input type="text" value="Search"/>		Sort By: Title		View:  		Add Dataset	
<input type="checkbox"/>	Title	↑	Edition	Creator	Created Date	Updated Date	Actions
<input type="checkbox"/>	DB - new-hires to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021	 ⋮
<input type="checkbox"/>	DB - northwind to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	 ⋮
<input type="checkbox"/>	Datafox to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	 ⋮
<input type="checkbox"/>	Export Tickets		http://openanzo.org/DatasetEditio...	System Administrator	May 11, 2021	May 11, 2021	 ⋮
<input type="checkbox"/>	Flight Data to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	 ⋮
<input type="checkbox"/>	LeagueData with LeagueDict to		http://csi.com/FileBasedLinkedDat...	System Administrator	May 11, 2021	May 11, 2021	 ⋮
<input type="checkbox"/>	Movie Data to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 12, 2021	May 12, 2021	 ⋮
<input type="checkbox"/>	Tickets to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021	 ⋮
Rows per page: 25 1-8 of 8 < >							

- Click the Dataset for which you want to modify 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

Description
None

Advanced ▼

Pipeline
Load Movie Data to Store <http://cambridgesemantics.com/Project/611fe632-1c0d-2dc6-30b4-233b7a9c17c4/611fe632-1c0d-2dc6-30...>

Managed Editions

Title	Description	Most Recent Published Date	Actions
Default Edition	Contains the latest successfully publishe...	05/12/2021 11:48AM	⋮

Saved Editions

Search Sort By: Title Create New Edition

No editions found

- 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

<input type="checkbox"/>	Name	Published Date	Job Type
<input checked="" type="checkbox"/>	Load Region	05/10/2021 11:42AM	Standard
<input type="checkbox"/>	Load Products	05/10/2021 11:42AM	Standard
<input checked="" type="checkbox"/>	Load Territories	05/10/2021 11:42AM	Standard
<input type="checkbox"/>	Load Summary of Sales t	05/10/2021 11:41AM	Standard
<input type="checkbox"/>	Load Shippers	05/10/2021 11:41AM	Standard
<input checked="" type="checkbox"/>	Load Suppliers	05/10/2021 11:40AM	Standard

Rows per page: 25 1-7 of 7

3 Items Selected

Data Component for selected job included in this Edition

<input checked="" type="checkbox"/>	Name	Published Date
<input checked="" type="checkbox"/>	LoadSuppliers_Suppliers	05/10/2021 11:40AM

Rows per page: 25 1-1 of 1

1 Items Selected

CANCEL SAVE

4. Specify a name for the Edition in the **Title** field and include an optional description in the **Description** field.
5. 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.
6. 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**.

Related Topics

[Introduction to Editions](#)

[Creating an Edition](#)

[Adding a Dataset to a Graphmart](#)

[Creating a New Graphmart](#)

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:

<div> <div> <div></div> <div>Search</div> </div> <div> <div>Sort By: Title</div> <div>View:  </div> </div> </div> <div>Add Dataset</div>							
	Title	Edition	Creator	Created Date	Updated Date	Actions	
	DB - new-hires to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021	 	
	DB - northwind to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	 	
	Datafox to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	 	
	Export Tickets	http://openanzo.org/DatasetEditio...	System Administrator	May 11, 2021	May 11, 2021	 	
	Flight Data to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	 	
	LeagueData with LeagueDict to	http://csi.com/FileBasedLinkedDat...	System Administrator	May 11, 2021	May 11, 2021	 	
	Movie Data to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 12, 2021	May 12, 2021	 	
	Tickets to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021	 	
Rows per page: 25 1-8 of 8 < >							

- 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
<div>Description</div> <div>None</div> <div>Advanced </div> <div>Pipeline</div> <div>Load Tickets to Store http://cambridgesemantics.com/Project/c1c694bd-849d-91d0-eb34-bfef8c2d8894/c1c694bd-849d-91d0-eb34-... </div> <div>Managed Editions</div> <div> <div>Title</div> <div>↑</div> <div>Description</div> <div>Most Recent Published Date</div> <div>Actions</div> <div></div> </div> <div>Default Edition</div> <div>Contains the latest successfully published...</div> <div>05/06/2021 11:34AM</div> <div></div>					
<div>Saved Editions</div> <div> <div>Search</div> <div>Sort By: Title</div> <div>Create New Edition</div> </div> <div> <div>Title</div> <div>↑</div> <div>Description</div> <div>Last Modified Date</div> <div>Most Recent Published Date</div> <div>Actions</div> <div></div> </div> <div>Add Movie Component</div> <div>05/14/2021 10:21AM</div> <div>05/06/2021 11:34AM</div> <div></div> <div>Exclude Users</div> <div>05/14/2021 10:20AM</div> <div>05/06/2021 11:34AM</div> <div></div>					

- 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.

Related Topics

[Introduction to Editions](#)

[Creating an Edition](#)

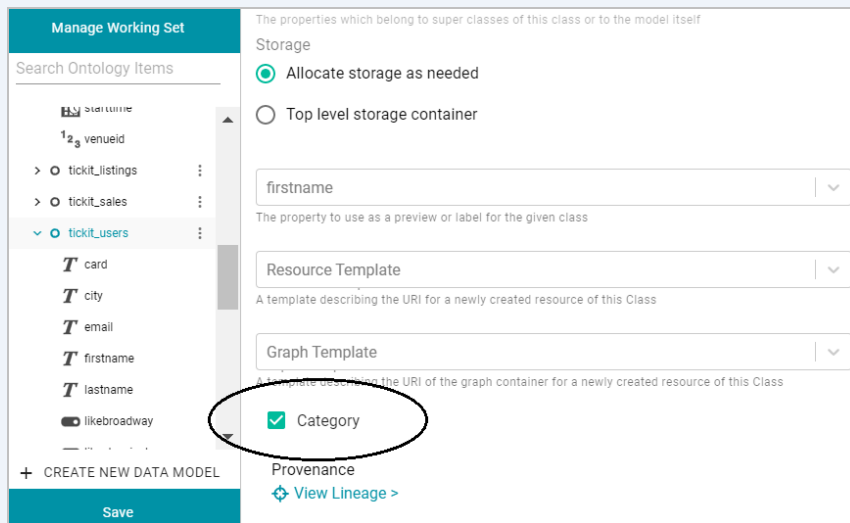
[Modifying an Edition](#)

Configuring Dataset Categories

Anzo's Category manager provides a way to define metadata about a data set that can be used to classify or catalog the data. Categories describe the properties in a data set but are independent of the instance data. When categories are configured for a data set in the catalog, they are displayed as choices in the list of quick filters that are available when sorting data sets. This topic provides instructions for configuring data set categories.

Note

Before you can configure categories for a data set, the **Category** setting must be enabled for the classes in the data model for that data set. If necessary, open the model for editing and select the **Category** checkbox for each class that you want to list as a category. For example:



Make sure that you save the model changes. You do not need to re-ingest the data source. The Category tab for that data set becomes available once the model is saved. For more information about changing a model, see [Editing a Model](#).

Follow the steps below to configure categories.

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

Sort By: Title

View:

Add Dataset

<div></div>	Title	Edition	Creator	Created Date	Updated Date	Actions	
<div></div>	1}. DB - new-hires to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021	<div></div>	<div></div>
<div></div>	1}. DB - northwind to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	<div></div>	<div></div>
<div></div>	1}. Datafox to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	<div></div>	<div></div>
<div></div>	1}. Export Tickets	http://openanzo.org/DatasetEditio...	System Administrator	May 11, 2021	May 11, 2021	<div></div>	<div></div>
<div></div>	1}. Flight Data to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	<div></div>	<div></div>
<div></div>	1}. LeagueData with LeagueDict to	http://csi.com/FileBasedLinkedDat...	System Administrator	May 11, 2021	May 11, 2021	<div></div>	<div></div>
<div></div>	1}. Movie Data to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 12, 2021	May 12, 2021	<div></div>	<div></div>
<div></div>	1}. Tickets to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021	<div></div>	<div></div>

Rows per page: 25 1-8 of 8

2. Click the data set for which you want to configure categories. Then click the **Category** tab. For example:

Tickets

Not Versioned

Profile Data

+ Add to Cart

Create Graphmart

Overview

Explore

Graphmarts

Pipelines

Dashboards

Versions

Category

Categories

Manage Categories

There are no categories applied to this object.

Category/Class Properties
Select a Category or Class on the left to see details here

3. Click the **Manage Categories** button. The Manage Categories dialog box is displayed, which lists each of the classes that are designated as categories in the data model. For example:

Manage Categories

☐ ticket_categories

☐ ticket_dates

☐ ticket_events

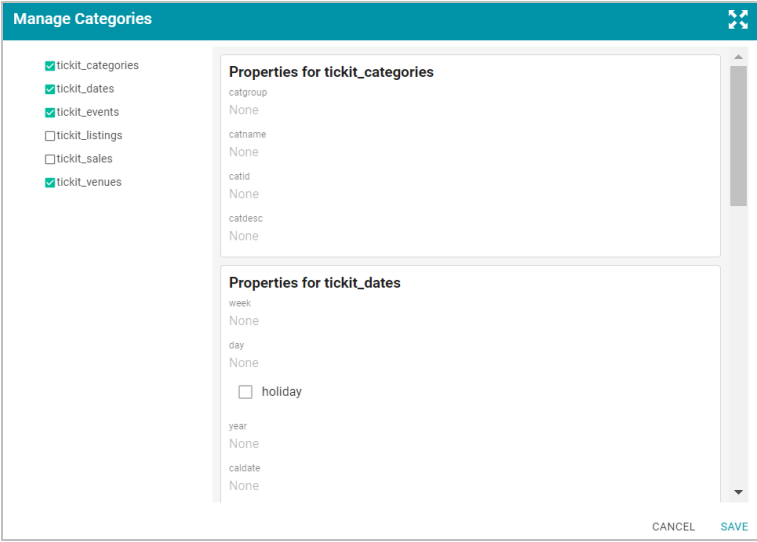
☐ ticket_listings

☐ ticket_sales

☐ ticket_venues

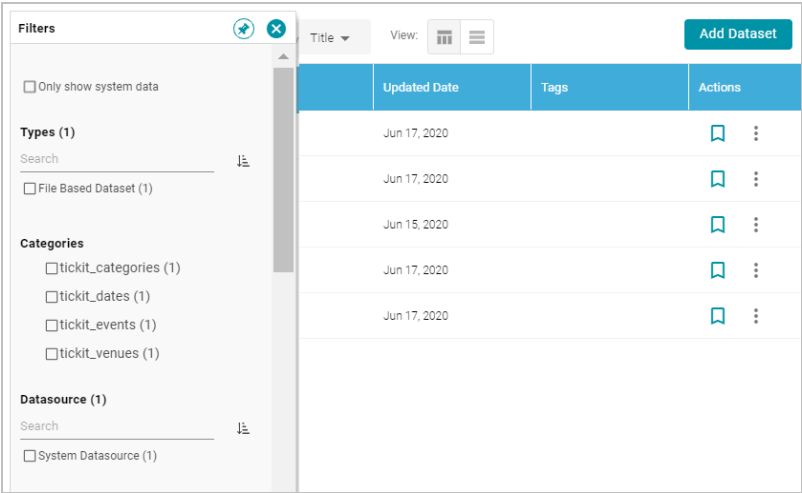
CANCEL SAVE

4. On the left side of the screen, select the checkbox next to each class that you want to display as a category. When you select a class, the properties for that class are displayed on the right side of the screen. For example:



5. In the list of properties for each class, you can edit the values to categorize the data for that property. For example, if you know that the data set has date values that fall in a specific date range, you can specify that range in a date-related property, such as the "year" property in the image above. To add a description for a property, click the value field under the property to make the field editable. The characters that are supported depend on the data type of the property. Click the checkmark icon (✓) to save the change. Repeat this step for any of the properties that you want to describe.
6. When you have finished adding values, click **Save** to save the configuration and close the Manage Categories dialog box. Categories can be modified any time from the Category tab.

Categories are displayed as quick filters in the Filters panel that is available when sorting the data set list on the Datasets screen. Open the Filters panel by clicking the filter icon (🔍) in the top left corner of the screen. For example:



When a category is selected, the properties for that class are also displayed in the Filters panel.

Related Topics

[Configuring Data Source Categories](#)

Generating a Dataset Data Profile

Similar to generating a profile for a data source (see [Generating a Source Data Profile](#)), Anzo provides the ability to profile a graph data set in its final format. When metrics are generated for graph data, Anzo profiles the entire data set and reports metrics for the classes and properties in the model as well as statistics about the values for the properties. Generating a graph data profile helps users perform data discovery, assess the quality of the onboarded data, and decide whether to use the data set in a particular graphmart. The report can also assist users in determining the types of data layers to create and writing the queries to include in the steps.

Important

To generate a graph data profile, AnzoGraph must be online. If you have dynamic AnzoGraph deployments enabled, Anzo will provision AnzoGraph automatically when metrics are 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

Sort By: Title

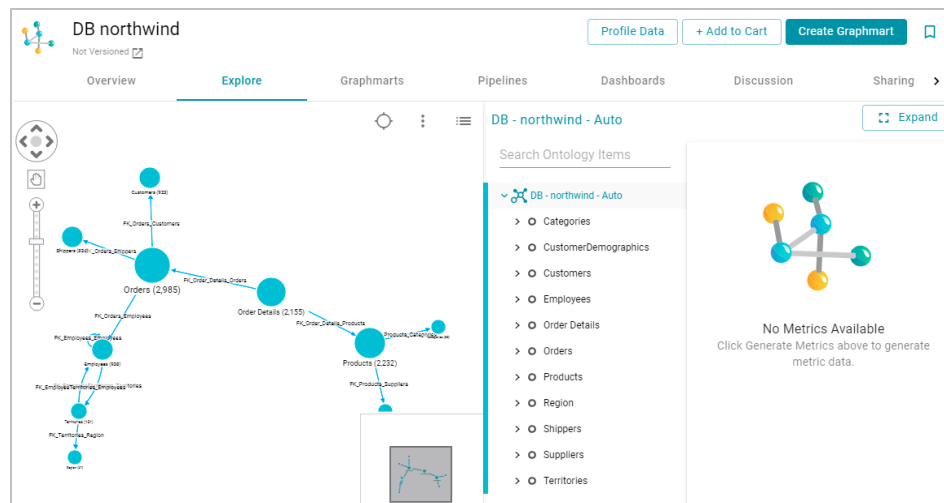
View:

Add Dataset

<div><div></div><div></div></div>	Title	↑	Edition	Creator	Created Date	Updated Date	Actions
1	DB - new-hires to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021	<div><div></div><div></div></div>
2	DB - northwind to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	<div><div></div><div></div></div>
3	Datafox to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	<div><div></div><div></div></div>
4	Export Tickets		http://openanzo.org/DatasetEditio...	System Administrator	May 11, 2021	May 11, 2021	<div><div></div><div></div></div>
5	Flight Data to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021	<div><div></div><div></div></div>
6	LeagueData with LeagueDict to		http://csi.com/FileBasedLinkedDat...	System Administrator	May 11, 2021	May 11, 2021	<div><div></div><div></div></div>
7	Movie Data to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 12, 2021	May 12, 2021	<div><div></div><div></div></div>
8	Tickets to Store		http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021	<div><div></div><div></div></div>

Rows per page: 251-8 of 8<>

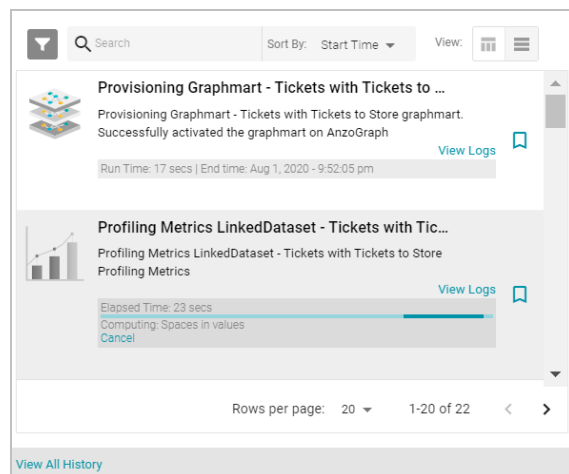
2. On the Datasets screen, click the name of the data set for which you want to generate metrics. Anzo displays the Explore screen for the source. For example:



Note

The instance counts for the classes in the graph view on the left side of the screen are the initial, non-unique counts from the ETL engine. Most likely the data has not yet been deduplicated. After generating metrics, the instance counts may change.

- Click the **Profile Data** button at the top of the screen. Anzo provisions a temporary graphmart and loads the data into AnzoGraph. AnzoGraph computes the data profiling metrics. The process may take several minutes. 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:

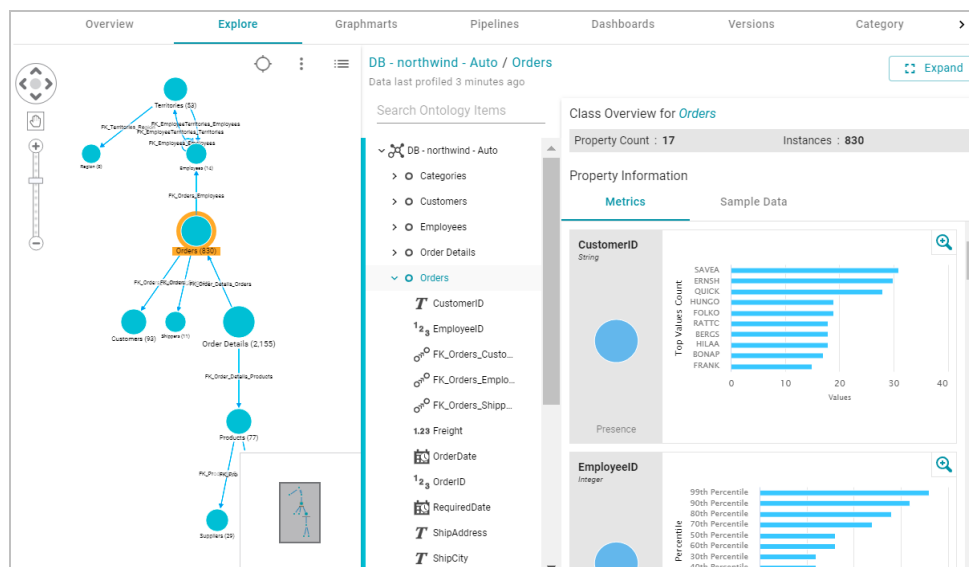


Once the metrics are generated, Anzo removes the graphmart from AnzoGraph and the new information becomes available to explore in Anzo.

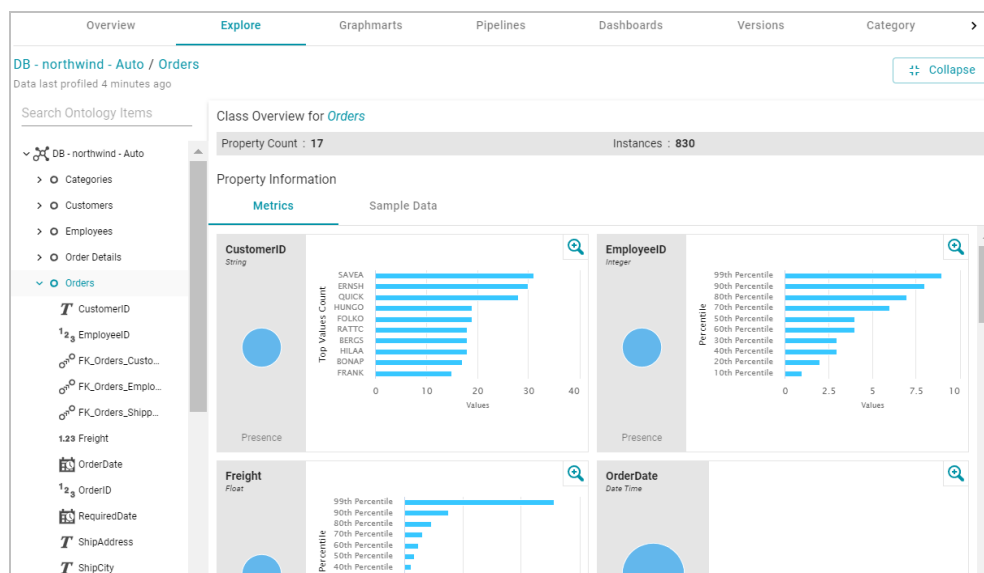
Note

Graph data profiles are always generated against the Default Edition of the data set. Saved Editions are not included.

- To display the metrics, select a node or edge in the graph view on the left side of the screen or expand the model in the middle of the screen and select a class or property. For example:



You can click the **Expand** button on the right side of the screen to collapse the graph view and expand the metrics view. For example:



Select any class or property to view its metrics. When a class is selected, Anzo displays the number of properties and total number of instances as well as one or more of the following metrics for each property in the class. The metrics that are calculated depend on the data type of the properties:

- **Percentile Metric:** This metric presents the data distribution for a property in percentiles.
- **Top Value Counts Metric:** This metric displays the count (as a histogram) of the 10 most frequently occurring values for a property.
- **Presence Metric:** This metric displays the number of values present and not present for a property.

When a property is selected, Anzo displays the metrics described above and one or more of the following metrics, depending on the data type of the property:

- **Extrema Metric:** Shows the smallest and largest values.
- **Geometric Mean Metric:** Shows the geometric mean of the values.
- **Median Metric:** Shows the middle value.
- **Mode Metric:** Shows the value that appears most often.
- **Std Deviation Metric:** Shows the standard deviation in the set of values.
- **Unique Values Metric:** Shows the number of unique values.

Related Topics

[Creating a New Graphmart](#)

[Adding Data Layers to Graphmarts](#)

Working with Graphmarts

The topics in this section provide guidance on working with Graphmarts and Data Layers.

- [Creating a New Graphmart](#)
- [Copying a Graphmart](#)
- [Graphmart Settings Reference](#)
- [Generating a Graphmart Data Profile](#)
- [Adding a Dataset to a Graphmart](#)
- [Working with Data Layers](#)
- [Sharing Access to Graphmarts](#)
- [Creating Data on Demand Endpoints](#)

Related Topics

[Sharing Access to Artifacts](#)

[Versioning and Migrating Artifacts](#)

[Blending Data from Remote Sources \(Preview\)](#)

Creating a New Graphmart

Follow the steps below to create and activate a new Graphmart.

Tip

You can also create a Graphmart by copying an existing one. For instructions, see [Copying a Graphmart](#).

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:

<input type="checkbox"/>	Title	Edition	Creator	Created Date	Updated Date	Actions	
<input type="checkbox"/>	DB - new-hires to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021		
<input type="checkbox"/>	DB - northwind to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021		
<input type="checkbox"/>	Datafox to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021		
<input type="checkbox"/>	Export Tickets	http://openanzo.org/DatasetEdito...	System Administrator	May 11, 2021	May 11, 2021		
<input type="checkbox"/>	Flight Data to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 10, 2021	May 10, 2021		
<input type="checkbox"/>	LeagueData with LeagueDict to	http://csi.com/FileBasedLinkedDat...	System Administrator	May 11, 2021	May 11, 2021		
<input type="checkbox"/>	Movie Data to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 12, 2021	May 12, 2021		
<input type="checkbox"/>	Tickets to Store	http://csi.com/FileBasedLinkedDat...	System Administrator	May 6, 2021	May 6, 2021		

Rows per page: 25 1-8 of 8

2. In the Dataset 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:

<input type="checkbox"/>	Title	Edition	Creator	Created Date	Updated Date	Actions	
<input type="checkbox"/>	DB - new-hires to Store	http://csi.com/FileBasedLin...	System Administrator	May 6, 2021	May 6, 2021		
<input type="checkbox"/>	DB - northwind to Store	http://csi.com/FileBasedLin...	System Administrator	May 10, 2021	May 10, 2021		
<input type="checkbox"/>	Datafox to Store	http://csi.com/FileBasedLin...	System Administrator	May 10, 2021	May 10, 2021		
<input type="checkbox"/>	Export Tickets	http://openanzo.org/Datas...					
<input checked="" type="checkbox"/>	Flight Data to Store	http://csi.com/FileBasedLin...	System Administrator	May 10, 2021	May 10, 2021		
<input type="checkbox"/>	LeagueData with LeagueDI	http://csi.com/FileBasedLin...	System Administrator	May 11, 2021	May 11, 2021		

Rows per page: 25 1-7 of 7

1 Items Selected

Bookmark Delete Create Graphmart

- Click the **Create Graphmart** button. Anzo displays the Create Graphmart screen.

Anzo populates the Title field by appending "Graphmart" to the data set name.

- On the Create Graphmart screen, you have the option to edit the **Title** and add an optional **Description**.
- By default the current working edition (Managed Edition) of the Dataset is selected. If you want to select a different Edition, follow these steps:
 - click **Modify Edition**. The Modify Edition dialog box is displayed. For example:

- To choose a different Edition, click the drop-down list at the top of the screen and select the Edition to use.
- If you want to make changes to the selected 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.

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. For more information about changing or creating Editions, see [Managing Dataset Editions](#).

- d. When you are finished making changes, choose one of the following options for saving the changes:
- 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 and click **Save**.

Create New Edition

Title *

Description

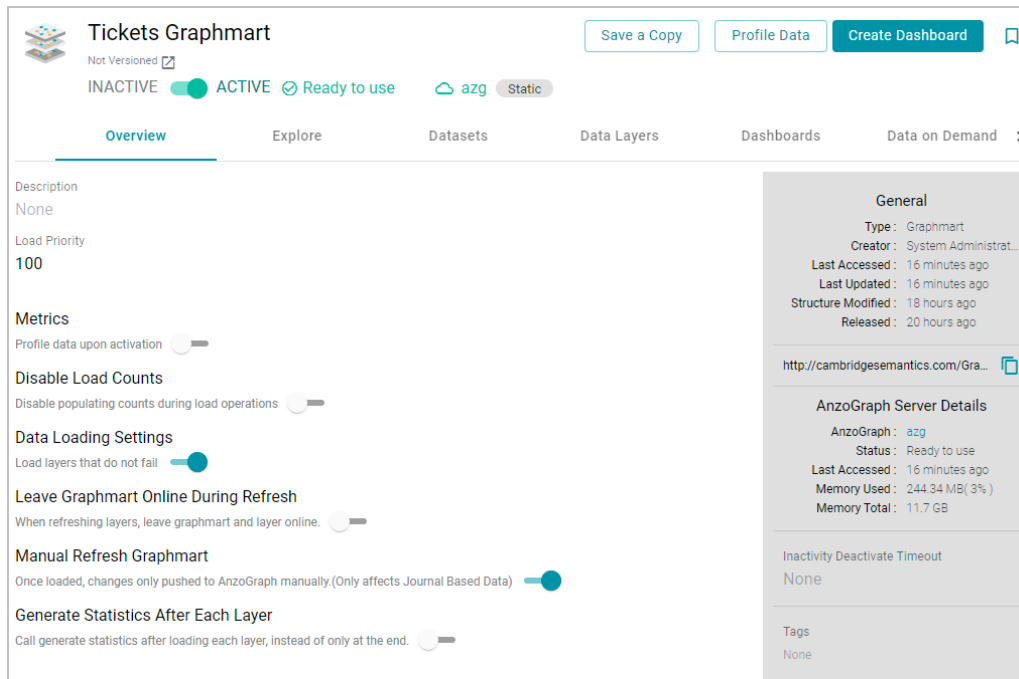
CANCEL

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 screen. For example, the image below shows a Saved Edition that has been modified. A copy of the Edition with the modifications was added to the Datasets list.

Datasets				
Title	↑	Edition	Description	Last Published Date
Tickets		Excludes Users (copy) (1)(Modified)		07/02/2020 03:27PM
				MODIFY EDITION

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 Graphmart Overview screen. For example:



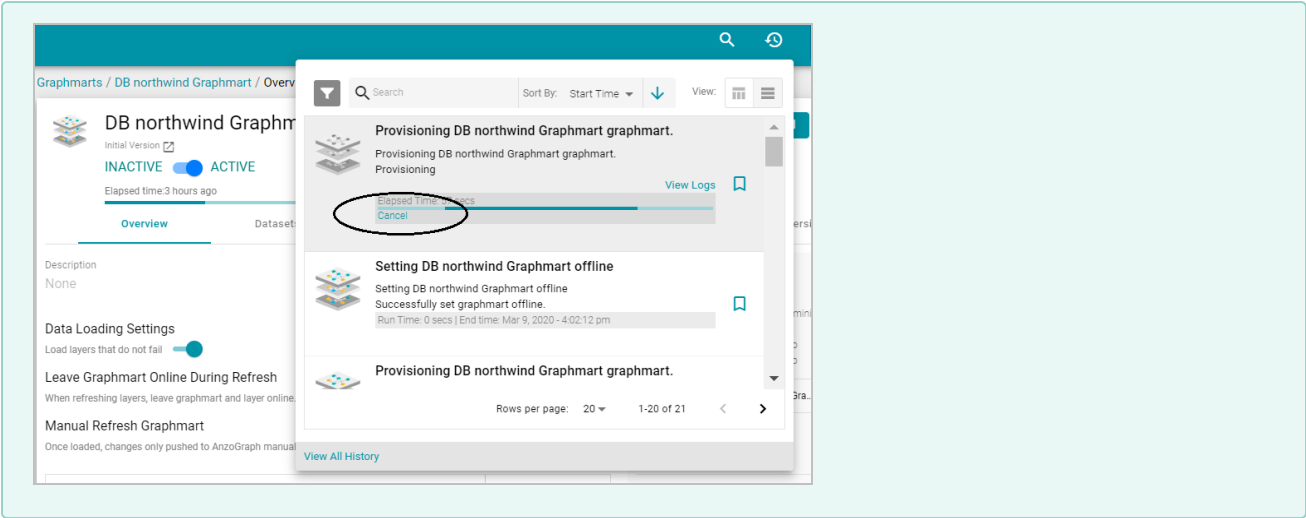
7. Modify any of the Data Load and Graphmart settings as needed. For details about the options, see [Graphmart Settings Reference](#).
8. You can add any number of Data Layers to enhance the data in the Graphmart. For information, see [Adding Data Layers to Graphmarts](#).
9. If the Graphmart is not yet loaded to AnzoGraph, slide the slider at the top of the screen from Inactive to **Active** to start loading the data into memory and executing the Data Layer steps.

Note

If you 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**.

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 activated, the data is available to access and analyze. For more information, see [Accessing and Analyzing Data](#).

Related Topics

- [Graphmart Settings Reference](#)
- [Adding a Dataset to a Graphmart](#)
- [Adding Data Layers to Graphmarts](#)
- [Copying a Graphmart](#)
- [Accessing and Analyzing Data](#)

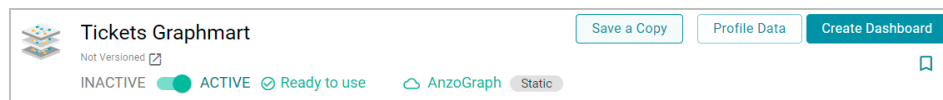
Copying a Graphmart

Follow the instructions below if you want to create a Graphmart by making a copy of an existing Graphmart and its Data Layers. For instructions on creating a new Graphmart from scratch, see [Creating a New Graphmart](#).

1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing Graphmarts. For example:

<div><div><div></div><div>Search</div></div><div>Sort By: Title</div><div>View: <div><div></div><div></div></div></div><div>Add Graphmart</div></div>									
<input type="checkbox"/>	Title	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions	
<input type="checkbox"/>	DB - northwind to Store Graphmart	System Admi...	Ready to us	9,172	2 hours ago	2 hours ago			
<input type="checkbox"/>	Datafox Graphmart	System Admi...							
<input type="checkbox"/>	Flight Data Graphmart	System Admi...	Ready to us	265,696	2 minutes ago	2 minutes ago			
<input type="checkbox"/>	Tickets Graphmart	System Admi...	Ready to us	4,088,422	5 hours ago	5 hours ago			

2. On the Graphmarts screen, click the name of the Graphmart that you want to copy. Anzo displays the Graphmart Overview. At the top of the screen, click the Save a Copy button.



The Clone dialog box is displayed. For example:

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. By default, the **Would you like to clone the permissions?** option is enabled, meaning the access control settings from the existing Graphmart will be applied to the new Graphmart. If you do not want to clone the permissions from the existing Graphmart, clear the checkbox.
5. 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 New Graphmart](#)

[Sharing Access to Graphmarts](#)

[Creating Data on Demand Endpoints](#)

[Adding Data Layers to Graphmarts](#)

Graphmart Settings Reference

This topic describes the configuration settings that are available on the Overview tab for each Graphmart.

- [Load Priority](#)
- [Profile Data Upon Activation](#)
- [Disable Populating Counts During Load Operations](#)
- [Load Layers that Do Not Fail](#)
- [Leave Graphmart Online During Refresh](#)
- [Manual Refresh Graphmart](#)

- [Generate Statistics After Each Layer](#)
- [Require Elasticsearch](#)

Load Priority

If you want Anzo to prioritize the order in which Graphmarts are activated when reconnecting to AnzoGraph or resetting and reloading the AnzoGraph server, 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 **100**.

Profile Data Upon Activation

This setting is disabled by default and controls whether a Graphmart Data Profile is automatically generated each time the Graphmart is activated. For information about data profiles, see [Generating a Graphmart Data Profile](#).

Disable Populating Counts During Load Operations

This setting is disabled by default controls whether Anzo periodically sends `select (count(*) as ?count) . . .` 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.

Load Layers that Do Not Fail

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 Data Layers that succeed and skip any Layers that fail. When disabled, the entire Graphmart activation is aborted if any Layer fails.

Leave Graphmart Online During Refresh

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 **Refresh** button to refresh a Graphmart (or the Refresh icon on a Data Layer), Anzo copies the existing Data 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.

Note

This setting applies only to **Refresh** operations. If **Leave Graphmart Online During Refresh** is enabled and a user clicks **Reload**, the Data Layers will not remain online. During reloads all of the data is dropped and then loaded again.

Manual Refresh Graphmart

This setting is enabled by default and controls whether changes to a data set 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 Data 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 data set are automatically loaded to AnzoGraph without requiring a manual refresh.

Generate Statistics After Each Layer

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 Data 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.

Require Elasticsearch

This setting is disabled by default. If you plan to include Unstructured Datasets in this Graphmart or configure Data Layers to incorporate Graph Data Interface queries against Elasticsearch, you can enable this option to ensure that Anzo validates the connection to Elasticsearch whenever this Graphmart is activated, reloaded, or refreshed.

Related Topics

[Creating a New Graphmart](#)

[Copying a Graphmart](#)

[Sharing Access to Graphmarts](#)

Generating a Graphmart Data Profile

Similar to generating a profile for a Dataset (see [Generating a Dataset Data Profile](#)), Anzo provides the ability to 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 Data 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 profile, AnzoGraph must be online. If you have dynamic AnzoGraph deployments enabled, Anzo will provision AnzoGraph 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

Sort By:

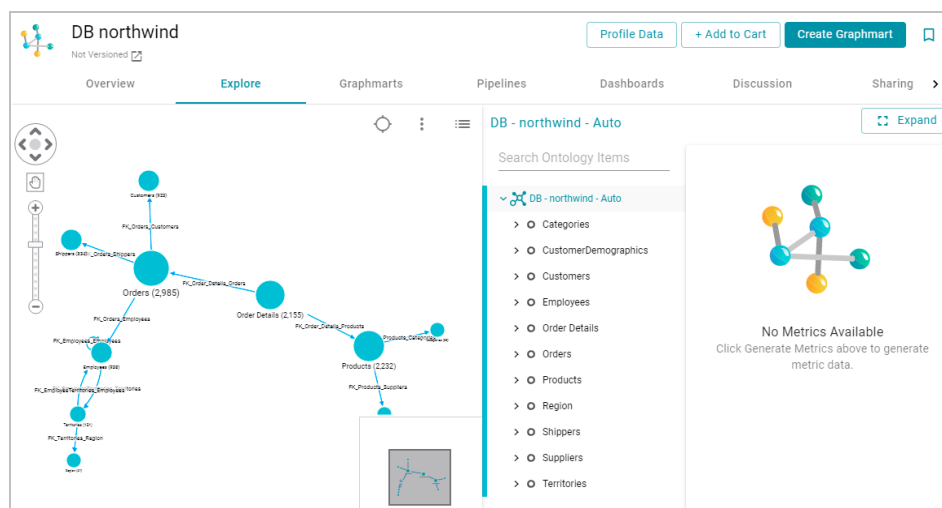
Title

View:

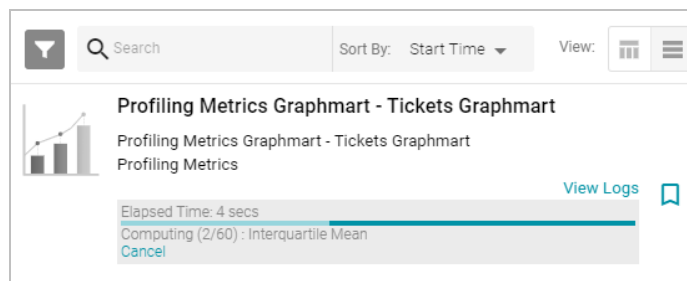
Add Graphmart

	Title	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions
	DB - northwind to Store Graphmart	System Admi...	<div><div></div>Ready to us</div>	9,172	2 hours ago	2 hours ago		<div><div></div><div></div></div>
	Datafox Graphmart	System Admi...	<div><div></div></div>					<div><div></div><div></div></div>
	Flight Data Graphmart	System Admi...	<div><div></div>Ready to us</div>	265,696	2 minutes ago	2 minutes ago		<div><div></div><div></div></div>
	Tickets Graphmart	System Admi...	<div><div></div>Ready to us</div>	4,088,422	5 hours ago	5 hours ago		<div><div></div><div></div></div>

- On the Graphmarts screen, click the name of the Graphmart for which you want to generate metrics. Anzo displays the Overview screen for the Graphmart. For example:

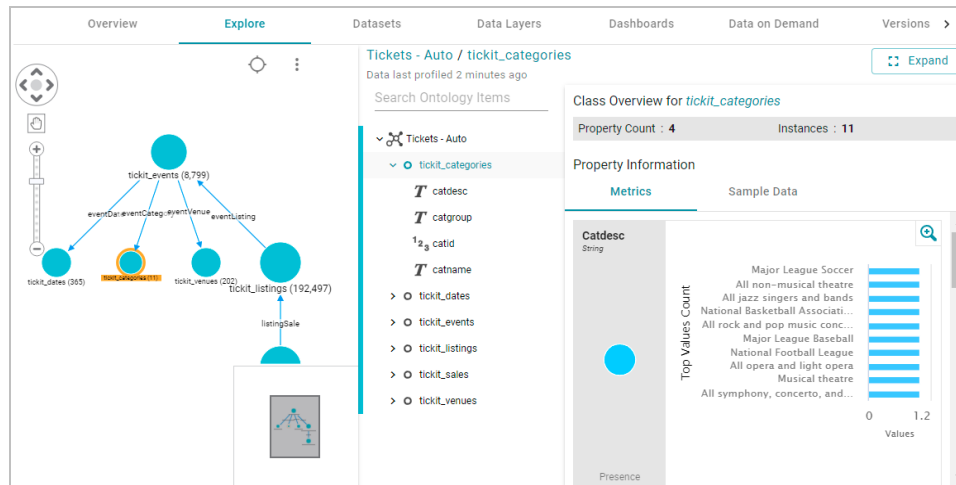


- If necessary, activate the Graphmart so that the **Profile Data** button becomes active, and then click **Profile Data**. AnzoGraph computes the data profiling metrics. The process may take several minutes. You can check the status of the process in the Activity Log. The Activity Log also presents the option to stop the profiling process by clicking **Cancel** under the progress bar for the task. For example:



Once the metrics are generated, the profile becomes available to explore in Anzo.

- To display the metrics, click the **Explore** tab. Then select a node or edge in the graph view on the left side of the screen or expand the model in the middle of the screen and select a class or property. For example:



You can click the **Expand** button on the right side of the screen to collapse the graph view and expand the metrics view.

Select any class or property to view its metrics. When a class is selected, Anzo displays the number of properties and total number of instances as well as one or more of the following metrics for each property in the class. The metrics that are presented depend on the data type of the properties that are in the class.

- **Percentile Metric:** This metric presents the data distribution for a property in percentiles.
- **Top Value Counts Metric:** This metric displays the count (as a histogram) of the 10 most frequently occurring values for a property.
- **Presence Metric:** This metric displays the number of values present and not present for a property.

When a property is selected, Anzo displays the metrics described above and one or more of the following metrics, depending on the data type of the property:

- **Extrema Metric:** Shows the smallest and largest values.
- **Geometric Mean Metric:** Shows the geometric mean of numeric values.
- **Mean Metric:** Shows the arithmetic mean of numeric values.
- **Median Metric:** Shows the arithmetic median of numeric values.
- **Mode Metric:** Shows the value that appears most often.
- **Std Deviation Metric:** Shows the standard deviation in the set of values.
- **Unique Values Metric:** Shows the number of unique values.

Related Topics

[Accessing and Analyzing Data](#)

Adding a Dataset to a Graphmart

This topic provides instructions for adding a new Dataset to an existing Graphmart from the Datasets tab for the Graphmart.

Tip

You can also add a Dataset by creating a Data Layer Step that loads the data. For more information, see [Configuring a Load Data Step](#).

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	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions
<input type="checkbox"/>	DB - northwind to Store Graphmart	System Admi...	Ready to us	9,172	2 hours ago	2 hours ago		
<input type="checkbox"/>	Datafox Graphmart	System Admi...						
<input type="checkbox"/>	Flight Data Graphmart	System Admi...	Ready to us	265,696	2 minutes ago	2 minutes ago		
<input type="checkbox"/>	Tickets Graphmart	System Admi...	Ready to us	4,088,422	5 hours ago	5 hours ago		

2. On the Graphmarts screen, click the name of the Graphmart that you want to add data to. Anzo displays the Graphmart Overview.
3. Click the **Datasets** tab. The screen lists the Datasets in the Graphmart. For example:

Title	Edition	Description	Last Published Date	Actions
DB - emrdb with emrdb to S...	Default Edition	Contains the latest successf...		MODIFY EDITION REMOVE

4. Click the **Add Dataset** button. Anzo opens the Select Datasets dialog box.

<input type="checkbox"/>	Title	Description	Updated Date	Tags	Actions
<input type="checkbox"/>	DB northwind		Jul 1, 2020		
<input type="checkbox"/>	Tickets		Jul 2, 2020		

Rows per page: 20 1-2 of 2

CANCEL OK

5. In the dialog box, select the checkbox next to the Dataset that you want to add, then click **OK**. Anzo adds the Dataset to the Graphmart and creates a new Data Layer with a Load Data Step that loads the Dataset.

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 **Modify Edition**. The Modify Edition dialog box is displayed. For example:

Modify Edition

Title: Default Edition
Description: Contains the latest successfully published data for all jobs

Job Data included in this Edition			
	Name	Published Date	Job Type
<input checked="" type="checkbox"/>	Load new-employ	05/13/2021 09...	Incremental

Data Component for selected job included in this Edition		
	Name	Published Date
<input type="checkbox"/>	Loadnewemployees_newemployees	05/13/2021 09:30AM
<input checked="" type="checkbox"/>	Loadnewemployees_newemployees	05/06/2021 05:53PM

Rows per page: 25 1-1 of 1

1 Items Selected

Rows per page: 25 1-2 of 2

1 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 Job checkboxes on the left side of the screen. Each time you select a Job checkbox, 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.

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 data set 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**.

Create New Edition



Title *


Description

CANCEL 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. For example, the image below shows a Saved Edition that has been modified. A copy of the edition with the modifications was added to the

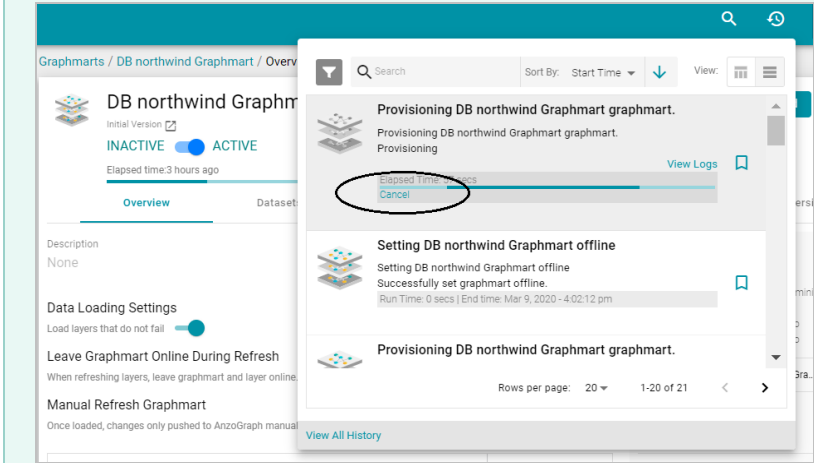
Member Datasets list.

Overview Explore Datasets Data Layers Dashboards Data on Demand >				
Member Datasets				
<div><div><div></div><div>Search</div></div><div>Sort By: Title</div><div>View: <div></div></div><div>Add Dataset</div></div>				
Title	↑	Edition	Description	Last Published Date Actions
DB - emrdb with emrdb to S...		Default Edition	Contains the latest successf...	MODIFY EDITION 
Tickets		Excludes Users (copy) (1)		MODIFY EDITION 

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 [Accessing and Analyzing Data](#).

Related Topics

- [Creating a New Graphmart](#)
- [Creating Data on Demand Endpoints](#)
- [Adding Data Layers to Graphmarts](#)

Working with Data Layers

The topics in this section introduce the concepts to know when working with Data Layers and provide instructions for creating, configuring, and adding Steps to Data Layers.

- [Introduction to Data Layers](#)
- [Adding Data Layers to Graphmarts](#)
- [Configuring Data Layers](#)
- [Adding Steps and Views to Data Layers](#)
- [Configuring Steps and Views](#)

Related Topics

[Sharing Access to Graphmarts](#)

Introduction to Data Layers

Data Layers enhance Graphmarts dynamically by loading additional Datasets, masking certain data, 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 Masking on a layer to hide sensitive information.
- You can configure Execution Conditions on Data 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.

For instructions on creating Data Layers, see [Adding Data Layers to Graphmarts](#).

What is a Step?

The Steps in a layer perform all of the data operations, such as loading a Dataset, 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:

- **Export Step:** Exports the Graphmart data in memory to a file-based linked data set (FLDS).
- **Load Data Step:** Loads a Dataset from the Dataset catalog into a Data Layer graph. When you create a Graphmart from the catalog, Anzo automatically creates a Data Layer with a Load Data 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.
- **View:** Enables advanced 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 and Views to Data Layers](#).

Related Topics

[Adding Data Layers to Graphmarts](#)

[Configuring Data Layers](#)

[Adding Steps and Views to Data Layers](#)

[Sharing Access to Graphmarts](#)

Adding Data Layers to Graphmarts

You can add a Data Layer to a Graphmart by creating a new layer from scratch or by copying a Data Layer from any Graphmart into another Graphmart. For instructions on adding layers to a Graphmart, select an option below:

- [Creating a New Data Layer](#)
- [Cloning an Existing Data Layer](#)

Related Topics

[Configuring Data Layers](#)

Creating a New Data Layer

Follow the steps below to create a new Data Layer in a Graphmart.

Tip 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:

<input type="checkbox"/>	Title	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions
	DB - northwind to Store Graphmart	System Admi...	Ready to use	9,172	2 hours ago	2 hours ago		
	Datafox Graphmart	System Admi...						
	Flight Data Graphmart	System Admi...	Ready to use	265,696	2 minutes ago	2 minutes ago		
	Tickets Graphmart	System Admi...	Ready to use	4,088,422	5 hours ago	5 hours ago		

2. Click the name of the Graphmart that you want to add a Data Layer to. The Overview tab is displayed. For example:

DB - northwind to Store Graphmart
Save a Copy
Profile Data
Create Dashboard

Not Versioned
INACTIVE
ACTIVE
Ready to use
azg
Static

Overview
Explore
Datasets
Data Layers
Dashboards
Data on Demand

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 ☒

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. ☐

General
Type: Graphmart
Creator: System Administrat...
Last Accessed: 21 minutes ago
Last Updated: 21 minutes ago
Structure Modified: May 10, 2021 12:28...
Released: May 10, 2021 11:43...

<http://cambridgesemantics.com/Gra...>

AnzoGraph Server Details
AnzoGraph: azg
Status: Ready to use
Last Accessed: 21 minutes ago
Memory Used: 245.03 MB (3%)
Memory Total: 11.7 GB

Inactivity Deactivate Timeout
None

Tags
None

3. Click the **Data Layers** tab. Anzo displays the existing data layers. For example:

Overview	Explore	Datasets	Data Layers	Dashboards	Data on Demand
RELOAD HIDE SIMPLE LOAD LAYERS EXPAND ALL ADD LAYER					
<div> 1. DB - northwind to Store ✓ </div> <div> 1.1 Load DB - northwind to Store ✓ </div>					

- Click **Add Layer** and select **New Layer**. Anzo displays the Create dialog box.

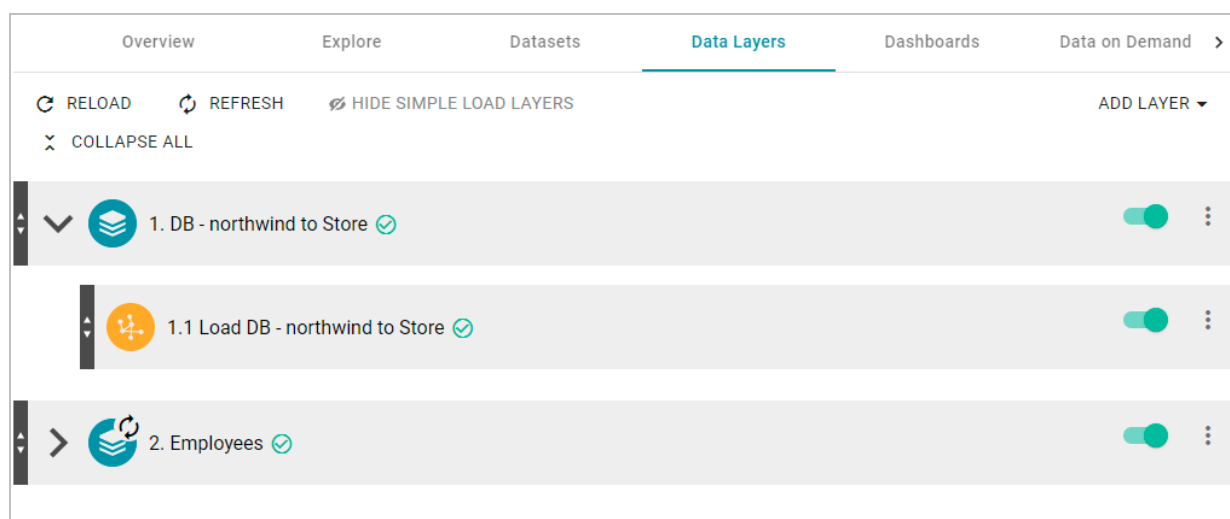
- Specify a name for the layer in the **Title** field and an optional description in the **Description** field.
- Determine how you want to control changes to this Data Layer's dependent Data Models:
 - If you want Anzo to automatically deploy to AnzoGraph any changes to the related models without having 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 [Creating a New Graphmart](#) for information about Graphmart settings.

- If you want Model changes to be deployed to AnzoGraph only when this Data Layer (or entire Graphmart) is manually refreshed or reloaded, leave the **Auto Deploy Ontology Changes** checkbox empty (disabled).
- Determine whether to **Disable Loading Counts** for this Data 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 during data loads.
 - 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. For example, the image below shows a new layer called Employees:



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.

Next, proceed to [Configuring Data Layers](#) if you want to configure options such as Execution Conditions and Masking for the Layer. See [Adding Steps and Views to Data Layers](#) instructions on creating Steps.

Related Topics

[Introduction to Data Layers](#)

[Configuring Data Layers](#)

[Adding Steps and Views to Data Layers](#)

[Sharing Access to Graphmarts](#)

Cloning an Existing Data Layer

Follow the steps below to create a new Data Layer by copying an existing one from any Graphmart.

Tip 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:

Search

Sort By: Title

View:

Add Graphmart

	Title	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions
	DB - northwind to Store Graphmart	System Admi...	Ready to us	9,172	2 hours ago	2 hours ago		
	Datafox Graphmart	System Admi...						
	Flight Data Graphmart	System Admi...	Ready to us	265,696	2 minutes ago	2 minutes ago		
	Tickets Graphmart	System Admi...	Ready to us	4,088,422	5 hours ago	5 hours ago		

2. Click the name of the Graphmart that you want to add a Data Layer to. The Overview tab is displayed. For example:

DB - northwind to Store Graphmart

Not Versioned

INACTIVE

ACTIVE

Ready to use

azg

Static

Save a Copy

Profile Data

Create Dashboard

Overview

Explore

Datasets

Data Layers

Dashboards

Data on Demand

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

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.

General

Type: Graphmart

Creator: System Administrat...

Last Accessed: 21 minutes ago

Last Updated: 21 minutes ago

Structure Modified: May 10, 2021 12:28...

Released: May 10, 2021 11:43...

http://cambridgesemantics.com/Gra...

AnzoGraph Server Details

AnzoGraph: azg

Status: Ready to use

Last Accessed: 21 minutes ago

Memory Used: 245.03 MB(3%)

Memory Total: 11.7 GB

Inactivity Deactivate Timeout

None

Tags

None

3. Click the **Data Layers** tab. Anzo displays the existing data layers. For example:

Overview

Explore

Datasets

Data Layers

Dashboards

Data on Demand

RELOAD

HIDE SIMPLE LOAD LAYERS

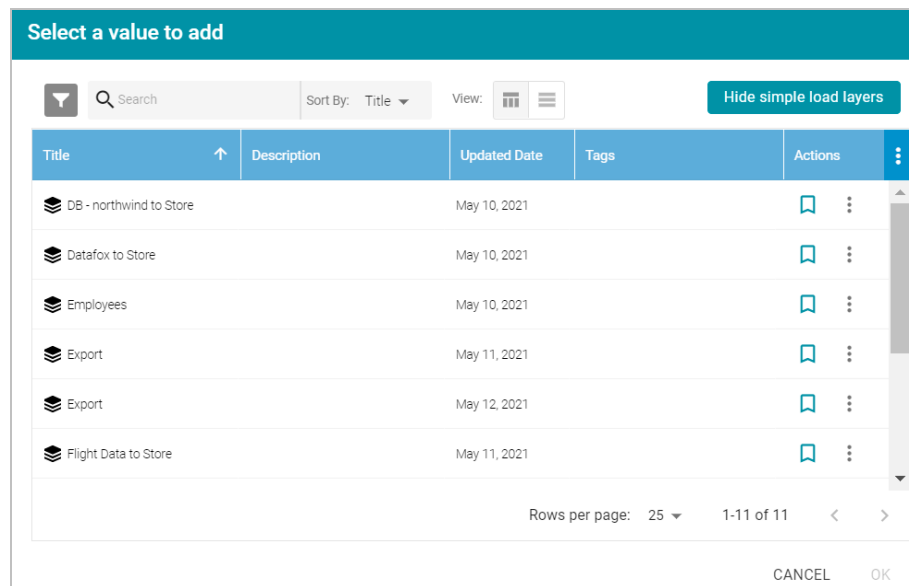
EXPAND ALL

ADD LAYER

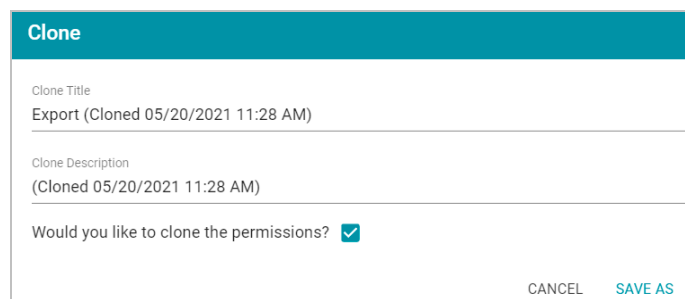
1. DB - northwind to Store

1.1 Load DB - northwind to Store

4. click **Add Layer** and select **Add Existing**. Anzo opens the Select a value to add dialog box, which lists the existing layers for all Graphmarts. For example:

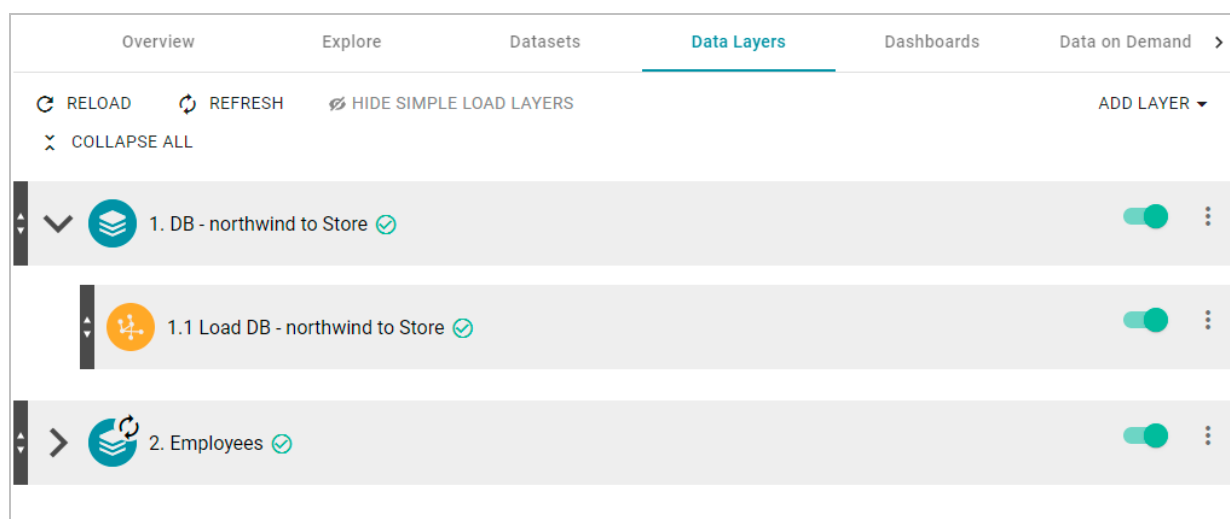


5. Select the layer that you want to copy and click **OK**. The Clone dialog box is displayed. For example:



6. In the Clone dialog box, you have the option to edit the **Clone Title** and/or **Clone Description**.
7. By default, the cloned version of the layer is set to copy the permissions from the original layer. If you would not like the cloned version to inherit the same permissions, clear the **Would you like to clone the permissions** checkbox.
8. 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. For example, the image below shows a new layer called Employees:



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.

Next, proceed to [Configuring Data Layers](#) if you want to configure options such as Execution Conditions and Masking for the Layer. See [Adding Steps and Views to Data Layers](#) instructions on creating Steps.

Related Topics

[Introduction to Data Layers](#)

[Configuring Data Layers](#)

[Adding Steps and Views to Data Layers](#)

[Sharing Access to Graphmarts](#)

Configuring Data Layers

The topics in this section provide information about configuring Data Layers, such as to set up Contexts for Graph Data Interface connections, configuring Data Access, defining Execution Conditions, and Masking certain predicates.

- [Configuring Contexts for Data Source Connections](#)
- [Defining Execution Conditions](#)
- [Configuring Advanced Data Access Settings](#)
- [Masking Data in a Data Layer](#)

Related Topics

[Configuring Steps and Views](#)

Configuring Contexts for Data Source Connections

When you use the Graph Data Interface (GDI) for Data Virtualization, you may connect to Data Sources that require input of sensitive connection and authorization information such as keys, tokens, and user credentials. When configuring a Data Layer (or Step), you have the option to use a Context to store the sensitive information as key-value pairs. Queries can then reference the keys from the Context so that the sensitive details are abstracted from any requests that are sent to the Data Source. This topic provides information on configuring Contexts and referring to Context keys in a query.

- [Configuring a Context](#)
- [Referencing Context Keys in a Query](#)

Configuring a Context

Contexts are configured from the **Context** tab that is available when creating or editing a Data Layer or Step. The image below shows the Context tab for a Data Layer.

The screenshot shows the 'Context' configuration interface. At the top, there are five tabs: 'Details', 'Execution Condition', 'Data Access', 'Masking', and 'Context' (which is selected). Below the tabs, the 'Context Providers' section contains a dropdown menu with 'Context Providers' selected. The 'Context Keys' section has an 'Add Key' button and a table with columns 'Name' and 'Value'. The table is empty, with the text 'There are no MANUAL keys created yet.' displayed below the header. At the bottom right, there are 'CANCEL' and 'SAVE' buttons.

Context Providers

Connections in Anzo implement the **Context Provider** interface. For example, File Store connections, Anzo Data Store connections, and Data Source connections provide contexts (in the form of JSON objects) that contain key-value pairs which define connection details such as URLs, database names, usernames and passwords, tokens, etc. These contexts are 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.

Note

If you specify a Context Provider, the key-value pairs from the selected provider are not populated in the Context Key list on the screen. However, the keys are used automatically when a query is run against that provider.

Context Keys

Context Keys 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

When defining Context Keys at the Data Layer level, make sure that the Key Names are unique. A Data Layer may have different Steps that virtualize data from different sources and reference the same Context. The image below, for example, creates URL, username, and password Context Keys.

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 Keys in a Query

The format that you use for referencing a Context Key in a query depends on the type of AnzoGraph plugin or extension that is being called by the query. Generally, Data Layer Contexts are only used in Steps that contain Graph Data Interface (GDI) queries. When referencing Context Keys in GDI queries, use the following format:

```
{{@<context_key_name>}}
```

For example, the following GDI query references three Context Keys: `adventureworks-url`, `adventureworks-username`, and `adventureworks-password`.

```
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 DISTINCT *
WHERE
{
  SERVICE <http://cambridgesemantics.com/services/DataToolkit>
  {
    ?data a s:DbSource ;
      s:url "{{@adventureworks-url}}" ;
      s:username "{{@adventureworks-username}}" ;
      s:password "{{@adventureworks-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 Data Layers to Graphmarts](#)

[Adding Steps and Views to Data Layers](#)

[Configuring Steps and Views](#)

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 [Configuring a 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.

Details **Execution Condition** Data Access Masking

☐ Enable layer based on boolean condition

Select condition variable | v

Conditional Variable

Execute only if returned value is | v

Conditional Variable If Result

CANCEL 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 the [Options](#) tab section in [Configuring a 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

[Adding Data Layers to Graphmarts](#)

[Adding Steps and Views to Data Layers](#)

[Configuring Steps and Views](#)

Configuring 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:

Edit

Details Execution Condition **Data Access** Masking Context

View Permissions

☒ Inherit from Load Step's Dataset (DB - northwind to Store)

☐ 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

CANCEL SAVE

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](#)

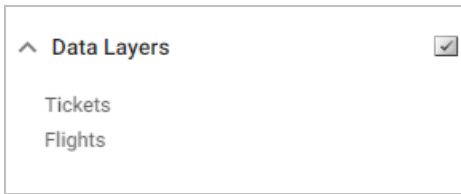
Display data outputted by the layer in Hi-Res Analytics dashboards

This setting controls whether the data generated by the steps is available to query and display in Hi-Res Analytics:

- When the setting is **enabled** (the default value), the layer's data is available to Hi-Res Analytics.
- When the setting is **disabled**, other data layers in the graphmart can use the layer's data, but the data is not available to Hi-Res Analytics.

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 Hi-Res Analytics.
- When the setting is **disabled**, the layer's data is always used in Hi-Res Analytics 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 Hi-Res Analytics dashboards:

- When the setting is **enabled** (the default value), the layer is listed in the Data Layers panel in Hi-Res Analytics 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 list 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 generated by the steps in the layer is visible in Hi-Res Analytics:

- When the setting is **enabled** (the default value), the layer is listed in the Data Layers panel in Hi-Res Analytics and is selected by default.
- 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.

Hide from Other Layers

This setting controls whether the other Data 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.

Related Topics

[Configuring Data Layers](#)

Masking Data in a Data Layer

Follow the instructions below if you want to hide or mask sensitive information by selecting properties or predicates to hide in a Data Layer. When you mask predicates at the Data Layer level, Anzo does load the triples associated with those predicates so that other steps and layers in the Graphmart can use that data in calculations. However, those masked triples are excluded from Hi-Res Analytics.

Note

Masking can only be configured for Data Layers that include at least one Load Data Step or Query Step. You cannot configure masking on layers that contain only View or RDFS+ Inference steps. For information about adding steps to layers, see [Adding Steps and Views to Data Layers](#).

- 1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing Graphmarts. For example:

	Title	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions
	DB - northwind to Store Graphmart	System Admi...	Ready to us	9,172	2 hours ago	2 hours ago		
	Datafox Graphmart	System Admi...						
	Flight Data Graphmart	System Admi...	Ready to us	265,696	2 minutes ago	2 minutes ago		
	Tickets Graphmart	System Admi...	Ready to us	4,088,422	5 hours ago	5 hours ago		

- 2. Click the name of the Graphmart that you want to change.
- 3. Click the **Data Layers** tab. Anzo displays the existing data layers. For example:

Overview	Explore	Datasets	Data Layers	Dashboards	Data on Demand
RELOAD	HIDE SIMPLE LOAD LAYERS	EXPAND ALL			ADD LAYER
	1. DB - northwind to Store				
	1.1 Load DB - northwind to Store				

- 4. On the layer for which you want to mask data, click the menu icon (⋮) and select **Edit Layer**. The Edit dialog box is displayed. For example:

Edit

Details

Execution Condition

Data Access

Masking

>

Title *

DB - northwind to Store

The title of the layer

Description

A brief description of the layer

☐ Auto Deploy Ontology Changes

☐ Disable Loading Counts

URI : <http://cambridgesemantics.com/Layer/22e9624f25a3404285a56319cecb0b6e>

CANCEL

SAVE

5. Click the **Masking** tab:

Details

Execution Condition

Data Access

Masking

Context

>

Masked Predicate

CANCEL

SAVE

6. On the Masking screen, click the **Masked Predicate** drop-down list. The list includes the predicates or properties from the Models selected in the Data Layer's steps. You can also start typing in the field to find properties by name. Select a property to add it to the Masked Predicate field. Repeat this step to mask additional properties. You can remove a property from the masked list by clicking the X to the right of the property name.

For example, the following image shows a Data Layer that masks API key and AWS Access Key values. When users view Hi-Res Analytics dashboards that include this layer, the two properties are not available to select and display.

Details

Execution Condition

Data Access

Masking

>

Masked Predicate

API key

AWS Access Key

CANCEL

SAVE

7. Click **Save** to save the masking configuration and return to the Data Layers screen.



Related Topics

- [Configuring Data Layers](#)
- [Sharing Access to Graphmarts](#)

Adding Steps and Views to Data Layers

The Steps in a Data Layer perform the data operations, such as loading, creating, deleting, or changing data. You can add any number of steps to a Layer. For instructions on adding a Step to a Data Layer, select an option below:

- [Creating a New Step or View](#)
- [Cloning an Existing Step or View](#)

Related Topics

- [Configuring Steps and Views](#)

Creating a New Step or View

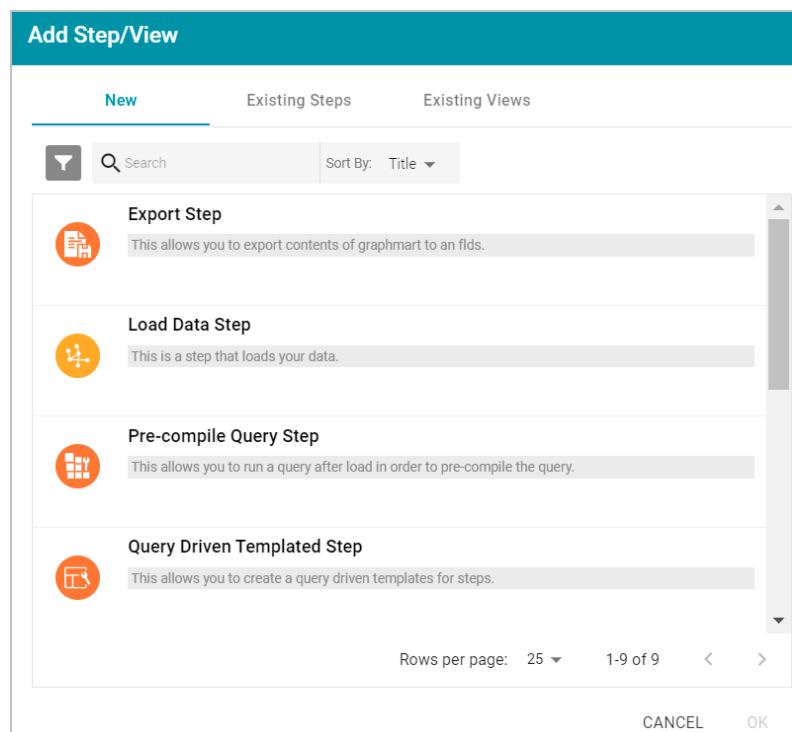
Follow the steps below to create a new Step in a Data Layer.

Tip For instructions on creating a new step by copying an existing one, see [Cloning an Existing Step or View](#).

1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing Graphmarts. For example:

<div><div><div></div></div><div>Search</div><div>Sort By: Title</div><div>View: <div><div></div><div></div></div></div><div>Add Graphmart</div></div>										
	Title	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions		
	DB - northwind to Store Graphmart	System Admi...	<div>Ready to us</div>	9,172	2 hours ago	2 hours ago		<div></div> <div></div>		
	Datafox Graphmart	System Admi...	<div></div>					<div></div> <div></div>		
	Flight Data Graphmart	System Admi...	<div>Ready to us</div>	265,696	2 minutes ago	2 minutes ago		<div></div> <div></div>		
	Tickets Graphmart	System Admi...	<div>Ready to us</div>	4,088,422	5 hours ago	5 hours ago		<div></div> <div></div>		

2. Click the name of the Graphmart that you want to add a Step to. The Overview tab for the Graphmart is displayed. Click the **Data Layers** tab.
3. Locate the Data Layer for which you want to add a Step or View, and then click the menu icon (⋮) for the layer and select **Add Step/View**. Anzo opens the Add Step/View dialog box.



4. Select the type of step that you want to add, and then click **OK**.

The Create dialog box for the chosen type of Step is presented. The configuration options differ depending on the type of Step you added. See [Configuring Steps and Views](#) for instructions on configuring the new Step or View.

Related Topics

[Cloning an Existing Step or View](#)

[Configuring Steps and Views](#)

Cloning an Existing Step or View

Follow the steps below to create a new Step or View by copying an existing one from any Graphmart.

Tip

You cannot clone and reuse existing Load Data Steps. For instructions on creating a Step from scratch, see [Creating a New Step or View](#).

1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing Graphmarts. For example:

Search

Sort By: Title

View:

Add Graphmart

<div></div>	Title	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions
	DB - northwind to Store Graphmart	System Admi...	<div>Ready to us</div>	9,172	2 hours ago	2 hours ago		<div></div> <div></div>
	Datafox Graphmart	System Admi...	<div></div>					<div></div> <div></div>
	Flight Data Graphmart	System Admi...	<div>Ready to us</div>	265,696	2 minutes ago	2 minutes ago		<div></div> <div></div>
	Tickets Graphmart	System Admi...	<div>Ready to us</div>	4,088,422	5 hours ago	5 hours ago		<div></div> <div></div>

2. Click the name of the Graphmart that you want to add a Step to. The Overview tab for the Graphmart is displayed. Click the **Data Layers** tab.
3. Locate the Data Layer for which you want to add a Step or View, and then click the menu icon (⋮) for the layer and select **Add Step/View**. Anzo opens the Add Step/View dialog box.

Add Step/View

NewExisting StepsExisting Views

Search

Sort By: Title

Export Step

This allows you to export contents of graphmart to an fids.

Load Data Step

This is a step that loads your data.

Pre-compile Query Step

This allows you to run a query after load in order to pre-compile the query.

Query Driven Templated Step

This allows you to create a query driven templates for steps.

Rows per page: 251-9 of 9

CANCEL

OK

4. If you want to clone an existing Step, click the **Existing Steps** tab, select the Step to clone, and click **OK**. To clone a View, click the **Existing Views** tab, select the View to clone, and click **OK**. Anzo displays the Clone dialog box. For example:

© 2023 Cambridge Semantics, Inc.

5. On the Clone dialog box, you have the option to edit the **Clone Title** and/or **Clone Description**.
6. By default, the clone is set to copy the permissions from the original step. If you would not like the cloned version to inherit the same permissions, clear the **Would you like to clone the permissions** checkbox.
7. Click **Save As** to add the cloned Step or View to the Data Layer.

To change the configuration of the copied Step or View, click the menu icon (⋮) on the Step or View and click **Edit Step**. The configuration options differ depending on the type of Step you added. See [Configuring Steps and Views](#) for instructions on configuring Steps.

Related Topics

[Creating a New Step or View](#)

[Configuring Steps and Views](#)

Configuring Steps and Views

The topics in this section provide information about configuring each type of Data Layer Step.

- [Configuring an Export Step](#)
- [Configuring a Load Data Step](#)
- [Configuring a Pre-Compile Query Step](#)
- [Configuring a Query-Driven Templated Step](#)
- [Configuring a Query Step](#)
- [Configuring an RDFS+ Inference Step](#)
- [Configuring a Templated Step](#)
- [Configuring a Validation Step](#)
- [Configuring a View Step](#)

Related Topics

[Adding Steps and Views to Data Layers](#)

[Configuring Data Layers](#)

Configuring an 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 file store. The sections below describe each of the tabs and configuration

options that are available when you create or edit 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 Data Layer after adding the step does not create the ontology graph that the Export Step requires.

- [Details](#)
- [Elastic Search](#)
- [Execution Condition](#)

Details

The **Details** tab includes all of the required settings that configure options such as the name of the step, the source data to act upon, and the target Dataset.

The screenshot shows the 'Details' tab of a configuration window. At the top, there are three tabs: 'Details' (selected), 'Elastic Search', and 'Execution Condition'. Below the tabs, there are several input fields: 'Title *' with a placeholder 'The title of the step', 'Description' with a placeholder 'A brief description of the step', an 'Enabled' checkbox which is checked, a 'Source' dropdown menu showing 'Self' and 'All Previous Layers Within Graphmart' (selected), and a 'Data models' dropdown menu. At the bottom right, there are 'CANCEL' and 'SAVE' buttons.

Title

The required name of the Step.

Description

And optional 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 Data Layer is loaded or refreshed. If you want to disable the Step so that it is not processed, clear the **Enabled** checkbox.

The required name of the step.

Source

The Source is the source data that this Step should act upon. Steps can build upon the data generated by Steps in other Data 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 Anzo exports only the data that is generated in the Layer to which this Step belongs.
- **All Previous Layers Within Graphmart:** Choosing this option means that Anzo exports data that is generated by all of the Data Layers in the Graphmart that precede this layer.

- **Previous Layer Within Graphmart:** Choosing this option means that Anzo exports only the data that is generated by the one Layer that precedes this Layer.
- **Layer Name:** The Source drop-down list also includes options for specific Layer names. You can choose a specific Layer to export only the data that is generated by that Layer.

Data Models

This optional field specifies the Data Model or Models to include in the export. If you do not select a Model, Anzo automatically exports any associated Models when the data is exported.

Target FLDS

This is the target Dataset for this export. If a Dataset 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 Dataset, 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 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.

- If you want Anzo to *copy* the binary store to the Target FLDS, leave **Always Move Binary Store** disabled (unchecked).
- If you want Anzo to *move* the binary store to the Target FLDS, select the **Always Move Binary Store** checkbox to enable it.

Keep Elasticsearch Index Online

This option applies to exports of Unstructured Graphmarts and controls whether the Elasticsearch index that is associated with the Dataset remains stored in Elasticsearch or is removed from Elasticsearch once it is exported.

Generate Metrics

This option controls whether a Data Profile is generated before the data is exported. Since the data must be loaded in AnzoGraph to compute the metrics, you have the option to generate them during the export. If you load the exported files in the future, the Data Profile is also loaded.

Elastic Search

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 >

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)

CANCEL SAVE

Elasticsearch Index Settings

A custom 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

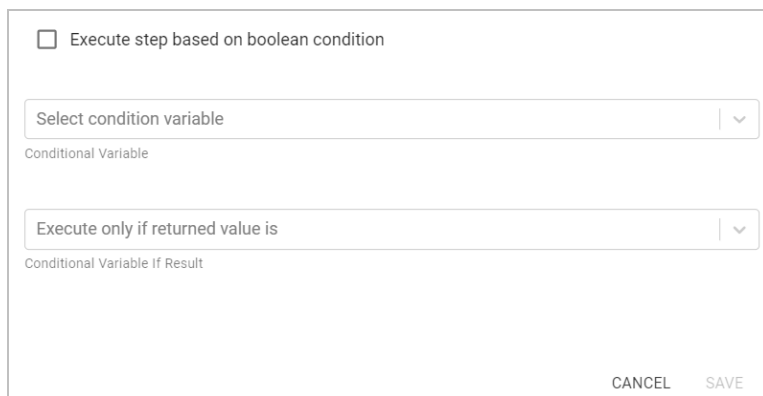
The maximum number of FLDSes to index concurrently.

Execution Condition

If you want this Step to be executed conditionally, based on the result of a specified Validation Condition, you can configure an Execution Condition on the **Execution Condition** tab that is available when creating or editing a Step. The image below shows the Execution Condition tab.

Note

In order to set up an Execution Condition, the Graphmart needs to have at least one Validation Step that defines a Condition Variable. Condition Variables can be used across all Data Layers in the Graphmart. For guidance on configuring a Validation Step, see [Configuring a Validation Step](#).



The screenshot shows a configuration dialog box for an Execution Condition. It contains the following elements:

- A checkbox labeled "Execute step based on boolean condition".
- A dropdown menu labeled "Select condition variable" with a downward arrow. Below it, the text "Conditional Variable" is displayed.
- A second dropdown menu labeled "Execute only if returned value is" with a downward arrow. Below it, the text "Conditional Variable If Result" is displayed.
- At the bottom right, there are two buttons: "CANCEL" and "SAVE".

Enable Layer Based on Boolean Condition

This setting indicates whether to enable this Step 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 Step is disabled.

Conditional Variable

This field specifies the variable that you want to base this Execution Condition on. The variable is the result of a Validation Step Query in the Graphmart.

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 Step 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.

If you added the Export Step to an active Graphmart, make sure that you reload the entire Graphmart.

Related Topics

[Adding Steps and Views to Data Layers](#)

Configuring a Load Data Step

This topic provides guidance on configuring a Load Data Step to use for adding a Dataset to a Graphmart. The sections below describe each of the tabs and configuration options that are available when you create or edit a Load Data Step.

Note The Dataset to load must be a system dataset or be available in the Dataset catalog.

- [Details](#)
- [Execution Condition](#)
- [Filter](#)

Details

The **Details** tab includes all of the required settings that configure options such as the name of the step and the Dataset to load.

The screenshot shows the 'Details' configuration window for a step. It has four tabs: 'Details' (active), 'Execution Condition', 'Filter', and 'Context'. The 'Enabled' toggle is turned on. The 'Title *' field is required and has a placeholder 'The title of the step'. The 'Description' field is optional with a placeholder 'A brief description of the step'. The 'Linked Dataset *' is a dropdown menu with a toggle on the right. Below it are two checkboxes: 'Watch Flds Directory' and 'Skip Elastic Search Snapshot Restoration if Index Already Exists'. At the bottom right are 'CANCEL' and 'SAVE' buttons.

Title

The required name of the Step.

Description

And optional description of the Step.

Enabled

When creating a new Step, the Step is **Enabled** by default, indicating that the Step will run when the Data 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 that you want this Step 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** (×). The Linked Dataset drop-down list will display the system datasets in addition to the Datasets in the catalog.

If this is a new Load Data Step, the current working edition (Managed Edition) of the Dataset is selected. If you want to select a different Edition, you can click **Modify Edition** and follow the steps in [Modifying an Edition](#).

Watch FLDS Directory

If you want this Step to watch the FLDS directory on the file store and indicate when any of the load files change, select the **Watch FLDS Directory** checkbox. If **Watch FLDS Directory** is enabled and changes are detected in the FLDS directory, Anzo will mark this Step (and Data Layer) as needing a refresh.

Skip Elastic Search Snapshot Restoration if Index Already Exists

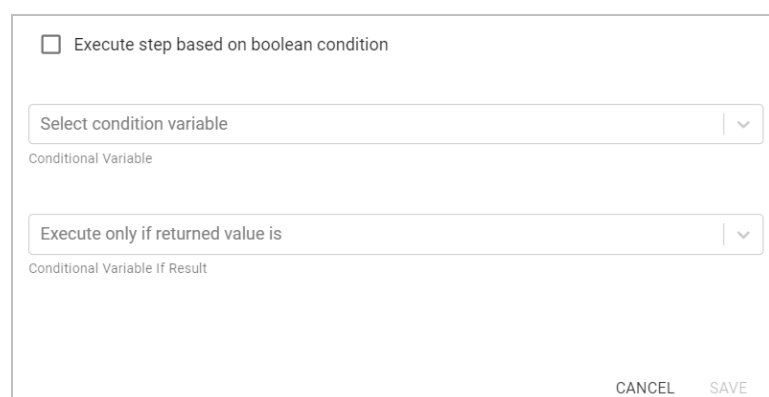
This option applies to loads of Unstructured Datasets and controls whether Anzo first checks to see if an index with the alias for the Dataset already exists in Elasticsearch. If it does exist, Anzo will not reload the index snapshot into Elasticsearch.

Execution Condition

If you want this Step to be executed conditionally, based on the result of a specified Validation Condition, you can configure an Execution Condition on the **Execution Condition** tab that is available when creating or editing a Step. The image below shows the Execution Condition tab.

Note

In order to set up an Execution Condition, the Graphmart needs to have at least one Validation Step that defines a Condition Variable. Condition Variables can be used across all Data Layers in the Graphmart. For guidance on configuring a Validation Step, see [Configuring a Validation Step](#).



The screenshot shows a configuration dialog box for an Execution Condition. It contains the following elements:

- A checkbox labeled "Execute step based on boolean condition".
- A dropdown menu labeled "Select condition variable" with a downward arrow. Below it, the text "Conditional Variable" is displayed.
- A dropdown menu labeled "Execute only if returned value is" with a downward arrow. Below it, the text "Conditional Variable If Result" is displayed.
- At the bottom right, there are two buttons: "CANCEL" and "SAVE".

Enable Layer Based on Boolean Condition

This setting indicates whether to enable this Step 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 Step is disabled.

Conditional Variable

This field specifies the variable that you want to base this Execution Condition on. The variable is the result of a Validation Step Query in the Graphmart.

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 Step 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.

Filter

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 Linked Dataset that you chose on the Details tab, 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 window. The 'Filter Type' section has two radio buttons: 'Multiple Select' (which is selected) and 'Query'. Below this is a 'Masked Predicate' dropdown menu. At the bottom right of the window are 'CANCEL' and 'SAVE' buttons.

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**, which can also be configured at the Data Layer level (as described in [Masking Data in a Data Layer](#)). 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. Repeat this step to mask 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#tickit_users_personid> ?id .
  FILTER (?id=1)
}
```

Related Topics

[Adding Steps and Views to Data Layers](#)

Configuring a 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.

The sections below describe each of the tabs and configuration options that are available when you create or edit a Pre-Compile Query Step.

- [Details](#)
- [Query](#)
- [Execution Condition](#)
- [Context](#)

Details

The **Details** tab includes options such as the name of the step, the source data to act upon, and conditions on what to do if the Step fails.

DetailsQueryExecution ConditionContext

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 xAll Previous Layers Within Graphmart x

x | v

Source data to act upon

☐ Pre-Run Generate Statistics

CANCEL

SAVE

Title

The required name of the Step.

Description

And optional 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 Data 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

Select this option if you want Anzo to abort the load of the Data Layer if this Step fails. The Graphmart and other successful Data Layers continue to load.

If the precompile 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.

Tip

If you want Anzo to load the Data Layer if this Step fails, leave both **If the precompile query fails...** settings blank.

Source

The Source is the source data that this Step should act upon. Steps can build upon the data generated by Steps in other Data 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 Query 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 Query runs against the data that is generated by all of the successful Layers in the Graphmart 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 run the Query against 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.

Query

The **Query** tab includes the query that you want to be pre-compiled.

Details **Query** Execution Condition Context >

Precompile query *

```
1
2 #fromSources is replaced with the URIs of the Layer's Sources at runtime
3 SELECT *
4 ${fromSources}
5 WHERE{ ?s ?p ?o
6 }
```

Query used to pre-compile

CANCEL SAVE

The tab provides the syntax for writing a SPARQL SELECT query. You can edit the template to write the query, or you paste in query contents that you copied from a log file, dashboard, the Query Builder, etc.

Note

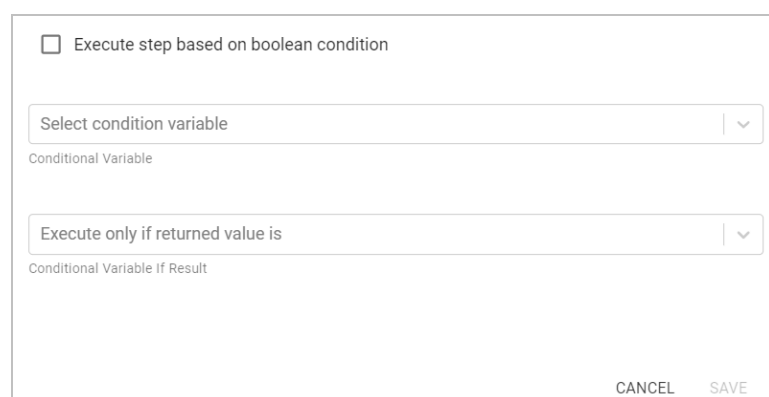
Make sure that you include the `${fromSources}` parameter in the query. Anzo automatically populates the query with the appropriate source graph URIs according to the Source configured from the Details tab.

Execution Condition

If you want this Step to be executed conditionally, based on the result of a specified Validation Condition, you can configure an Execution Condition on the **Execution Condition** tab that is available when creating or editing a Step. The image below shows the Execution Condition tab.

Note

In order to set up an Execution Condition, the Graphmart needs to have at least one Validation Step that defines a Condition Variable. Condition Variables can be used across all Data Layers in the Graphmart. For guidance on configuring a Validation Step, see [Configuring a Validation Step](#).



The screenshot shows a configuration dialog box for an Execution Condition. It contains the following elements:

- A checkbox labeled "Execute step based on boolean condition".
- A dropdown menu labeled "Select condition variable" with a downward arrow. Below it, the text "Conditional Variable" is displayed.
- A dropdown menu labeled "Execute only if returned value is" with a downward arrow. Below it, the text "Conditional Variable If Result" is displayed.
- At the bottom right, there are two buttons: "CANCEL" and "SAVE".

Enable Layer Based on Boolean Condition

This setting indicates whether to enable this Step 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 Step is disabled.

Conditional Variable

This field specifies the variable that you want to base this Execution Condition on. The variable is the result of a Validation Step Query in the Graphmart.

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 Step 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.

Context

When you use the Graph Data Interface (GDI) for Data Virtualization, you may connect to Data Sources that require input of sensitive connection and authorization information such as keys, tokens, and user credentials. The **Context** tab gives you the option to configure a Context to store the sensitive information as key-value pairs. Queries can then reference the keys from the Context so that the sensitive details are abstracted from any requests that are sent to the Data Source.

The screenshot shows a configuration window with two main sections. The top section, titled 'Context Providers', contains a dropdown menu currently showing 'Context Providers'. The bottom section, titled 'Context Keys', features an 'Add Key' button and a table with columns 'Name' and 'Value'. The table is currently empty, displaying the message 'There are no MANUAL keys created yet.' At the bottom right of the window are 'CANCEL' and 'SAVE' buttons.

Context Providers

Connections in Anzo implement the **Context Provider** interface. For example, File Store connections, Anzo Data Store connections, and Data Source connections provide contexts (in the form of JSON objects) that contain key-value pairs which define connection details such as URLs, database names, usernames and passwords, tokens, etc. These contexts are 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.

Note

If you specify a Context Provider, the key-value pairs from the selected provider are not populated in the Context Key list on the screen. However, the keys are used automatically when a query is run against that provider.

Context Keys

Context Keys 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

The image below, for example, creates URL, username, and password Context Keys.

Context Keys		Add Key	
Name	Value		
mysql-password	admin123		
mysql-username	sysadmin		
mysql-northwind-url	jdbc:mysql://10.111.4.9:3306/NORTHWIND		

Note

The format that you use for referencing a Context Key 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 Graph Data Interface (GDI) queries. When referencing Context Keys in GDI queries, use the following format:

```
{{@<context_key_name>}}
```

Related Topics

[Adding Steps and Views to Data Layers](#)

[SPARQL Query Templates and Best Practices](#)

Configuring 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, rather than rewriting the query, 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 must be 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.

Tip

For information about Templated steps with manually created key-value pairs, see [Configuring a Templated Step](#).

The sections below describe each of the tabs and configuration options that are available when you create or edit a Query-Driven Templated Step.

- [Details](#)
- [Parameters Query](#)
- [Template](#)
- [Execution Condition](#)
- [Context](#)

Details

The **Details** tab includes options such as the name of the step, the source data to act upon, and the Data Model to create the template against.

The screenshot shows the 'Details' tab of a configuration window. It contains the following elements:

- Tab Headers:** Details (active), Parameters Query, Template, Execution Condition, and a right arrow.
- Title ***: A text input field with the placeholder 'The title of the step'.
- Description**: A text input field with the placeholder 'A brief description of the step'.
- Enabled**: A checked checkbox.
- Source**: A dropdown menu showing 'Self' and 'All Previous Layers Within Graphmart' as options.
- Data models ***: A dropdown menu for selecting associated data models.
- Pre-Run Generate Statistics**: An unchecked checkbox.
- Buttons:** CANCEL and SAVE buttons at the bottom right.

Title

The required name of the Step.

Description

And optional 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 Data 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 is the source data that this Step should act upon. Steps can build upon the data generated by Steps in other Data 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 Query 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 Query runs against the data that is generated by all of the successful Layers in the Graphmart 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 run the Query against that Layer only.

Data Models

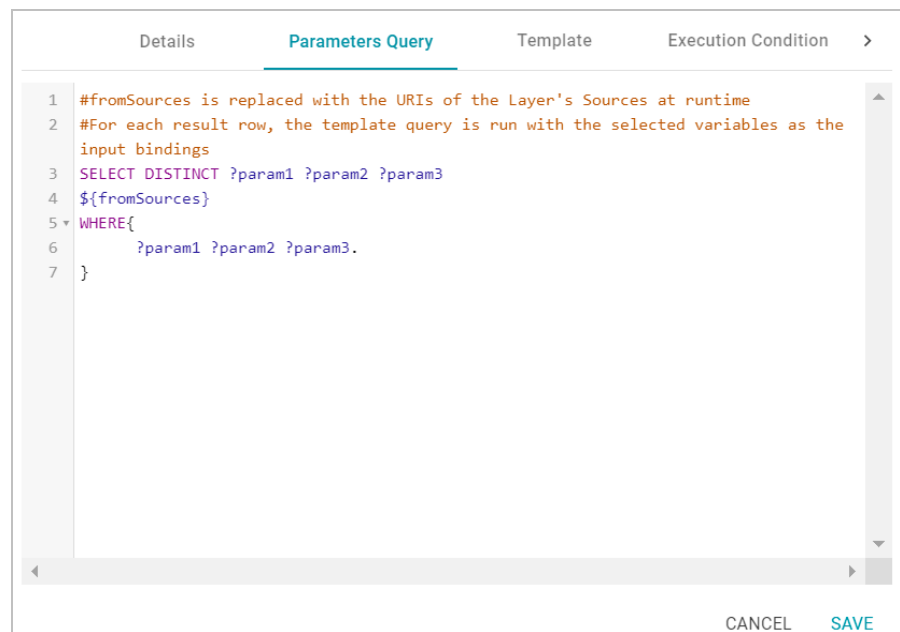
This required field specifies the Data Model or Models that you want to create this query template against. The list displays all of the Models for all of the Datasets in the Dataset catalog. By default, the field is set to **Exclude System Data** (+). If you want to choose a system Model, click the toggle button on the right side of the field to change it to **Include System Data** (x). The Data Models drop-down list will display the system Models in addition to the Dataset 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.

Parameters Query

The **Parameters Query** tab contains the query to use for determining the key-value pairs for the Source that was selected on the Details tab.



The tab provides the syntax for writing a SPARQL SELECT DISTINCT query. You can 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 from the Details tab.

Template

The **Template** tab contains the query to run for each of the key-value pairs identified by the Parameters Query.

```

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 }
  
```

CANCEL SAVE

The template includes the syntax for writing SPARQL DELETE and INSERT queries and includes source and target graph parameters as well as the 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}`.

Anzo replaces the parameters at runtime. The query is executed n times, where n is the number of rows returned by the Parameters Query.

Execution Condition

If you want this Step to be executed conditionally, based on the result of a specified Validation Condition, you can configure an Execution Condition on the **Execution Condition** tab that is available when creating or editing a Step. The image below shows the Execution Condition tab.

Note

In order to set up an Execution Condition, the Graphmart needs to have at least one Validation Step that defines a Condition Variable. Condition Variables can be used across all Data Layers in the Graphmart. For guidance on configuring a Validation Step, see [Configuring a Validation Step](#).

The screenshot shows a configuration dialog box for an Execution Condition. It contains the following elements:

- A checkbox labeled "Execute step based on boolean condition".
- A dropdown menu labeled "Select condition variable" with a downward arrow. Below it, the text "Conditional Variable" is displayed.
- A dropdown menu labeled "Execute only if returned value is" with a downward arrow. Below it, the text "Conditional Variable If Result" is displayed.
- At the bottom right, there are two buttons: "CANCEL" and "SAVE".

Enable Layer Based on Boolean Condition

This setting indicates whether to enable this Step 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 Step is disabled.

Conditional Variable

This field specifies the variable that you want to base this Execution Condition on. The variable is the result of a Validation Step Query in the Graphmart.

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 Step 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.

Context

When you use the Graph Data Interface (GDI) for Data Virtualization, you may connect to Data Sources that require input of sensitive connection and authorization information such as keys, tokens, and user credentials. The **Context** tab gives you the option to configure a Context to store the sensitive information as key-value pairs. Queries can then reference the keys from the Context so that the sensitive details are abstracted from any requests that are sent to the Data Source.

The screenshot shows a configuration window with two main sections. The top section, titled 'Context Providers', contains a dropdown menu currently showing 'Context Providers'. The bottom section, titled 'Context Keys', features an 'Add Key' button and a table with columns 'Name' and 'Value'. The table is currently empty, displaying the message 'There are no MANUAL keys created yet.' At the bottom right of the window are 'CANCEL' and 'SAVE' buttons.

Context Providers

Connections in Anzo implement the **Context Provider** interface. For example, File Store connections, Anzo Data Store connections, and Data Source connections provide contexts (in the form of JSON objects) that contain key-value pairs which define connection details such as URLs, database names, usernames and passwords, tokens, etc. These contexts are 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.

Note

If you specify a Context Provider, the key-value pairs from the selected provider are not populated in the Context Key list on the screen. However, the keys are used automatically when a query is run against that provider.

Context Keys

Context Keys 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

The image below, for example, creates URL, username, and password Context Keys.

Context Keys		Add Key	
Name	Value		
mysql-password	admin123		
mysql-username	sysadmin		
mysql-northwind-url	jdbc:mysql://10.111.4.9:3306/NORTHWIND		

Note

The format that you use for referencing a Context Key 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 Graph Data Interface (GDI) queries. When referencing Context Keys in GDI queries, use the following format:

```
{{@<context_key_name>}}
```

Related Topics

[Adding Steps and Views to Data Layers](#)

[SPARQL Query Templates and Best Practices](#)

Configuring a 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 Data Layer. The sections below describe each of the tabs and configuration options that are available when you create or edit a Query Step.

- [Details](#)
- [Query](#)
- [Execution Condition](#)
- [Context](#)

Details

The **Details** tab includes options such as the name of the Step, the source data to act upon, and the Data Model to use with the query.

The screenshot shows the 'Details' tab of a configuration window. It contains the following elements:

- Title ***: A text input field with a placeholder 'The title of the step'.
- Description**: A text input field with a placeholder 'A brief description of the step'.
- Enabled**: A checked checkbox.
- Source**: A dropdown menu showing 'Self' and 'All Previous Layers Within Graphmart'.
- Data models ***: A dropdown menu.
- Associated data models**: A section containing a checkbox for 'Pre-Run Generate Statistics'.
- Buttons**: 'CANCEL' and 'SAVE' buttons at the bottom right.

Title

The required name of the Step.

Description

And optional 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 Data 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 is the source data that this Step should act upon. Steps can build upon the data generated by Steps in other Data 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 Query 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 Query runs against the data that is generated by all of the successful Layers in the Graphmart 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 run the Query against that Layer only.

Data Models

This required field specifies the Data Model or Models that you want to create this query template against. The list displays all of the Models for 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 Model, click the toggle button on the right side of the field to change it to **Include System Data** (x). The Data Models drop-down list will display the system Models in addition to the Dataset 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.

Query

The **Query** tab contains the query that you want this Step to run.

Details
Query
Execution Condition
Context

Transformation query *

```

1
2  #targetGraph is replaced with the Layers URI at runtime
3  #usingSources is replaced with the URIs of the Layer's Sources at runtime
4  DELETE{
5    GRAPH ${targetGraph}{
6
7    }
8  }
9  INSERT{
10   GRAPH ${targetGraph}{
11
12   }
13 }
14 ${usingSources}
15 WHERE{
16
17 }

```

Query used to perform the transformation

Preview in Query Builder

CANCEL SAVE

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 Data Layer.

Tip

See [SPARQL Query Templates and Best Practices](#) for guidance on writing SPARQL queries. For information about the SPARQL syntax for INSERT and DELETE queries, see [SPARQL 1.1 Update Language](#) in the W3C SPARQL 1.1 Update specification.

Execution Condition

If you want this Step to be executed conditionally, based on the result of a specified Validation Condition, you can configure an Execution Condition on the **Execution Condition** tab that is available when creating or editing a Step. The image below shows the Execution Condition tab.

Note

In order to set up an Execution Condition, the Graphmart needs to have at least one Validation Step that defines a Condition Variable. Condition Variables can be used across all Data Layers in the Graphmart. For guidance on configuring a Validation Step, see [Configuring a Validation Step](#).

The screenshot shows a configuration dialog box for an Execution Condition. It contains the following elements:

- A checkbox labeled "Execute step based on boolean condition".
- A dropdown menu labeled "Select condition variable" with a downward arrow. Below it, the text "Conditional Variable" is displayed.
- A dropdown menu labeled "Execute only if returned value is" with a downward arrow. Below it, the text "Conditional Variable If Result" is displayed.
- At the bottom right, there are two buttons: "CANCEL" and "SAVE".

Enable Layer Based on Boolean Condition

This setting indicates whether to enable this Step 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 Step is disabled.

Conditional Variable

This field specifies the variable that you want to base this Execution Condition on. The variable is the result of a Validation Step Query in the Graphmart.

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 Step 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.

Context

When you use the Graph Data Interface (GDI) for Data Virtualization, you may connect to Data Sources that require input of sensitive connection and authorization information such as keys, tokens, and user credentials. The **Context** tab gives you the option to configure a Context to store the sensitive information as key-value pairs. Queries can then reference the keys from the Context so that the sensitive details are abstracted from any requests that are sent to the Data Source.

The screenshot shows a configuration window with two main sections. The top section, titled 'Context Providers', contains a dropdown menu currently showing 'Context Providers'. The bottom section, titled 'Context Keys', features an 'Add Key' button and a table with columns 'Name' and 'Value'. The table is currently empty, with a message 'There are no MANUAL keys created yet.' at the bottom. At the bottom right of the window are 'CANCEL' and 'SAVE' buttons.

Context Providers

Connections in Anzo implement the **Context Provider** interface. For example, File Store connections, Anzo Data Store connections, and Data Source connections provide contexts (in the form of JSON objects) that contain key-value pairs which define connection details such as URLs, database names, usernames and passwords, tokens, etc. These contexts are 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.

Note

If you specify a Context Provider, the key-value pairs from the selected provider are not populated in the Context Key list on the screen. However, the keys are used automatically when a query is run against that provider.

Context Keys

Context Keys 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

The image below, for example, creates URL, username, and password Context Keys.

Name	Value	
mysql-password	admin123	 
mysql-username	sysadmin	 
mysql-northwind-url	jdbc:mysql://10.111.4.9:3306/NORTHWIND	 

Note

The format that you use for referencing a Context Key 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 Graph Data Interface (GDI) queries. When referencing Context Keys in GDI queries, use the following format:

```
{{@<context_key_name>}}
```

Related Topics

[Adding Steps and Views to Data Layers](#)

[SPARQL Query Templates and Best Practices](#)

[Blending Data from Remote Sources \(Preview\)](#)

Configuring an 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. The sections below describe each of the tabs and configuration options that are available when you create or edit an RDFS+ Inference Step.

- [Details](#)
- [Execution Condition](#)
- [Inference Rule Reference](#)



Details

The **Details** tab includes options such as the name of the Step, the source data to act upon, and the Data Model to run the RDFS+ inference rules against.

Details

Execution Condition

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 All Previous Layers Within Graphmart x

x

v

Source data to act upon

Data models *

^

v

Associated data models

☐ Pre-Run Generate Statistics

CANCEL

SAVE

Title

The required name of the Step.

Description

And optional 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 Data 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 is the source data that this Step should act upon. Steps can build upon the data generated by Steps in other Data 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 inferencing runs against only the data that is generated in the Layer this Step belongs to.
- **All Previous Layers Within Graphmart**: This option means that inferencing runs against the data that is generated by all of the Layers in the Graphmart that precede the Layer this step is in.
- **Previous Layer Within Graphmart**: This option means that inferencing 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 run inferencing against that Layer only.

Data Models

This required field specifies the Data Model or Models to associate with this Step. The list displays all of the Models for 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 Model, click the toggle button on the right side of the field to change it to **Include System Data** (x). The Data Models drop-down list will display the system Models in addition to the Dataset 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.

Execution Condition

If you want this Step to be executed conditionally, based on the result of a specified Validation Condition, you can configure an Execution Condition on the **Execution Condition** tab that is available when creating or editing a Step. The image below shows the Execution Condition tab.

Note

In order to set up an Execution Condition, the Graphmart needs to have at least one Validation Step that defines a Condition Variable. Condition Variables can be used across all Data Layers in the Graphmart. For guidance on configuring a Validation Step, see [Configuring a Validation Step](#).

The screenshot shows a configuration dialog box for an Execution Condition. It contains the following elements:

- A checkbox labeled "Execute step based on boolean condition".
- A dropdown menu labeled "Select condition variable" with a downward arrow. Below it, the text "Conditional Variable" is displayed.
- A dropdown menu labeled "Execute only if returned value is" with a downward arrow. Below it, the text "Conditional Variable If Result" is displayed.
- At the bottom right, there are two buttons: "CANCEL" and "SAVE".

Enable Layer Based on Boolean Condition

This setting indicates whether to enable this Step 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 Step is disabled.

Conditional Variable

This field specifies the variable that you want to base this Execution Condition on. The variable is the result of a Validation Step Query in the Graphmart.

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 Step 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.



Inference Rule Reference

This topic 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 WITH RULES clause 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**) in the WITH RULES clause.

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)
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 WITH RULES clause 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**) in the WITH RULES clause.

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)
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)

Rule	Description	IF	THEN
prp-rng	Infer the object's type from the predicate's range.	$T(?p, \text{rdfs:range}, ?c)$ $T(?x, ?p, ?y)$	$T(?y, \text{rdf:type}, ?c)$
prp-spo1	Relationships that are described by a subproperty also hold for the superproperty.	$T(?p1, \text{rdfs:subPropertyOf}, ?p2)$ $T(?x, ?p1, ?y)$	$T(?x, ?p2, ?y)$
prp-symp	The inverse is true for a property.	$T(?p, \text{rdf:type}, \text{owl:SymmetricProperty})$ $T(?x, ?p, ?y)$	$T(?y, ?p, ?x)$
prp-trp	Chains of relationships collapse into a single relationship.	$T(?p, \text{rdf:type}, \text{owl:TransitiveProperty})$ $T(?x, ?p, ?y)$ $T(?y, ?p, ?z)$	$T(?x, ?p, ?z)$

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.	$T(?c, \text{rdf:type}, \text{owl:Class})$	$T(?c, \text{rdfs:subClassOf}, ?c)$ $T(?c, \text{owl:equivalentClass}, ?c)$ $T(?c, \text{rdfs:subClassOf}, \text{owl:Thing})$ $T(\text{owl:Nothing}, \text{rdfs:subClassOf}, ?c)$
scm-dom1	A property with domain c also has domain c's superclasses.	$T(?p, \text{rdfs:domain}, ?c1)$ $T(?c1, \text{rdfs:subClassOf}, ?c2)$	$T(?p, \text{rdfs:domain}, ?c2)$
scm-dom2	A subproperty inherits the domains of the superproperties.	$T(?p2, \text{rdfs:domain}, ?c)$ $T(?p1, \text{rdfs:subPropertyOf}, ?p2)$	$T(?p1, \text{rdfs:domain}, ?c)$

Rule	Description	IF	THEN
scm-eqc1	Equivalent classes are subclasses of each other.	$T(?c1, \text{owl:equivalentClass}, ?c2)$	$T(?c1, \text{rdfs:subClassOf}, ?c2)$ $T(?c2, \text{rdfs:subClassOf}, ?c1)$
scm-eqc2	If two classes are subclasses, they are also equivalent classes.	$T(?c1, \text{rdfs:subClassOf}, ?c2)$ $T(?c2, \text{rdfs:subClassOf}, ?c1)$	$T(?c1, \text{owl:equivalentClass}, ?c2)$
scm-eqp1	Equivalent properties are subproperties of each other.	$T(?p1, \text{owl:equivalentProperty}, ?p2)$	$T(?p1, \text{rdfs:subPropertyOf}, ?p2)$ $T(?p2, \text{rdfs:subPropertyOf}, ?p1)$
scm-eqp2	If two properties are subproperties, they are also equivalent properties.	$T(?p1, \text{rdfs:subPropertyOf}, ?p2)$ $T(?p2, \text{rdfs:subPropertyOf}, ?p1)$	$T(?p1, \text{owl:equivalentProperty}, ?p2)$
scm-rng1	A property with range c also has range c's superclasses.	$T(?p, \text{rdfs:range}, ?c1)$ $T(?c1, \text{rdfs:subClassOf}, ?c2)$	$T(?p, \text{rdfs:range}, ?c2)$
scm-rng2	A subproperty inherits the ranges of its superproperties.	$T(?p2, \text{rdfs:range}, ?c)$ $T(?p1, \text{rdfs:subPropertyOf}, ?p2)$	$T(?p1, \text{rdfs:range}, ?c)$
scm-sco	owl:subClassOf relationships are transitive	$T(?c1, \text{rdfs:subClassOf}, ?c2)$ $T(?c2, \text{rdfs:subClassOf}, ?c3)$	$T(?c1, \text{rdfs:subClassOf}, ?c3)$
scm-spo	owl:subPropertyOf relationships are transitive.	$T(?p1, \text{rdfs:subPropertyOf}, ?p2)$ $T(?p2, \text{rdfs:subPropertyOf}, ?p3)$	$T(?p1, \text{rdfs:subPropertyOf}, ?p3)$

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)
eq-sym	Describes the symmetric property of the owl:sameAs axiom.	T(?x, owl:sameAs, ?y)	T(?y, owl:sameAs, ?x)
eq-trans	Describes the transitive property of the owl:sameAs axiom.	T(?x, owl:sameAs, ?y) T(?y, owl:sameAs, ?z)	T(?x, 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, owl:someValuesFrom, ?y1) T(?c1, owl:onProperty, ?p) T(?c2, owl:someValuesFrom, ?y2) T(?c2, owl:onProperty, ?p) T(?y1, rdfs:subClassOf, ?y2)	T(?c1, 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, owl:someValuesFrom, ?y) T(?c1, owl:onProperty, ?p1) T(?c2, owl:someValuesFrom, ?y) T(?c2, owl:onProperty, ?p2) T(?p1, rdfs:subPropertyOf, ?p2)	T(?c1, rdfs:subClassOf, ?c2)
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)

Related Topics

[Adding Steps and Views to Data Layers](#)

Configuring a Templated Step

Templated steps enable users to create reusable templates for creating additional query steps in different Data Layers or Graphmarts. In templated queries, key-value pairs are represented by parameters in a query. When reusing the step, users do not need to rewrite the query to target a different Data Source. Instead, they modify the values for the keys.

Tip

This type of query 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 [Configuring a Query-Driven Templated Step](#).

The sections below describe each of the tabs and configuration options that are available when you create or edit a Templated Step.

- [Details](#)
- [Template](#)



- [Execution Condition](#)
- [Context](#)

Details

The **Details** tab includes options such as the name of the Step, the source data to act upon, and the Data Model to create the template against.

The screenshot shows the 'Details' tab of a configuration window. It contains the following elements:

- Navigation tabs:** Details (active), Template, Execution Condition, Context.
- Title ***: A text input field with placeholder text 'The title of the step'.
- Description**: A text input field with placeholder text 'A brief description of the step'.
- Enabled**: A checked checkbox.
- Source**: A dropdown menu showing 'Self' and 'All Previous Layers Within Graphmart'.
- Data models ***: A dropdown menu.
- Associated data models**: A section containing a checkbox for 'Pre-Run Generate Statistics'.
- Buttons:** CANCEL and SAVE at the bottom right.

Title

The required name of the Step.

Description

And optional 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 Data 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 is the source data that this Step should act upon. Steps can build upon the data generated by Steps in other Data 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 Query 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 Query runs against the data that is generated by all of the successful Layers in the Graphmart 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 run the Query against that Layer only.

Data Models

This required field specifies the Data Model or Models that you want to create this query template against. The list displays all of the Models for 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 Model, click the toggle button on the right side of the field to change it to **Include System Data** (x). The Data Models drop-down list will display the system Models in addition to the Dataset 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.

Template

The **Template** tab defines the template query and the key-value pairs.

Details **Template** Execution Condition 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**

```

1
2 #targetGraph is replaced with the Layers URI at
   runtime
3 #usingSources is replaced with the URIs of the
   Layer's Sources at runtime
4 * DELETE{
5   GRAPH ${targetGraph}{
6
7   }
8 }
9 * INSERT{
10  GRAPH ${targetGraph}{
11
12  }
13 }
14 ${usingSources}
15 * WHERE{
16
17 }
  
```

Search keys... `🔍` Search values... `🔍`

No parameters found

CANCEL **SAVE**

Query

The left side of the screen includes the query template that this Step will run. The template includes the syntax for writing SPARQL INSERT and DELETE queries and includes source and target graph parameters that Anzo replaces at runtime. Edit the template text as needed. In the query, include the parameters in the format `${key_name}` that you intend to replace at runtime with the value that you define for the key. 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))
}
  
```

Tip

See [SPARQL Query Templates and Best Practices](#) for guidance on writing SPARQL queries.

Key/Value Table

Once the template query has been defined, populate the Key/Value Table with the keys from the query and the values that should replace the parameters that are specified in the query. Clicking the **Refresh Keys** button adds each key in the template to the table. 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		

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	http://cambridgesemantics.com/ont/autogen/c89d/Tickets#tickit_events_eventid
operator	Value

Execution Condition

If you want this Step to be executed conditionally, based on the result of a specified Validation Condition, you can configure an Execution Condition on the **Execution Condition** tab that is available when creating or editing a Step. The image below shows the Execution Condition tab.

Note

In order to set up an Execution Condition, the Graphmart needs to have at least one Validation Step that defines a Condition Variable. Condition Variables can be used across all Data Layers in the Graphmart. For guidance on configuring a Validation Step, see [Configuring a Validation Step](#).

The screenshot shows a configuration dialog box for an Execution Condition. It contains the following elements:

- A checkbox labeled "Execute step based on boolean condition".
- A dropdown menu labeled "Select condition variable" with a downward arrow. Below it, the text "Conditional Variable" is displayed.
- A dropdown menu labeled "Execute only if returned value is" with a downward arrow. Below it, the text "Conditional Variable If Result" is displayed.
- At the bottom right, there are two buttons: "CANCEL" and "SAVE".

Enable Layer Based on Boolean Condition

This setting indicates whether to enable this Step 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 Step is disabled.

Conditional Variable

This field specifies the variable that you want to base this Execution Condition on. The variable is the result of a Validation Step Query in the Graphmart.

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 Step 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.

Context

When you use the Graph Data Interface (GDI) for Data Virtualization, you may connect to Data Sources that require input of sensitive connection and authorization information such as keys, tokens, and user credentials. The **Context** tab gives you the option to configure a Context to store the sensitive information as key-value pairs. Queries can then reference the keys from the Context so that the sensitive details are abstracted from any requests that are sent to the Data Source.

The screenshot shows a configuration window with two main sections. The top section, titled 'Context Providers', contains a dropdown menu currently showing 'Context Providers'. The bottom section, titled 'Context Keys', features an 'Add Key' button and a table with columns 'Name' and 'Value'. The table is currently empty, with a message 'There are no MANUAL keys created yet.' at the bottom. At the bottom right of the window are 'CANCEL' and 'SAVE' buttons.

Context Providers

Connections in Anzo implement the **Context Provider** interface. For example, File Store connections, Anzo Data Store connections, and Data Source connections provide contexts (in the form of JSON objects) that contain key-value pairs which define connection details such as URLs, database names, usernames and passwords, tokens, etc. These contexts are 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.

Note

If you specify a Context Provider, the key-value pairs from the selected provider are not populated in the Context Key list on the screen. However, the keys are used automatically when a query is run against that provider.

Context Keys

Context Keys 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

The image below, for example, creates URL, username, and password Context Keys.

Context Keys		Add Key	
Name	Value		
mysql-password	admin123		
mysql-username	sysadmin		
mysql-northwind-url	jdbc:mysql://10.111.4.9:3306/NORTHWIND		

Note

The format that you use for referencing a Context Key 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 Graph Data Interface (GDI) queries. When referencing Context Keys in GDI queries, use the following format:

```
{{@<context_key_name>}}
```

Related Topics

[Adding Steps and Views to Data Layers](#)

[SPARQL Query Templates and Best Practices](#)

Configuring a Validation Step

This topic provides guidance on configuring a Validation Step to use for validating the data in a Data Layer. The sections below describe each of the tabs and configuration options that are available when you create or edit a Validation Step.

- [Details](#)
- [Options](#)
- [Query](#)
- [Context](#)



Details

The **Details** tab includes options such as the name of the Step, the validation source, and the source data to validate.

DetailsOptionsQueryContext

Title *

The title of the step

Description

A brief description of the step

☒ Enabled

Validation Source Datasource

Optional Datasource used to perform the validation query

Source

Self xAll Previous Layers Within Graphmart x

Source data to act upon

☐ Pre-Run Generate Statistics

CANCEL

SAVE

Title

The required name of the Step.

Description

And optional 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 Data 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 to use to perform the validation against if you do not want the query to run against the Graphmart data in AnzoGraph. When Validation Source Datasource is unset, the validation is performed against the Graphmart data in AnzoGraph.

© 2023 Cambridge Semantics, Inc.

Source

The Source is the source data that this Step should act upon. Steps can build upon the data generated by Steps in other Data 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 validation 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 validation runs against the data that is generated by all of the Layers in the Graphmart that precede the Layer this step is in.
- **Previous Layer Within Graphmart:** This option means that the validation 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 run the validation against that Layer only.

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.

Options

The **Options** 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. If you want to set up an Execution Condition in a Data Layer or Step, configure the Options tab to define the Condition Variable to use in the Execution Condition.

The screenshot shows the 'Options' tab in a configuration window. At the top, there are four tabs: 'Details', 'Options' (which is active), 'Query', and 'Context'. Under the 'Options' tab, 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.' At the bottom, there is a text input field labeled 'Optional variable name to which results are placed.' with the placeholder text 'Validation Result Variable Name'. In the bottom right corner, there are 'CANCEL' and 'SAVE' buttons.

Check Type

This setting configures the Step as either a **Validation** or a **Condition**. A **Validation** check validates the data according to the defined Query and can be configured to take action depending on whether the validation passes or fails. 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** on the Data Layer or Step level. A Variable Name can also be associated with a Validation check. Select the appropriate radio button to define the Check Type.

Tip See [Defining Execution Conditions](#) for information about Execution Conditions.

If the validation query fails, the layer will be marked as failed

This setting is presented if the **Check Type** is **Validation**. Select this option if you want Anzo to abort the load of the Data Layer if this Step fails. The Graphmart and other successful Data Layers continue to load.

If the validation query fails, the whole graphmart will be marked as failed

This setting is presented if the **Check Type** is **Validation**. Select this option if you want Anzo to abort the load of the entire Graphmart if this Step fails.

Tip

If you want Anzo to load the Data Layer if this Step fails, leave both **If the validation query fails...** settings blank.

Validation Result Variable Name

This option specifies 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 for a Data Layer or Step.

Query

The **Query** tab contains the query that you want the Step to run.

Details
Options
Query
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

CANCEL
SAVE

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 data options from the Details tab, Anzo automatically populates the query with the appropriate source graph URIs when the query runs.

Context

When you use the Graph Data Interface (GDI) for Data Virtualization, you may connect to Data Sources that require input of sensitive connection and authorization information such as keys, tokens, and user credentials. The **Context** tab gives you the option to configure a Context to store the sensitive information as key-value pairs. Queries can then reference the keys from the Context so that the sensitive details are abstracted from any requests that are sent to the Data Source.

The screenshot shows a configuration window with two main sections. The top section, titled 'Context Providers', contains a dropdown menu currently showing 'Context Providers'. The bottom section, titled 'Context Keys', features an 'Add Key' button and a table with columns 'Name' and 'Value'. The table is currently empty, with a message 'There are no MANUAL keys created yet.' at the bottom. At the bottom right of the window are 'CANCEL' and 'SAVE' buttons.

Context Providers

Connections in Anzo implement the **Context Provider** interface. For example, File Store connections, Anzo Data Store connections, and Data Source connections provide contexts (in the form of JSON objects) that contain key-value pairs which define connection details such as URLs, database names, usernames and passwords, tokens, etc. These contexts are 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.

Note

If you specify a Context Provider, the key-value pairs from the selected provider are not populated in the Context Key list on the screen. However, the keys are used automatically when a query is run against that provider.

Context Keys

Context Keys 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

The image below, for example, creates URL, username, and password Context Keys.

Context Keys		Add Key	
Name	Value		
mysql-password	admin123		
mysql-username	sysadmin		
mysql-northwind-url	jdbc:mysql://10.111.4.9:3306/NORTHWIND		

Note

The format that you use for referencing a Context Key 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 Graph Data Interface (GDI) queries. When referencing Context Keys in GDI queries, use the following format:

```
{{@<context_key_name>}}
```

Related Topics

[Adding Steps and Views to Data Layers](#)

Configuring a View Step

This topic provides guidance on configuring a View Step to use creating a custom view of the data that does not change the source or necessarily materialize any new data. View steps use SPARQL CONSTRUCT queries to create a view definition in AnzoGraph. The sections below describe each of the tabs and configuration options that are available when you create or edit a View Step.

- [Details](#)
- [Query](#)
- [Execution Condition](#)
- [Hi-Res Analytics](#)
- [Context](#)

Details

The **Details** tab includes the settings that configure options such as the name of the step, the source data to act upon, and the Data Models to create the view against.

The screenshot shows the 'Details' tab in the Anzo interface. It features a header with tabs: 'Details' (active), 'Query', 'Execution Condition', and 'Hi-Res Analytics'. Below the tabs are several input fields: 'Title *' with a placeholder 'The title of the view', 'Description' with a placeholder 'A brief description of the view', a checkbox labeled 'Materialize the view when activated, otherwise at runtime.' which is checked and has a green 'Enabled' label next to it, a 'Source' dropdown menu with a placeholder 'Source data to act upon', and a 'Data models *' dropdown menu with a placeholder 'Associated data models'. At the bottom right of the form are two buttons: 'CANCEL' and 'SAVE'.

Title

The required name of the Step.

Description

And optional description of the Step.

Materialize the view when activated, otherwise at runtime

If you want to store a copy of the data that the view creates (materialize the data), 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 Data 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 is the source data that this Step should act upon. Steps can build upon the data generated by Steps in other Data 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 Query 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 Query runs against the data that is generated by all of the successful Layers in the Graphmart 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 run the Query against that Layer only.

Data Models

This required field specifies the Data Model or Models that you want to create this view against. The list displays all of the Models for 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 Model, click the toggle button on the right side of the field to change it to **Include System Data** (🔍). The Data Models drop-down list will display the system Models in addition to the Dataset Models.

Query

The **Query** tab contains the query used to create the view.



The tab includes the template for a CONSTRUCT query. 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 Data Layer.

Execution Condition

If you want this Step to be executed conditionally, based on the result of a specified Validation Condition, you can configure an Execution Condition on the **Execution Condition** tab that is available when creating or editing a Step. The image below shows the Execution Condition tab.

Note

In order to set up an Execution Condition, the Graphmart needs to have at least one Validation Step that defines a Condition Variable. Condition Variables can be used across all Data Layers in the Graphmart. For guidance on configuring a Validation Step, see [Configuring a Validation Step](#).

The screenshot shows a configuration dialog box for an Execution Condition. It contains the following elements:

- A checkbox labeled "Execute step based on boolean condition".
- A dropdown menu labeled "Select condition variable" with a downward arrow. Below it, the text "Conditional Variable" is displayed.
- A dropdown menu labeled "Execute only if returned value is" with a downward arrow. Below it, the text "Conditional Variable If Result" is displayed.
- At the bottom right, there are two buttons: "CANCEL" and "SAVE".

Enable Layer Based on Boolean Condition

This setting indicates whether to enable this Step 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 Step is disabled.

Conditional Variable

This field specifies the variable that you want to base this Execution Condition on. The variable is the result of a Validation Step Query in the Graphmart.

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 Step 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.

Hi-Res Analytics

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.

< Details Query Execution Condition **Hi-Res Analytics** >

Warning: These are Advanced Settings. Change Wisely.

- ☒ 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
- ☒ By default, the data outputted by the view is visible ☐ Is Dynamic

Dynamic Refresh Interval

The interval in milli-seconds between when server will send out data updated events.

☐ Hide From Other Layers/Views

CANCEL SAVE

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 dashboard :

- 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 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 Hi-Res Analytics dashboards. The image below shows an example Data Layers panel:

^ Data Layers ☒

Tickets

Flights

- 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 list 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 steps in 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 Data 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.

Context

When you use the Graph Data Interface (GDI) for Data Virtualization, you may connect to Data Sources that require input of sensitive connection and authorization information such as keys, tokens, and user credentials. The **Context** tab gives you the option to configure a Context to store the sensitive information as key-value pairs. Queries can then reference the keys from the Context so that the sensitive details are abstracted from any requests that are sent to the Data Source.

The screenshot shows a configuration window with two main sections. The top section, titled 'Context Providers', contains a dropdown menu currently showing 'Context Providers'. The bottom section, titled 'Context Keys', features an 'Add Key' button and a table with columns 'Name' and 'Value'. The table is currently empty, with a message 'There are no MANUAL keys created yet.' at the bottom. At the bottom right of the window are 'CANCEL' and 'SAVE' buttons.

Context Providers

Connections in Anzo implement the **Context Provider** interface. For example, File Store connections, Anzo Data Store connections, and Data Source connections provide contexts (in the form of JSON objects) that contain key-value pairs which define connection details such as URLs, database names, usernames and passwords, tokens, etc. These contexts are 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.

Note

If you specify a Context Provider, the key-value pairs from the selected provider are not populated in the Context Key list on the screen. However, the keys are used automatically when a query is run against that provider.

Context Keys

Context Keys 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

The image below, for example, creates URL, username, and password Context Keys.

Context Keys		Add Key	
Name	Value		
mysql-password	admin123		
mysql-username	sysadmin		
mysql-northwind-url	jdbc:mysql://10.111.4.9:3306/NORTHWIND		

Note

The format that you use for referencing a Context Key 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 Graph Data Interface (GDI) queries. When referencing Context Keys in GDI queries, use the following format:

```
{{@<context_key_name>}}
```

Related Topics

[Adding Steps and Views to Data Layers](#)

[SPARQL Query Templates and Best Practices](#)

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](#) for information). Users who have permission to modify Graphmart access (as granted by the Default Access Policy) 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 Data 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 Data 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 is blank because the Graphmart is configured to follow the Default Access Policy. The **Pass permissions to** field lists any Data Layers ("Tickets to Store" in the image), Steps ("Load Tickets to Store"), and Data on Demand endpoints ("Venues") that inherit their Configuration permissions from the Graphmart.

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

Inherit permissions from

Pass permissions to

Venues x Tickets to Store x

Load Tickets to Store x

Tip

If this Graphmart had associated backup Versions, those Versions would also be listed in the **Pass permissions to** field.

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**.

Note

The **Default Layer View Permissions (for new Layers)** field sets the inheritance for new, user-created Data Layers. Anzo-created Load Data Layers inherit permissions from the Load Step's Dataset by default.

Below the inheritance settings, the **Permissions Overview** provides a detailed view of the permission inheritance for each Data Layer 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

Default Layer View Permissions (for new Layers)





Inherit from Graphmart [CHANGE](#)

Inherit from Graphmart [CHANGE](#)

Permissions Overview

Search

Sort By: Graphmart

Title	Type	Warnings	Inherits ACLs from	Roles that can view	Actions	
 Tickets Graphmart	Graphmart		Graphmart Configuration		EDIT PERMISSIONS	
 Venues	View		Graphmart		EDIT PERMISSIONS	
 Tickets to Store	Layer		Dataset Tickets to Store	Everyone, sysadmin,	EDIT PERMISSIONS	
 Event Venues	Endpoint		Graphmart		EDIT PERMISSIONS	

Rows per page: 251-4 of 4<>

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"> • See the Graphmart in the Anzo application. • Copy the Graphmart URI from the Overview tab. • Copy Data Layer URIs from the Data Layers tab. • See the existing Data on Demand endpoints on the Data on Demand tab. • View and clone the Dataset Editions that are included in the Graphmart. • Reload and refresh the Graphmart. • Create and import Graphmart versions.
	Meta View	<ul style="list-style-type: none"> • 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.
Modify	In addition to the View and Meta View permissions described above, the Modify set includes the Add/Edit and Delete permissions described below.	
	Add/Edit	<ul style="list-style-type: none"> • Rename the Graphmart and edit the description. • Create Data on Demand endpoints. • Add Datasets to the Graphmart. • Enable, disable, add, or edit Data Layers and Steps. • Activate and deactivate the Graphmart.
	Delete	<ul style="list-style-type: none"> • Remove Datasets from the Graphmart. • Delete Data Layers and Steps from the Graphmart. • Cannot delete the Graphmart.

Set	Permission	Allows a user to:
Admin		In addition to the View , Meta View , Add/Edit , and Delete permissions described above, the Admin set includes the Meta Add/Edit and Meta Delete permissions described below.
	Meta Add/Edit	<ul style="list-style-type: none"> 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.
	Meta Delete	<ul style="list-style-type: none"> Modify the sharing settings by removing permissions from a user or group. Delete the Graphmart.

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 Data 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

Sort By: Title

View:

Add Graphmart

<div></div>	Title	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions
	DB - northwind to Store Graphmart	System Admi...	Ready to us	9,172	2 hours ago	2 hours ago		<div></div>
	Datafox Graphmart	System Admi...						<div></div>
	Flight Data Graphmart	System Admi...	Ready to us	265,696	2 minutes ago	2 minutes ago		<div></div>
	Tickets Graphmart	System Admi...	Ready to us	4,088,422	5 hours ago	5 hours ago		<div></div>

2. In the list of Graphmarts, 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:

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

Pass permissions to

Inherit permissions from

Venues Tickets to Store Load Tickets to Store

User and Group Permissions

Permissions defined for specific users or groups are applied in addition to any inherited permissions.

Search users, roles or groups

System Administrator Admin

Everyone View

Define the permissions of this artifact.
Start by selecting a user or group on the left.

3. If you want to change how the Configuration permissions are inherited, 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. By default, Graphmarts inherit their permissions from the Graphmarts Registry Default Access Policy.
- To pass this artifact's permissions to other artifacts, select the artifacts to pass permissions to in the **Pass permissions to** field. By default, Graphmarts pass their permissions to all of the Data Layers, Data on Demand endpoints, and Versions that are in the Graphmart.

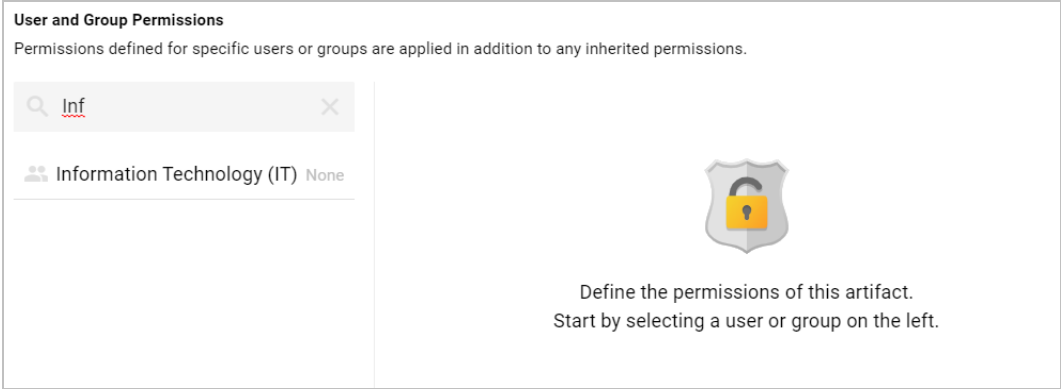
Tip

For more information about permission inheritance at the Graphmart level, see [Permission Inheritance](#).

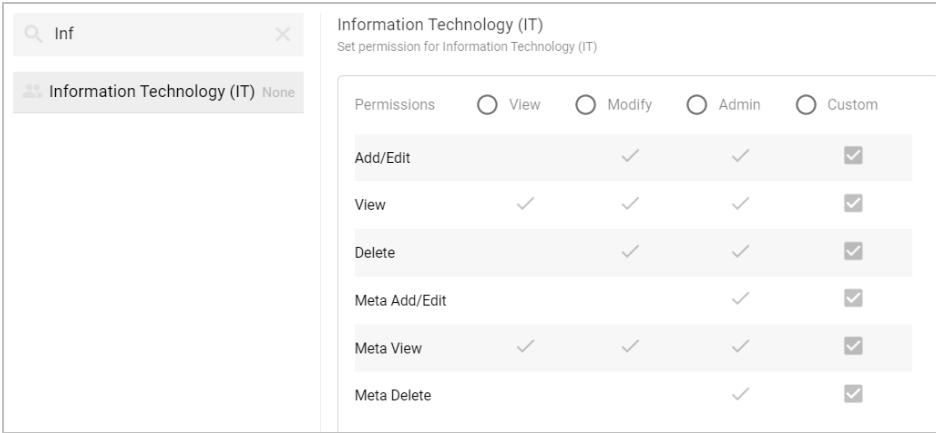
4. To modify Configuration access to this Graphmart with a particular user or group, type a value in the **Search users, roles or groups** field to find and display the user or group. The resulting list shows the current permission level that is set for each user or group in the search results. For example, the image below shows the current

© 2023 Cambridge Semantics, Inc.

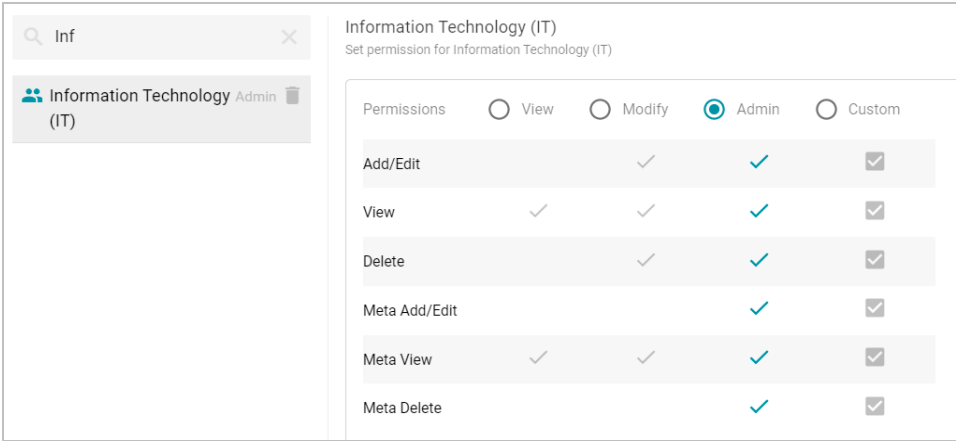
permissions for the IT group (None):



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:



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:



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.

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

Sort By: Title

View:

Add Graphmart

<div></div>	Title	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions
<div></div>	<div></div> DB - northwind to Store Graphmart	System Admi...	<div></div> Ready to use	9,172	2 hours ago	2 hours ago		<div></div> <div></div>
<div></div>	<div></div> Datafox Graphmart	System Admi...	<div></div>					<div></div> <div></div>
<div></div>	<div></div> Flight Data Graphmart	System Admi...	<div></div> Ready to use	265,696	2 minutes ago	2 minutes ago		<div></div> <div></div>
<div></div>	<div></div> Tickets Graphmart	System Admi...	<div></div> Ready to use	4,088,422	5 hours ago	5 hours ago		<div></div> <div></div>

2. In the list of Graphmarts, 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:

Configuration

Data Access

Warnings

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

Default Layer View Permissions (for new Layers)

Inherit from Graphmart [CHANGE](#)

Inherit from Graphmart [CHANGE](#)

Permissions Overview

	Search	Sort By: Graphmart			
Title	Type	Warnings	Inherits ACLs from	Roles that can view	Actions
Tickets Graphmart	Graphmart		Graphmart Configuration		EDIT PERMISSIONS
Venues	View		Graphmart		EDIT PERMISSIONS
Tickets to Store	Layer		Dataset Tickets to Store	Everyone, sysadmin,	EDIT PERMISSIONS
Event Venues	Endpoint		Graphmart		EDIT PERMISSIONS

Rows per page: 25 1-4 of 4

3. 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](#).

4. To change the permissions for an individual Data Layer, Data on Demand endpoint, or another Graphmart component that is listed in the Permissions Overview, click **Edit Permissions** in the row for that component.

Changes to Graphmart and Data 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.

Related Topics

[Creating a New Graphmart](#)

[Adding a Dataset to a Graphmart](#)

[Adding Data Layers to Graphmarts](#)

[Adding Steps and Views to Data Layers](#)

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:

Search

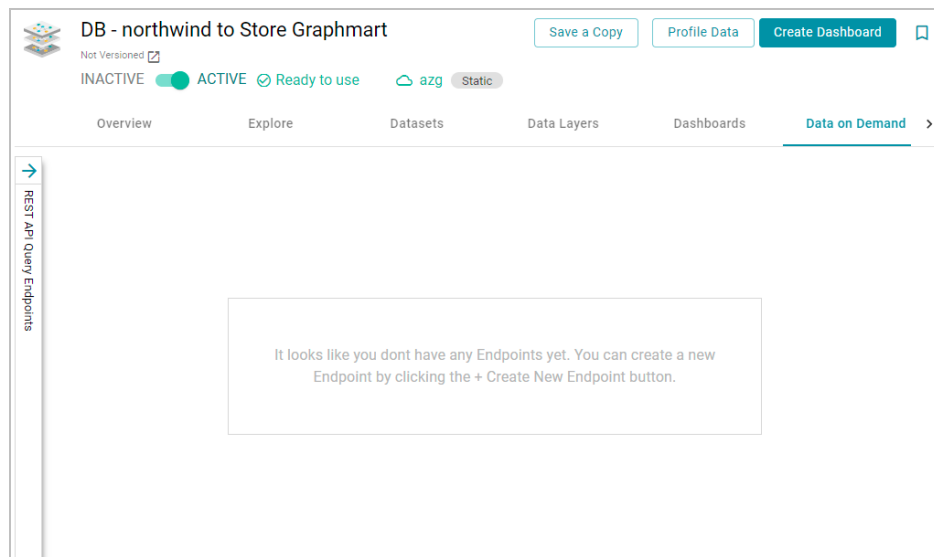
Sort By: Title

View:

Add Graphmart

	Title	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions
	DB - northwind to Store Graphmart	System Admi...	<div></div> Ready to us	9,172	2 hours ago	2 hours ago		<div></div> <div></div>
	Datafox Graphmart	System Admi...	<div></div>					<div></div> <div></div>
	Flight Data Graphmart	System Admi...	<div></div> Ready to us	265,696	2 minutes ago	2 minutes ago		<div></div> <div></div>
	Tickets Graphmart	System Admi...	<div></div> Ready to us	4,088,422	5 hours ago	5 hours ago		<div></div> <div></div>

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:



- If necessary, expand the REST API Query Endpoints panel on the left side of the screen, and then click the **Create New Endpoint** button. Anzo displays the Create REST API Query Endpoint screen.

Create REST API Query Endpoint

Endpoint Name *

Name of Endpoint

Endpoint Description

Description of Endpoint

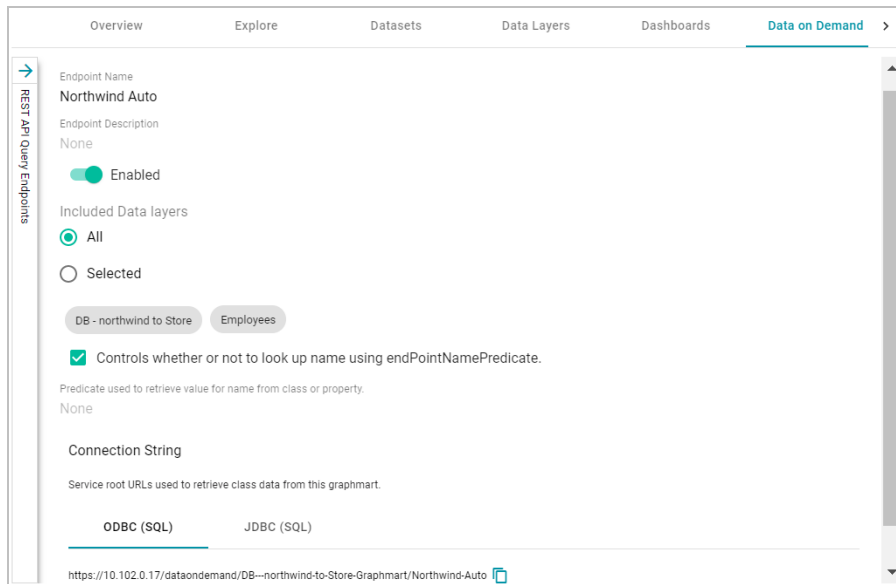
Endpoint Creation

☒ Auto-Generated
 ☐ Custom

CANCEL

SAVE

- Type a name for the endpoint in the **Endpoint Name** field. The endpoint name must be unique. In addition, add an optional description in the **Endpoint Description** field.
- By default, the Endpoint Creation type is set to **Auto-Generated**. Leave the radio button selected and click **Save** to create the endpoint and view the configuration details. For example:

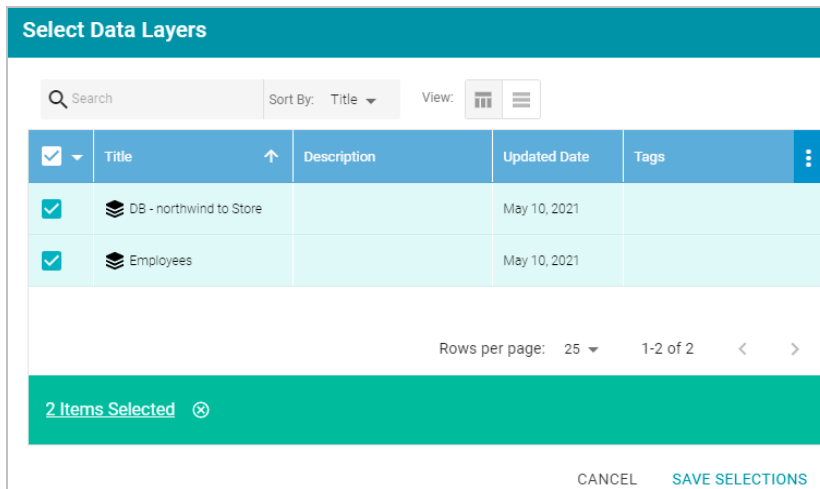


7. You can make changes to any of the following configuration options:
- By default the endpoint is set to **Enabled**, indicating that the endpoint is active. If you want to disable the endpoint, slide the Enabled slider to the left.

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."

- By default the **Included Data Layers** option is set to **All**, indicating that all of the Data Layers in the Graphmart will be 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:



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.

- The **Controls whether or not to look up name using endPointNamePredicate** setting controls which the 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 a different 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.

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/DB--northwind-to-Store-Graphmart/Northwind-Auto>

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"?>
<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/DB---northwind-to-Store-Graphmart/Northwind-Auto/$metadata">
  <app:workspace>
    <atom:title>Feeds.Default</atom:title>
    <app:collection href="new_employees" metadata:name="new_employees">
      <atom:title>new_employees</atom:title>
    </app:collection>
    <app:collection href="Summary_of_Sales_by_Quarter" metadata:name="Summary_of_Sales_by_Quarter">
      <atom:title>Summary_of_Sales_by_Quarter</atom:title>
    </app:collection>
    <app:collection href="Territories" metadata:name="Territories">
      <atom:title>Territories</atom:title>
    </app:collection>
    <app:collection href="Shippers" metadata:name="Shippers">
      <atom:title>Shippers</atom:title>
    </app:collection>
    <app:collection href="Suppliers" metadata:name="Suppliers">
      <atom:title>Suppliers</atom:title>
    </app:collection>
    <app:collection href="Products" metadata:name="Products">
      <atom:title>Products</atom:title>
    </app:collection>
    <app:collection href="Region" metadata:name="Region">
      <atom:title>Region</atom:title>
    </app:collection>
    <app:collection href="Summary_of_Sales_by_Year" metadata:name="Summary_of_Sales_by_Year">
      <atom:title>Summary_of_Sales_by_Year</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 \(Preview\)](#)

[Accessing Data on Demand Endpoints](#)

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 Hi-Res Analytics Table lens. 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:

<div><div><div><div></div></div><div>Search</div></div><div>Sort By: Title</div><div>View: <div><div></div><div></div></div></div><div>Add Graphmart</div></div>									
<div><div></div></div>	Title	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions	
	DB - northwind to Store Graphmart	System Admi...	<div><div></div>Ready to us</div>	9,172	2 hours ago	2 hours ago		<div><div></div></div>	<div><div></div></div>
	Datafox Graphmart	System Admi...	<div><div></div></div>					<div><div></div></div>	<div><div></div></div>
	Flight Data Graphmart	System Admi...	<div><div></div>Ready to us</div>	265,696	2 minutes ago	2 minutes ago		<div><div></div></div>	<div><div></div></div>
	Tickets Graphmart	System Admi...	<div><div></div>Ready to us</div>	4,088,422	5 hours ago	5 hours ago		<div><div></div></div>	<div><div></div></div>

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:

Tickets Graphmart

Not Versioned

INACTIVE

ACTIVE

Ready to use

azg

Static

Overview

Explore

Datasets

Data Layers

Dashboards

Data on Demand

REST API Query Endpoints

It looks like you dont have any Endpoints yet. You can create a new Endpoint by clicking the + Create New Endpoint button.

4. If necessary, expand the REST API Query Endpoints panel on the left side of the screen, and then click the **Create New Endpoint** button. 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

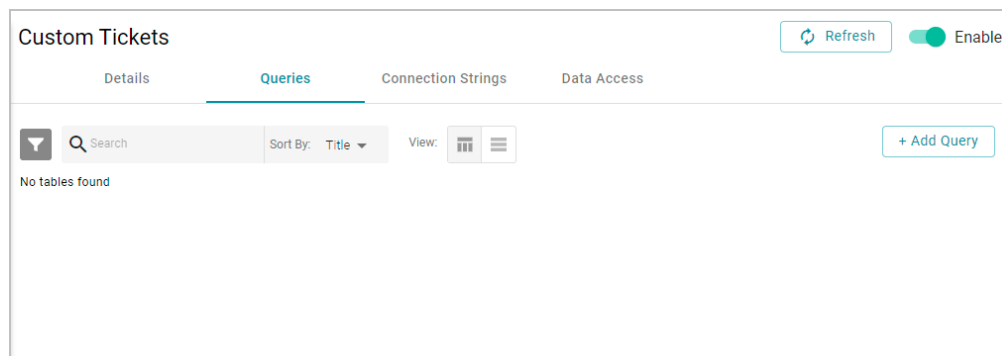
Auto-Generated

Custom

CANCEL

SAVE

5. Type a name for the endpoint in the **Endpoint Name** field. The endpoint name must be unique. In addition, add an optional description in the **Endpoint Description** field.
6. By default, the Endpoint Creation type is set to Auto-Generated. Select the **Custom** radio button and then click **Save**. The endpoint is created and is empty until you create queries. For example, the image below shows a new empty endpoint:



7. To start building a table, click the **Add Query** button. The New Query dialog box is displayed.

8. On the **Details** tab, configure the following details:
 - Type a name for the table in the **Title** field. The table name must be unique. In addition, add an optional description in the **Description** field.
 - If you want to store a copy of the data that this table 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 drops the temporary data.
 - 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.

- Click the **Query** tab to start building the table, which Anzo converts to a SPARQL query.

Edit Query

Details **Query**

Starting point: Select starting point...

Set a starting point to begin defining your query

CANCEL SAVE

- On the Query screen, get started by selecting the Class to use as the starting point. Click the **Starting Point** drop-down list and select a Class. Once you select the class, the **Add Column** button is displayed.
- Click **Add Column** to create a column in the table. The Edit Column dialog box is displayed:

Edit Column

Formula Value

Column Header Label

☐ Denormalize

Sample Data

No expression specified

CANCEL SAVE

- On the Edit Column screen, configure the following options:
 - Formula Value:** Click this field to choose the column header. Like building columns in dashboard Table lenses, 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:



- **Column Header Label:** This is the label that you want to use for displaying the Formula Value.
- **Denormalize:** This setting controls how multi-valued properties are formatted in exported output. When **Denormalize** is disabled (the default value) multi-value properties are output in arrays. When **Denormalize** is enabled, each value gets its own row. For example, the following image shows the output of multi-value properties when Denormalize is disabled:

```
TeamId,TeamName,TeamToPlayer_PlayerName,league_tbal44_key
1,Al Thomas,"[Fred Wynn,Steve Jones,James Smith]",aHR0cDovL2NzaS5jb20vVGhvbXNvMQ
2,Black Sox,"[Tim Hooper,Jared Bonds,Matt Butler]",aHR0cDovL2NzaS5jb20vVGhvbXNvMg
3,Braves,"[Billy Roper,Alex Granderson,Ted Sale]",aHR0cDovL2NzaS5jb20vVGhvbXNvMw
4,Somerville,"[Chris Underwood,Mike Magazine,Ted James]",aHR0cDovL2NzaS5jb20vVGhvbXNvNA
```

And the example below shows the output of multi-value properties when Denormalize is enabled:

TeamId	TeamName	PlayerName	table51_key
1	Al Thomas	James Smith	aHR0cDovL2NzaS5jb20vVGhvbXNvMQ
1	Al Thomas	Fred Wynn	aHR0cDovL2NzaS5jb20vVGhvbXNvMQ
1	Al Thomas	Steve Jones	aHR0cDovL2NzaS5jb20vVGhvbXNvMQ
2	Black Sox	Matt Butler	aHR0cDovL2NzaS5jb20vVGhvbXNvMg
2	Black Sox	Jared Bonds	aHR0cDovL2NzaS5jb20vVGhvbXNvMg
2	Black Sox	Tim Hooper	aHR0cDovL2NzaS5jb20vVGhvbXNvMg
3	Braves	Billy Roper	aHR0cDovL2NzaS5jb20vVGhvbXNvMw
3	Braves	Alex Granderson	aHR0cDovL2NzaS5jb20vVGhvbXNvMw
3	Braves	Ted Sale	aHR0cDovL2NzaS5jb20vVGhvbXNvMw
4	Somerville	Mike Magazine	aHR0cDovL2NzaS5jb20vVGhvbXNvNA
4	Somerville	Chris Underwood	aHR0cDovL2NzaS5jb20vVGhvbXNvNA
4	Somerville	Ted James	aHR0cDovL2NzaS5jb20vVGhvbXNvNA

Tip

If you do not Denormalize the data, you can use the **Format** tab to specify the character to use for separating values in the arrays that are returned for multi-value properties. By default, the Value Separator is set to comma (,).

13. Add new columns to the table by clicking **Add Column** on the right side of the screen. 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.

Details

Query

Starting point: ticket_venues

Venue	Event	Event Category	Venue City
Toyota Park	Akon, Allman Brothers ...	Pop	Bridgeview
Pizza Hut Park	.38 Special, Aimee Man...	Pop	Frisco
U.S. Cellular Field	Bamboozle Left, Ben H...	Pop	Chicago
Progressive Field	Air Supply, Blues Broth...	Pop	Cleveland
Comerica Park	Allman Brothers Band, ...	Pop	Detroit
Kauffman Stadium	Air Supply, Black Kids, ...	Pop	Kansas City

Rows per page: 25 1-25 of 202

Manage Configuration

Columns

Filters

Format

+ Add Column

Venue

Event

Event Category

Venue City

ON

ON

ON

ON

CANCEL SAVE

14. 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, this Data on Demand 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:

<

Dashboards

Data on Demand

Versions

Discussion

Sharing

Event Venues

Refresh

Enable

Details

Queries

Connection Strings

Data Access

Connection String

Service root URLs used to retrieve class data from this graphmart.

ODBC (SQL)

JDBC (SQL)

https://10.102.0.17/dataondemand/Tickets-Graphmart/Event-Venues

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://10.10.0.10/dataondemand/Tickets-Graphmart/Event-Venues`, and the table name is Venues, the connection string is `https://10.10.0.10/dataondemand/Tickets-Graphmart/Event-Venues/Venues`.

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.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<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/Tickets-Graphmart/Event-Venues/$metadata#Venues">
  <a:id>https://10.102.0.17/dataondemand/Tickets-Graphmart/Event-Venues/Venues</a:id>
  <a:entry>
    <a:id>Tickets-Graphmart/Event-Venues/Venues('aHR0cDovL2NzaS5jb20vdG1ja2l0X3ZlbnVlcy8x')</a:id>
    <a:title/>
    <a:summary/>
    <a:updated>2021-06-04T17:39:43Z</a:updated>
    <a:author>
      <a:name/>
    </a:author>
    <a:link rel="edit" href="Tickets-Graphmart/Event-Venues/Venues('aHR0cDovL2NzaS5jb20vdG1ja2l0X3ZlbnVlcy8x')"/>
    <a:category scheme="http://docs.oasis-open.org/odata/ns/scheme" term="#Event Venues.Venues"/>
    <a:content type="application/xml">
      <m:properties>
        <d:venueName>Toyota Park</d:venueName>
        <d:eventName m:type="#Collection(String)">
          <m:element>Engelbert Humperdinck</m:element>
          <m:element>Zappa Plays Zappa</m:element>
          <m:element>Cute Is What We Aim For</m:element>
          <m:element>Toby Keith</m:element>
          <m:element>El Monstero y Los Masked Avengers</m:element>
          <m:element>Keith Urban</m:element>
          <m:element>Bow Wow</m:element>
          <m:element>Carrie Underwood</m:element>
          <m:element>Hot Tuna</m:element>
          <m:element>Jim Brickman</m:element>
          <m:element>Ana Gabriel</m:element>
          <m:element>Donna Summer</m:element>
          <m:element>Josh Turner</m:element>
          <m:element>Extreme</m:element>
          <m:element>Greg Kihn Band</m:element>
          <m:element>Loretta Lynn</m:element>
          <m:element>Monsters of Rock</m:element>
          <m:element>Built To Spill</m:element>
          <m:element>Akon</m:element>
        </d:eventName>
      </m:properties>
    </a:content>
  </a:entry>
</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](#)

[Accessing Data on Demand Endpoints](#)

Blending Data from Remote Sources (Preview)

The Anzo Graph Data Interface (GDI) service (sometimes called the Data Toolkit) is an extremely flexible and configurable service component that enables users to write SPARQL queries that access a variety of remote data sources. The GDI service has built-in, native support for various file format types as well as HTTP/REST endpoints. And the GDI service can be extended to access relational database sources by adding JDBC drivers to AnzoGraph. The data that you retrieve can be incorporated into a data layer to augment the data that is stored in Anzo without requiring you to ingest all of the data into Anzo up front.

Note

The capabilities of the Graph Data Interface are potentially endless because it enables users to freely write a multitude of SPARQL queries against virtually any Data Source or endpoint. For this reason, we have labeled the GDI as a **Preview** release. Features are considered "Preview" when the implementation has recently been incorporated into the product, significant development is still underway, or when Quality Assurance testing cannot cover all possible use cases of the feature. When employing a Preview feature, Cambridge Semantics recommends that you thoroughly test your specific use cases in a development environment before relying on the feature in a production environment.

The topics in this section introduce you to the GDI and provide instructions for exploring, analyzing, and ingesting data from remote data sources.

- [Introduction to the Graph Data Interface](#)
- [Reading Remote Source Metadata](#)
- [Reading or Ingesting Remote Instance Data](#)

Introduction to the Graph Data Interface

This topic introduces you to the Graph Data Interface (GDI) by providing setup instructions and information about the supported data sources.

Note

The capabilities of the Graph Data Interface are potentially endless because it enables users to freely write a multitude of SPARQL queries against virtually any Data Source or endpoint. For this reason, we have labeled the GDI as a **Preview** release. Features are considered "Preview" when the implementation has recently been incorporated into the product, significant development is still underway, or when Quality Assurance testing cannot cover all possible use cases of the feature. When employing a Preview feature, Cambridge Semantics recommends that you thoroughly test your specific use cases in a development environment before relying on the feature in a production environment.

- [Graph Data Interface Setup](#)
- [Supported Data Sources](#)

Graph Data Interface Setup

AnzoGraph processes Graph Data Interface (GDI) service calls using a Java plugin that is provided by your Cambridge Semantics Customer Success Manager. The plugin, **gdi-*.jar**, needs to be copied to the `<install_path>/lib/udx` directory on the AnzoGraph leader server. For more information and setup instructions, see [Deploy the Graph Data Interface Java Plugin](#).

Supported Data Sources

The GDI natively supports reading or ingesting data from HTTP/REST endpoints. In addition, the following file types are supported:

- CSV and TSV
- JSON
- XML
- Parquet
- SAS (SAS Transport XPT and SAS7BDAT formats)

To extend the service to access relational databases, JDBC drivers can also be added to AnzoGraph. For more information, see [Deploy Optional Drivers for Accessing Database Sources](#).

Note

Kubernetes-based AnzoGraph deployments are pre-configured with the GDI plugin as well as JDBC drivers for the following database types:

- Apache Derby, Hive, and Impala
- Google BigQuery
- IBM DB2
- Microsoft SQL Server
- MariaDB/MySQL
- Hyper SQL Database (HSQLDB)
- PostgreSQL
- SAP Sybase (jTDS)

Related Topics

[Reading Remote Source Metadata](#)

[Reading or Ingesting Remote Instance Data](#)

Reading Remote 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) service to explore the source's

metadata. The service can be used to return a list of the catalogs (schemas), models, columns, data types, and other data source specific information.

Note

Graph Data Interface service calls are processed by AnzoGraph using a Java plugin. Before running GDI service queries, make sure that the AnzoGraph cluster is configured to use the plugin. For more information, see [Graph Data Interface Setup](#).

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 <http://cambridgesemantics.com/services/DataToolkit>
    {
        [] s:select ?metadata .

        ?data a s:source_type ;
              s:connection_parameters .

        ?metadata a s:MetadataSource ;
                 s:from ?data ;

        ?metadata_selector [
            ?metadata_type (datatype) ;
            ... ;
        ]
    }
}
```

```

    ] .
  }
}

```

PREFIX Clause

The PREFIX clause declares the prefixes that are standard for all GDI service queries. You can declare additional prefixes to use in the query, but the PREFIX clause must include the following statements:

```

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

The result clause defines the results to return. For metadata queries, the result clause is typically **SELECT ***.

SERVICE <http://cambridgesemantics.com/services/DataToolkit>

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.

[] s:select ?metadata

Include this required triple pattern in metadata queries. The select property specifies the source that should be used to return data.

source_type

The ?data a s:source_type triple pattern specifies the type of data source that the query targets. For example, ?data a s:DbSource targets a database data source. The list below describes the commonly used types:

- **DbSource** for any type of database.
- **FileSource** for flat files. The supported file types are CSV and TSV, JSON, XML, Parquet, and SAS (SAS Transport XPT and SAS7BDAT formats).
- **HttpSource** for HTTP endpoints.

connection_parameters

The source connection parameters are the values that are required for accessing the source, such the database connection URL, path to a file source, username, password, key, token, etc.

For example, the pattern below specifies the connection details for a database source:

```
?data a s:DbSource ;
  s:url "jdbc:postgresql://10.100.2.9:5555/k1_hosp_db?
user=postgres&password=postgres123"
```

The example below specifies the connection details for a file-based source, a directory of CSV files:

```
?data a s:FileSource ;
  s:url "/opt/shared-files/sales-csv"
```

For a single file, specify the filename in the URL. For example, `"/opt/shared-files/sas/edu_inc.sas7bdat"`.

metadata_selector

The rest of the WHERE clause defines the metadata to retrieve. The **metadata_selector** specifies the type of metadata to return. The following list describes the valid selectors:

- **catalogs**: This selector narrows the results to schema-related metadata such as the schema names. Even when additional metadata types (described in the row below) are specified as objects, only catalog (schema) information is returned.
- **fields**: This selector is the broadest and most flexible option. Using the **fields** selector enables users to return any and all of the source metadata information, depending on the specified metadata types (described in the row below).
- **models**: This selector narrows the results to model-related metadata such as the model names. Even when additional metadata types (described in the row below) are specified as objects, only model information is returned.

metadata_type (datatype)

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 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.

- **?model (xsd:string)**: Returns model names in string format. For file sources, this property returns file names.
- **?field (xsd:string)**: Returns column names.
- **?catalog (xsd:string)**: Returns schema names.
- **?datatype (owl:Thing)**: Returns the data types of the columns.
- **?format (xsd:string)**: Returns the format of the source.
- **?cardinality (xsd:string)**: Returns the cardinality of relationships between tables: optional, many, or required.
- **?count (xsd:int)**: Returns the number of times the field appears in the source.
- **?order (xsd:int)**: Returns the order in which the field was encountered.

The [Metadata Query Examples](#) section below provides sample metadata queries that access various data sources.

Metadata Query Examples

This section includes sample metadata queries that run against different types of data sources.

- [View Database Schemas](#)
- [Explore a Database Schema](#)
- [Explore a Directory of SAS Files](#)
- [Explore an HTTP Endpoint](#)
- [Explore a Directory of CSV Files](#)

View 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=Mysql1@#" .

    ?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 ([View 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=Mysql11@#" .

    ?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	
http://www.w3.org/2001/XMLSchema#int		
Alphabetical list of products	Discontinued	
http://www.w3.org/2001/XMLSchema#boolean		
Alphabetical list of products	SupplierID	
http://www.w3.org/2001/XMLSchema#int		
Alphabetical list of products	UnitPrice	
http://www.w3.org/2001/XMLSchema#decimal		
Alphabetical list of products	ProductName	
http://www.w3.org/2001/XMLSchema#string		
Alphabetical list of products	QuantityPerUnit	
http://www.w3.org/2001/XMLSchema#string		
Alphabetical list of products	UnitsOnOrder	
http://www.w3.org/2001/XMLSchema#short		
Alphabetical list of products	CategoryName	
http://www.w3.org/2001/XMLSchema#string		
Alphabetical list of products	ProductID	
http://www.w3.org/2001/XMLSchema#int		
Alphabetical list of products	ReorderLevel	
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 an HTTP endpoint. The query runs against the [Dark Sky API](#), which compiles worldwide weather data.

```
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://api.darksky.net/forecast/bdbe3f638eb908c9b94919537dad5945/30.374563,-
89.975892" .

    ?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
```

For instructions on querying the instance data based on the data source metadata, see [Reading or Ingesting Remote Instance Data](#).

Related Topics

[Introduction to the Graph Data Interface](#)

[Reading or Ingesting Remote Instance Data](#)

Reading or Ingesting Remote Instance Data

Depending on the type of SPARQL query that you write, the Graph Data Interface (GDI) service can be used to ingest instance data or it can be used to analyze source data without updating the database.

Note

Graph Data Interface service calls are processed by AnzoGraph using a Java plugin. Before running GDI service queries, make sure that the AnzoGraph cluster is configured to use the plugin. For more information, see [Graph Data Interface Setup](#).

This topic provides details about the syntax to use when writing GDI queries and includes examples that demonstrate the data integration capabilities for different types of data sources.

- [GDI Query Syntax](#)
- [Deep-Dive into Hierarchical Bindings](#)
- [GDI Query Examples](#)

GDI Query Syntax

To invoke the GDI service in a data layer, add a **Query Step** or **View Step** to the layer. For instructions on creating a Query Step, see [Configuring a Query Step](#). For instructions on creating a View Step, see [Configuring a View Step](#).

The following query syntax shows the structure of a GDI 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
[ { GRAPH ${targetGraph} { ... } ]
[ ${usingSources} ]
WHERE
{
    SERVICE [ TOPDOWN ] <http://cambridgesemantics.com/services/DataToolkit>
    {
        ?data a s:source_type ;
        s:connection_parameters ;
        s:input_parameters ;
        # output_parameters
        ?variable ([ "binding" ] [datatype] [ "format" ]) ;
        ... ;
        .
    }
    # Additional clauses such as BIND, VALUES, FILTER
}

```

PREFIX Clause

The PREFIX clause declares the prefixes that are standard for all GDI service queries. You can declare additional prefixes to use in the query, but the PREFIX clause must include the following statements:

```

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

The result clause defines the type of SPARQL query to run and the set of results to return. This clause defines whether you want to read (SELECT, CONSTRUCT, or CREATE VIEW) from the remote source or ingest the data into Anzo (INSERT or CREATE MATERIALIZED VIEW).

GRAPH \${targetGraph}

Include the **GRAPH** keyword and target graph parameter **\${targetGraph}** 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.

`${usingSources}`

Include the source graph parameter **`${usingSources}`** when writing a query that passes values from the data that is in the graphmart to the remote data source. Anzo automatically populates the query with the appropriate FROM clauses and graph URIs when the query runs. When passing literal values to the remote source, you do not need to include the source graph parameter. The [SERVICE \[TOPDOWN \]](http://cambridgesemantics.com/services/DataToolkit) [description below](http://cambridgesemantics.com/services/DataToolkit) includes more information about passing input to remote sources.

WHERE

The WHERE clause includes the required GDI SERVICE call and defines the data source, input and output parameters, and any additional clauses for ingesting or analyzing the data. The descriptions below describe the patterns in the WHERE clause.

SERVICE [TOPDOWN] [<http://cambridgesemantics.com/services/DataToolkit>](http://cambridgesemantics.com/services/DataToolkit)

The SERVICE call invokes the GDI service. The optional TOPDOWN keyword is used to pass input values 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. The GDI service makes repeated calls to pass in each of the specified values and retrieve the data that is based on those values.

source_type

The `?data a s:source_type` triple pattern specifies the type of data source that the query will run against. For example, `?data a s:DbSource`. The list below describes the commonly used types:

- **DbSource** for any type of database.
- **FileSource** for flat files. The supported file types are CSV and TSV, JSON, XML, Parquet, and SAS (SAS Transport XPT and SAS7BDAT formats) .
- **HttpSource** for HTTP endpoints.

connection_parameters

The source connection parameters are the values that are required for accessing the source, such the database connection URL, path to a file source, username, password, key, token, etc. For example, the pattern below specifies the connection details for a database source:

```
?data a s:DbSource ;
  s:url "jdbc:postgresql://10.100.2.9:5555/kl_hosp_
db?user=postgres&password=postgres123"
```

The example below specifies the connection details for a file-based source, a directory of CSV files:

```
?data a s:FileSource ;
  s:url "/opt/shared-files/sales-csv"
```

For a single file, specify the file name in the URL. For example, `"/opt/shared-files/sas/edu_inc.sas7bdat"`.

input_parameters

In addition to the source connection parameters, you can include other input parameters based on the data source type. For example, the list below describes the commonly used input parameters:

- **timeout *ms***: Specifies the timeout (in milliseconds) to use for requests against the source. For example, `s:timeout 5000`.
- **refresh *s***: The number of seconds to wait before refreshing the source. If unspecified no caching will be performed and the source will be queried each time.
- **limit *n***: The maximum number of results to retrieve from the source. For example, `s:limit 1000`.
- **debug *true***: (For View Steps) Indicates that the GDI service should not cache the resulting view definition when the view is created. By default, to avoid running expensive metrics-gathering queries against the source system more than once, the GDI service caches the view definition (`s:debug` is false) when a view is first created. That means if you debug, modify, and re-run the View Step, the original view definition remains cached instead of being updated. Including `s:debug true` as part of the service call instructs the GDI service not to cache the definition.
- **selector "*path*"**: The selector is a source-specific binding component that identifies the path to the source object. For example, `s:selector "currently"` targets the "currently" class of data from a source. And `s:selector "hourly.data"` specifies a hierarchical path. For more information about the selector property and bindings, see the [Deep-Dive into Hierarchical Bindings](#) section below.

output_parameters

The output parameters, in `?variable (["binding"] [datatype] ["format"])` format, define the triple patterns to output. When the specified `?variable` matches the source column name, the GDI uses the variable as the source data selector. If you specify an alternate variable name, then a **binding** needs to be specified to map the new variable to the source.

In the object position in parenthesis, you also have the option to transform the data using the **datatype** and **format** options. The list below describes each option:

- **binding**: The binding 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, then the binding is needed to map the new variable to that source column.

For example for a flat source like CSV, the following pattern simply binds the source column AIRLINE to the lowercase variable `?airline`: `?airline ("AIRLINE")`. For a database source, this example binds the `?subject` variable by navigating to the SUBJECT column in the FILM table in the dbo schema: `?subject`

`("dbo.FILM.SUBJECT")`. And for an HTTP source, this example binds the `?time` variable to the time object under the minutely data path: `?time ("minutely.data.time")`.

Note

For **FileSource** and **HttpSource**, 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 **average/day**, the variable and binding pattern could be written as `?averagePerDay ("average\\ /day")`.

For **DbSource**, 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 `(_|A-Z|a-z)(_|A-Z|a-z|0-9)*` should be quoted, such as `("'Adventure.Works'.Sales.'Daily.Totals'")`.

- **datatype**: The datatype is the data type to convert the column to. If you do not specify a data type, the GDI infers the type. See [Property Range Guidelines](#) for information about data types.
- **format**: The format option is used to specify the format to use for data types such as `xsd:date` or `xsd:dateTime`. 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 `"ddMMYY"` to display date values such as "01JAN19."

Deep-Dive into Hierarchical Bindings

As part of the GDI's flexibility, there are multiple ways to express binding hierarchies in GDI queries. One way is to use brackets (`[]`) to group triple patterns into binding trees. For example, the WHERE clause snippet below organizes output parameters into an `hourly.data` hierarchy:

```
WHERE
{
  SERVICE [ TOPDOWN ] <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
  [
    s:selector "data" ;
```

```

        ?time (xsd:long) ;
        ?summary (xsd:string) ;
        ?nearestStormDistance (xsd:int) ;
        ?rainIntensity ("precipIntensity" xsd:double) ;
        ?rainProbability ("precipProbability" xsd:double) ;
        ?temperature (xsd:double) ;
        ?feelsLike ("apparentTemperature" xsd:double) ;
        ?humidity (xsd:double) ;
        ?pressure (xsd:double) ;
        ?windSpeed (xsd:double) ;
    ] ;
] .
}
}

```

Each level of the binding hierarchy above has a **s:selector** property that is used to navigate the data. The selector is optional, however. If a selector is not specified, the output triple patterns default to the name of the variable that introduces that level of the hierarchy.

As an alternative to grouping triple patterns in bracketed trees, the s:selector property can specify a path. For example, the WHERE clause snippet below rewrites the example above to express the same hierarchy by specifying a path as the value for s:selector:

```

WHERE
{
    SERVICE [ TOPDOWN ] <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) ;
            ?nearestStormDistance (xsd:int) ;
            ?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 working with schema-less (or schema-flexible) sources like JSON, you can also capture a tree of data as a JSON string. For example, using the query snippet above, if the properties under hourly were unknown, the snippet could be rewritten as follows. This query would bind all of the data below hourly to the **?hourly** variable and return a JSON string representation of the properties and instance data:

```
WHERE
{
  SERVICE [ TOPDOWN ] <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 () .
  }
}
```

For example, the results look like this:

latitude	longitude	timezone	hourly
30.374563	-97.975892	America/Chicago	{ "summary": "\"Humid and partly 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 [ TOPDOWN ] <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 ("@" ) ;
  ] .
}

```

The results look like this:

latitude	longitude	timezone	time	summary	hourly_data
30.374563	-97.975892	America/Chicago	1595559600	Clear	<pre> {"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","win dSpeed":"7.48", "windGust":"16.71","windBearing":"109","cloudCover":"0.06","uvIndex":"0","visibility":"10" ,"ozone":"285.2"} </pre>
30.374563	-97.975892	America/Chicago	1595563200	Clear	<pre> {"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","winds peed":"7.05","windGust":"17.56", "windBearing":"110","cloudCover":"0.12","uvIndex":"0","visibility":"10","ozone":"284.9"} </pre>
30.374563	-97.975892	America/Chicago	1595566800	Partly Cloudy	<pre> {"time":"1595566800","summary":"\"Partly Cloudy\"", "icon":"\"partly-cloudy-night\"", "precipIntensity":"3.0E-4","precipProbability":"0.01", "precipType":"\"rain\"", "temperature":"85.63","apparentTemperature":"89.21", </pre>

```
"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"}
...
```

GDI Query Examples

The SELECT query below also reads data from an HTTP source, the [Dark Sky API](#), which compiles worldwide weather statistics. The API 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 includes values to be input to the source to narrow the results to specific locations. 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 API service requires parameters to be specified in the connection URL, the `s:url` value includes `?lat` and `?long` as parameters for 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://api.darksky.net/forecast/bdbe3f638eb908c9b94919537dad5945/{?lat}},
    {?long}" ;
    s:selector "currently" ;
    ?lat ("latitude") ;
    ?long ("longitude") ;
    ?time () ;
    ?summary () ;
    ?temp ("temperature") ;
    ?rainChance ( "precipProbability" ) ;
    ?rainIntensity ( "precipIntensity" ) ;
    ?humidity () ;
```



```

    ?pressure () ;
    ?windSpeed () ;
    ?windGust () ;
    ?windBearing () ;
    ?nearestStorm ("nearestStormDistance") ;
    ?stormBearing ("nearestStormBearing") ;
    ?visibility () .
}
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 Query 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 "jdbc:postgresql://10.10.5.3:5555/kl_hosp_
db?user=postgres&password=postgres123" ;
        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(IRI("http://example.com/inputevent_cv/{{?row_id}}") AS ?InputEvent_cv)
    BIND(IRI("http://example.com/patients/{{?subject_id}}") AS ?patient)
    BIND(IRI("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)  
  }  
}
```

Related Topics

[Configuring a Query Step](#)

[Configuring a View Step](#)

[Introduction to the Graph Data Interface](#)

[Reading Remote Source Metadata](#)

Sharing Access to Artifacts

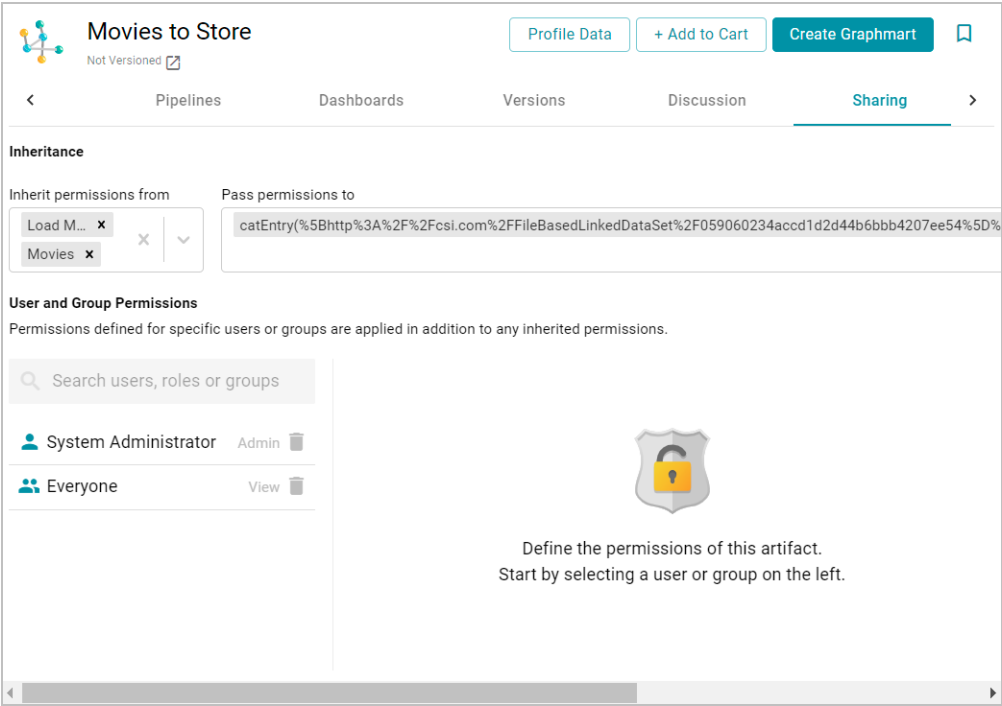
All Anzo artifacts—Data Sources, Schemas, Models, Mappings, Pipelines, Graphmarts, etc.—that you create can be shared with other groups (or users) from the **Sharing** tab in the Anzo application. This topic provides an overview of the Sharing tab and basic instructions for configuring artifact permissions.

Note

For specifics about sharing multifaceted artifacts like Graphmarts that include Data 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 in the Anzo application and clicking **Sharing**. For example, the image below shows the Sharing tab for a Dataset in the Datasets catalog.



Inheritance

The top of the screen displays the permission inheritance settings:



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 **Copy permissions from**. For example, this sample Dataset artifact shown below is configured to copy permissions from the Load Movie Data Pipeline and Movies Data Source.

Inherit permissions from

Load Movies to Store 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](#).

You can also configure an artifact to **Copy permissions to** other artifacts. For example, by default Graphmarts are configured to copy their permissions to their Data Layers, Steps, and Data on Demand endpoints.

Pass permissions to

Venues x

Tickets to Store x

Load Tickets to Store x

x

▼

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.

Search users, roles or groups

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 type="radio"/> View	<input type="radio"/> Modify	<input type="radio"/> Admin	<input type="radio"/> Custom
Add/Edit		✓	✓	✓
View	✓	✓	✓	✓
Delete		✓	✓	✓
Meta Add/Edit			✓	✓
Meta View	✓	✓	✓	✓
Meta Delete			✓	✓

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:

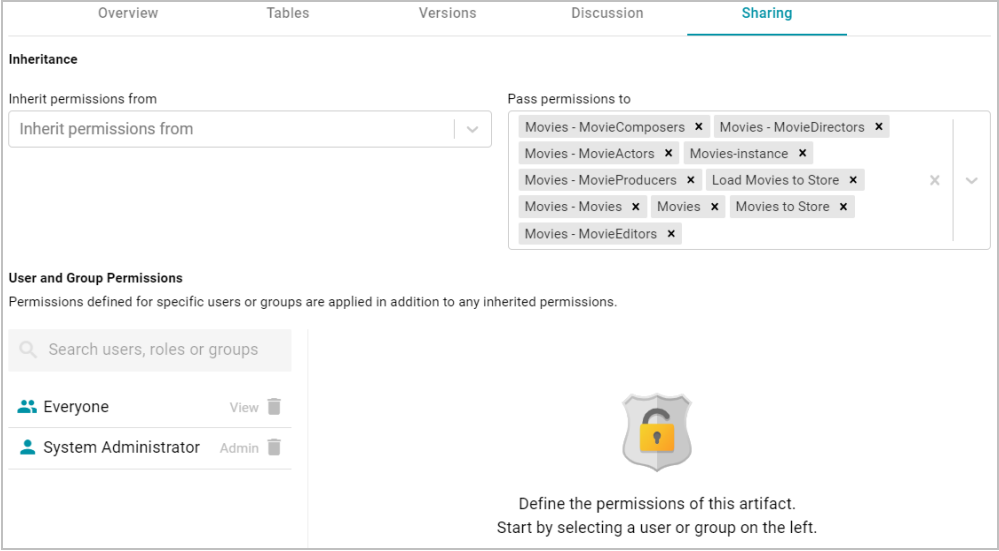
Set	Permission	Allows a user to:
View	View	<ul style="list-style-type: none">• See the artifact in the Anzo application.• Create versions of the artifact.
	Meta View	<ul style="list-style-type: none">• Relates only to an artifact's permissions. A user with Meta View can see the permissions on the Sharing tab but they cannot modify, add, or remove permissions.
Modify	In addition to the View and Meta View permissions described above, the Modify set includes the Add/Edit and Delete permissions described below.	
	Add/Edit	<ul style="list-style-type: none">• Change an artifact, such as to rename it or edit its description.• Add an entity to an artifact. For example, add a Schema to a Data Source or a Data Layer to a Graphmart.
	Delete	<ul style="list-style-type: none">• Remove an entity from the artifact. For example, delete a Data Layer from a Graphmart or a Schema from a Data Source.• Does not give permission to remove the parent artifact. For example, a user can remove a Schema from a Data Source but cannot delete the Data Source.

Set	Permission	Allows a user to:
Admin		In addition to the View , Meta View , Add/Edit , and Delete permissions described above, the Admin set includes the Meta Add/Edit and Meta Delete permissions described below.
	Meta Add/Edit	<ul style="list-style-type: none">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.
	Meta Delete	<ul style="list-style-type: none">Remove permissions from a user or group.Delete the parent artifact and its related entities.

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. For example, the image below shows the Sharing screen for a Data Source. By default the Data Source is configured to pass its permissions to the Mappings, Model, and Pipeline for this source:

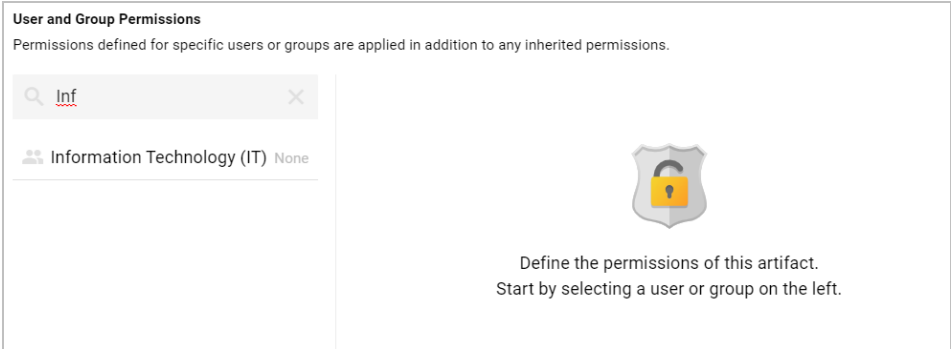


2. If you want to change the inheritance for the artifact, use the fields at the top of the screen:
 - To apply all of the permissions from another artifact to this one, select the artifact to inherit from in the **Inherit permissions from** field.
 - To pass this artifact's permissions to other artifacts, select the artifacts to pass permissions to in the **Pass permissions to** field.

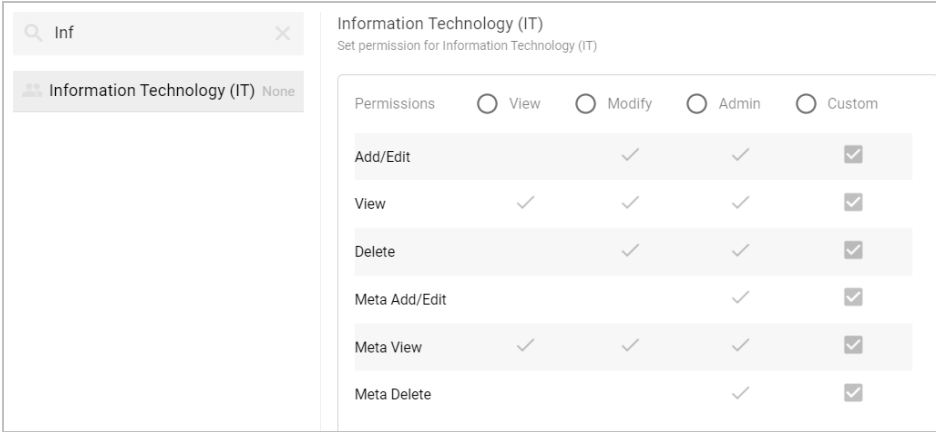
Note

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](#).

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**):



4. 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:



5. 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:

Inf

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](#)

Accessing and Analyzing Data

Once data has been onboarded, modeled, and blended into the Dataset catalog and graphmarts, users have several options for accessing and analyzing the data. Anzo provides 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 in the user interface enables users to find specific statements or write and run SPARQL queries. Users can also access data remotely from the SPARQL endpoint, HTTP client interface, 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](#)
- [Accessing Data on Demand Endpoints](#)
- [Accessing Data with the Query Builder](#)
- [Accessing Data from the SPARQL Endpoint](#)
- [Accessing Data from the HTTP Client Interface](#)
- [SPARQL Query Templates and Best Practices](#)

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. This section also includes reference information about the available filters and lenses as well as the functions that are available for calculating the values to display.

- [Introduction to Hi-Res Analytics](#)
- [Getting Started: Exploring and Visualizing Data](#)
- [Creating Dashboards](#)
- [Creating a Lens](#)
- [Creating Dashboard Filters](#)
- [Combining Data from Multiple Classes](#)
- [Calculating Values in Lenses and Filters](#)
- [Searching for Text in Unstructured Documents](#)
- [Sharing Access to a Dashboard or Lens](#)
- [Exporting a Lens](#)
- [Deleting a Lens](#)

- [Function and Formula Reference](#)
- [Lens Type Reference](#)

Related Topics

[Routing Hi-Res Analytics to a Custom URL](#)

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 Creating a Lens .
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 Creating Dashboard Filters .

Term	Description
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.
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 Calculating Values in Lenses and Filters .

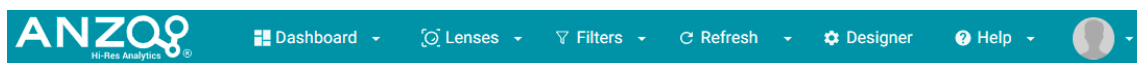
User Interface Overview

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.

The screenshot shows the ANZO 5.2 user interface. The top navigation bar includes the ANZO logo, a search icon, and menu items for Dashboard, Lenses, Filters, Refresh, Designer, and Help. Below the navigation bar, there are tabs for Northwind (g), Orders (g), and Tickets (g). The main content area is divided into a left sidebar and a main table view. The sidebar contains sections for Graphmart, Data Types (Customers (93)), Data Layers (DB northwind), and City (show sort details). The main table view displays a table with columns: Address, City, CompanyName, ContactName, Country, and Phone. The table contains 20 rows of data, including entries from Toulouse, Walla Walla, Marseille, London, Lille, Boise, Vancouver, Lyon, Tsawwassen, Strasbourg, Paris, Eugene, Anchorage, Albuquerque, Seattle, and London. The bottom of the table shows pagination information: Rows per page: 20, 1 - 20 of 93, Page 1 of 5.

Main Toolbar

The toolbar at the top of the screen provides the following options:

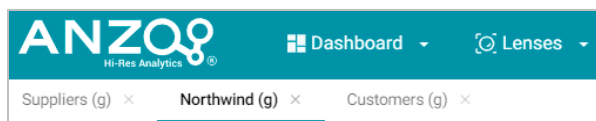


- **Dashboard:** Accesses dashboard functions, including Save.
- **Lenses:** Creates or opens lenses.
- **Filters:** Creates or manages selected filters.
- **Refresh:** Accesses the automatic refresh check box. Select this box to refresh data automatically. New data will appear and change according to changes elsewhere.
- **Designer:** Controls dashboard layout and design.

- **Help:** Opens help options.
- **User:** Click to access the **Logout** button.

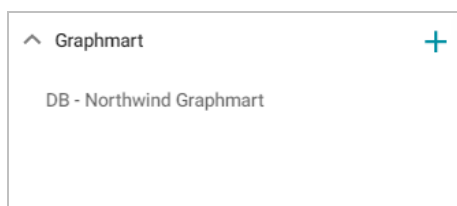
Dashboard Tabs

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.



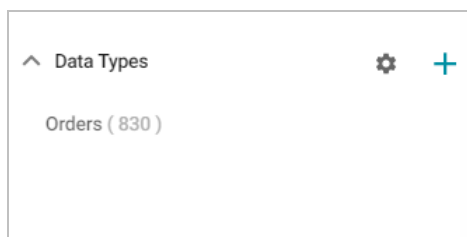
Graphmart

The Graphmart panel displays the selected Graphmart for the dashboard:



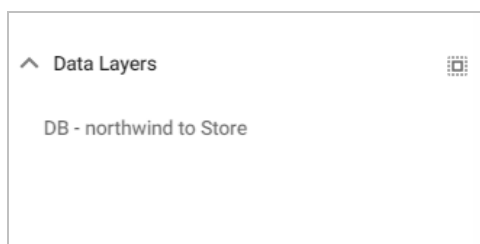
Data Types

The Data Types panel displays the selected data type from the Graphmart.



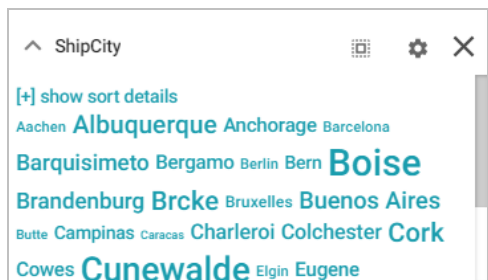
Data Layers

The Data Layers panel displays the Data Layers in the Graphmart.



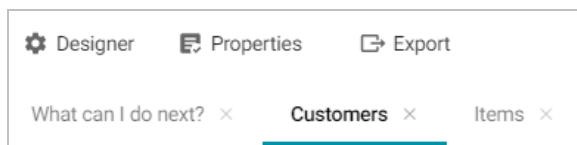
Filters

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 objects, and the toolbar enables you work with the object properties.



Related Topics

[Getting Started: Exploring and Visualizing Data](#)

[Creating Dashboards](#)

[Creating a Lens](#)

[Creating Dashboard Filters](#)

[Combining Data from Multiple Classes](#)

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:

<div> <div> <div></div> <div>Search</div> </div> <div>Sort By: Title</div> <div>View:  </div> </div> <div>Add Graphmart</div>								
<input type="checkbox"/>	Title	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions
	DB - northwind to Store Graphmart	System Admi...	 Ready to us	9,172	2 hours ago	2 hours ago		 
	Datafox Graphmart	System Admi...						 
	Flight Data Graphmart	System Admi...	 Ready to us	265,696	2 minutes ago	2 minutes ago		 
	Tickets Graphmart	System Admi...	 Ready to us	4,088,422	5 hours ago	5 hours ago		 


- On the Graphmarts screen, click the name of the Graphmart for which you want to create a dashboard. Anzo displays the graphmart overview. For example:


DB - Northwind Graphmart

Save a Copy

Profile Data

Create Dashboard



Not Versioned

INACTIVE

ACTIVE

Ready to use

AnzoGraph

Static

Overview

Explore

Datasets

Data Layers

Dashboards

Data on Demand

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

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.

General

Type : Graphmart

Creator : System Administrat...


Last Accessed : 2 hours ago

Last Updated : 3 hours ago

Structure Modified : 3 hours ago

Released : 3 hours ago

http://cambridgesemantics.com/Gra...



AnzoGraph Server Details

AnzoGraph : AnzoGraph

Status : Ready to use

Last Accessed : 2 hours ago

Memory Used : 189.7 MB(2%)

Memory Total : 11.7 GB

Inactivity Deactivate Timeout

None

Tags

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**.
- 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:*

Northwind Orders

Description:

Graphmart:*

DB - Northwind Graphmart

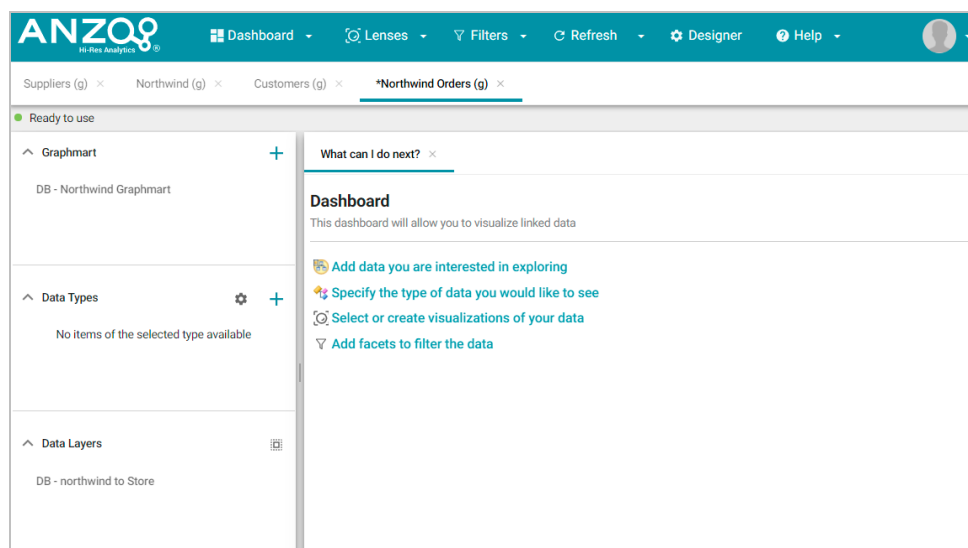
CANCEL

PREVIOUS

FINISH

© 2023 Cambridge Semantics, Inc.

- 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:



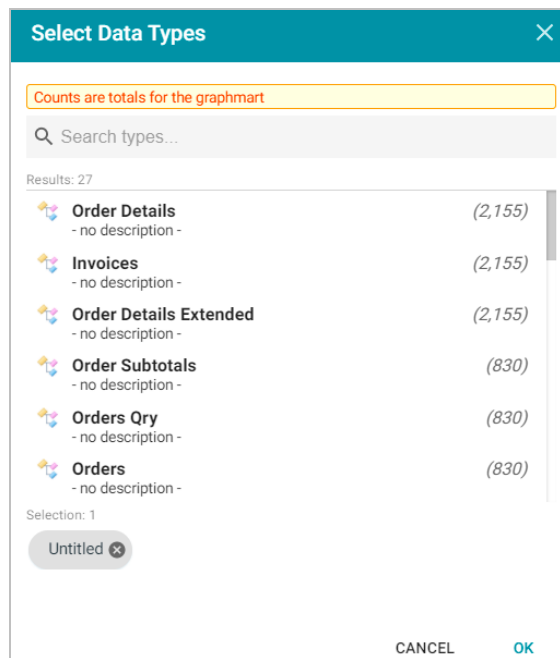
- In the main toolbar, click the **Dashboard** button and select **Save**. Proceed to [Explore the Data](#) below for guidance on next steps.

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

- 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:

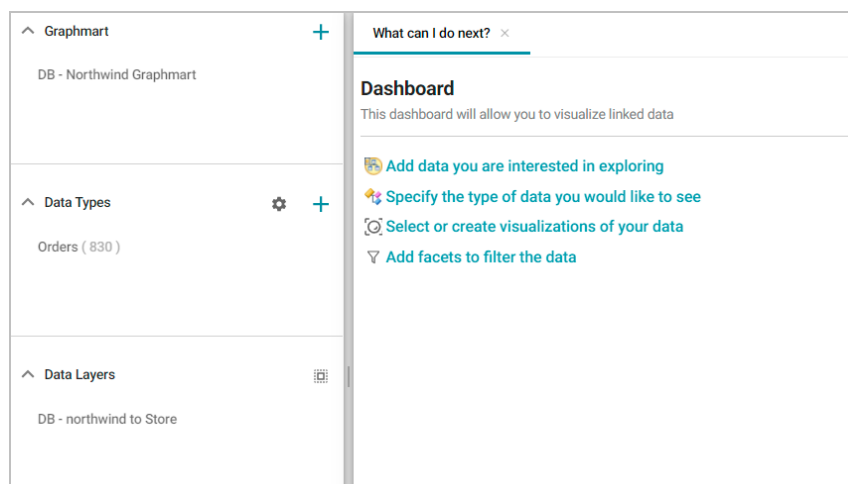


2. Select one data type. The property that you choose determines the fields that become available to filter on.

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. For example:



Proceed to [Create Filters to See the Values for Properties](#) below for next steps.

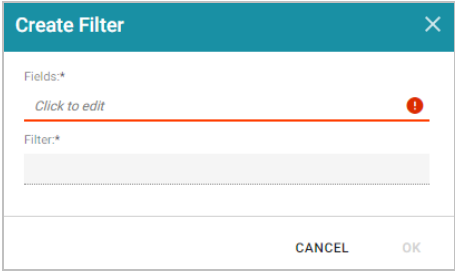
Tip

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, for example, 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.

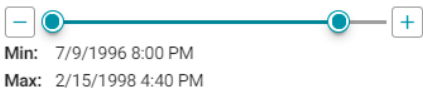



- 2. In the Create Filter dialog box, click the **Fields** field and select the property or property path to filter on.
- 3. Then click the **Filter** field to select the filter type. The list of available choices depends on the data type of the property you selected in Fields. The table below describes each filter type.

Filter Types

Filter	Description
<p>Cloud</p> <p>Argentina Austria Belgium Brazil Canada Denmark Finland France Germany Ireland Italy Mexico Norway Poland Portugal Spain Sweden Switzerland UK USA Venezuela</p>	<p>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.</p>

Filter	Description
List <input type="checkbox"/> Aachen (6) <input type="checkbox"/> Albuquerque (18) <input checked="" type="checkbox"/> Anchorage (10) <input checked="" type="checkbox"/> Barcelona (5) <input type="checkbox"/> Barquisimeto (14) <input type="checkbox"/> Bergamo (10) <input checked="" type="checkbox"/> Berlin (6)	<p>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.</p>
Single Select List AK (10) BC (17) CA (4) Co. Cork (19) DF (2) Essex (13)	<p>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.</p>
Limit Include the <input type="text" value="10"/> <input type="text" value="Largest"/>	<p>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.</p>
Date Range 7/3/1996 8:00 PM - 12/30/1996 8:00 PM (151) 12/30/1996 8:00 PM - 6/28/1997 8:00 PM (184) 6/28/1997 8:00 PM - 12/25/1997 8:00 PM (215) 12/25/1997 8:00 PM - 6/23/1998 8:00 PM (280) 4 choices	<p>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.</p>
Numeric Range 10,248 - 10,348 (100) 10,348 - 10,448 (100) 10,448 - 10,548 (100) 10,548 - 10,648 (100) 10,648 - 10,748 (100)	<p>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.</p>

Filter	Description
Range Slider  Min: 7/9/1996 8:00 PM Max: 2/15/1998 4:40 PM	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.
Relative Time Last 3 months	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.
Search 	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.
Presence Exists (69) Does not exist (24)	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.
Quartile 4(Range: 4,228.00-20,000.00) (48,125) 3(Range: 2,009.00-4,228.00) (48,124) 2(Range: 822.00-2,009.00) (48,124) 1(Range: 20.00-822.00) (48,124)	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.
Hierarchy <ul style="list-style-type: none"> - <input type="checkbox"/> Metabolism (3) <ul style="list-style-type: none"> <input type="checkbox"/> Carbohydrate transport and metabolism (1) <input type="checkbox"/> Amino acid transport and metabolism (1) + <input type="checkbox"/> (+)-neomenthol dehydrogenase (1) + <input type="checkbox"/> (+)-delta-cadinene synthase isozyme C2 (1) 	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 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.

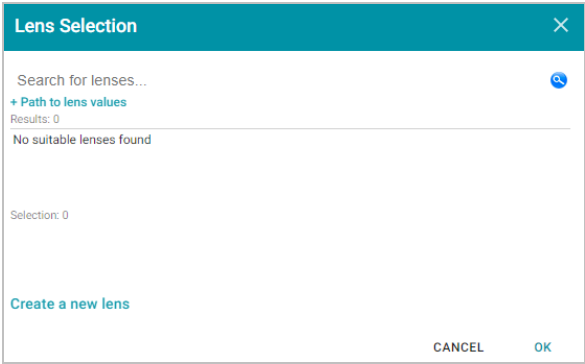
Filter	Description
<div>Types<div>Search...</div><div>FrameClass (93) Class (93) Resource (93)</div></div>	Types Filters are used to filter data according to the types of data (classes) that are connected by a specified path. This type of filter is available only for paths and not properties.

For information about configuring each type of filter, see [Creating Dashboard Filters](#).

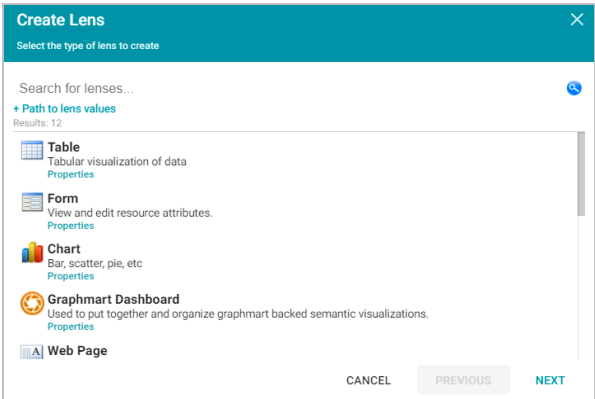
Create Visualizations of the Data

Once you have a good understanding of the values and relationships that exist in the data set, 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.

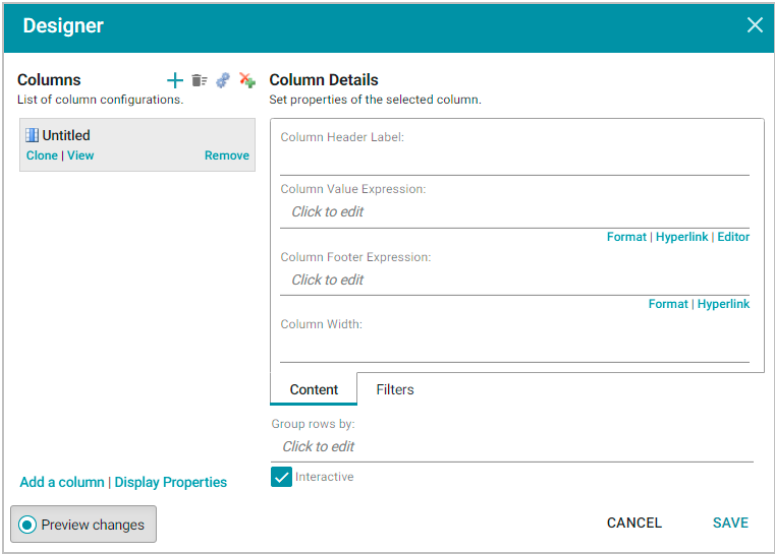


2. In the dialog box, click **Create a new lens**. Anzo displays the Create Lens dialog box.













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 [Lens Type Reference](#).

Lens	Description
AnzoKO Web Page	AnzoKO Web Page lenses include the Knockout JavaScript framework and enable you to display data on a web page that you create using HTML, CSS, and JavaScript.
Chart <div><div> Column</div><div> Heat Map</div><div> Bar</div><div> Pie</div><div> Line</div><div> Polar</div><div> Bubble</div><div> Scatter</div><div> Funnel</div><div> Area</div></div>	Anzo offers several types of Chart lenses. These lenses are useful for displaying large 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.

Lens

Description

Drill Down

	dateid	↓	eventid
▼	2,191		3,305
▼	2,191		1,594
▼	2,191		3,305
▼	2,191		7,192

Form

aircraft type

Airliner


designed by

Reginald Kirshaw Pierson


maximum speed

180.246528


List




Abbas Kiarostami




Abe Vigoda




Adrian Dunbar



Adriana Asti



Aki Kaurismäki



Al Lettieri

Query

Resource Tree Navigator

Order Details

FK_Order_Details...

FK_Order_Details...

Vins et alcools ...

FK_Orders_Custom...

FK_Orders_Employ...

FK_Orders_Shippe...

Drill Down lenses create clickable data points that enable you to drill down to view additional details. You can specify multiple lenses within the Drill Down lens so that clicking a data point presents the data in a different view.

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.

Note

Form lenses are valid in Linked Data Set Dashboards. In Graphmart Dashboards, form lenses do not display instance data.

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.

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.

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.

Lens

Description

Table

Category	Event	numtickets
Musicals	Legally Blonde	22
Musicals	Flower Drum Song	10
Musicals	Spamalot	20
Musicals	Mamma Mia!	12
Musicals	Pal Joey	12
Musicals	A Chorus Line	22

Table lenses present results in a basic table grid consisting of rows and columns. Table lenses are useful for presenting data aggregates or summaries.

Web Page

Web Page lenses enable you to display results on a web page that you create using HTML, CSS, and JavaScript.

Related Topics

[Creating Dashboards](#)

[Creating Dashboard Filters](#)

[Creating a Lens](#)

[Sharing Access to a Dashboard or Lens](#)

Creating Dashboards

Once your Datasets have been added to a Graphmart and the Graphmart is active, there are two types of dashboards that you can create to explore and analyze the 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 \(Preview\)](#)

Related Topics

[Creating a Lens](#)

[Creating Dashboard Filters](#)

Sharing Access to a Dashboard or Lens

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="checkbox"/>	Title	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions
	DB - northwind to Store Graphmart	System Admi...	Ready to us	9,172	2 hours ago	2 hours ago		Bookmark More
	Datafox Graphmart	System Admi...	Ready to us					Bookmark More
	Flight Data Graphmart	System Admi...	Ready to us	265,696	2 minutes ago	2 minutes ago		Bookmark More
	Tickets Graphmart	System Admi...	Ready to us	4,088,422	5 hours ago	5 hours ago		Bookmark More

2. On the Graphmarts screen, click the name of the Graphmart for which you want to create a dashboard. Anzo displays the graphmart overview. For example:

DB - Northwind Graphmart
Save a Copy Profile Data Create Dashboard [Bookmark](#)

Not Versioned
INACTIVE ACTIVE Ready to use AnzoGraph Static

Overview Explore Datasets Data Layers Dashboards Data on Demand

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 ☒

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. ☐

General

Type : Graphmart

Creator : System Administrat...

Last Accessed : 2 hours ago

Last Updated : 3 hours ago

Structure Modified : 3 hours ago

Released : 3 hours ago

<http://cambridgesemantics.com/Gra...>

AnzoGraph Server Details

AnzoGraph : [AnzoGraph](#)

Status : Ready to use

Last Accessed : 2 hours ago

Memory Used : 189.7 MB(2%)

Memory Total : 11.7 GB

Inactivity Deactivate Timeout

None

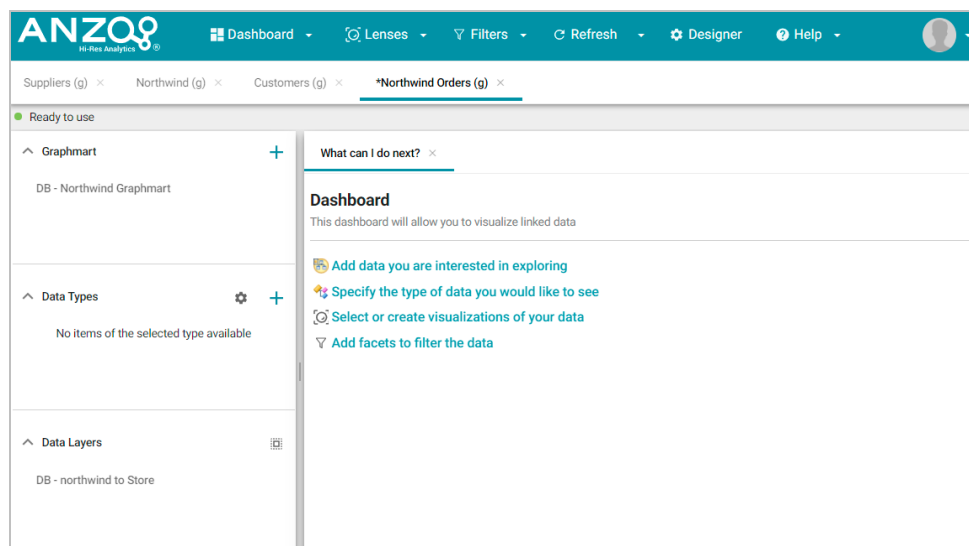
Tags

None

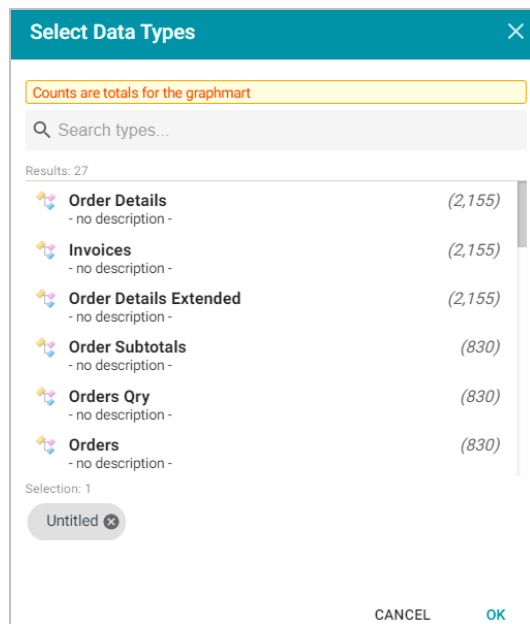
3. Click the **Create Dashboard** button. The Hi-Res Analytics application opens and displays the Create Dashboard dialog box. Leave **Graphmart Dashboard** selected and click **Next**.

4. Next, type a name for the dashboard in the **Title** field and enter an optional **Description**.

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 data set:

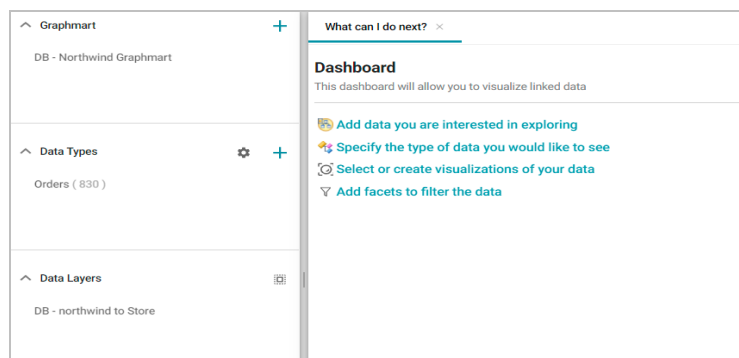


- In the Select Data Types dialog box, select the data type or class of data that you want to display on the dashboard. Anzo uses the type, along with any filters, to populate the visualizations (lenses) that you add to the dashboard.

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.

- 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 and additional options become available on the What can I do next tab. For example:



- 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 a Lens](#) and [Creating Dashboard Filters](#) for instructions on adding lenses and filters to the dashboard.

Related Topics

- [Creating a Lens](#)
- [Creating Dashboard Filters](#)
- [Combining Data from Multiple Classes](#)
- [Sharing Access to a Dashboard or Lens](#)

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:

<div><div><div><div></div></div><div>Search</div></div><div>Sort By: Title</div><div>View: <div><div></div><div></div></div></div><div>Add Graphmart</div></div>							
	Title	Description	Status	Statements	Last Accessed	Last Updated	
	Automobile Supply Chain Risk		Ready to use	32,118	an hour ago	an hour ago	
	DB - emrddb to store Graphmart						
	Movies to store Graphmart		Ready to use	176,754	4 minutes ago	4 minutes ago	
	Tickets to store Graphmart		Ready to use	4,240,857	an hour ago	an hour ago	

2. On the Graphmarts screen, click the name of the Graphmart for which you want to create a dashboard. The Graphmart Overview is displayed. For example:

The screenshot shows the AnzoGraph interface for a graphmart named "Automobile Supply Chain Risk". At the top, there are buttons for "Save a Copy", "Profile Data", and "Create Dashboard". Below these, the status is "INACTIVE" with a toggle switch set to "ACTIVE" and a "Ready to use" indicator. The connection is "azg" and "Static".

The main area has tabs for "Overview", "Explore", "Datasets", "Data Layers", "Dashboards", and "Data on Demand". The "Overview" tab is selected, showing a description of "None", a load priority of "100", and various settings like "Metrics", "Disable Load Counts", "Data Loading Settings", "Leave Graphmart Online During Refresh", "Manual Refresh Graphmart", and "Generate Statistics After Each Layer".

On the right, there is a sidebar with "General" information (Type: Graphmart, Creator: System Administrator, Last Accessed: an hour ago, Last Updated: an hour ago, Structure Modified: an hour ago, Released: Jul 27, 2021 5:03 PM) and "AnzoGraph Server Details" (AnzoGraph: azg, Status: Ready to use, Last Accessed: 6 minutes ago, Memory Used: 255.46 MB(5%), Memory Total: 5.56 GB). There is also an "Inactivity Deactivate Timeout" set to "None" and "Tags" set to "None".

- 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.

The "Create Dashboard" dialog box is shown, with a title bar and a close button. Below the title bar, it says "Select the type of dashboard to create". There are two options listed:

- Graphmart Dashboard**: Set up a new Graphmart dashboard. Graphmart dashboards display data from graphmarts. (Properties)
- Network Navigator Dashboard**: Interactively explore Graphmart data. (Properties)

The "Network Navigator Dashboard" option is highlighted. At the bottom, there are buttons for "CANCEL", "PREVIOUS", and "NEXT".

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:*

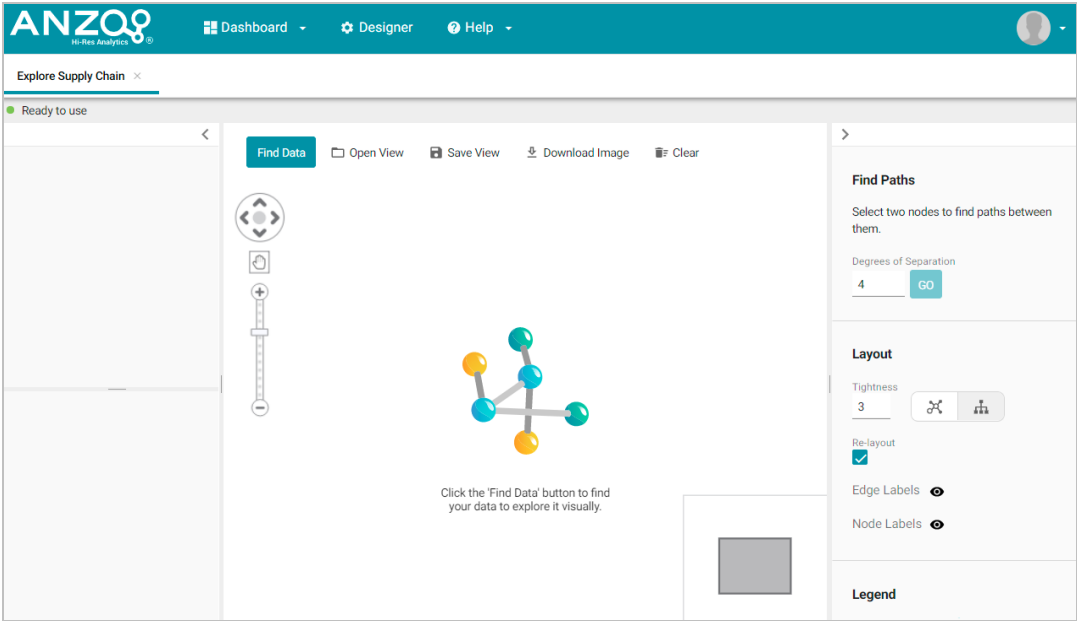
Automobile Supply Chain Risk

CANCEL

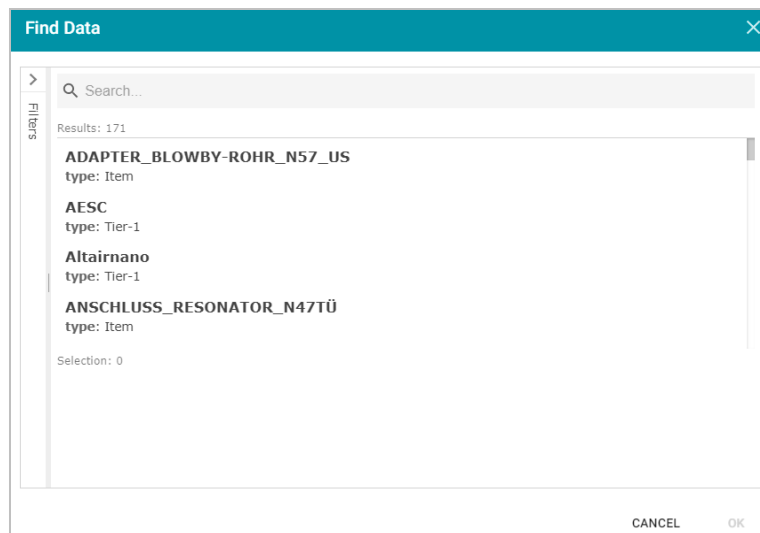
PREVIOUS

FINISH

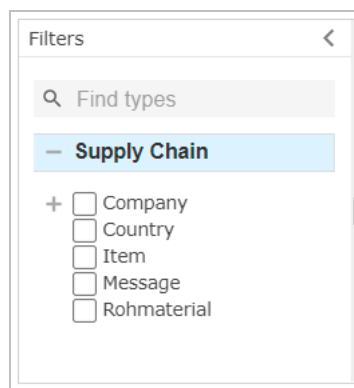
5. Click **Finish** to create the dashboard. The dashboard appears as a new tab in the application. For example:



6. 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.

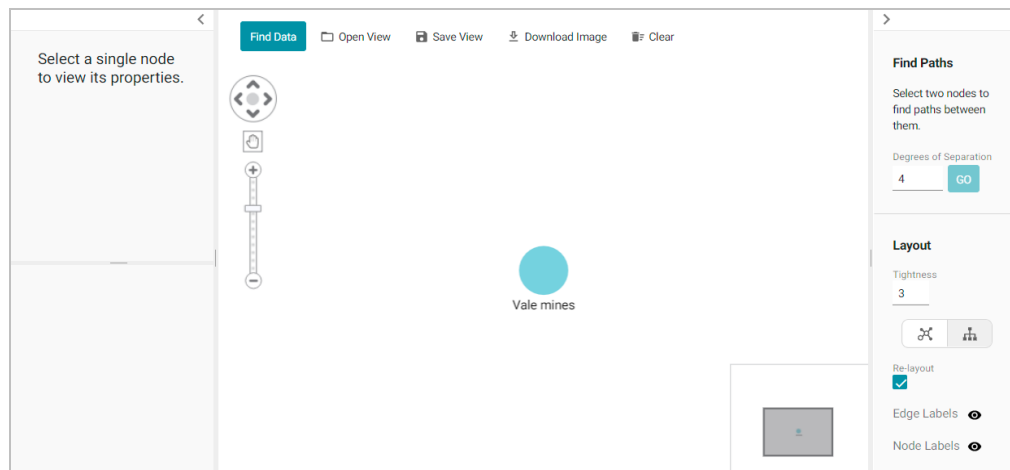


To narrow down the list of values by class or type of data, you can click **Filter** to open the Filter panel on the left side of the screen. The Filter panel lists the classes and subclasses from the model. For example, the image below shows the classes for a knowledge graph that contains supply chain 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.

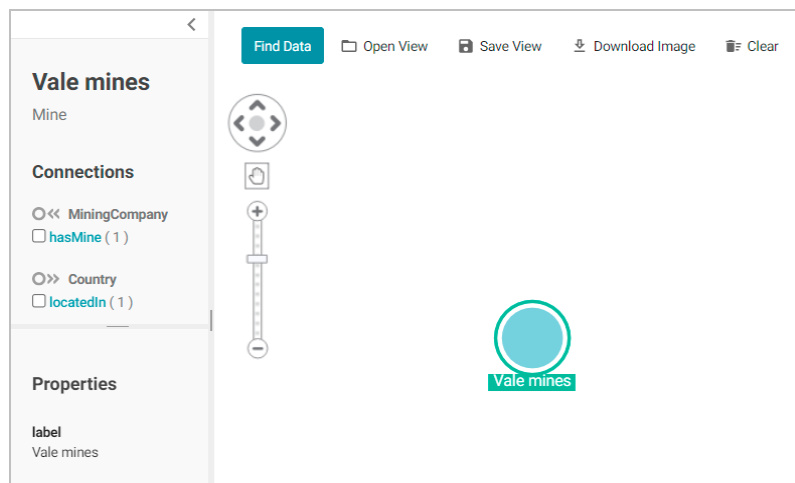
- Once you have selected the starting node in the Find Data dialog box, click **OK** to add the node to the Network View. For example, in the image below the value **Vale mines** was selected as the starting point, and the Vale mines node is added to the View:



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.

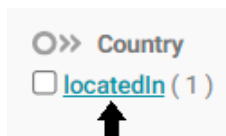


The image shows that Vale mines has an incoming connection (or backward path ◀◀) from **Mining Company** via **hasMine** and an outgoing connection (or forward path ▶▶) to **Country** via **locatedIn**.

- There are two options for exploring the connections:

See a list of connected nodes without adding them to the View

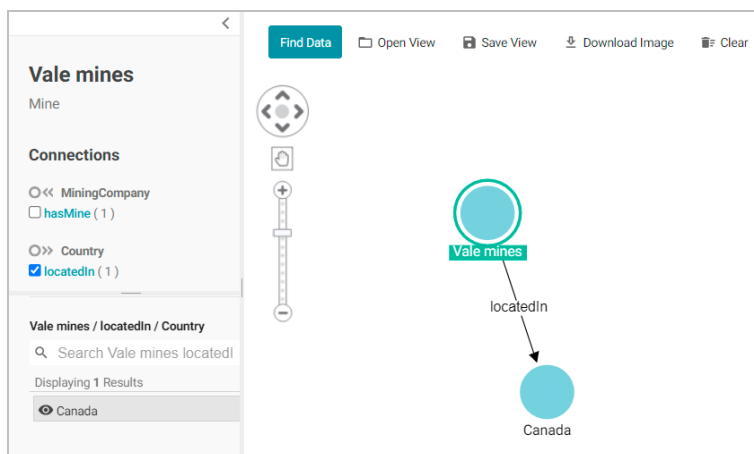
To see a list of nodes that are connected by a path without adding the nodes and path to the View, you can click the path name link.



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.

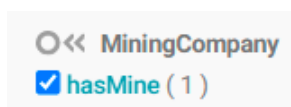


If you decide you want a node and path to be added to the View, you can click the hidden node to make it visible. For example, clicking **Canada** adds the path and node to the View:

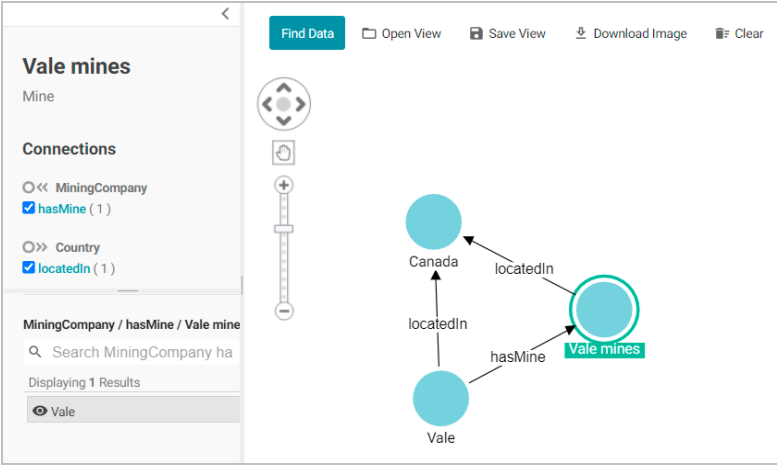


Add a path and node directly to the View

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:



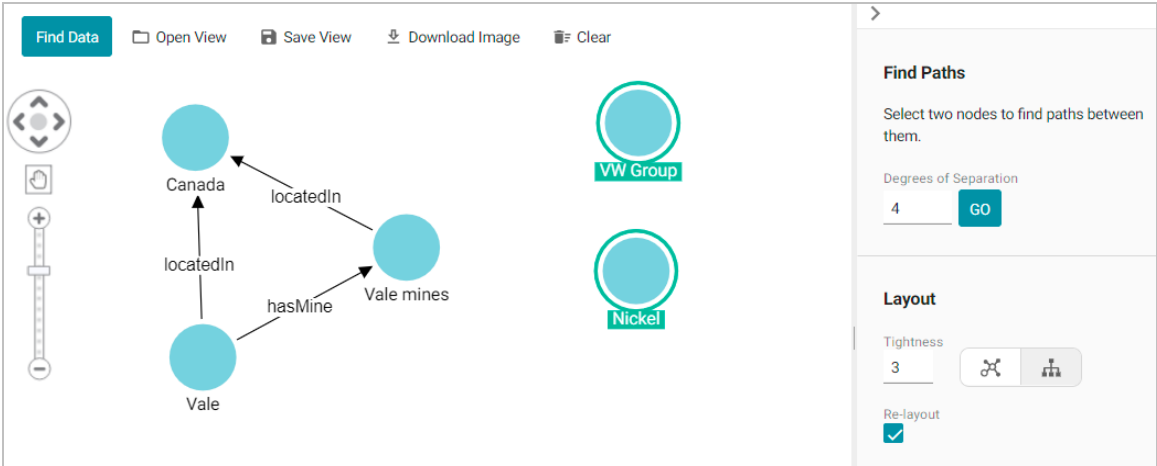
Tip

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.

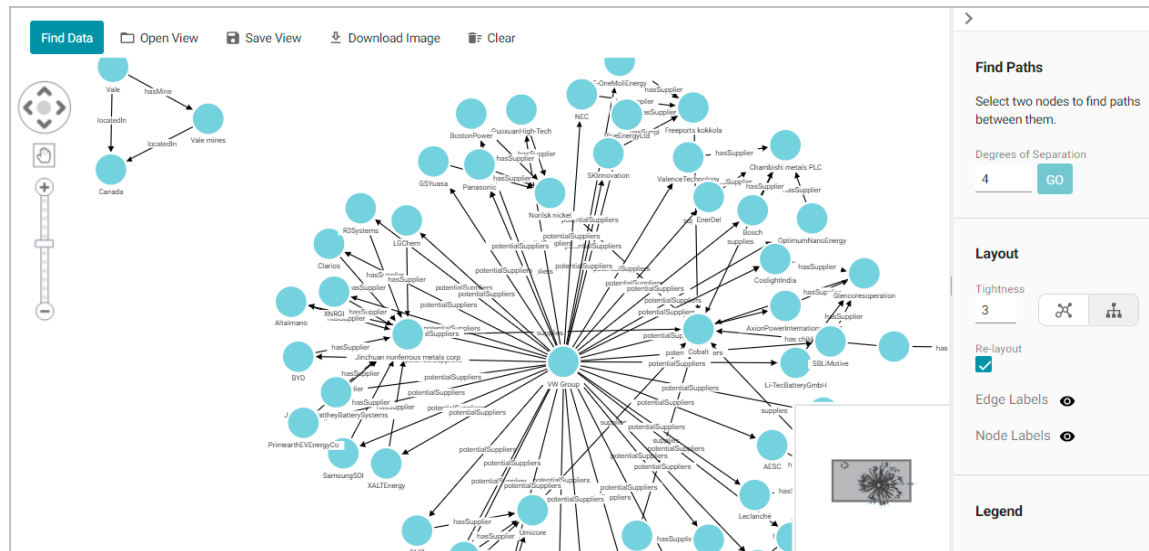
10. To continue to explore and expand the Network View, you can select another node and repeat the previous step to identify additional paths to follow. Or you also have the option to select two nodes and do a search for paths between them. To find paths, find and add to the view, if necessary, the two 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:



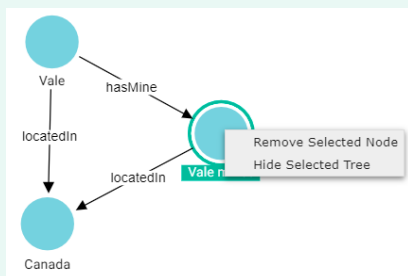
11. On the right side of the screen under **Find Paths**, adjust the **Degrees of Separation** value as needed (the default is 4) and then click **Go**. Anzo runs an All Paths graph algorithm to find all of the paths that exist between the two nodes—where the maximum number of hops is the value in the **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

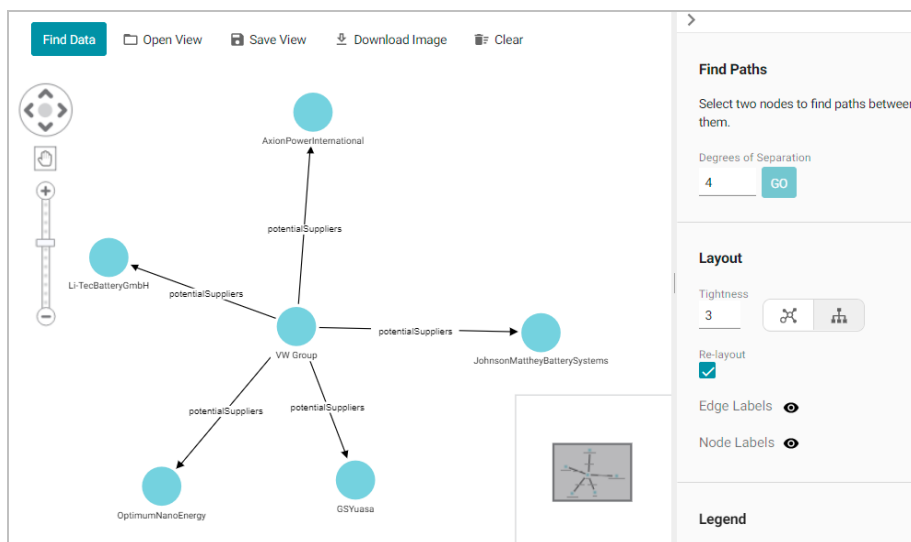
You can remove a single node or a hide a group of nodes by right-clicking a node and selecting an option from the pop-up menu (shown below). **Remove Selected Node** deletes the node and its incoming and outgoing connections. **Hide Selected Tree** hides the selected node and all of its connected nodes and paths.



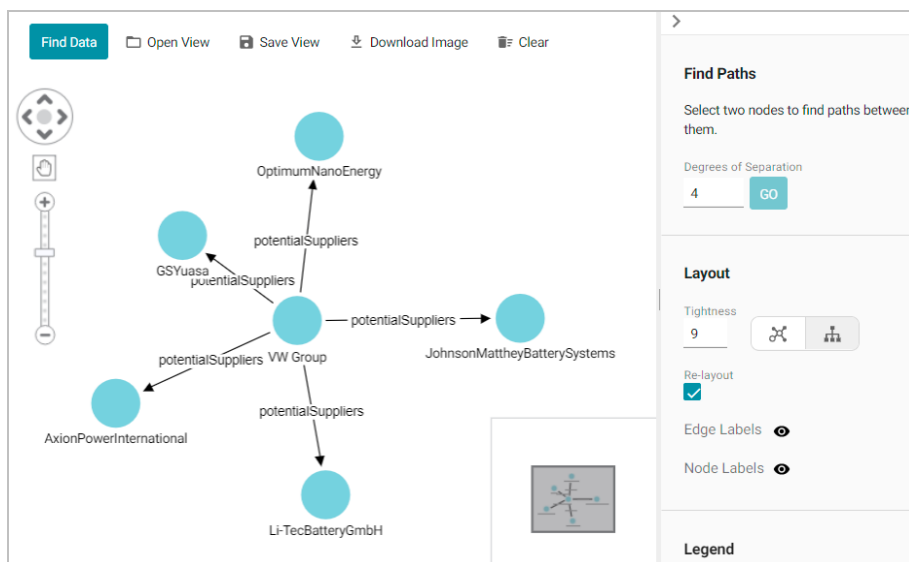
12. You can continue to explore the network using the methods described above. In addition, you can adjust the layout of the View using the options under **Layout** on the right side of the screen:

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:

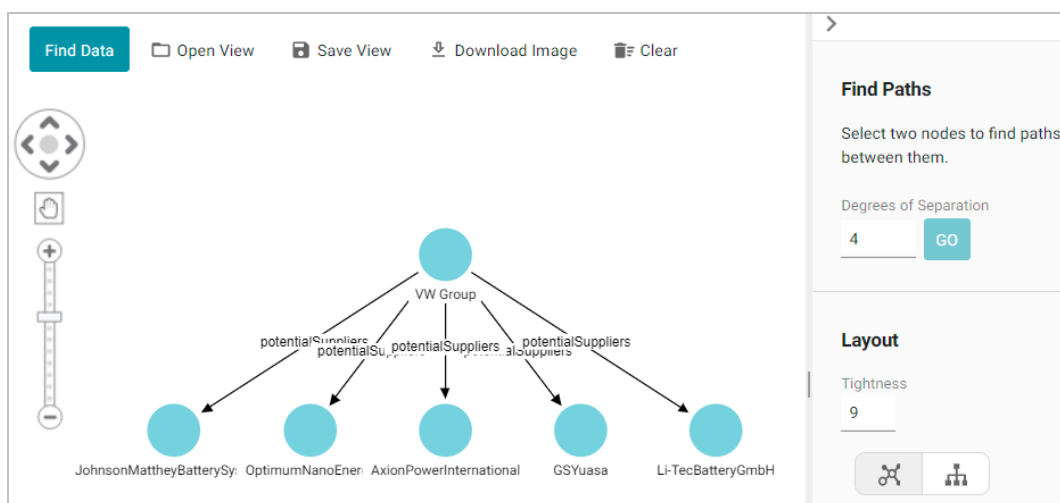


Standard vs. Hierarchical Layout

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](#)

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](#)

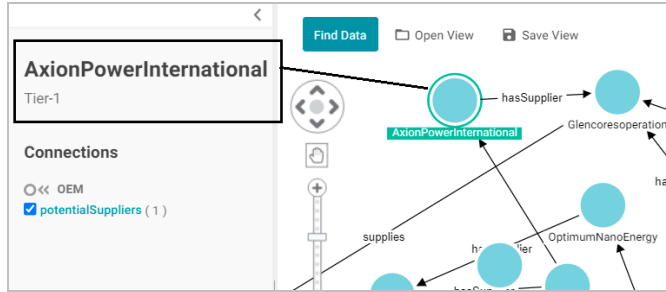
Note

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 \(Preview\)](#).

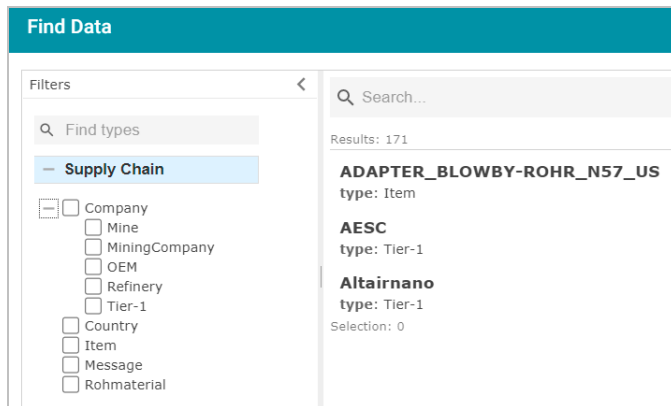
Identifying Types and Preparing Icons

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.

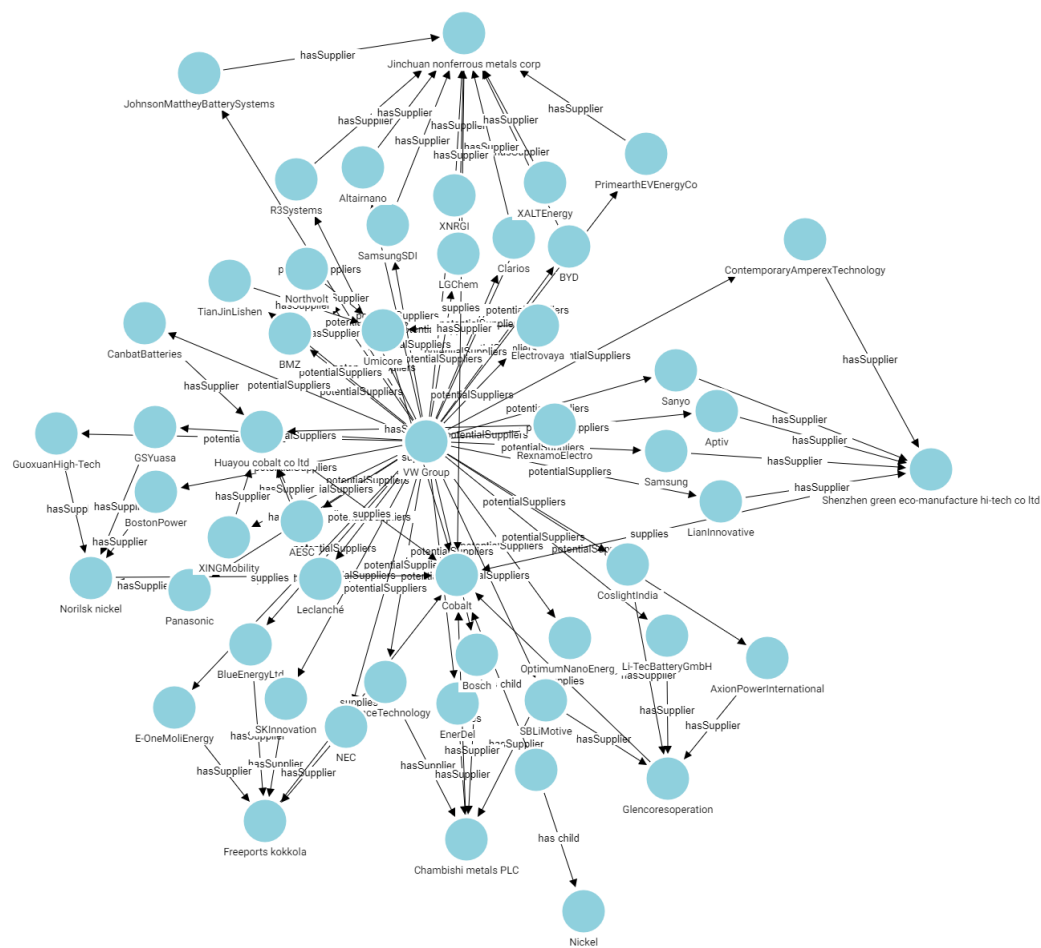


- You can click **Find Data** and open the **Filter** panel. Expand the contents to see the classes and subclasses. For example:



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.



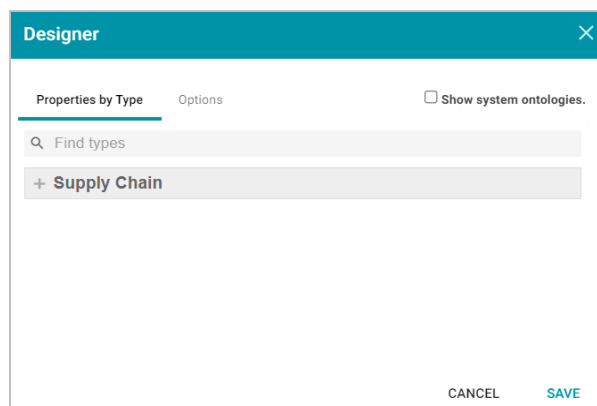
Uploading and Applying Icons to a Dashboard

Follow the instructions below to upload images to the icon library and then apply them to the classes in your Network View.

Note

When you apply icons to the classes (types) in a dashboard, those icons will apply to all Network Views that include those types.

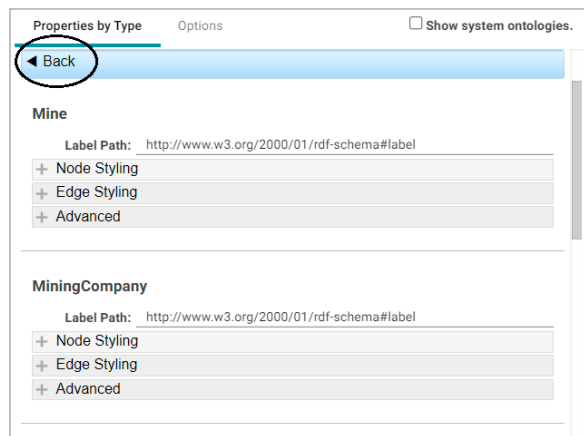
1. Open in the Hi-Res Analytics application the Network Navigator Dashboard that you want to add icons to.
2. In the main toolbar, click the **Designer** button. The **Properties by Type** tab is displayed in the Designer. For example:



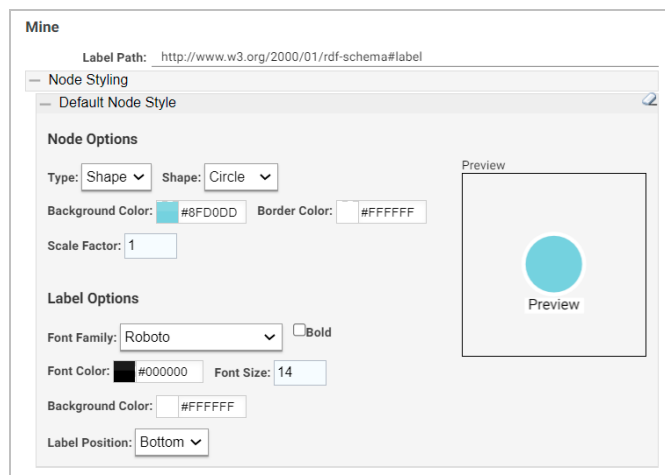
3. Expand the view of the Model to display each of the types (classes). If the Model contains subclasses, you can view them by clicking the **View Sub Types** link next to the parent class (shown below).



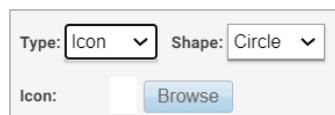
When viewing the subtypes, you can click **Back** to return to the parent level. For example:



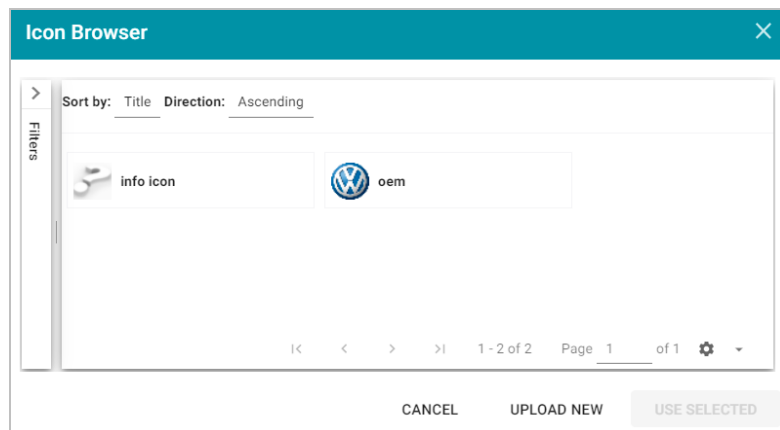
- Under the class that you want to apply an icon to, expand **Node Styling**. Then expand **Default Node Style**. For example, the image below shows the default node style configuration for the **Mine** type.



- Under Node Options, click the **Type** drop-down list and select **Icon**. An Icon field with a **Browse** button is displayed below Type:



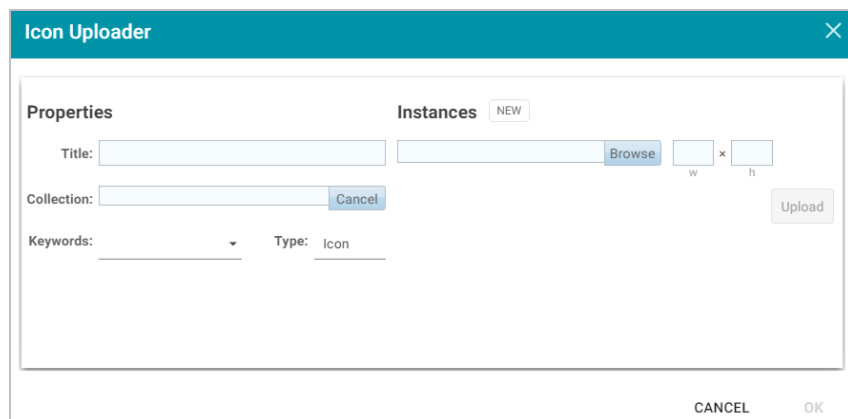
- Click **Browse** to open the Icon Browser. The Browser lists the icons that have been uploaded to the application and gives you the option to upload new icons. For example, the image below shows that two icons have been uploaded previously and are in the library.



Note

In the image, the icon that is shown as a question mark is an SVG image that does not render properly in the Icon Browser but does display in the Network View.

- 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:



- 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.

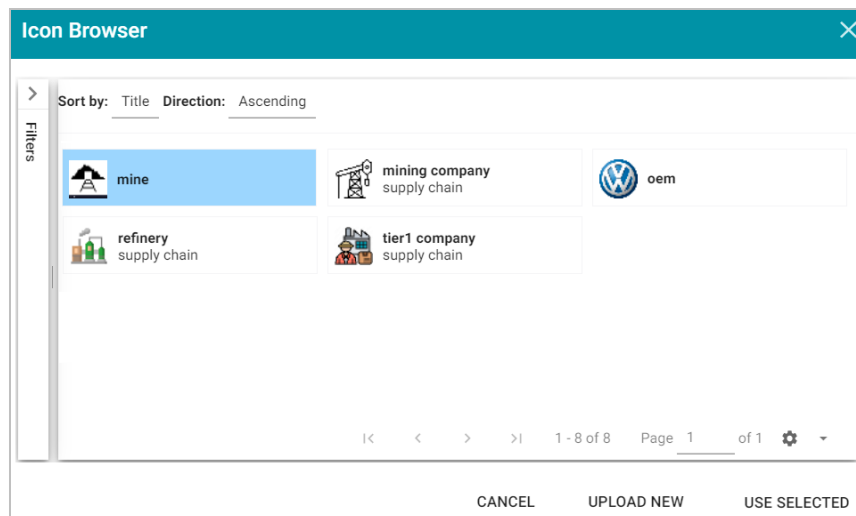
The screenshot shows the 'Properties' and 'Instances' tabs of the icon upload interface. The 'Properties' tab is active, showing fields for Title, Collection, Keywords, and Type. The 'Instances' tab is also visible, showing a list of instances with a 'NEW' button. The 'Title' field is set to 'mine'. The 'Collection' field is set to 'supply chain'. The 'Keywords' field is empty. The 'Type' field is set to 'Icon'. The 'Instances' tab shows a list of instances with a 'mine.jpeg' file, a 'Browse' button, and dimensions of 1000 x 1080. An 'Upload' button is located at the bottom right of the 'Instances' tab.

- 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.

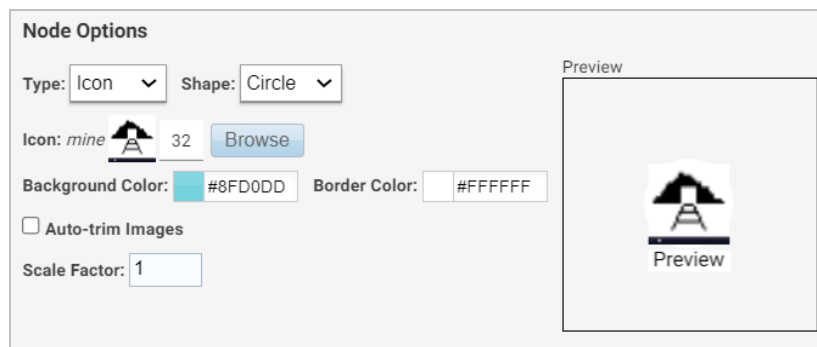
Tip

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.



11. Click **Use Selected** to apply the icon and return to the Designer where you can preview the icon and configure additional node options.



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.

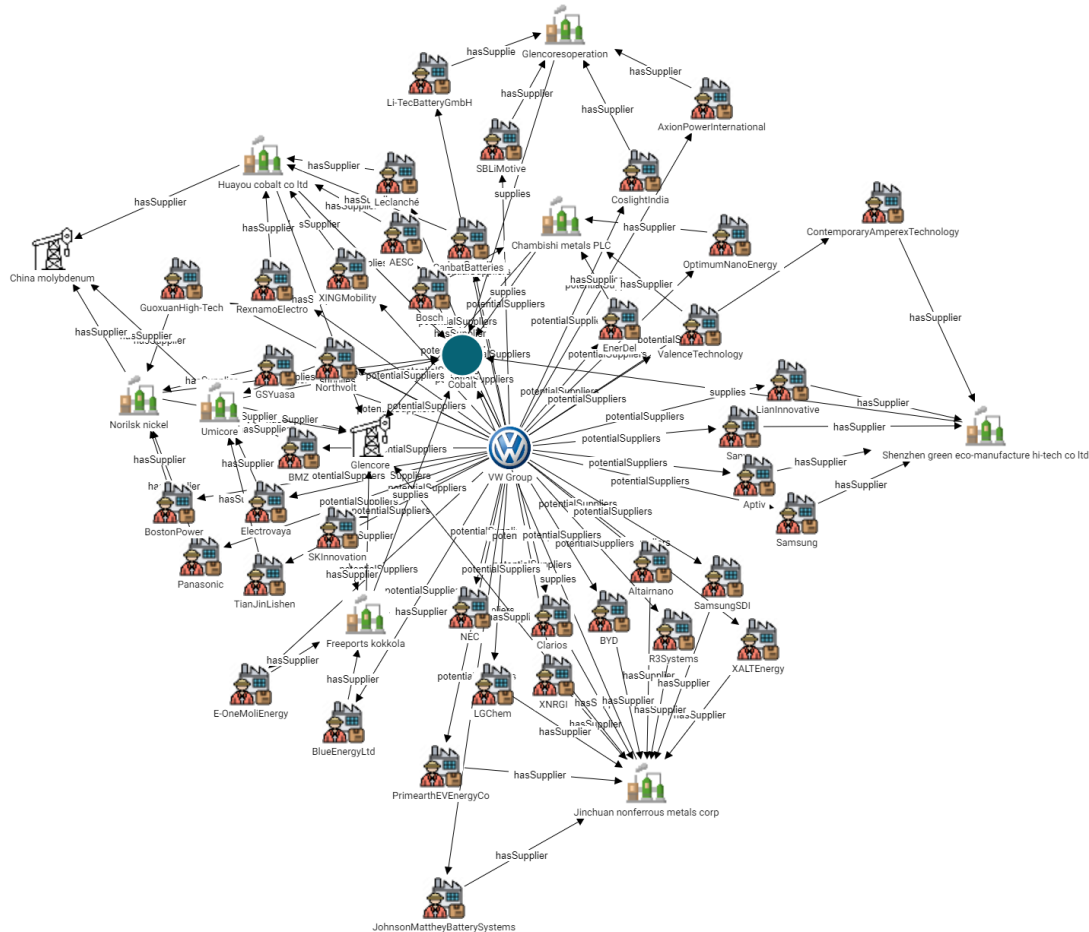
Auto-Trim Images

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.

- 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.



Related Topics

[Creating a Network Navigator Dashboard \(Preview\)](#)

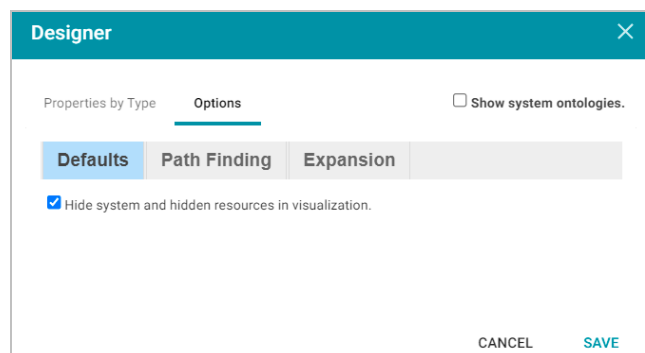
[Auto-Expanding a Network View](#)

Auto-Expanding a Network View

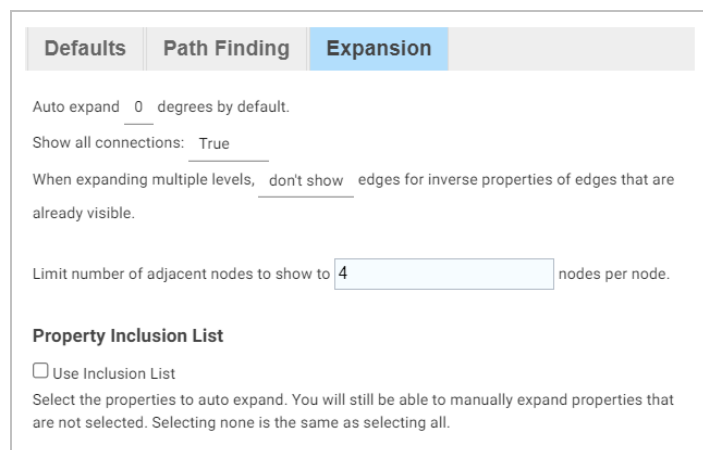
When you add nodes to a Network View, the Network Navigator Dashboard is configured by default to add only the selected nodes and paths to the View without automatically expanding the paths to show related nodes. You have the option to enable the **Auto-Expand** feature in the Dashboard Designer, however, so that the network is automatically

expanded by the specified number of degrees any time a new node or path is added to the View. To configure a dashboard to auto-expand, follow the steps below.

1. Open in the Hi-Res Analytics application the Network Navigator Dashboard that you want to configure.
2. In the main toolbar, click the **Designer** button. The Dashboard Designer opens and the **Properties by Type** tab is displayed. Click the **Options** tab:



3. Click the **Expansion** tab to view the expansion options. For example, the image below shows the default configuration.



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

TBD

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](#)

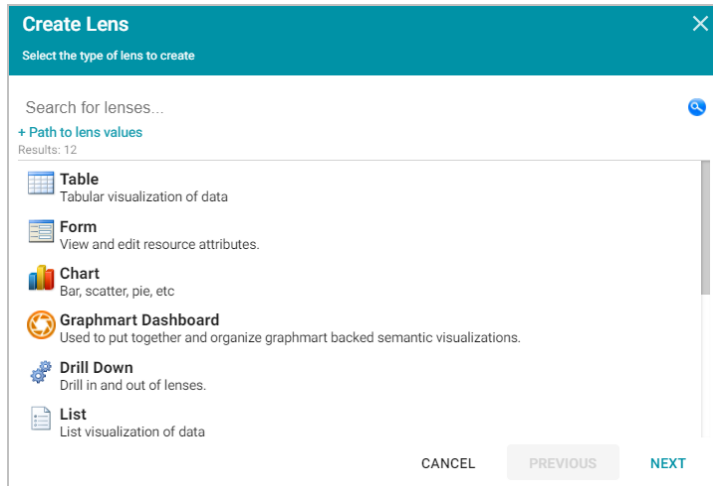
Creating a Lens

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. This topic provides instructions for creating and cloning lenses.

- [Creating a New Lens](#)
- [Cloning a Lens](#)

Creating a New 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 window opens.



3. Determine the type of lens that you want to create. The following list describes each lens type. For more information about each lens type, see [Lens Type Reference](#).
 - **AnzoKO Web Page:** Includes the [Knockout JavaScript](#) framework and displays data on a web page that you create using HTML, CSS, and JavaScript.
 - **Chart:** Displays results in rectangular columns, 3D bubbles, scatter charts, heat maps, or other chart types. A chart lens has the widest format range of any lens type. These lenses are useful for displaying large amounts of complex data. The ability to add an axis enables you to compare data, such as comparing monthly sales data for multiple stores.
 - **Dashboard:** Dashboard lenses display a dashboard within a dashboard.
 - **Drill Down:** Creates clickable data points that enable you to drill down to view additional details. You can specify multiple lenses within the Drill Down lens so that clicking a data point presents the data in a different view. When you select a Drill Down lens, the Designer opens and prompts you to select the lenses that you want to use for the drill down functionality. The top lens in the Designer becomes the primary lens with the clickable data points. You can drag lenses to re-order them. You then separately configure each of the lenses that you selected. There is no further configuration for the drill down.
 - **Form:** Enables you to create an editable or read-only form on the dashboard. Form lenses can be useful for displaying many details about a record. Form lenses are read-only when used with graphmarts.
 - **List:** Displays results as icons in a folder view, similar to the Microsoft Windows® Explorer interface. List lenses enable you to add images for each data value.
 - **Query:** 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 data set, data type, or filters defined on the dashboard.

- **Resource Tree Navigator:** Displays results in a hierarchical tree view. Click parent data points to open the successive child data points. You can also click certain objects to view the object's data properties in the left panel. This lens is useful for presenting small amounts of hierarchical data.
 - **Table:** Presents results in a basic table grid consisting of rows and columns. Table lenses are useful for presenting data aggregates or summaries.
 - **Web Page:** Displays results on a custom web page that you create using HTML, CSS, and JavaScript.
4. On the Create Lens dialog box, select the type of lens that you want to add to the dashboard, and then click **Next**. Anzo displays the General Information dialog box.

5. Type a **Title** and optional **Description** for the lens.
6. Click **Finish**. The lens Designer dialog box opens to enable you to configure the lens.
7. Configure the lens and then click **Save** to save the configuration and add the lens to the dashboard. For information about using formulas to compute the values to display in the lens, see [Calculating Values in Lenses and Filters](#).

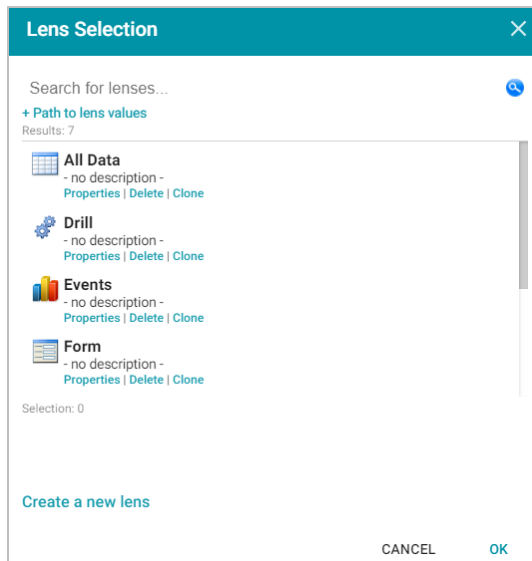
Cloning a Lens

Cloning a lens makes a copy of the lens that can be changed without affecting the original lens or other dashboards.

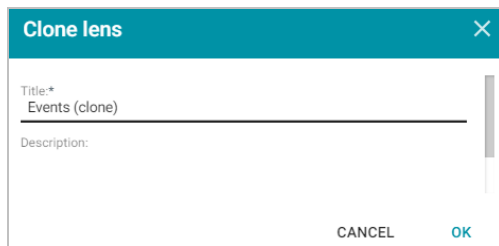
Note

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:



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.

Related Topics

[Lens Type Reference](#)

[Combining Data from Multiple Classes](#)

[Sharing Access to a Dashboard or Lens](#)

[Exporting a Lens](#)

[Deleting a Lens](#)

[Creating Dashboard Filters](#)

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 a Lens](#)

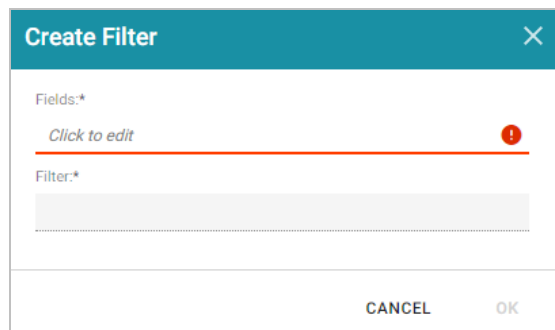
[Function and Formula Reference](#)

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.

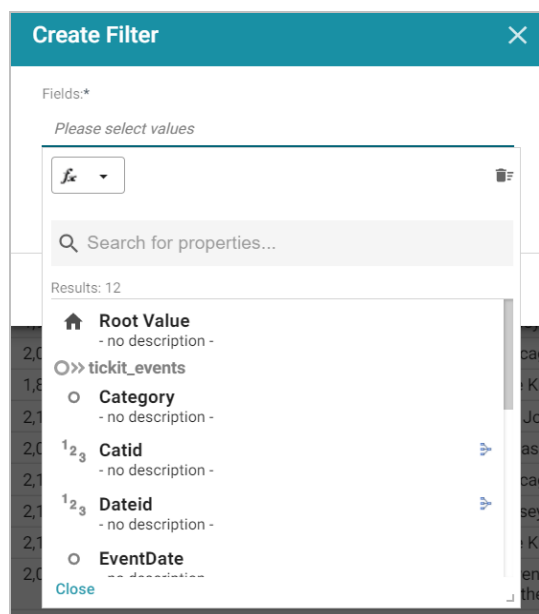
Follow the instructions below to create a Cloud Filter.

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.



- 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 Data 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.

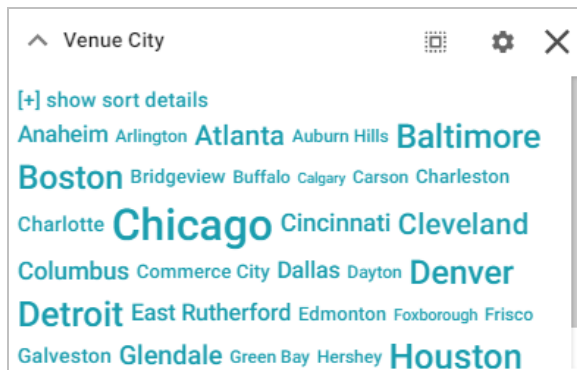
- Linked classes are represented by incoming (◀) and outgoing (▶) connection icons. The properties in those classes with a path to another class are denoted with a circle icon (○). Selecting a linked property navigates to that class and displays its properties.
 - When a property or path is selected, 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 **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:

The screenshot shows the 'Filter Properties' dialog box. It has a title field, a label field with a 'Click to edit' link, and an 'Exclude' checkbox. Below these are two checked checkboxes: 'Show counts' and 'Respond to other filters'. At the bottom, there is a 'Subfilters' section with a '+ Create Filter' link.

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 Cloud 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 **Value** to sort string values alphabetically, or select **Count** to order results according to the total number of results for each value.
 - **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.
- **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** (⚙️): Clicking this icon opens the filter Designer so that you can view or change the filter configuration.
- **Close** (X): Clicking **Close** removes the filter from the Dashboard. This action cannot be undone.

Related Topics

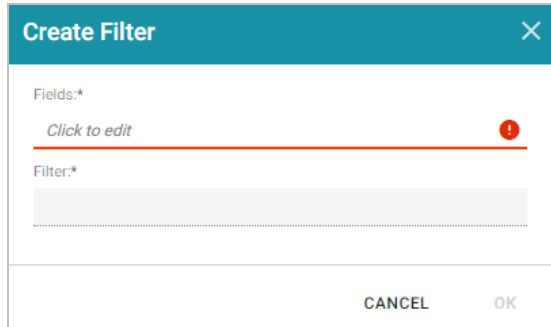
[Creating Dashboard Filters](#)

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.

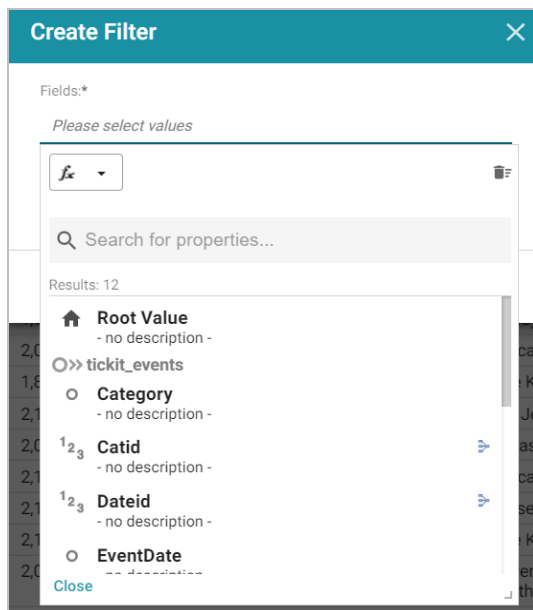
Follow the instructions below to create a Date Range Filter.

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 Data 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. For this example Date Range Filter, selecting **EventDate** at the bottom of the screen, 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](#).
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 **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:

Filter Properties

Title:

Interval Unit:*

 MILLENNIUM

Interval:*

Exclude: ☐ Show Bars: ☐

 Show counts: ☒ Respond to other filters: ☒

Format

Format Type:

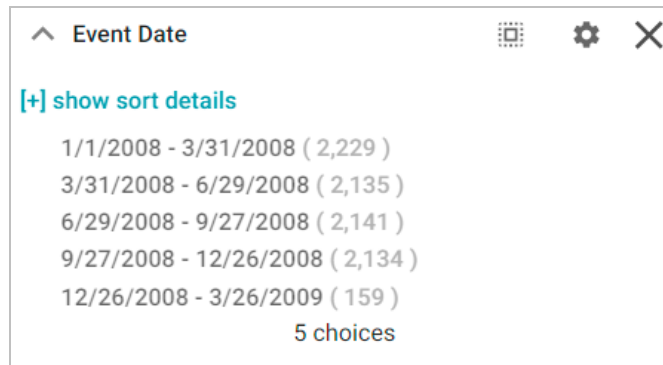
 Automatic

Subfilters

[+ Create Filter](#)

6. 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 Date Range 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.
 - **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.
- **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** (⚙): Clicking this icon opens the filter Designer so that you can view or change the filter configuration.
- **Close** (X): Clicking **Close** removes the filter from the Dashboard. This action cannot be undone.

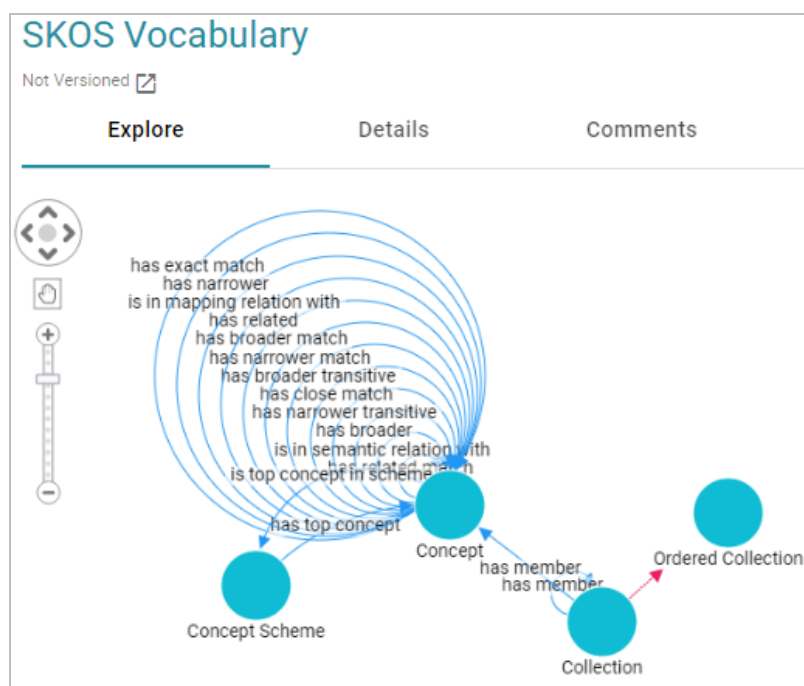
Related Topics

[Creating Dashboard Filters](#)

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 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.



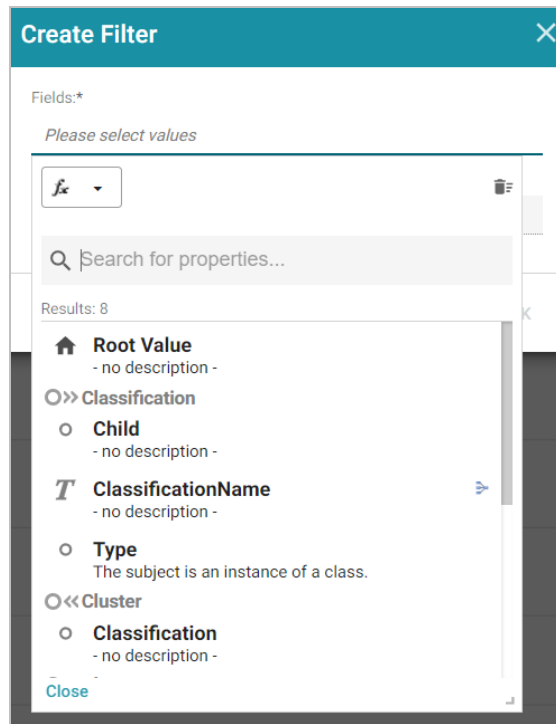
Follow the instructions below to create a Hierarchy Filter.

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.

The **Create Filter** dialog box is shown. It has a title bar with a close button (X). The main area contains two fields: **Fields:*** and **Filter:***. The **Fields:*** field has a red underline and a red exclamation mark icon, with the text *Click to edit* below it. The **Filter:*** field is empty. At the bottom, there are **CANCEL** and **OK** buttons.

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 Data Type for the Dashboard is Classification:



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.

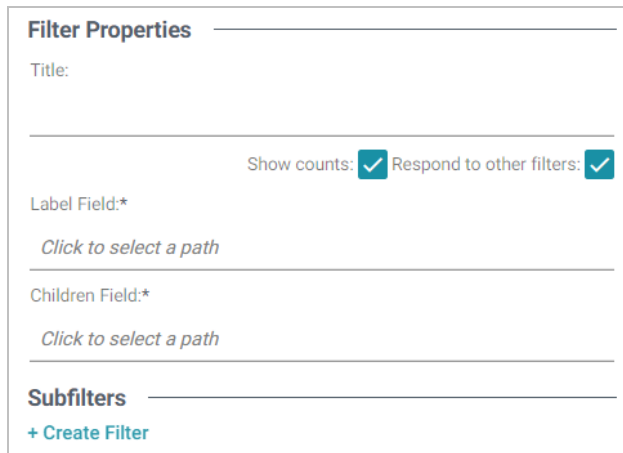
Tip

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:

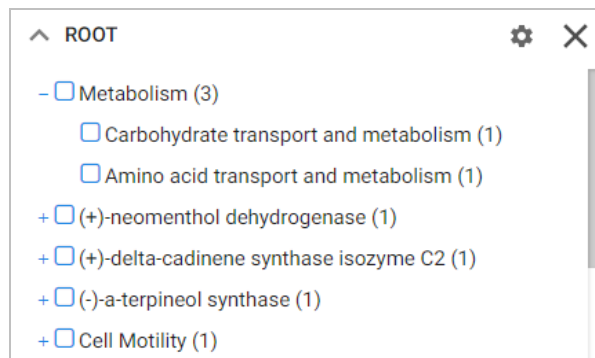


The screenshot shows a dialog box titled "Filter Properties". It contains the following elements:

- A "Title:" label followed by a text input field.
- Two checkboxes: "Show counts:" (checked) and "Respond to other filters:" (checked).
- A "Label Field:*" label followed by a text input field containing the text "Click to select a path".
- A "Children Field:*" label followed by a text input field containing the text "Click to select a path".
- A "Subfilters" section with a "+ Create Filter" link.

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.
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 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:

- **Clear** (☒): This option is available when one or more results are selected. Clicking **Clear** removes all selections.
- **Designer** (⚙️): Clicking this icon opens the filter Designer so that you can view or change the filter configuration.
- **Close** (X): Clicking **Close** removes the filter from the Dashboard. This action cannot be undone.

Related Topics

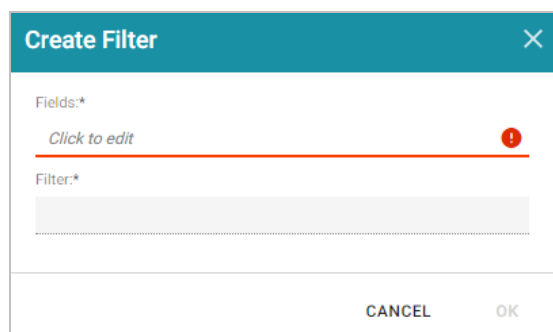
Creating Dashboard Filters

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.

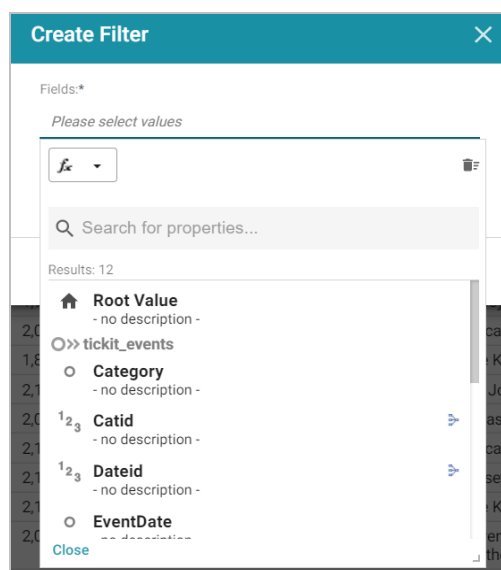
Follow the instructions below to create a Limit Filter.

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.



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 Data Type for the Dashboard is `ticket_events`:



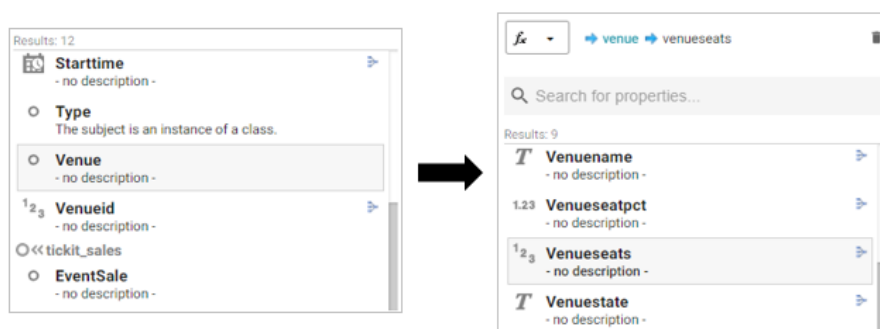
The list below describes the icons and options that are available when choosing a property:

- 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. For this example Limit Filter, selecting the **Venue** path (shown below) navigates to the `ticket_venues` class where an integer type property, `Venueseats`, is selected.

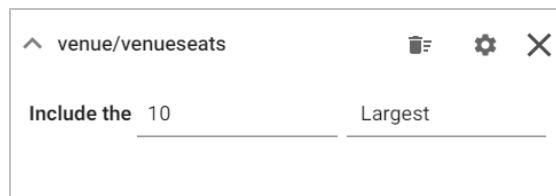


- When a property or path is selected, the breadcrumbs at the top of the dialog box show you the property path. You can click the Clear icon (☒) to clear the path and start again.
 - After you have selected a property, you can apply a function or formula to that property to calculate the values that 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:

The screenshot shows a 'Filter Properties' dialog box. It contains a 'Title:' label followed by a text input field. Below the input field is a 'Limit by resource:' checkbox, which is checked. At the bottom of the dialog, there is a 'Subfilters' section with a '+ Create Filter' link.

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.



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:

- **Clear** (🗑️): Clicking **Clear** removes the value in the **Include the** field.
- **Designer** (⚙️): Clicking this icon opens the filter Designer so that you can view or change the filter configuration.
- **Close** (X): Clicking **Close** removes the filter from the Dashboard. This action cannot be undone.

Related Topics

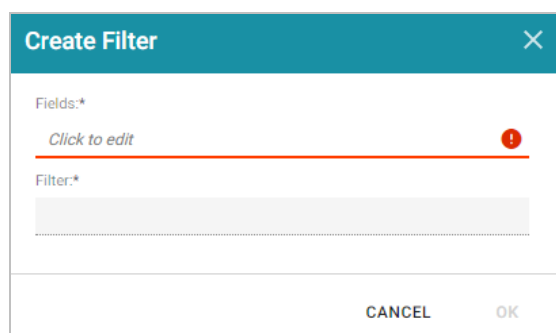
[Creating Dashboard Filters](#)

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.

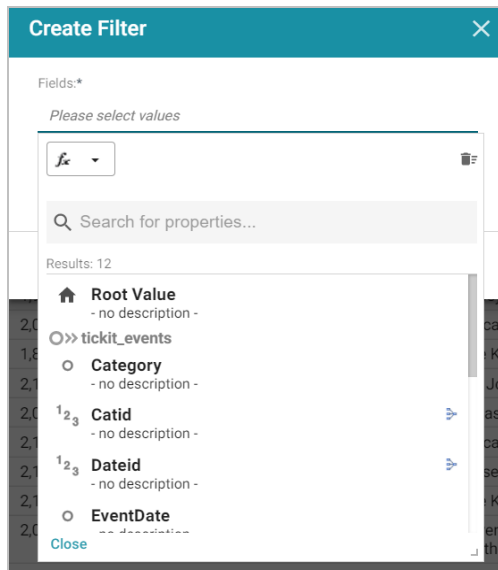
Follow the instructions below to create a List Filter.

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.



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 Data 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 **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:

Filter Properties

Title:

Label field:

Click to edit

Exclude: ☐ Show Bars: ☐ Show Blanks: ☐

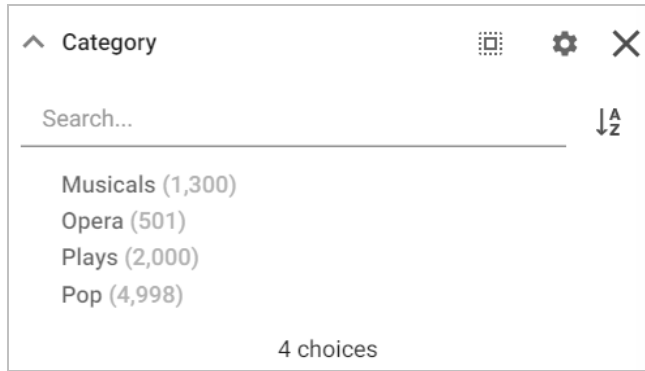
Show counts: ☒ Respond to other filters: ☒

Subfilters

[+ Create Filter](#)

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.
 - **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 List 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** (↕): 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.
- **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** (⚙): Clicking this icon opens the filter Designer so that you can view or change the filter configuration.
- **Close** (X): Clicking **Close** removes the filter from the Dashboard. This action cannot be undone.

Related Topics

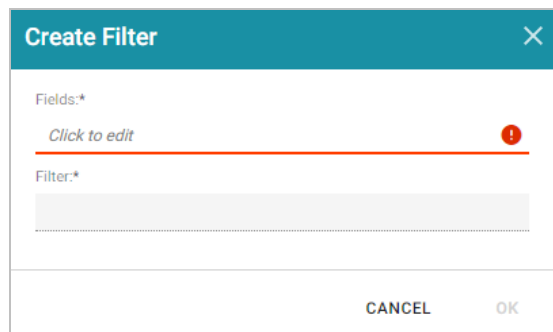
[Creating Dashboard Filters](#)

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.

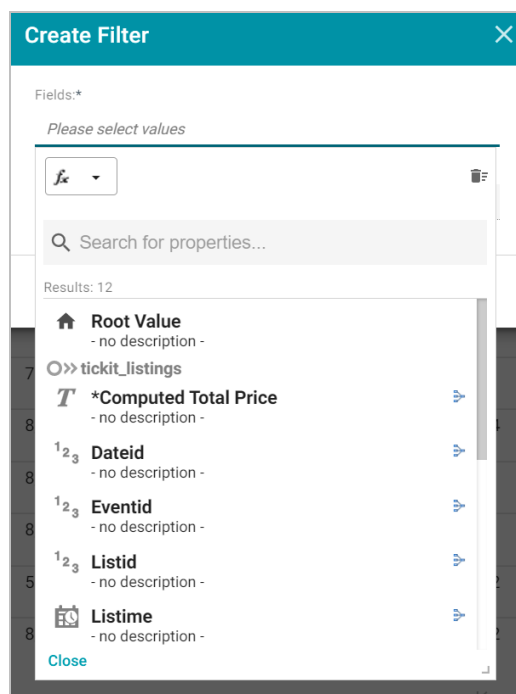
Follow the instructions below to create a Numeric Range Filter.

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.



3. 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 Data Type for the Dashboard is `ticket_listings`:



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:

Filter Properties

Title:

Interval:*

Exclude: ☐ Show Bars: ☐

Show counts: ☒ Respond to other filters: ☒

Format

Type:

Automatic

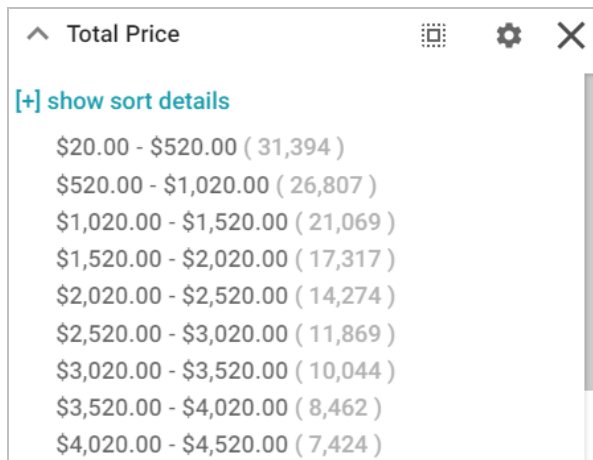
Subfilters

+ Create Filter

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.

- **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.
- **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 (⚙️)**: Clicking this icon opens the filter Designer so that you can view or change the filter configuration.
- **Close (X)**: Clicking **Close** removes the filter from the Dashboard. This action cannot be undone.

Related Topics

[Creating Dashboard Filters](#)

Creating a Presence Filter

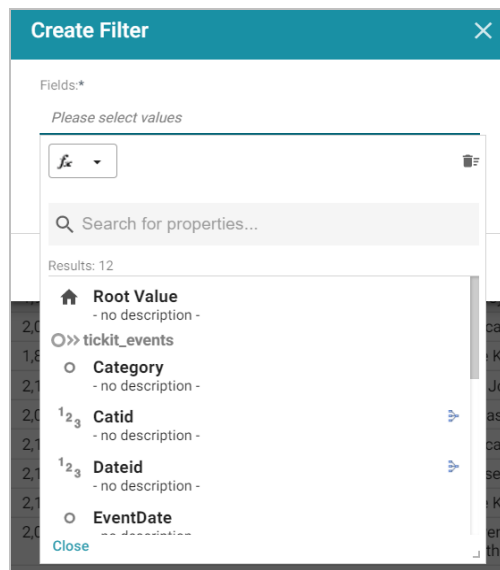
Presence Filters group results based on whether the value exists or does not exist. This type of filter is useful for testing whether there are records that are missing a particular value. Presence Filters are available for paths and properties of all data types.

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.

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 Data Type for the Dashboard is ticket_events:



The list below describes the icons and options that are available when choosing a property:

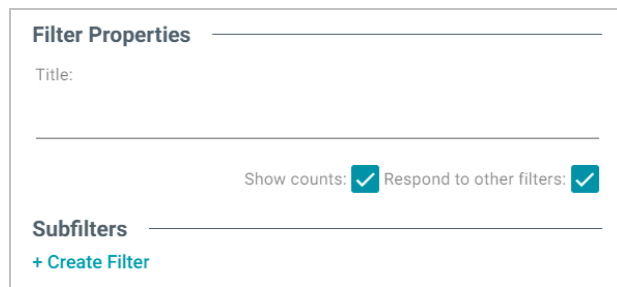
- The **Root Value** (🏠) is the instance URI for the root resource—the URI for the instances of the class that was chosen as the Data Type for the Dashboard.

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 **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:



Filter Properties

Title:

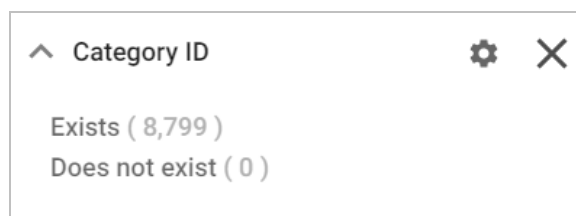
Show counts: ☒ Respond to other filters: ☒

Subfilters

[+ Create Filter](#)

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.



Category ID

Exists (8,799)

Does not exist (0)

When working with the filter on the Dashboard, the following options are available for sorting and configuration:

- **Clear** (☒): This option is available when one or more items are selected. Clicking Clear removes the selections.
- **Designer** (⚙): Clicking this icon opens the filter Designer so that you can view or change the filter configuration.
- **Close** (X): Clicking **Close** removes the filter from the Dashboard. This action cannot be undone.

Related Topics

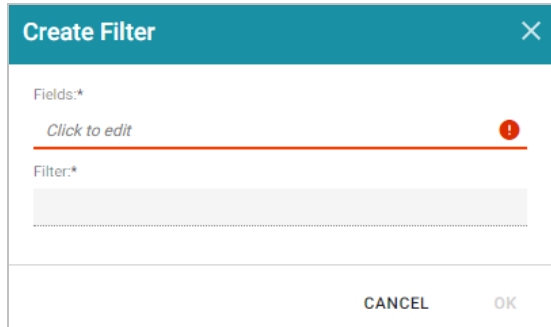
[Creating Dashboard Filters](#)

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.

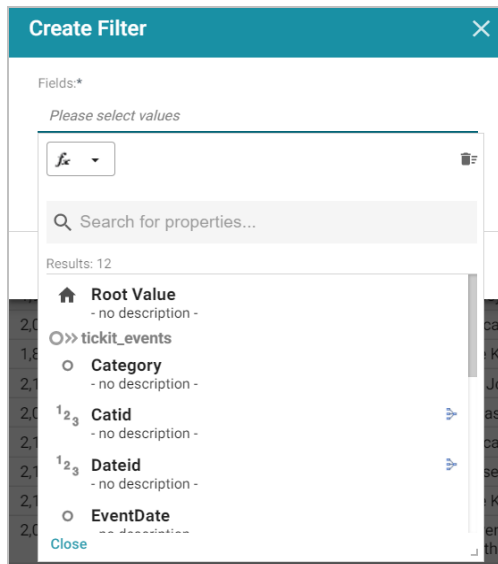
Follow the instructions below to create a Quartile Filter.

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.



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 Data 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 **Quartile** from the drop-down list. The dialog box is refreshed to show the Filter Properties that are available for the filter type:

Filter Properties

Title: _____

6. 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.
7. 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 Quartile Filter that groups the price paid for tickets to events into four equal ranges.

Price Paid

1 (Range: 20-194)	(8,076)
2 (Range: 194-386)	(8,210)
3 (Range: 386-758)	(8,204)
4 (Range: 758-12,624)	(8,106)

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:

- **Select All Visible** (☒): This option selects all of the items that are listed in the filter.
- **show/hide filters**: This option is displayed when a value is selected. It shows or hides the selection.
- **Clear** (☒): This option is available when one or more items are selected. Clicking Clear removes the selections.
- **Designer** (⚙): Clicking this icon opens the filter Designer so that you can view or change the filter configuration.
- **Close** (X): Clicking **Close** removes the filter from the Dashboard. This action cannot be undone.

Related Topics

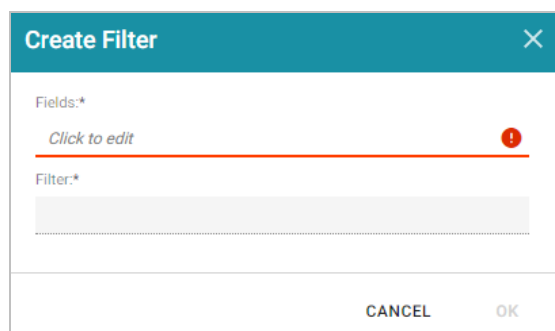
Creating Dashboard Filters

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.

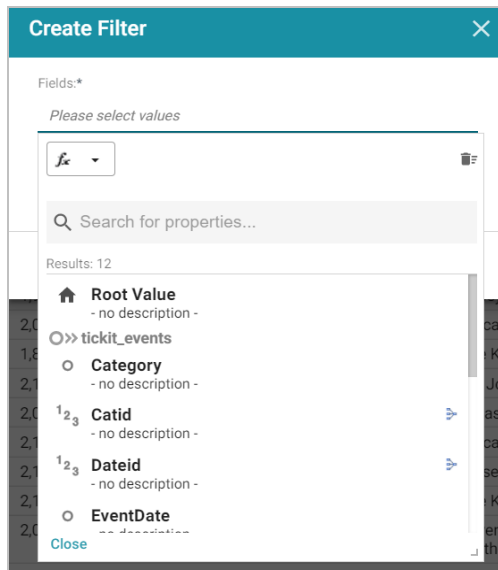
Follow the instructions below to create a Range Slider Filter.

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.



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 Data Type for the Dashboard is ticket_events:



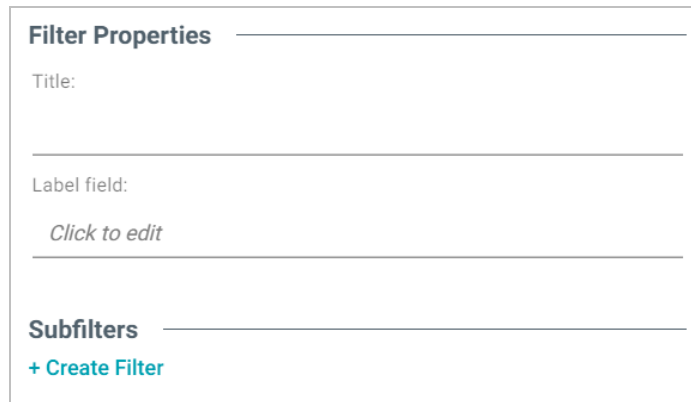
The list below describes the icons and options that are available when choosing a property:

- 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:



Filter Properties

Title:

Label field:

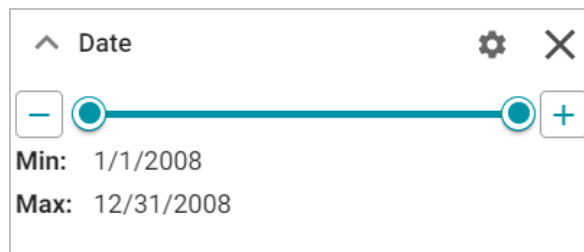
Click to edit

Subfilters

[+ Create Filter](#)

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.



^ Date ⚙️ ✕

— ● — ● +

Min: 1/1/2008

Max: 12/31/2008

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:

- **Clear (☒):** This option is available once the slider has been adjusted. Clicking Clear resets the range back to the default configuration.
- **Designer (⚙️):** Clicking this icon opens the filter Designer so that you can view or change the filter configuration.
- **Close (✕):** Clicking **Close** removes the filter from the Dashboard. This action cannot be undone.

Related Topics

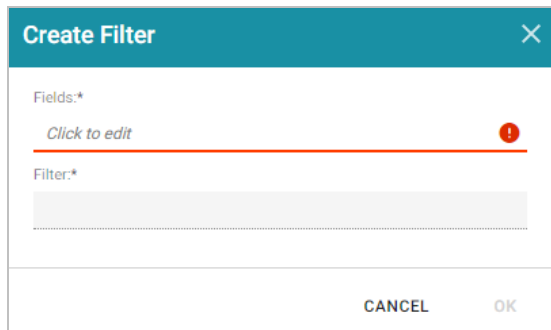
[Creating Dashboard Filters](#)

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.

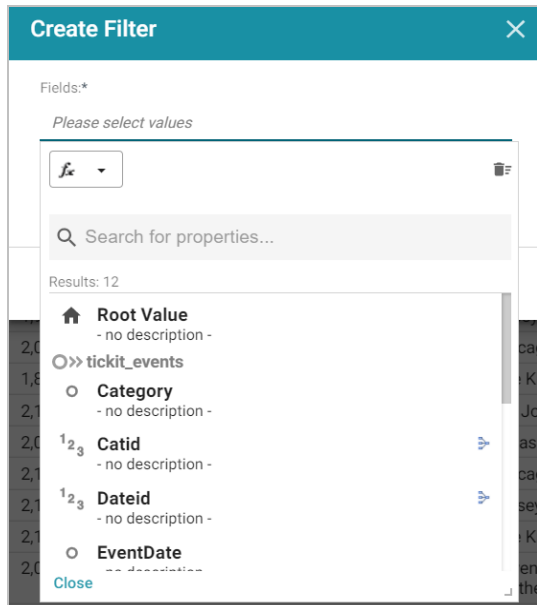
Follow the instructions below to create a Relative Time Filter.

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.



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 Data 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 **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:

Filter Properties

Title:

Label field:

Click to edit

Respond to other filters:

☒

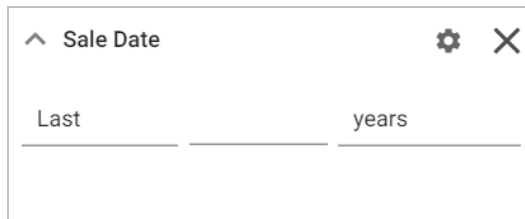
Subfilters

+ Create Filter

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 to events.



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:

- **Clear** (☐): This option is available when once the filter options are selected. Clicking Clear resets the filter to the default values.
- **Designer** (⚙): Clicking this icon opens the filter Designer so that you can view or change the filter configuration.
- **Close** (X): Clicking **Close** removes the filter from the Dashboard. This action cannot be undone.

Related Topics

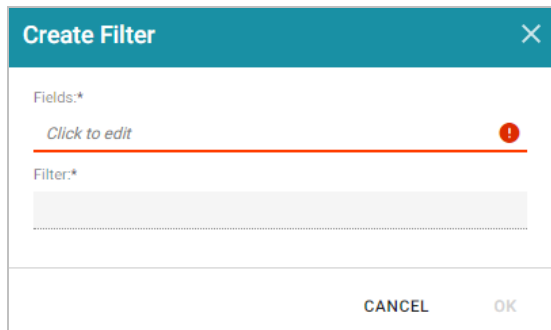
[Creating Dashboard Filters](#)

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.

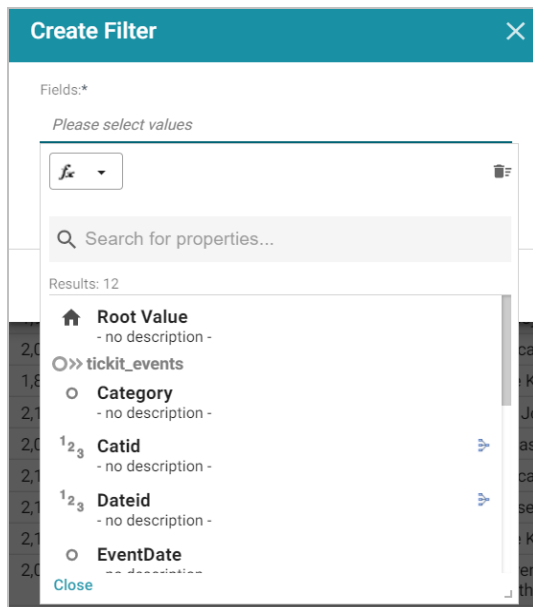
Follow the instructions below to create a Search Filter.

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.



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 Data 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 **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:

Filter Properties

Title:

Subfilters

[+ Create Filter](#)

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.

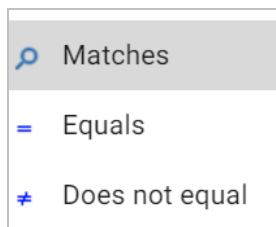
For example, the filter in the image below defines Category as the property to search on.

Category

=

Filter

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:

- **Clear** (🧼): This option is available when once the filter options are selected. Clicking Clear removes the search value.
- **Designer** (⚙️): Clicking this icon opens the filter Designer so that you can view or change the filter configuration.
- **Close** (X): Clicking **Close** removes the filter from the Dashboard. This action cannot be undone.

Related Topics

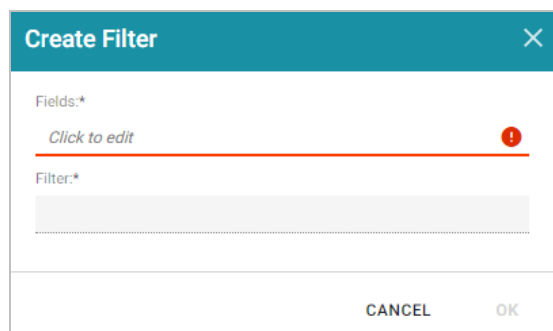
Creating Dashboard Filters

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.

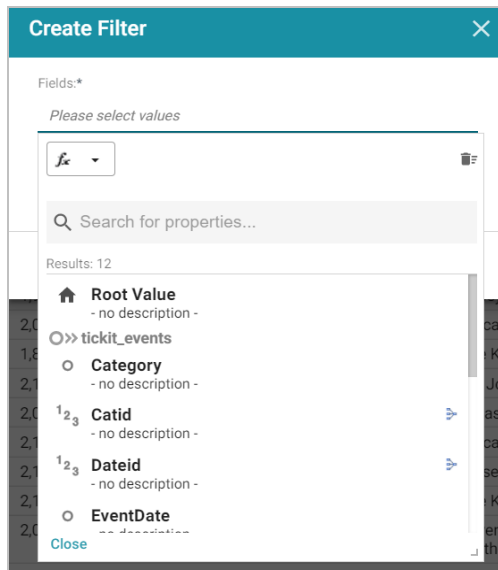
Follow the instructions below to create a Single Select List Filter.

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.



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 Data 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 **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:

Filter Properties

Title:

Label field:

Click to edit

Exclude: ☐ Show Bars: ☐ Show Blanks: ☐

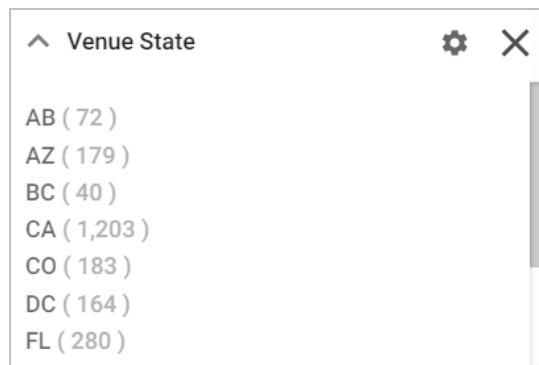
Show counts: ☒ Respond to other filters: ☒

Subfilters

[+ Create Filter](#)

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.
 - **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:

- **Clear** (☒): This option is available when one or more items are selected. Clicking Clear removes the selections.
- **Designer** (⚙️): Clicking this icon opens the filter Designer so that you can view or change the filter configuration.
- **Close** (X): Clicking **Close** removes the filter from the Dashboard. This action cannot be undone.

Related Topics

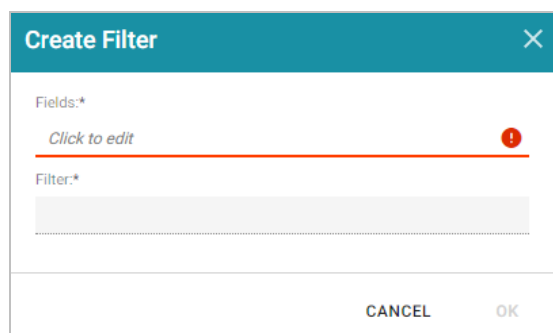
[Creating Dashboard Filters](#)

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.

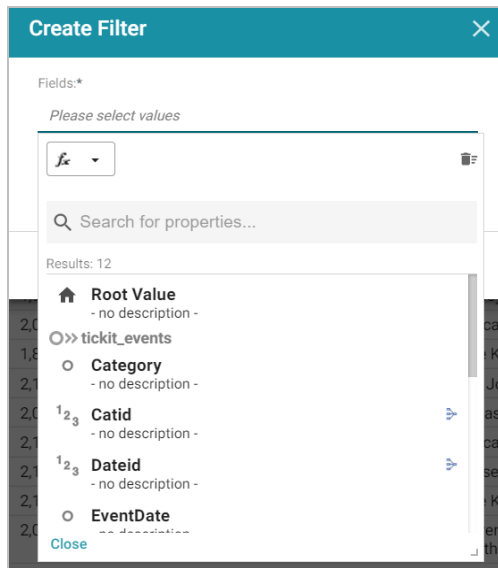
Follow the instructions below to create a Types Filter.

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.



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 Data Type for the Dashboard is `ticket_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](#).
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 **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:

Filter Properties

Title:

Exclude: ☐ Show Bars: ☐ Show Blanks: ☐

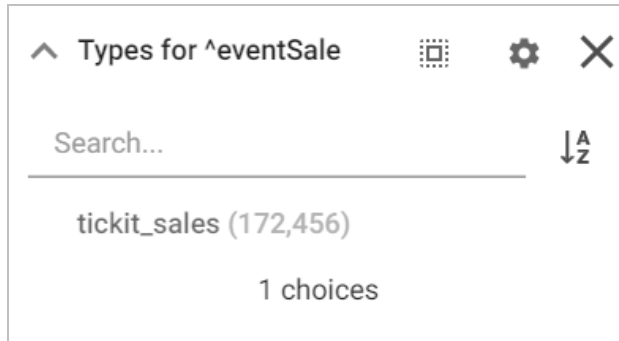
Show counts: ☒ Respond to other filters: ☒

Subfilters

[+ Create Filter](#)



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.
 - **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.
 - **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** ( - **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.
- **Select All Visible** (

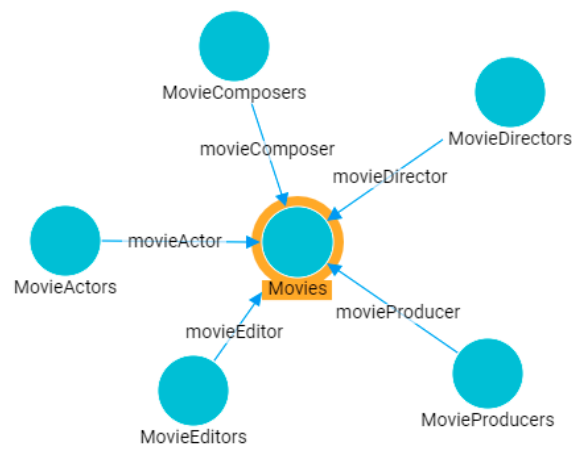
Related Topics

[Creating Dashboard Filters](#)

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, **MovieActors**, **MovieDirectors**, **MovieComposers**, 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:

Graphmart

Movies Graphmart

Data Types

Movies (8,157)

Data Layers

Movies to Store

Designer

Properties

Export

What can I do next?

Movie List

MovieID	MovieTitle
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

< >

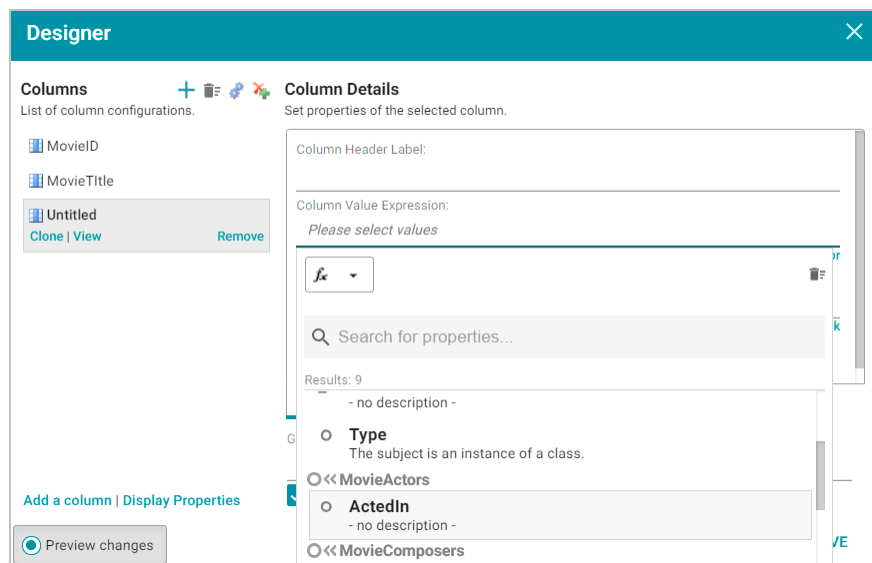
1 - 20 of 8157

Page 1 of 408

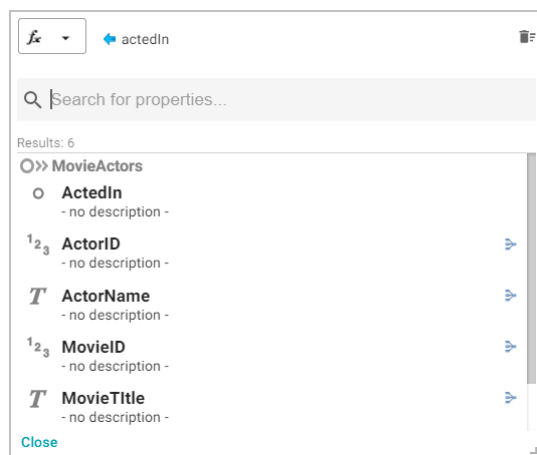
⚙

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 **MovieActors** 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 **MovieActor** 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.

DesignerPropertiesExport

What can I do next?Movie Actors

MovieID	MovieTitle	Actor
47862985	Get Rich or Die Tryin' (film)	50 Cent, Adewale Akinnuoye-Agbaje, Bill Duke, Joy Bryant, Omar Benson Miller, Terrence Howard, Viola Davis
12117643	The Signal (2007 film)	A. J. Bowen, Anessa Ramsey, Justin Welborn
8509794	Meet the Deedles	A. J. Langer, Dennis Hopper, John Ashton (actor), Paul Walker, Robert Englund, Steve Van Wormer
10746393	No Reservations (film)	Aaron Eckhart, Abigail Breslin, Catherine Zeta-Jones, Patricia Clarkson
1241914	Erin Brockovich (film)	Aaron Eckhart, Albert Finney, Julia Roberts
6856644	Your Friends & Neighbors	Aaron Eckhart, Amy Brenneman, Ben Stiller, Catherine Keener, Jason Patric, Nastassja Kinski
37052715	Olympus Has Fallen	Aaron Eckhart, Angela Bassett, Ashley Judd, Cole Hauser, Dylan McDermott, Melissa Leo, Morgan Freeman, Radha Mitchell, Rick Yune, Robert Forster
939561	Suspect Zero	Aaron Eckhart, Ben Kingsley, Carrie-Anne Moss, Barry Lonnig, Kevin Chamberlain

1 - 20 of 8157Page 1 of 408

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:

Graphmart

Movies Graphmart

Data Types

Movies (8,157)

Data Layers

Movies to Store

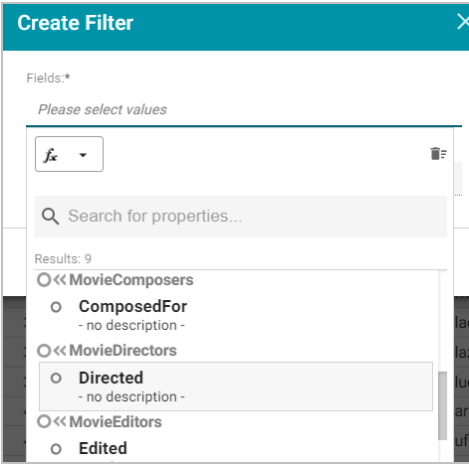
DesignerPropertiesExport

What can I do next?Movie List

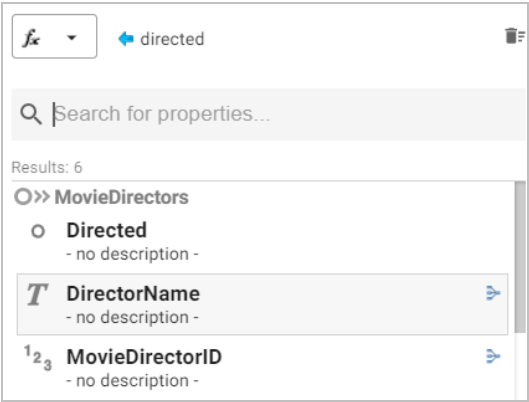
MovieID	MovieTitle
3217	Army of Darkness
3746	Blade Runner
3837	Blazing Saddles
3947	Blue Velvet (film)
4227	Barry Lyndon
4231	Buffy the Vampire Slayer (film)
4560	Braveheart
4726	Batman (1989 film)
4727	Batman (1966 film)
4728	Batman Returns
4729	Batman & Robin (film)
4730	Batman Forever
5313	Crouching Tiger, Hidden Dragon
7906	Destry Rides Again

1 - 20 of 8157Page 1 of 408

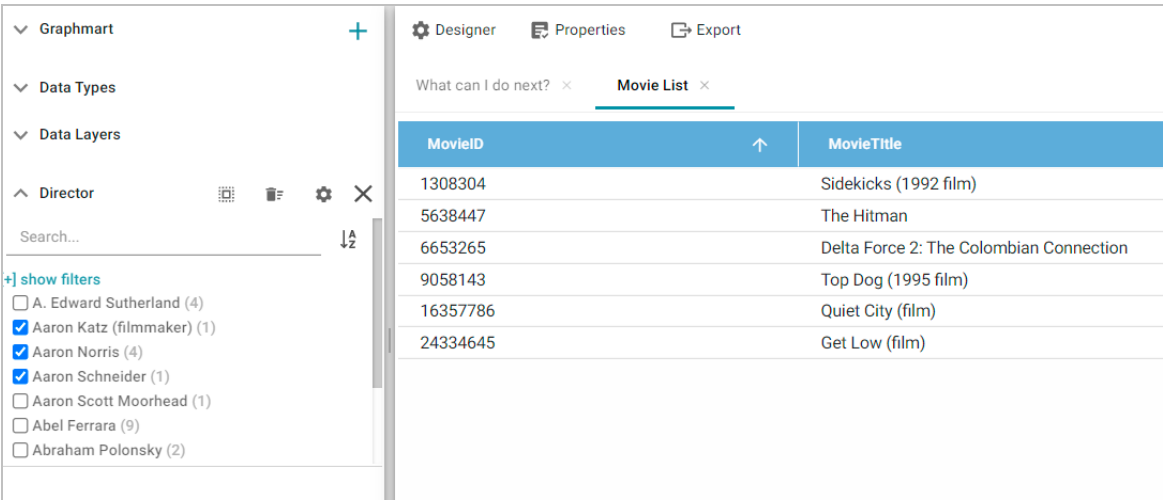
A filter can be configured to leverage the relationships from the base class to the connected classes. For example, adding a filter that navigates the **Directed** path to access the **MovieDirectors** class could be used to display, and filter on, the name of the director for each movie. To navigate the relationship in the Create Filter dialog box, the **Directed** path from the **MovieDirectors** class is selected for the Field to filter on:



Once the path is chosen, the properties from the **MovieDirectors** class are displayed:



Selecting the **DirectorName** 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.



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 `MovieActors` or `MovieProducers` classes. For more information about creating or editing lenses and filters, see [Creating a Lens](#) and [Creating Dashboard Filters](#).

Related Topics

[Creating Dashboards](#)

[Creating a Lens](#)

[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. Hi-Res Analytics enables you to save your formulas as computed properties that can be reused on other dashboards, lenses, and filters.

This section 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

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.

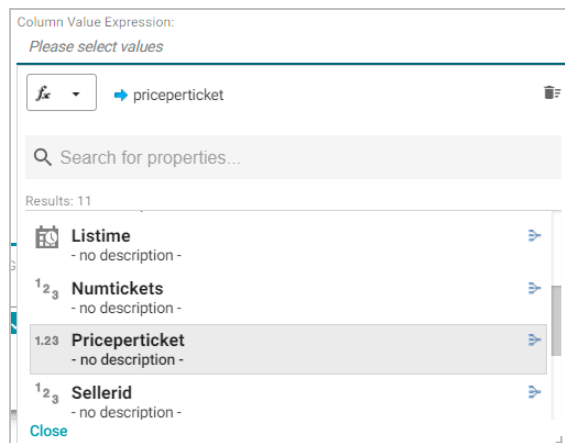
When you apply formulas to properties in lenses, the calculation results depend on the data type of the dashboard or lens. If the property belongs to a class that allows multiple values, Anzo performs the calculation on each set of multiple values and returns the results as one record in the lens. If the class includes single value properties, the calculation is performed separately for each single value.


- [Applying Functions and Formulas to Properties](#)
- [Saving Formulas for Reuse](#)
- [Reusing Computed Properties](#)

Applying Functions and Formulas to Properties

Follow these instructions to use a function or formula to compute the values in a lens or filter.

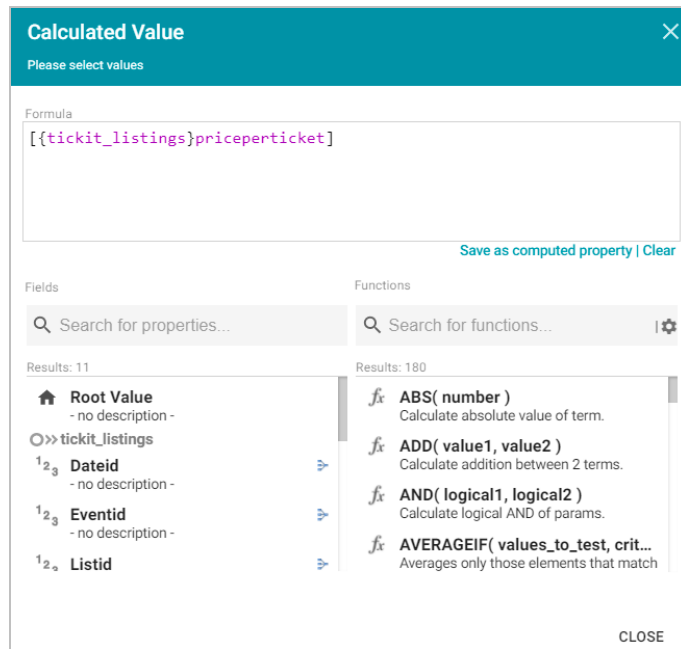
1. Create a new lens or filter or open the Designer for an existing lens or filter.
2. In the drop-down list for selecting properties or fields, select the property or path for which you want to compute the values. For example:



3. Click the function button () to display the list of standard functions. The list varies depending on the data type of the selected property.



4. Click a function to apply it to the property that you chose. For information about each of the available functions, see [Function and Formula Reference](#).
5. 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.

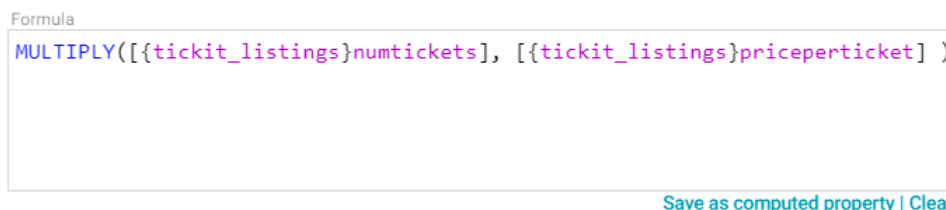


Tip

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.

6. 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 [Function and Formula Reference](#).
7. 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:



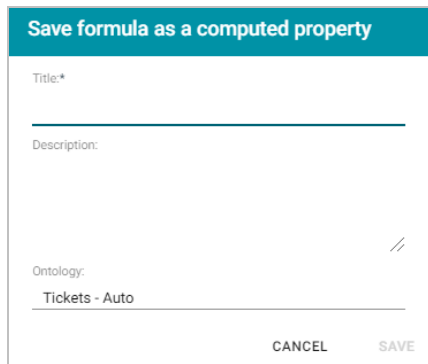
8. When you are finished writing a formula, you have two options:
 - If you want to use the formula now without saving it for later use, click **Close** to close the Calculated Value dialog box. Then complete the lens or filter configuration.

- 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.

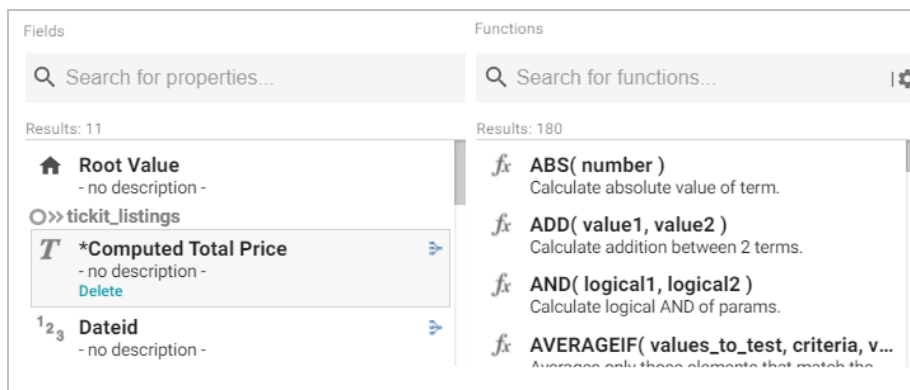


The dialog box titled "Save formula as a computed property" contains the following fields:

- Title:** A text input field with an asterisk (*) indicating it is required.
- Description:** A text input field.
- Ontology:** A dropdown menu currently showing "Tickets - Auto".

At the bottom right, there are two buttons: "CANCEL" and "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.



The screenshot shows two panels side-by-side:

- Fields:** Contains a search bar "Search for properties...". Below it, "Results: 11" are listed. The first is "Root Value" with "- no description -". The second is "tickit_listings" with a sub-property "*Computed Total Price" (marked with an asterisk) and "- no description -". Below that is "Dateid" with "- no description -".
- Functions:** Contains a search bar "Search for functions...". Below it, "Results: 180" are listed. The first few are:
 - ABS(number)**: Calculate absolute value of term.
 - ADD(value1, value2)**: Calculate addition between 2 terms.
 - AND(logical1, logical2)**: Calculate logical AND of params.
 - AVERAGEIF(values_to_test, criteria, v...**: Averages only those elements that match the...

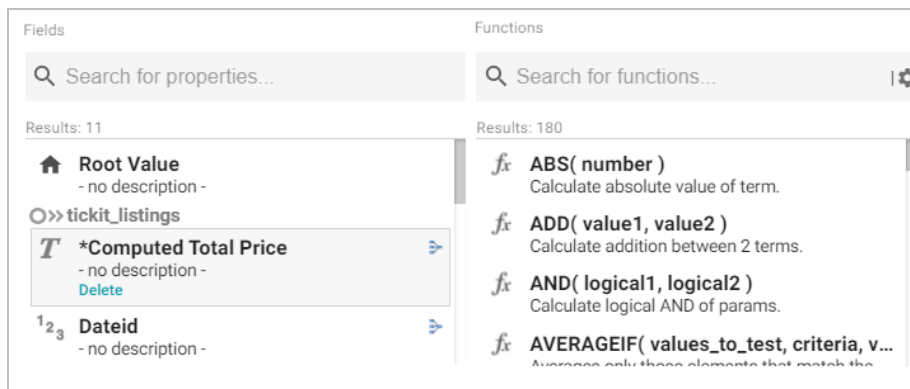
6. Click **Close** to close the dialog box.

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 a Lens](#)

[Function and Formula Reference](#)

Searching for Text in Unstructured Documents

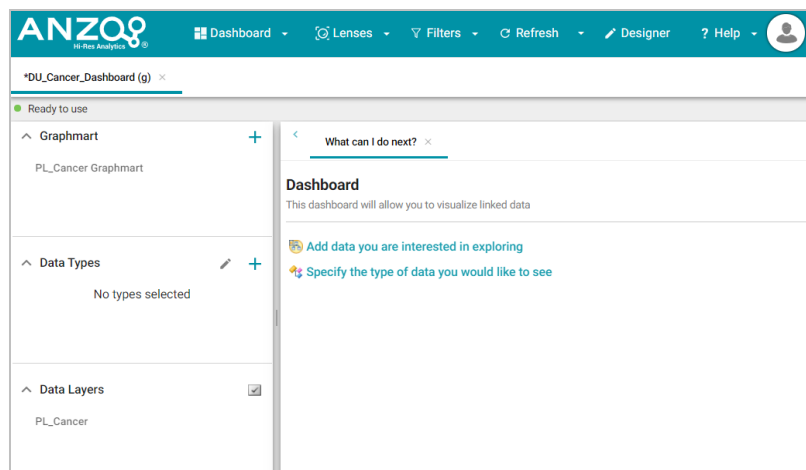
Anzo Hi-Res Analytics incorporates the [Elasticsearch](#) search engine to enable you to perform full text searches on unstructured documents. This topic provides instructions for creating a dashboard with text search capability and running a search across unstructured documents.

For information about running a pipeline to create an unstructured document data set, see [Onboarding Unstructured Data](#).

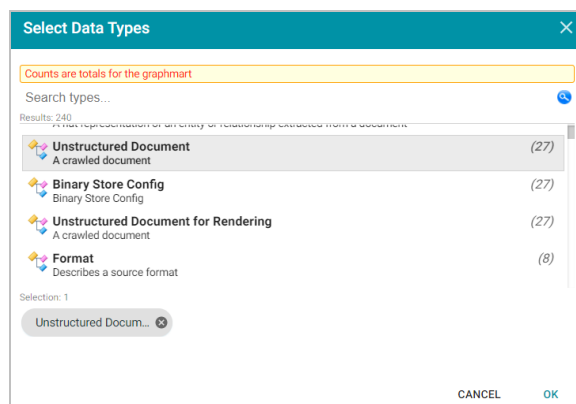
1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays a list of the existing Graphmarts. For example:

<div> <div> <div> <div></div> <div>Search</div> </div> <div>Sort By: Title</div> <div>View: <div></div></div> </div> <div>Add Graphmart</div> </div>									
<input type="checkbox"/>	Title	Creator	Status	Statements	Last Accessed	Last Updated	Tags	Actions	
	DB - northwind to Store Graphmart	System Admi...	Ready to us	9,172	2 hours ago	2 hours ago			
	Datafox Graphmart	System Admi...							
	Flight Data Graphmart	System Admi...	Ready to us	265,696	2 minutes ago	2 minutes ago			
	Tickets Graphmart	System Admi...	Ready to us	4,088,422	5 hours ago	5 hours ago			

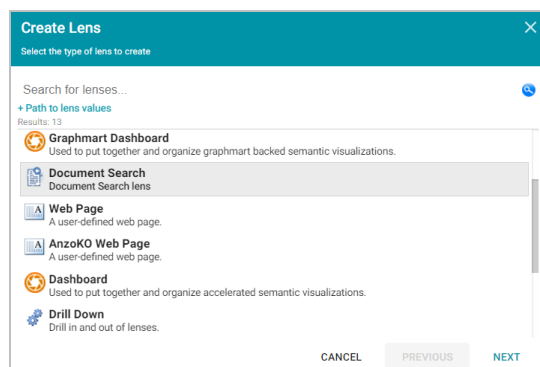
- On the Graphmarts screen, click the name of the graphmart that contains the unstructured documents. Anzo displays the graphmart overview screen.
- Click the **Create Dashboard** button. The Hi-Res Analytics application opens and displays the Create Dashboard dialog box. Leave **Graphmart Dashboard** selected and click **Next**.
- 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:



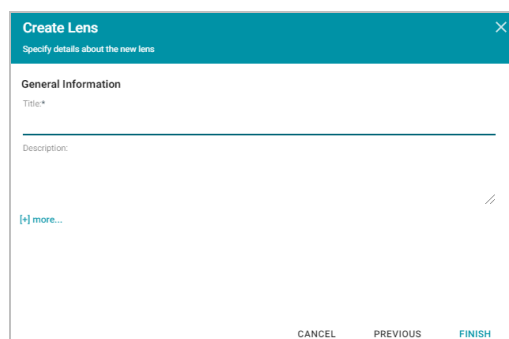
- In the Data Types panel, click the plus icon (+) to open the Select Data Types dialog box. In the dialog box, select **Unstructured Document**. For example:



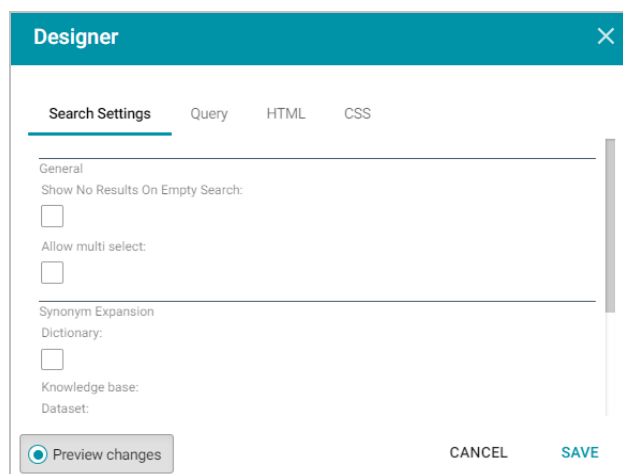
6. Click **OK**. Anzo adds the data type to the Data Types panel.
7. Next, click the **Lenses** button in the main toolbar and select **New** from the drop-down list. Anzo opens the Create Lens dialog box.



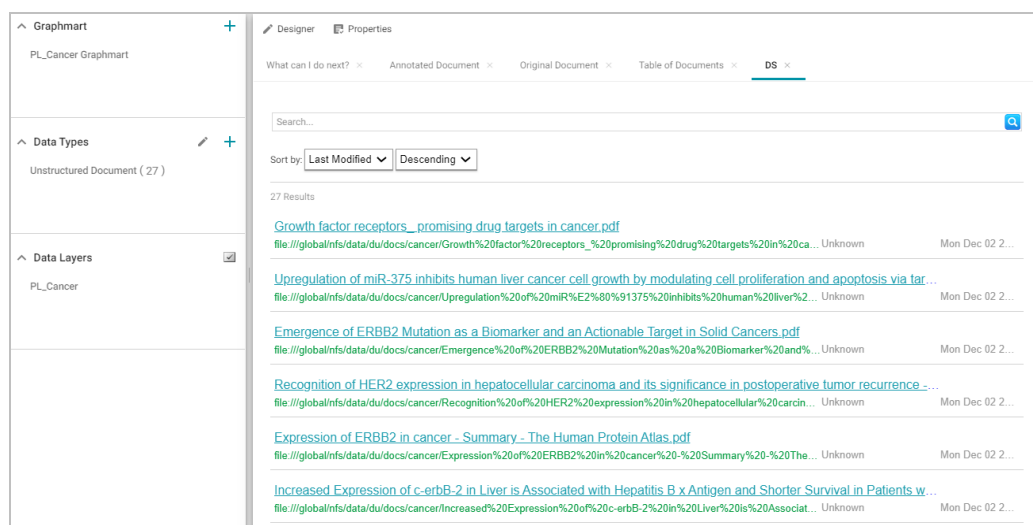
8. In the Create Lens dialog box, select **Document Search** and then click **Next**. Anzo displays the General Information dialog box.



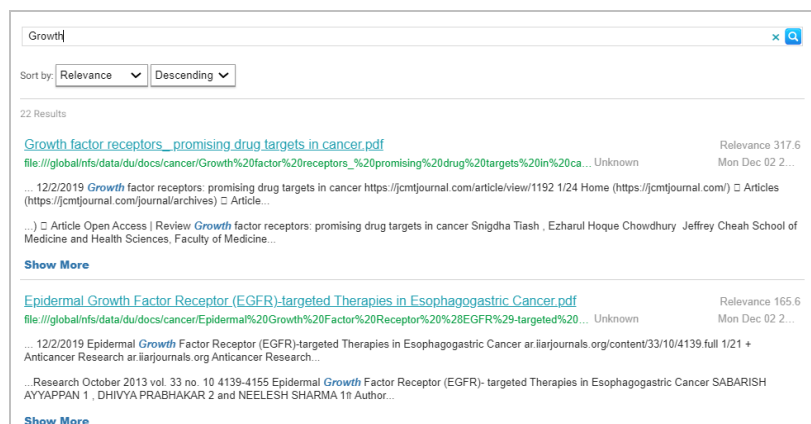
9. Type a name for the lens in the **Title** field and include an optional **Description**. Then click **Finish**. Anzo opens the Document Search Designer where you can configure the search settings or customize the style sheet, query, and HTML, if necessary. For example:



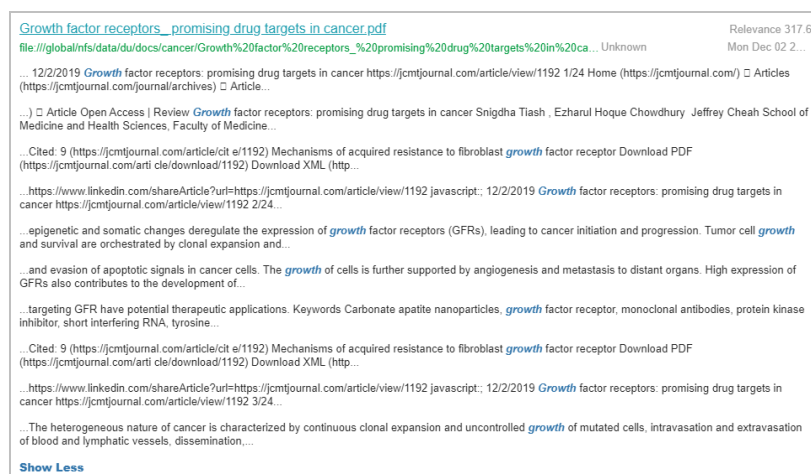
10. In the Designer, change the optional search settings as needed. The list below describes each option:
 - **Show No Results on Empty Search:** Determines whether documents are listed in the search results before a search is run. When enabled, the Document Search lens remains blank until a search is 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:



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:

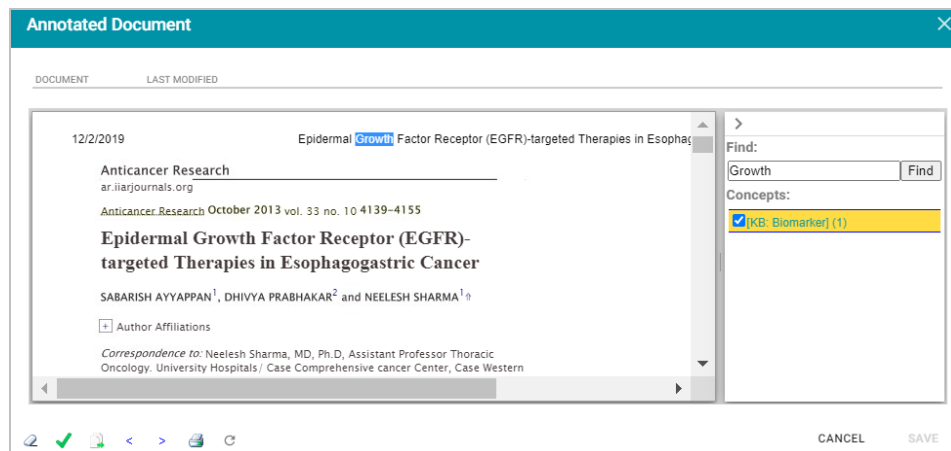


Clicking **Show More** expands the result to display additional matches. For example:



- 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 `cool?`, 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 a Lens](#)

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	<ul style="list-style-type: none"> • Search for and open accessible dashboards. • Save As a new dashboard. • Share the dashboard. • View dashboard Properties. • View lens Properties. • Export lenses.
Modify	In addition to the Read permission described above, the Modify set includes the Write and Delete permissions described below.	
	Write	<ul style="list-style-type: none"> • Use the dashboard Designer to change the dashboard. • Clone lenses.
	Delete	<ul style="list-style-type: none"> • Delete the dashboard.
Admin	In addition to the Read , Write , and Delete permissions described above, the Admin set includes the Manage permission described below.	
	Manage	<ul style="list-style-type: none"> • 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.

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:
View	Read	<ul style="list-style-type: none"> Search for and open accessible lenses. View lens Properties. Export lenses.
	In addition to the Read permission described above, the Modify set includes the Write and Delete permissions described below.	
Modify	Write	<ul style="list-style-type: none"> Use the lens Designer to change the lens. Rename the lens. Clone the lens.
	Delete	<ul style="list-style-type: none"> Delete the lens.

Set	Permission	Allows a user to:
Admin	In addition to the Read , Write , and Delete permissions described above, the Admin set includes the Manage permission described below.	
	Manage	<ul style="list-style-type: none">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.

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
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

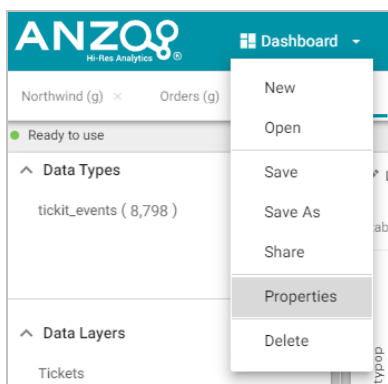
Data can be restricted at a higher level than a dashboard. Though users might have access to view your dashboards and lenses, graphmart permissions determine whether they can view the data that the dashboard displays.

1. In the Anzo application, expand the **Access** menu and click **Hi-Res Analytics**. Anzo displays the Hi-Res Analytics screen, which lists the existing dashboards. For example:

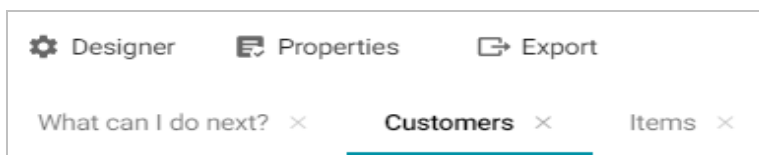
<div> <div> <div> <div></div> <div>Search</div> </div> <div>Sort By: Title</div> <div>View: <div></div></div> </div> </div>								
	Title	Description	Datasets Count	Instance Status	Query Engine Sta	Updated Date	Tags	Actions
	Listings		0			Jun 24, 2020		<div></div>
	Northwind		1	Ready to use	Ready to use	Jun 18, 2020		<div></div>
	Orders		1	Ready to use	Ready to use	Jun 18, 2020		<div></div>
	Tickets		2	Ready to use	Ready to use	Jun 24, 2020		<div></div>

- Click the name of the dashboard for which you want to modify access. Anzo opens the dashboard in the Hi-Res Analytics application.
- 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**.

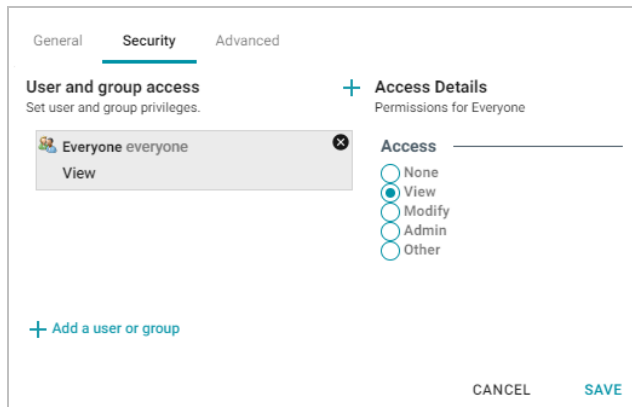
Note Sharing a dashboard automatically shares the lenses in that dashboard.



- 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.

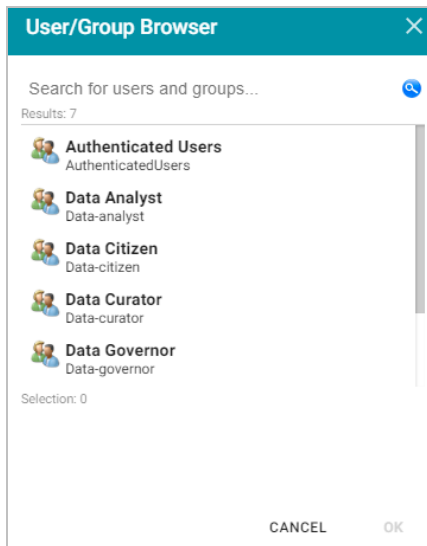


- In the Properties dialog box, click the **Security** tab. This tab lists the available groups and users who can view this dashboard or lens.



5. 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).

6. 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 a Lens](#)

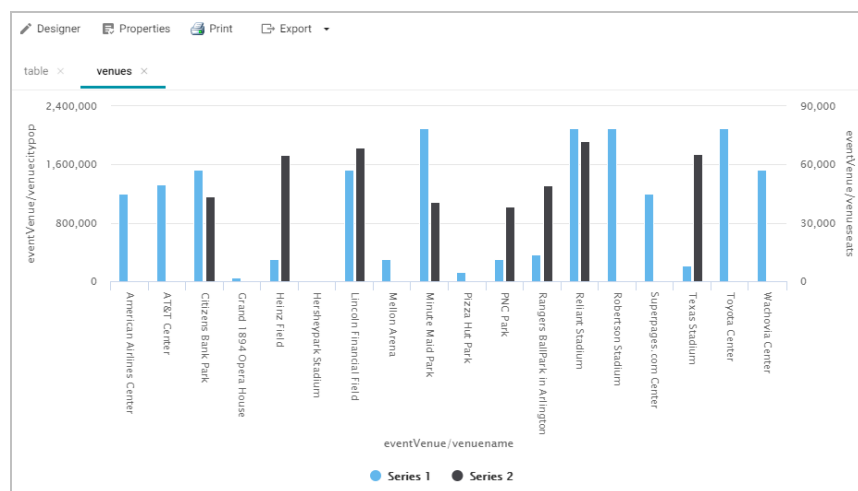
[Exporting a Lens](#)

[Deleting a Lens](#)

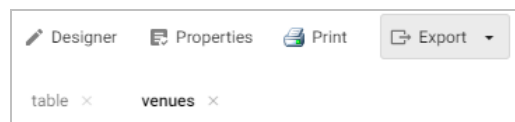
Exporting a Lens

If you have dashboards with Table and Chart lenses, you can export those lenses from the Hi-Res Analytics application. Charts can be exported as images in JPEG, PNG, or SVG format, and tables can be exported to CSV or JSON 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.



4. 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.

5. If the lens is a table, the Export Options dialog box is displayed:



The image shows a dialog box titled "Export Options" with a close button (X) in the top right corner. It contains four input fields: "File name:" (empty), "Format:" (set to "CSV"), "Multi Valued Column Separator:" (set to "[pipe]"), and a checked checkbox for "Export Headers". At the bottom right, there are "CANCEL" and "OK" buttons.

6. 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.
7. Click **OK** to download the file to your computer.

Related Topics

[Creating a Lens](#)

[Deleting a Lens](#)

[Creating Dashboard Filters](#)

Deleting a Lens

This topic provides information about the permissions that are required for deleting lenses as well as instructions for deleting a lens.

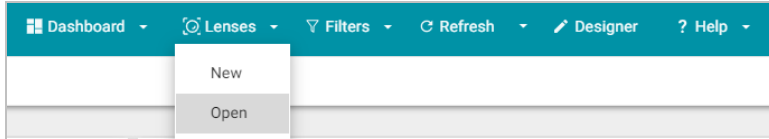
Required Permissions for Deleting Lenses

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.

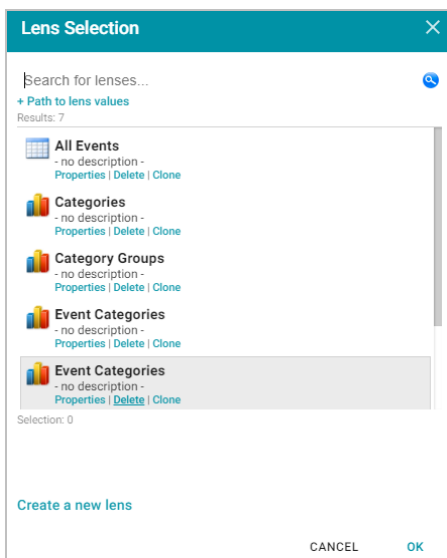
1. In the Hi-Res Analytics application, click the **Lenses** menu in the main toolbar and select **Open**.



The Lens Selection dialog box is displayed.

2. 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:



3. 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.

Related Topics

[Sharing Access to a Dashboard or Lens](#)

[Creating a Lens](#)

Function and Formula Reference

This section describes the standard and advanced functions that are available when working with Hi-Res Analytics. For information about using functions in dashboards, see [Calculating Values in Lenses and Filters](#).

- [Functions on Strings](#)
- [Functions on RDF Terms](#)
- [Functions on Numerics](#)

- [Functions on Dates, Times, and Durations](#)
- [Functions on Boolean Values](#)
- [Window Aggregate Functions](#)

Functions on Strings

The table below details the Anzo functions for string data types.

Function (syntax) Argument: Data Type	Description	Return Type
BUSINESS_ENTITY_EXCLUDER(text) text: string	Removes from strings suffixes that represent business entities.	String
CONCATURL(text, ...) text: string	Concatenates the values for the specified properties or expressions and returns the concatenation as an xsd:anyURI value.	URI
CONCATENATE(text, ...) text: string	Concatenates the values for the specified properties or expressions and returns the concatenation as an xsd:string value.	String
CONTAINS(text, pattern) text: string pattern: string	Determines if the values for a property contain the specified string. Results are grouped under True or False.	Boolean
ENCODE_FOR_URI(text) text: string	Returns results encoded as URIs.	URI
ESCAPEHTML(text) text: string	Escapes the specified string for use in HTML.	String
FIND(find_text, within_text, start_num) find_text: string within_text: string start_num: integer	Determines if the specified text exists in another text string.	Integer

Function (syntax) Argument: Data Type	Description	Return Type
GROUPCONCAT(separator, valueSeparator, serialize, valueLimit, rowLimit, delimitBlanks, text) separator: string valueSeparator: string serialize: boolean valueLimit: integer rowLimit: integer delimitBlanks: boolean text: string	Performs a string concatenation all of the values that are bound to a property.	String
LANG(value) value: string	Returns any language tags that exist for the specified property's literal values and groups the results under any language tags or "blank" if a language tag does not exist for a record.	String
LANGMATCHES(language_tag, language_range) language_tag: string language_range: string	Determines whether any of the values for a property contain a language tag from the specified range of tags.	Boolean
LCASE(value) value: string literal	Converts string values to lower case in the filter.	String
LEFT(text, number_of_characters) text: string number_of_characters: integer	Returns the specified number of characters starting from the left of the string.	String
LEN(value) value: string	Calculates the length of the string values.	Integer

Function (syntax) Argument: Data Type	Description	Return Type
LEVENSHTEIN_DIST(value1, value2) value1: string value2: string	Calculates the Levenshtein distance or measure of similarity between the specified strings. The distance is the number of edits required to transform the first string into the second string.	Integer
LOWER(text, language, country, variant) text: string language: string country: string variant: string	Converts string values to lower case letters.	String
MD5(value) value: term	Calculates the MD5 hash of string values.	String
MID(text, start_num, num_chars) text: string start_num: integer num_chars: integer	Returns the specified number of characters from a string, starting from the chosen position in the string.	String
REGEX(text, pattern, [flags]) text: string pattern: string flags: string	Determines whether the specified string matches a regular expression pattern. You can use the optional flags argument to include one or more modifier flags that further define the pattern. For information about flags, see the Flags section of the W3C XQuery 1.0 and XPath 2.0 Functions and Operators specification.	Boolean
REGEXP_SUBSTR(text, pattern, [startPosition], [nthAppearance]) text: string pattern: string startPosition: integer nthAppearance: integer	Searches a string for a regular expression pattern and returns the matching substring. In the syntax, text is the string to search, pattern is the regex pattern to match, the optional startPosition specifies the number of characters from the beginning of the string to start searching for matches (the default value is 1), and the optional nthAppearance specifies which occurrence of the pattern to match (the default value is 1).	String

Function (syntax) Argument: Data Type	Description	Return Type
REPLACE(text, pattern, replacement, flags) text: string pattern: string replacement: string flags: string	Extends the REGEX function to provide the ability to take a replacement pattern and return the replaced string.	String
RIGHT(text, num_chars) text: string num_chars: integer	Returns the specified number of characters, starting at the end of a string.	String
RULE_BASED_LOCALITY_SENSITIVE_HASH(text) text: string	Transforms the specified string by normalizing across spacing and characters, removing punctuation and special characters, and cleaning common English affixes.	String
SEARCH(text, pattern, required, wildcard, remove, escape) text: string pattern: string required: string wildcard: string remove: string escape: string	Uses text search semantics to determine whether the specified text matches a regular expression pattern.	Boolean
STRAFTER(text, pattern) text: string pattern: string	Returns the part of a string that comes after the pattern that you specify.	String
STRBEFORE(text, pattern) text: string pattern: string	Returns the part of a string that comes before the pattern that you specify.	String

Function (syntax) Argument: Data Type	Description	Return Type
STRDT(value, <i>URI</i> ("data_type")) value: string data_type: string	Casts a string value to the specified data type. A URI function, such as TOUR, IRI, or URI, is required to specify the data type, which is a URI. For example, the following formula casts a regionkey column from a string to an integer: <code>STRDT ([{region}regionkey], TOUR ("xsd:int"))</code>	Term
STREND(text, pattern) text: string pattern: string	Determines whether the specified string ends with the given pattern.	Boolean
STRLANG(text, language) text: string language: string	Constructs a literal value with the specified language tag.	String
STRSTARTS(text, pattern)	Determines whether the specified string value starts with the given pattern.	Boolean
STRUUID()	Returns a string that is the result of generating a Universally Unique Identifier (UUID).	String
SUBSTITUTE(text, old_text, new_text, instance_num) text: string old_text: string new_text: string instance_num: integer	Substitutes new text for old text in a string.	String
TOUR(value) value: string	Casts a string value to a URI.	URI
TRIM(text) text: string	Removes all spaces from values except for single spaces between words.	String

Function (syntax) Argument: Data Type	Description	Return Type
UPPER(text, language, country, variant) text: string language: string country: string variant: string	Converts all lower case letters to upper case letters.	String

Functions on RDF Terms

The table below details the Anzo functions for RDF term types: literal values, URIs, and blank nodes.

Function (syntax) Argument: Data Type	Description	Return Type
ADD(term1, term2) term1: term term2: term	Adds the results from the expressions that you specify.	Term
AVERAGEIF(values_to_test, criterion, values_to_average) values_to_test: term criterion: term values_to_average: integer	Calculates the averages of the values that meet the specified criterion.	Integer
AVERAGEIFS(values_to_average, values_to_test, criteria, ...) values_to_average: numeric values_to_test: term criteria: term	Similar to the AVERAGEIF function but enables you to specify multiple criteria.	Integer
BNODE(term) term: term	For use with Presence, Hierarchy, and Types filters to determine whether blank nodes exist for properties. You can also perform the BNODE function on literal values.	Term

Function (syntax) Argument: Data Type	Description	Return Type
BOOLEAN(term) term: term	Creates an xsd:boolean type based on label of the input term.	Boolean
BOUND(term) term: term	Determines which records include a value for the specified property and returns "True" for records that include a value or "False" for records that do not include a value.	Boolean
CASE(value, criteria, ..., result, ..., default) value: term criteria: term result: term default: term	Enables you to add IF/THEN logic. CASE expressions evaluate a series of conditions for the properties that you specify and return results when the test returns true. The optional "default" argument is a default value to return if none of the tests pass.	Term
CEILING(number) number: term	Calculates the ceiling (the next whole number up from the value if the value has a fractional part) of the values that exist for the selected property and then groups the results into the list of ceiling values. CEILING returns the value itself if it is a whole number.	Term
CHOOSE_BY_MAX(test, value) test: term value: term	Calculates the maximum values from the first expression or property and returns the values from the second expression or property that correspond to the maximum values. For example, in an imaginary sales data set, the following formula returns the IDs for the buyers who spent the most: CHOOSE_BY_MAX ([{Sales}Price Paid], [{Sales}Buyer Id])	Term
CHOOSE_BY_MIN(test, value) test: term value: term	Calculates the minimum values from the first expression or property and returns the values from the second expression or property that correspond to the minimum values. For example, in an imaginary sales data set, the following formula returns the IDs for the buyers who spent the least: CHOOSE_BY_MIN ([{Sales}Price Paid], [{Sales}Buyer Id])	Term

Function (syntax) Argument: Data Type	Description	Return Type
COALESCE(value, ...) value: term	Evaluates any number of expressions and returns the results for the first expression that does not raise an error. Errors occur if an expression evaluates to an unbound variable or a non-RDF term.	Term
COUNT(value) value: term	Counts the number of values for the selected property.	Integer
COUNT_DISTINCT(value) value: term	Counts the number of unique values for the selected property.	Integer
COUNTIF(value, criterion) value: term criterion: term	Calculates the counts of the values that meet the specified criterion.	Integer
COUNTIFS(value, criteria, ...) value: term criteria: term	Similar to the COUNTIF function but enables you to specify multiple criteria.	Integer
DATATYPE(term) term: literal value	For use with Presence, Hierarchy, and Types filters.	URI
DATEVALUE(date_text) date_text: term	Groups results under the specified literal date value.	Date
EQUAL(value1, value2) value1: term value2: term	Determines whether value1 is equal to value2.	Boolean
GE(value1, value2) value1: term value2: term	Performs a greater than or equal to (\geq) comparison between value1 and value2.	Boolean
GT(value1, value2) GT functions on numerics, booleans, dateTimes, and terms in this priority order	Performs a greater than ($>$) comparison between value1 and value2.	Boolean

Function (syntax) Argument: Data Type	Description	Return Type
IF(test, value_if_true, value_if_false, value_if_error) test: boolean value_if_true: term value_if_false: term value_if_error: term	Evaluates one expression and returns a second expression depending on the answer.	Term
IFERROR(value, value_if_error, ...) value: term value_if_error: term	Synonym for COALESCE.	Term
IN(value, test_value, ...) value: term test_value: term	Determines whether any of the values for the first property are found in the other specified expressions or properties. Anzo groups the results under True or False.	Boolean
ISBLANK(value) value: term	Determines whether the property has blank node values and groups the results under True or False.	Boolean
ISDATATYPE(value, data_type) value: term data_type: URI	Determines whether the values for a property are the specified data type and groups the results under True or False.	Boolean
ISERROR(value) value: term	Determines whether the argument evaluates to an error and groups the results under True or False.	Boolean
ISIRI(value) ISURI(value) value: term	Determines whether the argument is an IRI. ISIRI and ISURI return true if the value is an IRI or URI (and is not blank) and false if it is not.	Boolean
ISLITERAL(value) value: term	Determines whether the property has literal values.	Boolean

Function (syntax) Argument: Data Type	Description	Return Type
ISNUMERIC(value) value: term	Determines whether the property has numeric values.	Boolean
LE(value1, value2) value1: term value2: term	Performs a less than or equal to (\leq) comparison between value1 and value2.	Boolean
LOCALNAME(URI)	Returns only the local name portion of a URI.	String
LONG(value) value: term	Displays numeric values in xsd:long format.	Long
LT(value1, value2) value1: term value2: term	Performs a less than ($<$) comparison between value1 and value2.	Boolean
MAX(value, ...) value: term	Aggregate function that calculates the maximum values for each aggregate group.	Term
MAXVAL(value, ...) value: literal	Computes the maximum values for the specified arguments.	Literal
MD5(value) value: term	Calculates the MD5 hash of string values.	String
METADATAGRAPHURI(URI)	Returns the metadata graph URI for the specified input URI.	URI
MIN(value, ...) value: term	Aggregate function that calculates the minimum values	Term
MINVAL(value, ...) value: literal	Computes the minimum values for the specified arguments.	Literal
MODE(value) value: term	Aggregate function that returns the number that occurs most frequently in each aggregate group.	Numeric

Function (syntax) Argument: Data Type	Description	Return Type
NAMESPACE(URI)	Returns the namespace for the specified URI.	String
NOT(value) value: boolean	Performs logical negation on the specified expression.	Boolean
NOT_EQUAL(value1, value2) value1: term value2: term	Performs a not equal (!=) comparison between value1 and value2.	Boolean
NOT_IN(value, test_value, ...) value: term test_value: term	Tests whether the value is not found in the test_value list of expressions.	Boolean
OR(logical1, logical2) logical1: boolean logical2: boolean	Calculates the logical OR of the input values.	Boolean
PARTITIONINDEX(value, start, interval) value: literal start: literal interval: literal	Returns the zero-based index of the bucket in which the value falls. The buckets start at the specified start and are sized according to the specified interval. The first bucket is (start, start+interval): 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 value is less than start or if the comparison is indeterminate for date and time data types.	Integer
SAMETERM(term1, term2) term1: term term2: term	Determines whether the specified RDF terms are the same.	Boolean
SAMPLE(term) term: term	Returns an arbitrary value from the group to represent the given variable.	Term
SERIALIZE(term) term: term	Returns the string representation of the specified term.	String
SHA1(term)	Calculates the SHA1 hash of the specified term.	String

Function (syntax) Argument: Data Type	Description	Return Type
SHA224(term)	Calculates the SHA224 hash of the specified term.	String
SHA256(term)	Calculates the SHA256 hash of the specified term.	String
SHA384(term)	Calculates the SHA384 hash of the specified term.	String
SHA512(term)	Calculates the SHA512 hash of the specified term.	String
STR(term)	Returns a string representation of the values for the selected property.	String
STRLEN(term)	Calculates the length of the specified term.	Integer
SUBSTR(term, start, [length]) term: term start: integer length: integer	Returns a substring of the specified term. The start argument indicates the character position to start the substring with. The first character in the term is position 1. The optional length argument specifies the number of characters to return.	String
SUMIF(values_to_test, criterion, values_to_sum) values_to_test: term criterion: term values_to_sum: numeric	Calculates the sums of the values that match the specified criterion.	Integer
SUMIFS(values_to_sum, values_to_test, criteria, ...) values_to_sum: numeric values_to_test: term criteria: term	Similar to the SUMIF function but enables you to specify multiple criteria.	Integer
TEXT(value, format) value: term format: string	Formats a term value as text.	String
UCASE(term) term: term	Returns the specified term as an uppercase string value.	String

Function (syntax) Argument: Data Type	Description	Return Type
UNBOUND()	Returns an unbound term	Term
UUID()	Generates a new IRI from the Universally Unique Identifier (UUID) Uniform Resource Name (URN) namespace.	URI

Functions on Numerics

The table below details the Anzo functions for numeric data types. "Term" indicates an RDF term type value: a literal value, URI, or blank node.

Function (syntax) Argument: Data Type	Description	Return Type
ABS(number) number: numeric value	Calculates the absolute values that exist for the selected property.	Numeric
ADD(term1, term2) term1: term term2: term	Adds the results from the expressions that you specify.	Term
AVERAGEIF(values_to_test, criterion, values_to_average) values_to_test: term criterion: term values_to_average: integer	Calculates the averages of the values that meet the specified criterion.	Integer
AVERAGEIFS(values_to_average, values_to_test, criteria, ...) values_to_average: numeric values_to_test: term criteria: term	Similar to the AVERAGEIF function but enables you to specify multiple criteria.	Integer
AVG(number) number: numeric value	Calculates the averages of the values that exist for the selected property.	Numeric

Function (syntax) Argument: Data Type	Description	Return Type
CEILING(number) number: term	Calculates the ceiling (the next whole number up from the value if the value has a fractional part) of the values that exist for the selected property and then groups the results into the list of ceiling values. CEILING returns the value itself if it is a whole number.	Term
CHOOSE_BY_MAX(test, value) test: term value: term	Calculates the maximum values from the first expression or property and returns the values from the second expression or property that correspond to the maximum values. For example, in an imaginary sales data set, the following formula returns the IDs for the buyers who spent the most: <code>CHOOSE_BY_MAX ([{Sales}Price Paid], [{Sales}Buyer Id])</code>	Term
CHOOSE_BY_MIN(test, value) test: term value: term	Calculates the minimum values from the first expression or property and returns the values from the second expression or property that correspond to the minimum values. For example, in an imaginary sales data set, the following formula returns the IDs for the buyers who spent the least: <code>CHOOSE_BY_MIN ([{Sales}Price Paid], [{Sales}Buyer Id])</code>	Term
COS(angle) angle: double	Calculates the cosines of the values that exist for the selected property.	Double
DECIMAL(value) value: term	Returns numeric results in decimal format.	Numeric
DIVIDE(value1, value2) value1: numeric value2: numeric	Divides the values for the first property or expression by the values for the second property or expression and groups the results into the list of division values.	Numeric
DOUBLE(value) value: term	Displays the results of the specified numeric property in xsd:double format.	Double
EXP(number) number: double	Raises the results to the power of the specified number.	Double

Function (syntax) Argument: Data Type	Description	Return Type
FACT(number) number: integer	Calculates the factorial of the results by the specified number.	Integer
FLOAT(value) value: term	Returns numeric results in float format and groups the results into the list of float values	Float
FLOOR(number) number: term	Calculates the floor (the closest whole number down from the value if the value has a fractional part) of the values that exist for the selected property and then groups the results into the list of floor values. FLOOR returns the value itself if it is a whole number.	Term
FORMATDATE(value, format) value: term format: string	Formats a numeric or date value into date text.	String
FORMATFRACTION(value, tolerance, seperate_whole_number) value: term tolerance: double seperate_whole_number: boolean	Returns results in fraction format rather than decimal format.	String
FORMATNUMBER(value, format) value: term format: numeric format	Formats a numeric value into text in the specified format.	
GE(value1, value2) value1: term value2: term	Performs a greater than or equal to (\geq) comparison between value1 and value2.	Boolean

Function (syntax) Argument: Data Type	Description	Return Type
GT(value1, value2) GT functions on numerics, booleans, dateTimes, and terms in this priority order	Performs a greater than (>) comparison between value1 and value2.	Boolean
HAMMING_DIST(value1, value2) value1: long value2: long	Calculates the hamming distance between two values.	Integer
HAVERSINE_DIST(lat1, lon1, lat2, lon2) lat1: double lon1: double lat2: double lon2: double	Computes the haversine distance between two latitude and longitude values.	Double
INTEGER(value) value: term	Returns numeric results in integer format.	Integer
LE(value1, value2) value1: term value2: term	Performs a less than or equal to (<=) comparison between value1 and value2.	Boolean
LN(value) value: double	Calculates the natural logarithm of numeric values.	Double
LOG(number, base) number: double base: double	Calculates the specified base logarithm of numeric values.	Double
LONG(value) value: term	Displays numeric values in xsd:long format.	Long

Function (syntax) Argument: Data Type	Description	Return Type
LT(value1, value2) value1: term value2: term	Performs a less than (<) comparison between value1 and value2.	Boolean
MAX(value, ...) value: term	Aggregate function that calculates the maximum values for each aggregate group.	Term
MAXVAL(value, ...) value: literal	Computes the maximum values for the specified arguments.	Literal
MEDIAN(value) value: numeric	Aggregate function that calculates the median value for each aggregate group.	Numeric
MIN(value, ...) value: term	Aggregate function that calculates the minimum values	Term
MOD(number, divisor) number: integer divisor: integer	Calculates the modulo or remainder of the division between two numeric values.	Integer
MODE(value) value: term	Aggregate function that returns the number that occurs most frequently in each aggregate group.	Numeric
MODEPERCENT(value) value: numeric	Aggregate function that calculates the percentage of the values that belong to the mode.	Numeric
MULTIPLY(value1, value2) value1: numeric value2: numeric	Multiplies value1 by value2.	Numeric
NPV(rate, year, value) rate: numeric year: numeric value: numeric	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).	Numeric
PI()	Returns the value for PI.	Double

Function (syntax) Argument: Data Type	Description	Return Type
POWER(number, power) number: numeric power: numeric	Raises the specified number to the specified power.	Double
QUOTIENT(numerator, denominator) numerator: numeric denominator: numeric	Calculates the quotient for the specified values.	Integer
RAD(angle) angle: double	Converts degrees to radians.	Double
RAND()	Returns a random double value between 0 and 1.	Double
RANDBETWEEN(bottom, top) bottom: numeric top: numeric	Returns a random integer between the specified values (inclusive). If the input values are decimal types, Anzo returns a random integer between the ceil(bottom) and floor(top).	Integer
ROUND(number) number: double	Rounds a numeric value to the nearest integer.	Integer
ROUNDDOWN(number, num_digits) number: numeric num_digits: integer	Rounds a numeric value down by the specified number of digits.	Numeric
ROUNDUP(number, num_digits) number: numeric num_digits: integer	Rounds a numeric value up by the specified number of digits.	Numeric
SIN(angle) angle: double	Calculates the sine of the specified value.	Double
SQRT(number) number: double	Calculates the square root of the specified number.	Double

Function (syntax) Argument: Data Type	Description	Return Type
STDEV(number) number: numeric	Calculates the standard deviation of a group of numbers.	Numeric
STDEVP(number) number: numeric	Calculates the standard deviation product of a group of numbers.	Numeric
SUM(number) number: numeric	Calculates the sums of the values that exist for the selected property.	Integer
SUMIF(values_to_test, criterion, values_to_sum) values_to_test: term criterion: term values_to_sum: numeric	Calculates the sums of the values that match the specified criterion.	Integer
SUMIFS(values_to_sum, values_to_test, criteria, ...) values_to_sum: numeric values_to_test: term criteria: term	Similar to the SUMIF function but enables you to specify multiple criteria.	Integer
SUMPRODUCT(number) number: numeric	Calculates the sum of the product of the specified numeric values.	Numeric
SUMSQ(number) number: numeric	Calculates the square root of each number in the group and adds them all together.	Numeric
TAN(angle) angle: double	Calculates the tangent of the specified angle.	Double
TIME(hour, minute, second) hour: integer minute: integer second: integer	Converts the specified hour, minute, and second integer values as a time value.	Time

Function (syntax) Argument: Data Type	Description	Return Type
VAR(number) number: numeric	Calculates the variance for a group of numbers, i.e., how widely the values vary from the average of the values.	Numeric
VARP(number) number: numeric	Calculates the variance for a sample group of numbers, i.e., how widely the values vary from the average of the values.	Numeric

Functions on Dates, Times, and Durations

The table below details the Anzo functions for date, time, and duration data types.

Function (syntax) Argument: Data Type	Description	Return Type
DATE(year, month, day) year: integer month: integer day: integer	Groups results under the date (year, month, day) that you type.	Date
DATEPART(date_value) date_value: date or dateTime	Returns the date portion of a dateTime value.	Date
DATETIME(value) value: datetime, string (the string is parsed to datetime), or long (time in milliseconds since epoch)	Returns the appropriate dateTime based on the specified input value.	Date
DATEVALUE(date_text) date_text: term	Groups results under the specified literal date value.	Date
DAY(date_value) date_value: date or dateTime	Returns as an integer (1-31) the day portions of the values that exist for the selected property.	Integer
DAYSFROMDURATION(value) value: duration or numeric	Returns the day portion of duration values.	String

Function (syntax) Argument: Data Type	Description	Return Type
DUR_TO_MILLIS(value) value: date or dateTime	Displays date or date time values as the time in milliseconds.	Long
DURATION(number) number: long	Displays the specified values in duration format (PnYnMnDTnHnMnS).	Duration
DURATIONFORMAT(millis, format) millis: numeric format: duration	Displays the specified values in duration format and groups the results into the list of durations. This function enables you to specify the duration format to use. The default format is H:mm:ss.SSS.	String
DURATIONPERIODFORMAT (start, end, format) start: duration or numeric end: duration or numeric format: duration	Calculates the duration between the specified start and end values. This function also enables you to specify the duration format. The default format is PYYYYMMDDThhmmss.SSS.	String
FORMATDATE(value, format) value: term format: string	Formats a numeric or date value into date text.	String
GT(value1, value2) GT functions on numerics, booleans, dateTimes, and terms in this priority order	Performs a greater than (>) comparison between value1 and value2.	Boolean
HOUR(value) value: time or dateTime	Returns the hour portions of the values that exist for the selected property.	Integer

Function (syntax) Argument: Data Type	Description	Return Type
MASKEDDATETIME (value, year, month, day, hour, minute, second, millis) value: date or dateTime year: boolean month: boolean day: boolean hour: boolean minute: boolean second: boolean millis: boolean	Given an xsd:date or an xsd:dateTime value, this function returns the appropriate xsd:dateTime with the included parts of the date set to specific values.	DateTime
MILLIS (value) value: date or dateTime	Displays date or datetime values as the time in milliseconds.	Date
MINUTE (value) value: time	Returns the minute portions of the values that exist for the selected property.	Integer
MONTH (value) value: date	Returns as an integer (1-12) the month portions of the values that exist for the selected property.	Integer
NOW (timezone) timezone: string	Returns the current date and time.	DateTime
NOWMILLIS ()	Returns the current date and time in epoch milliseconds.	Long
PARSEDATETIME (date_string, output_type) date_string: string output_type: URI	Returns the specified string or literal value as a date, time, or datetime value.	DateTime
SECOND (time_value) time_value: date or dateTime	Returns the second portions of the values that exist for the selected property.	Integer

Function (syntax) Argument: Data Type	Description	Return Type
TIME(hour, minute, second) hour: integer minute: integer second: integer	Converts the specified hour, minute, and second integer values as a time value.	Time
TIMEPART(value) value: string, time, or dateTime	Returns the appropriate time based on the input value.	Time
TODAY()	Returns today's date.	Date
WEEKDAY(date_value, return_type) date_value: date return_type: integer	Returns the day of the week that corresponds to the specified date.	Integer
WEEKNUM(date_value, return_type) date_value: date return_type: integer	Returns the week of the year that the specified date occurs in.	Integer
YEAR(date_value) date_value: date or dateTime	Returns as an integer (1900-9999) the year portions of the values that exist for the selected property.	Integer
YEARMONTH(date_value) date_value: date	Returns the year-month of the specified date.	DateTime

Functions on Boolean Values

The table below details the Anzo functions for boolean data types.

Function (syntax) Argument: Data Type	Description	Return Type
AND(logical1, logical2) logical1: boolean logical2: boolean	Calculates the logical AND of the input values.	Boolean

Function (syntax) Argument: Data Type	Description	Return Type
NOT(value) value: boolean	Performs logical negation on the specified expression.	Boolean
OR(logical1, logical2) logical1: boolean logical2: boolean	Calculates the logical OR of the input values.	Boolean

Window Aggregate 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.

Except for WINDOW_NTILE, WINDOW_PERCENTILE, and WINDOW_QUARTILE, use the following syntax for window aggregates:

```
WINDOW_FUNCTION(value, partition_over, order_by, order, start_frame,
start_frame_type, start_frame_value, end_frame_type, end_frame_value)
```

The table below lists the supported window aggregates and provides the syntax for the WINDOW_NTILE, WINDOW_PERCENTILE, and WINDOW_QUARTILE functions.

Function (syntax)	Description	Return Type
WINDOW_AVG	Returns the average of the input values.	Numeric
WINDOW_COUNT	Returns the count of the specified values.	Integer
WINDOW_MAX	Returns the maximum of the input values.	Numeric
WINDOW_MIN	Returns the minimum of the input values.	Numeric
WINDOW_NTILE(ntile, value, order_by, partition_over)	Divides the rows in the partition into the specified number of ranked groups and returns the group that each value belongs to.	Numeric
WINDOW_PERCENTILE(value, order_by, partition_over)	Like using NTILE(100), this function divides the rows in the partition into 100 ranked groups and returns the group that each value belongs to.	Numeric

Function (syntax)	Description	Return Type
WINDOW_PRODUCT	Returns the product of the input values.	Numeric
WINDOW_QUARTILE(value, order_by, partition_over)	Like using NTILE(4), this function divides the rows in the partition into 4 ranked groups and returns the group that each value belongs to.	Numeric
WINDOW_SUM	Returns the sum of the input values.	Numeric

Related Topics

[Calculating Values in Lenses and Filters](#)

Lens Type Reference

Select a lens type from the list below to view reference information for that lens.

- [AnzoKO Web Page Lens](#)
- [Chart Lenses](#)
- [Drill Down Lens](#)
- [Form Lens](#)
- [List Lens](#)
- [Query Lens](#)
- [Resource Tree Navigator Lens](#)
- [Table Lens](#)
- [Web Page Lens](#)

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.

Related Topics

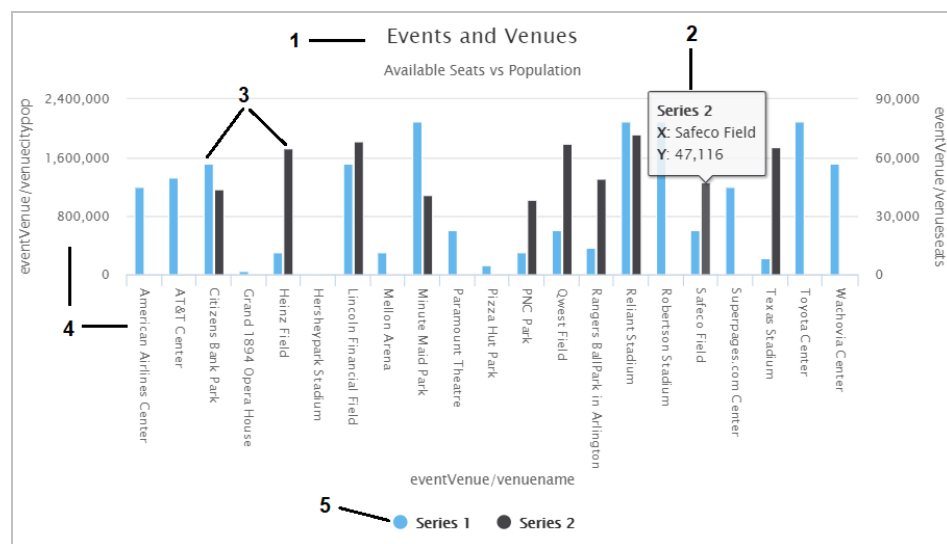
[Creating a Lens](#)

Chart Lenses

Anzo Hi-Res Analytics employs the [Highcharts](#) API to provide interactive chart lenses. This section provides information about chart concepts and describes the general, shared chart settings.

Chart Concepts

This section describes the high-level, basic chart options that you can configure.



- Title and Subtitle:** You can configure titles and subtitles for all charts.
- Tooltips:** Configurable tooltips display details when users place their cursor over an item in the chart.
- 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.
- Axes:** The X and Y axes define the horizontal and vertical coordinates for displaying the data.
- 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

>

☒ Interactive

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





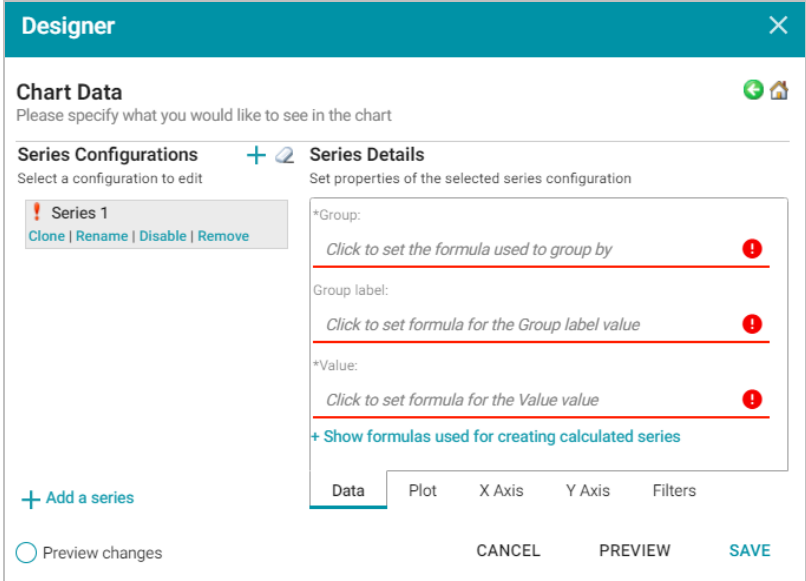
Icon	Description
	Click the eraser icon to erase all series. This action cannot be undone.
	Click the back icon to return to the previous screen.
	Click the home icon to return to the Chart Configuration screen.
	Click the plus icon to add a series.

Chart Data

The Chart Data screen allows you to configure the Properties and formulas that populate your chart. Most charts share the same settings.



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.

Field	Description
Show formulas used for...	<p>Creates a calculated series using the following fields:</p> <p>Series Group: Selects property and functions or formulas used to group data in addition to the Group setting.</p> <p>Series Label: Typically the same as Series Group. The property to use as the series label.</p>

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-

Show in legend:
☒

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 Values

Ascending

☐ 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.
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.
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.

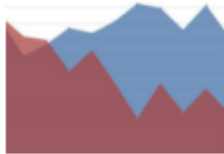

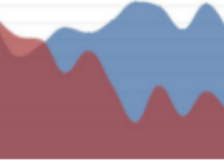
For more information about Series Details settings, see the documentation for specific lens types:

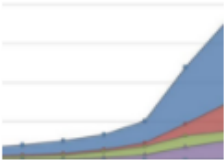

- [Area Chart](#)
- [Bar Chart](#)
- [Bubble Chart](#)
- [Column Chart](#)
- [Funnel Chart](#)
- [Heat Map](#)
- [Line Chart](#)

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.

Example	Type	Description
	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:
☐

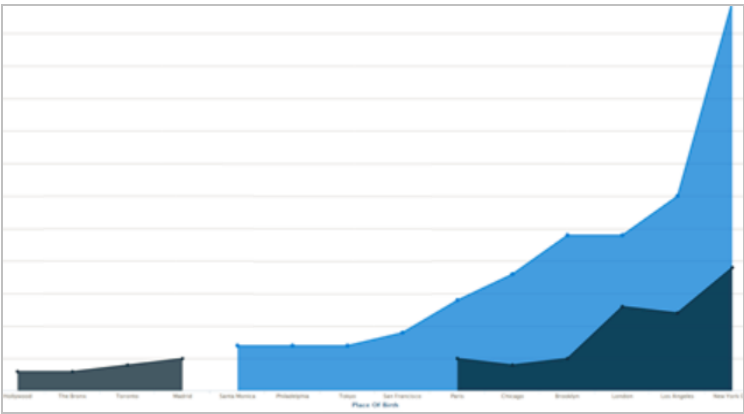
Threshold:
0

Data Plot X Axis Y Axis Filters

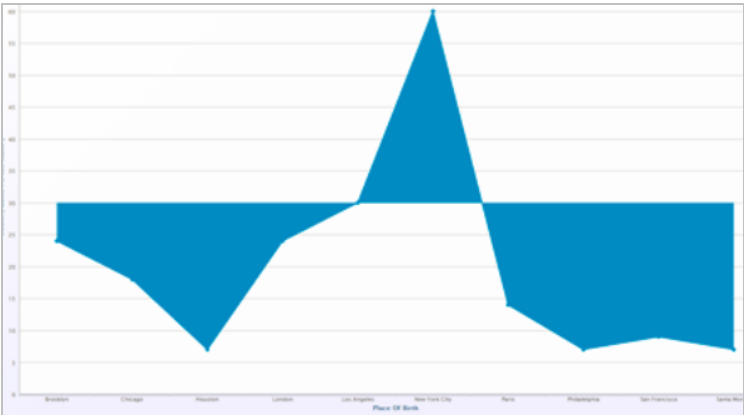
- **Series Chart Markers:** The options in this category enable you to customize the data points that appear on the series line:
 - **Enabled:** Enables or disables series chart markers.
 - **Symbol:** Selects a symbol to mark data points.
 - **Marker Radius:** Defines the marker size in pixels.
 - **Fill 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.





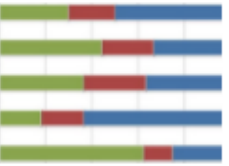
- **Threshold:** Defines the Y axis value to use as a base (starting point) for the shaded area. For example, a Threshold of 0 begins all shading at the value 0. A Threshold of 10 begins the shading at 10 and draws the area chart above or below the threshold as required. For example, the image below shows an area chart with a threshold of 30.



Bar Chart

There are three types of bar charts:

Example	Type	Description
	Clustered Bar	Compares values across categories.

Example	Type	Description
	Stacked Bar	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 Bar	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

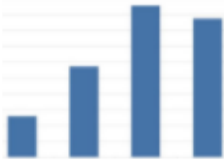
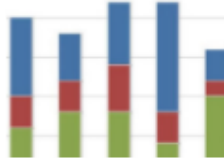

X Filters
No filters specified

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.



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

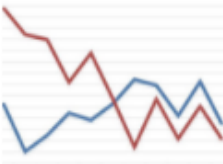
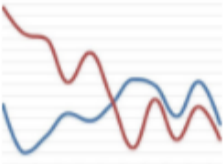
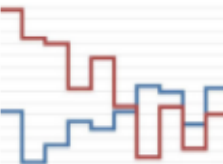
Axis Style
Axis style information

< t X Axis Y Axis Color Axis Filter > v

- **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.
	Spline	Connects value points with curved lines.
	Step Line	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:

Largest70-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.

Related Topics

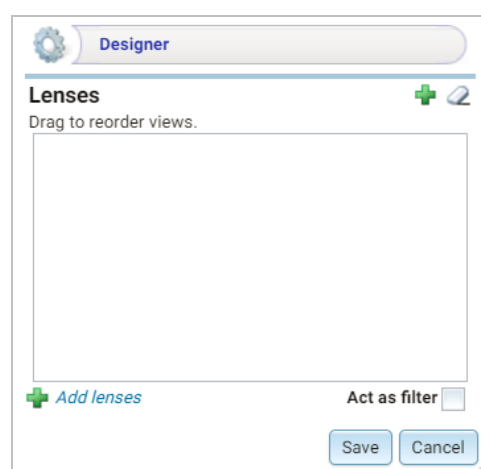
[Creating a Lens](#)

Drill Down Lens

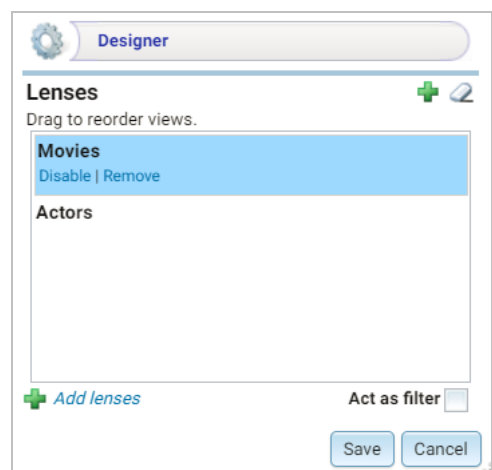
Drill Down lenses combine other lenses into a hierarchical interface. Clicking on an object in one lens opens the next lens in successive order.

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 this example, clicking the drill down icon () next to a movie ID displays the Actors lens, which shows the actor for that movie:

Designer Properties Export

Drill down example

Movies

	MovieID	MovieTitle
	6202854	Die, Mommie, Die!
	34643655	Turbo (film)
	9615983	Sidewalks of New York (2001 film)
	210224	Love with the Proper Stranger
	633531	The Contender (2000 film)
	358243	Class (film)
	1308068	The Mack
	98506	The Russians Are Coming, the Russians Are Coming
	25322634	Death at a Funeral (2010 film)
	1865502	Wrongfully Accused
	29664578	The Raven (2012 film)
	12807520	The Stone Killer
	29664578	The Raven (2012 film)
	854045	De-Lovely
	16780427	The Unholy Three (1930 film)
	26003401	The Curse (1987 film)
	5579814	Bad Company (1995 film)
	22548689	Comanche (1956 film)
	29664578	The Raven (2012 film)
	1202791	Raise Your Voice

Related Topics

[Creating a Lens](#)

Form Lens

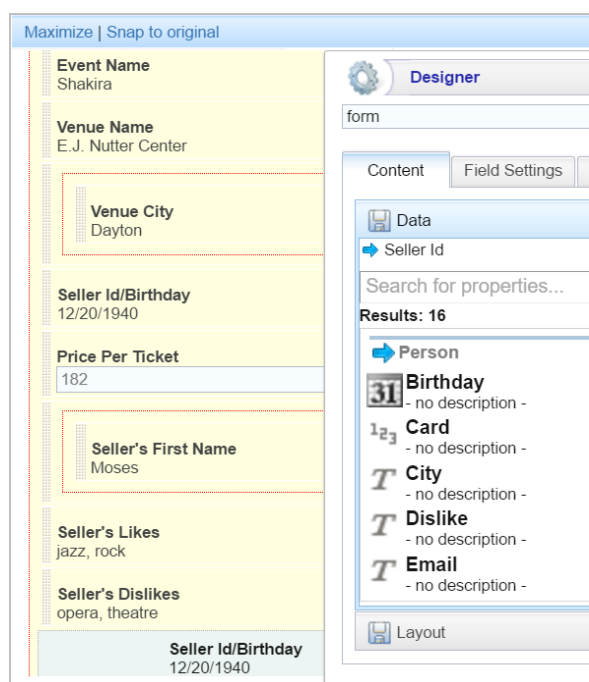
Form lenses enable you to create an editable or read-only form on the dashboard. Creating forms can be useful for displaying many details about each record instead of using a table where the large number of columns makes the data hard to read.

Note

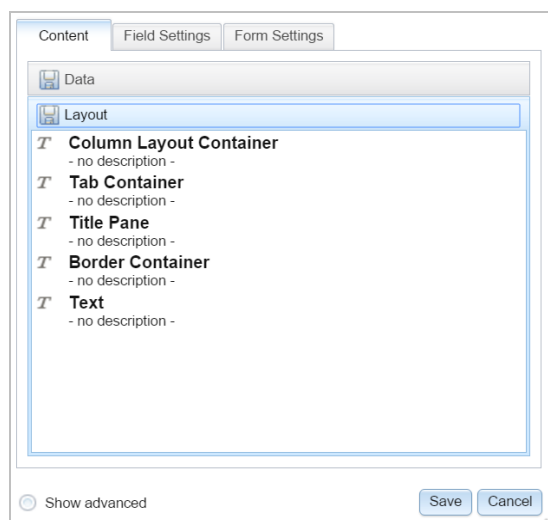
Form lenses are valid in Linked Data Set Dashboards. In Graphmart Dashboards, form lenses are read-only.

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. The Designer displays the available layout containers.



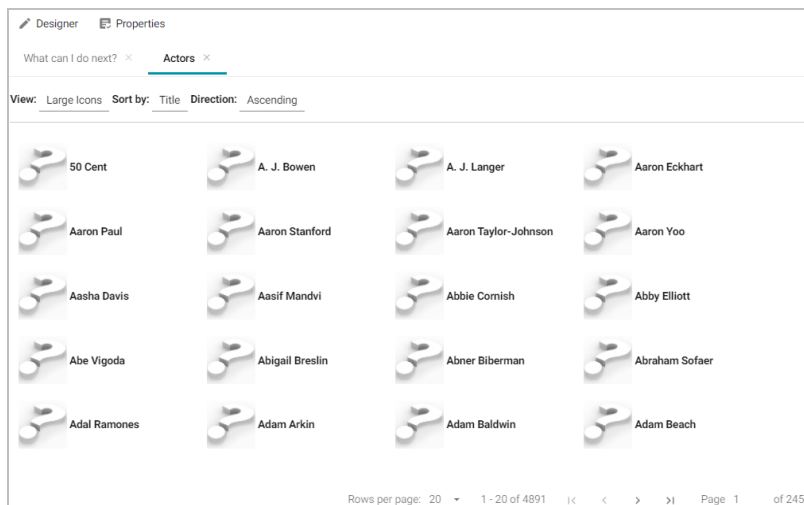
Drag a container onto the form to create the layout template. You can then drag properties into the template.

Related Topics

[Creating a Lens](#)

List Lens

List lenses display the values for the selected property in a list layout with icons, similar a directory explorer view. For example, the lens below lists actor names. The question mark icons are placeholders in this example:



Related Topics

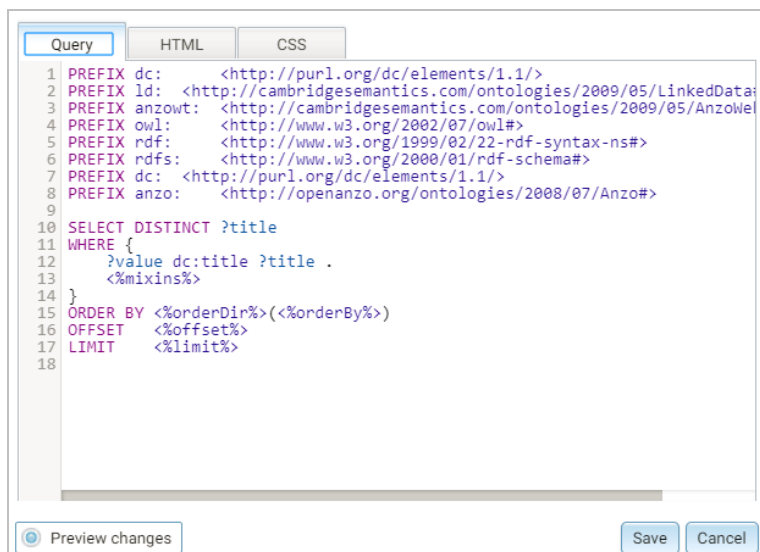
Creating a Lens

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.

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 Anzo functionality:

- **<%mixins%>**: Incorporates a filter function.
- **ORDER BY**: Incorporates a sort function.

See [SPARQL Query Templates and Best Practices](#) for guidance on writing SPARQL queries.

- **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.

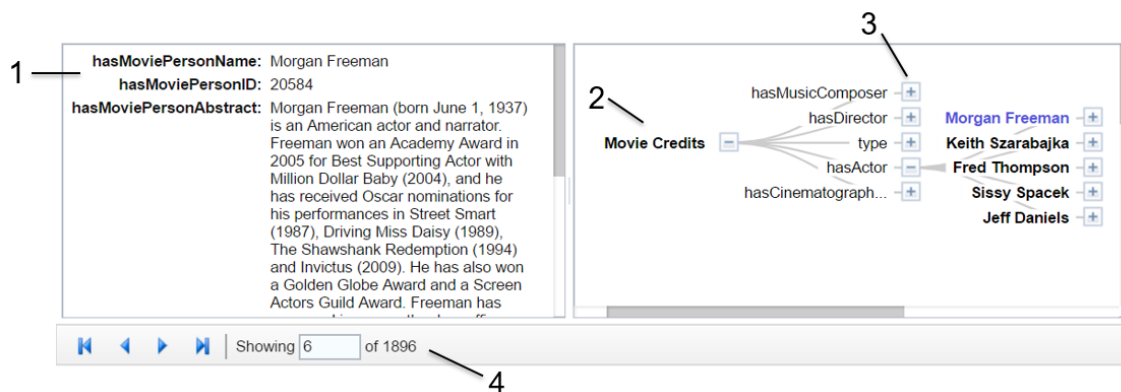
Related Topics

[SPARQL Query Templates and Best Practices](#)

[Creating a Lens](#)

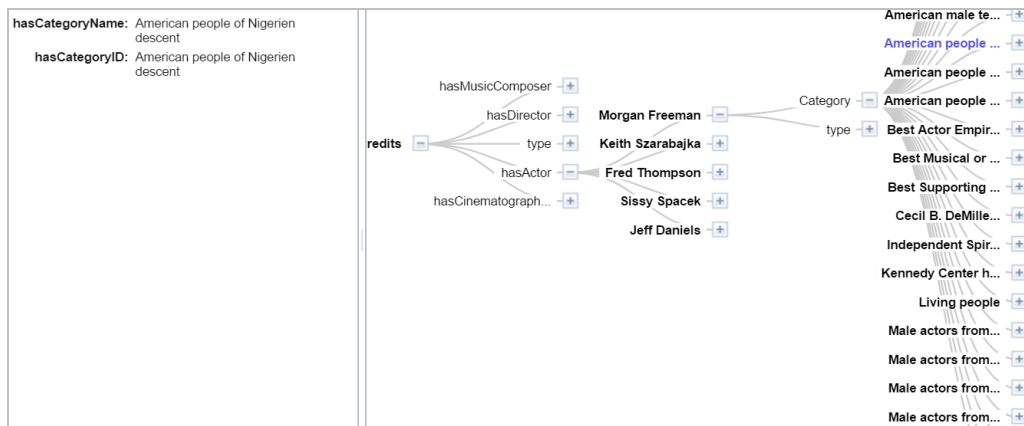
Resource Tree Navigator Lens

The resource tree navigator lens displays data in a tree format with points that you can click to open successive child data points.



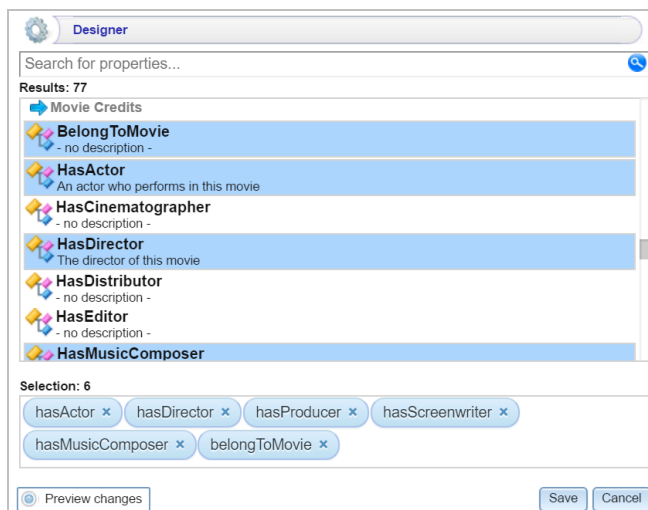
1. **Related data**: Displays the class data related to the selected data property. Data changes when a data end point is selected.
2. **Class property**: Displays the label property of the target class as the initial (start) point of the resource tree. Expand the tree to view child properties by clicking 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.

Click through to an end point and the data view changes to reflect the new class. The data point **American people of Nigerian descent** is selected, and the related class data appears on the left of the screen.



Resource Tree Navigator Lens Configuration

The Designer simply displays all of the properties that are linked to other classes. Select each property that you want the resource tree to include. Then click **Save**.



Related Topics

[Creating a Lens](#)

Table Lens

Table lenses display data in a standard row-and-column grid layout. You define each table column by selecting a property.

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.

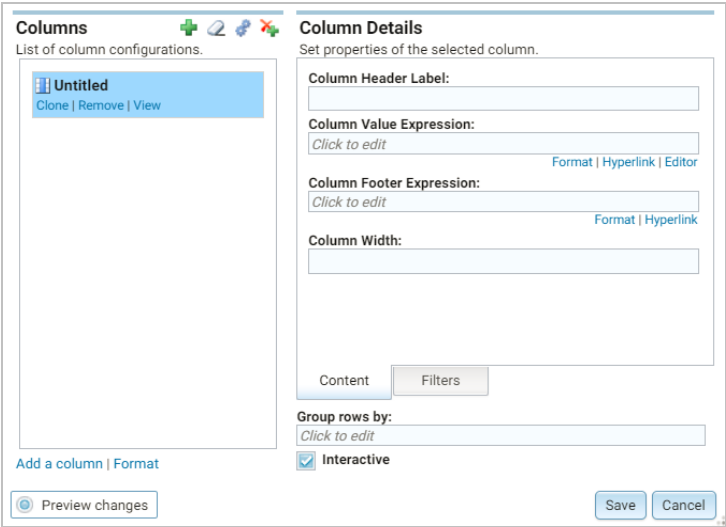






Table Designer Icons

The table below describes the functionality of the icons in the Table Designer.

Icon	Description
	Click the eraser icon to erase all columns. This action cannot be undone.
	Click the plus icon to add a column.
	Click the auto-generate columns icon to add all properties (for the data type selected on the dashboard) as columns.
	Click the Add delete button icon to add delete links to rows in the table. <div><p>Note</p><p>Delete links do not appear on Graphmart dashboards since graphmart data cannot be edited.</p></div>

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 auto-detected by the browser.

Related Topics

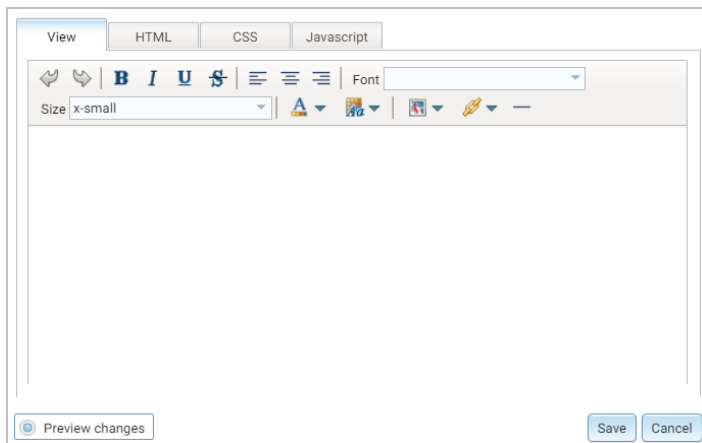
[Creating a Lens](#)

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 reflect in real time.

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.



```

1 <div style="text-align: center;"><b>
2 <i><font face="Times New Roman" size="4">My excellent lens</font></i></b></div>
3 <div xmlns:film="http://cambridgesemantics.com/ontologies/2009/08/Film#">
4 <span anzowbind:innerHTML="film:mpName"></span>
5 </div>

```

For more information about data binding, see the [Data Binding Example](#) section below.

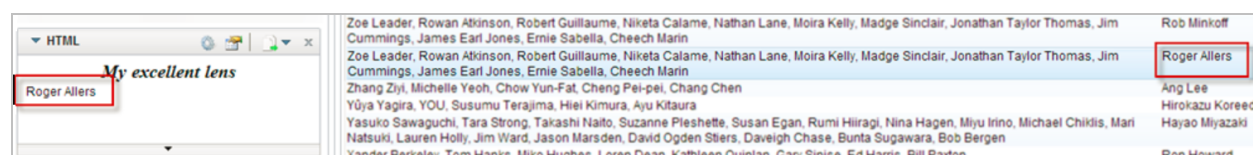
- **CSS:** This tab enables you to create a cascading style sheet to format the HTML and define the look and feel of the web page. Cambridge Semantics recommends that you define all CSS classes as namespaces to avoid global format changes.
- **Javascript:** This tab enables you to write JavaScript code 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.



Related Topics

[Creating a Lens](#)

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](#)
- [Accessing an Endpoint from an Application](#)
- [OData Reference](#)

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, as in 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](#)
- [Accessing Data via the ODBC or JDBC API](#)

JDBC Driver Considerations

This section describes important items to consider when using JDBC clients for accessing Data on Demand endpoints:

- [Join Performance](#)
- [Querying Multi-Value 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 [Configuring a 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-Value Properties

Some relational systems do not directly support Anzo's RDF graph data structures. For example, sometimes the JDBC driver presents multi-value properties as arrays, which can make displaying the results difficult in typical BI tools. There are often ways to restructure the data or the query to get the data you want in a relational system and work with standard BI tools. For guidance and advice using your specific data model, please work with Cambridge Semantics.

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](#).

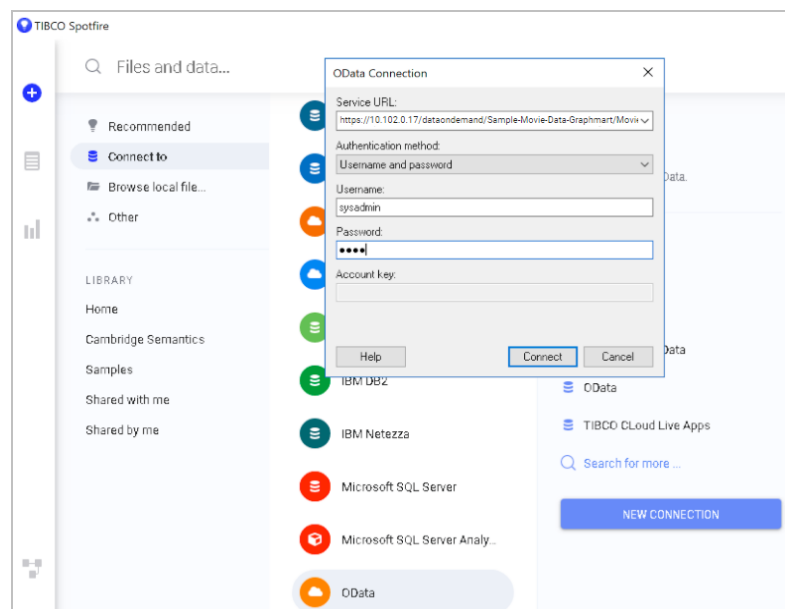
Note

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.

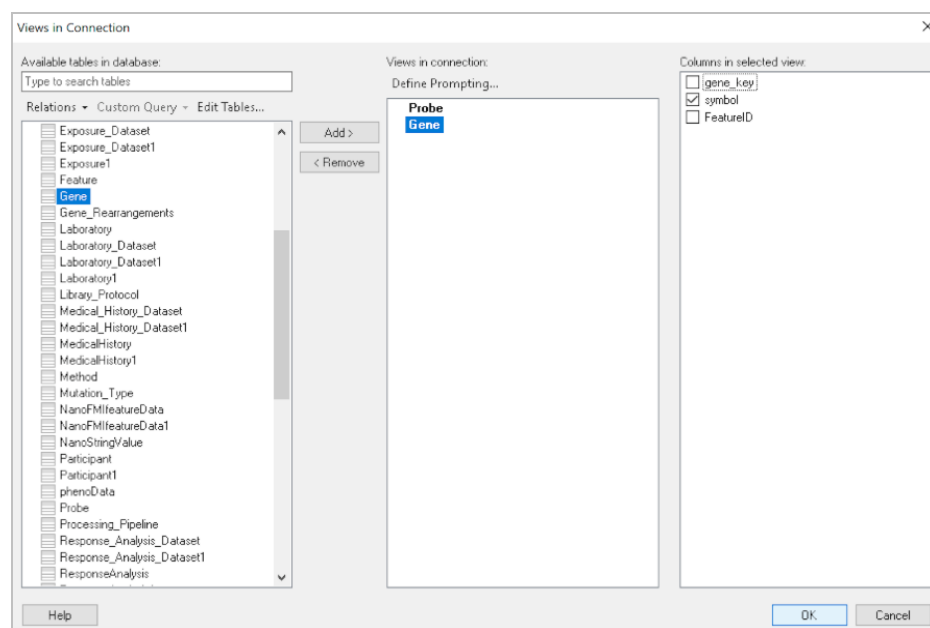
Accessing Data via the OData API

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, Spotfire 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:



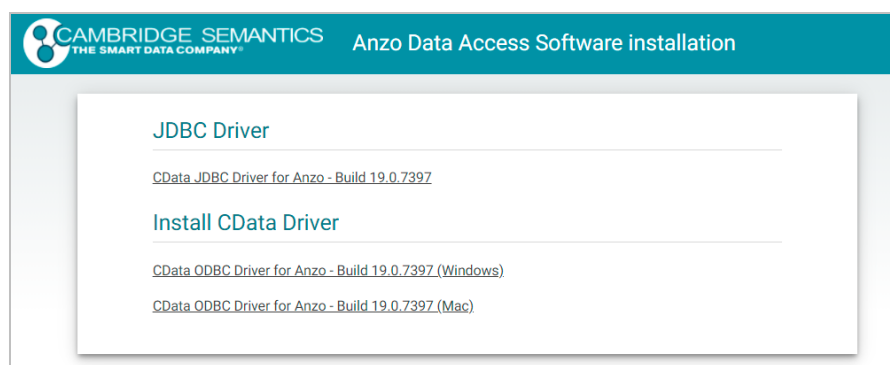
Once the properties are chosen, the data is loaded in Spotfire and can be used to inform existing analytics and data visualizations or create new ones.

Accessing Data via the ODBC or JDBC API

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:



Download the appropriate driver to the client server:

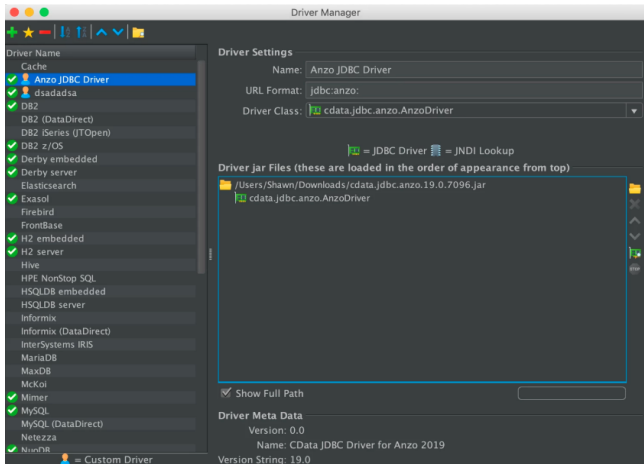
- The **CData JDBC Driver for Anzo** is used to connect to Anzo from most Java applications and database management tools.
- The **CData ODBC Driver for Anzo** for Windows or Mac is for use with applications and database management tools that support open database connectivity, such as Microsoft Excel or Tableau.

Configuring the Driver and Connecting to the Endpoint

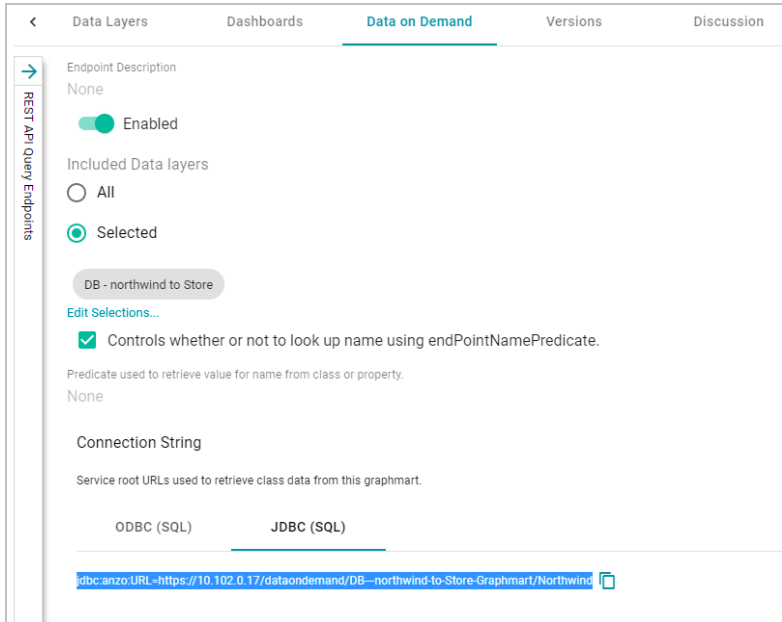
This section provides guidance configuring an ODBC or JDBC driver by showing examples of configuring DbVisualizer and Tableau to access a Data on Demand endpoint using Anzo's JDBC driver and configuring Power BI to access an endpoint using the ODBC driver.

Example JDBC Setup with DbVisualizer

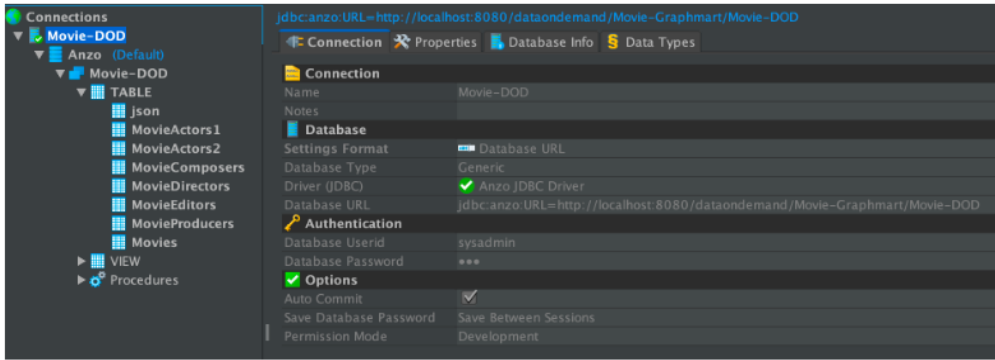
1. In DbVisualizer, go to **Tools** → **Driver Manager**.
2. In the Driver Manager, click the green plus icon to create a new driver.
3. Specify a name for the driver. For example, **Anzo JDBC Driver**.
4. In the URL Format field, specify the format **jdbc:anzo**.
5. In the **Driver File Paths** or **Driver 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.AnzoDriver**. For example:



- To connect to the endpoint in DbVisualizer, go to **Database → Create Database Connection**. Click **No Wizard** when prompted.
- Specify a name for the connection in the **Name** field.
- In the **Driver (JDBC)** field, select the Anzo JDBC driver connection.
- In the **Database URL** field, specify the JDBC URL from the Anzo Data on Demand endpoint configuration. For example: `jdbc:anzo:URL=https://10.100.0.10/dataondemand/DB-northwind-Graphmart/Northwind`

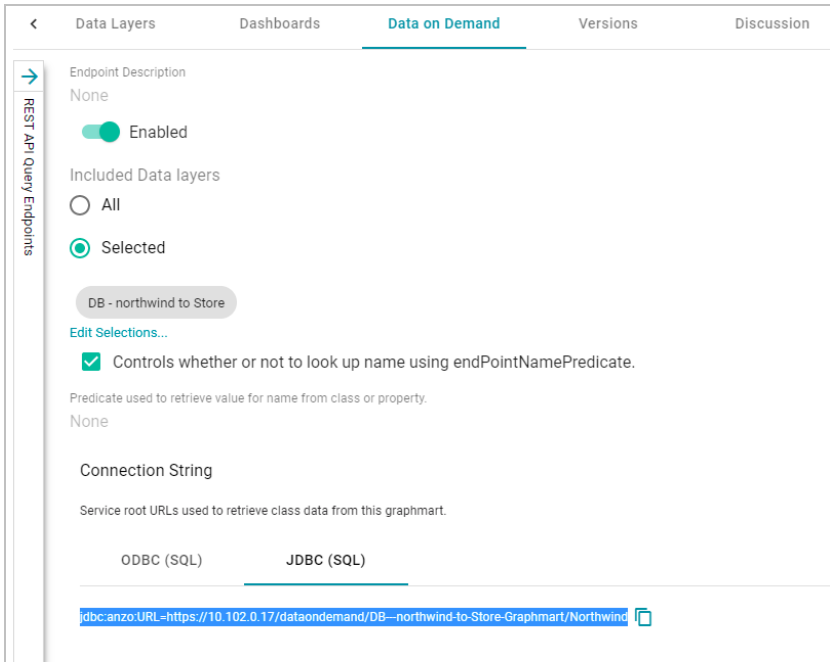


- Under **Authentication**, enter your Anzo user ID and password. You should now be able to connect to the endpoint and view the available schemas. For example:



Example JDBC Setup with Tableau

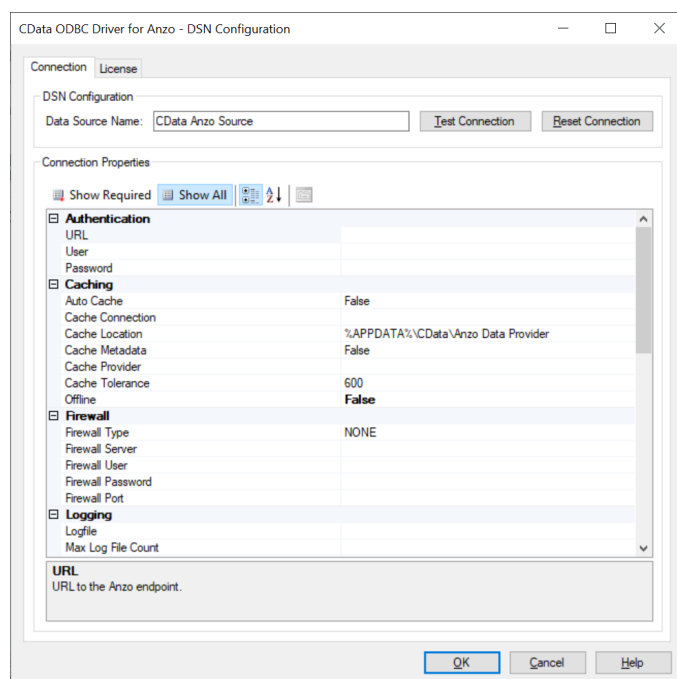
1. After downloading the CData JDBC Driver for Anzo cdata.jdbc.anzo.jar file, place the .jar in the appropriate directory depending on your operating system:
 - **Windows:** C:\Program Files\Tableau\Drivers
 - **MacOS:** ~/Library/Tableau/Drivers
 - **Linux:** /var/opt/tableau/tableau_server/data/tabsvc/vizqlserver/Datasources/
2. Restart Tableau and then go to **Add a Connection** → **To a Server**.
3. Click **Other Databases (JDBC)**.
4. In the URL field, specify the JDBC URL from the Anzo Data on Demand endpoint configuration. For example:
jdbc:anzo:URL=https://10.100.0.10/dataondemand/DB-northwind-Graphmart/Northwind



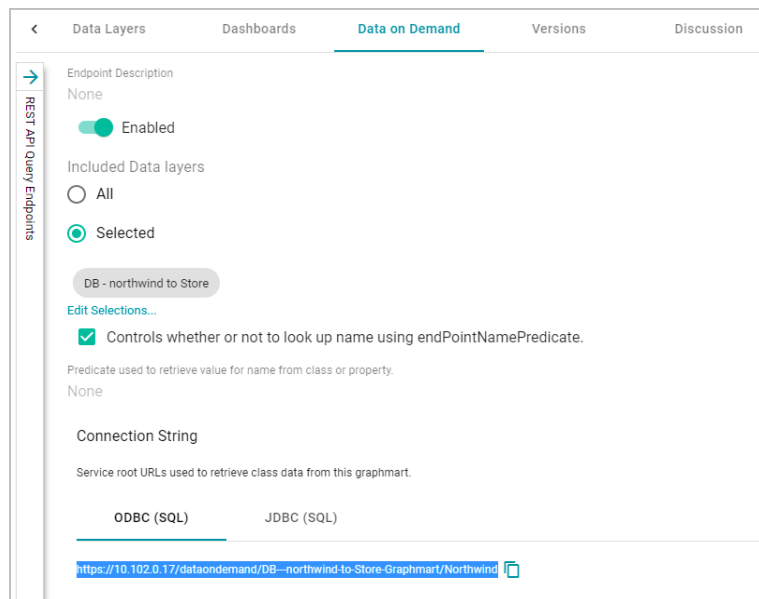
5. Enter your Anzo username and password and click **Sign In**. You should now be able to connect to the endpoint and view the available schemas.

Example ODBC Setup with Microsoft Power BI

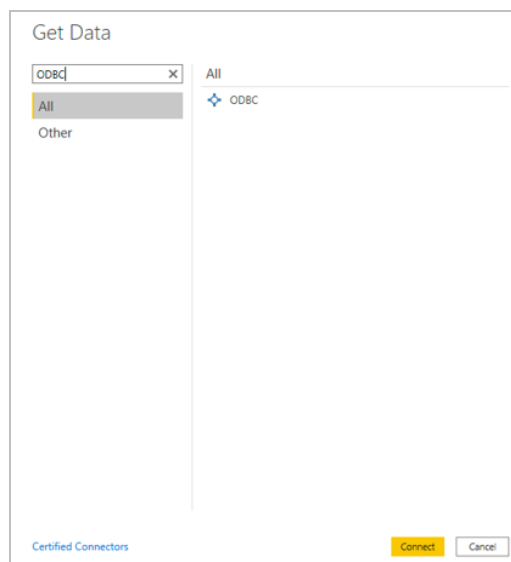
1. After downloading the Windows CData ODBC Driver for Anzo executable file, run the executable to start the installation wizard. The wizard guides you through installing the driver.
2. At the end of the installation, make sure the **Configure ODBC Data Source** checkbox is selected and click **Finish**. 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:



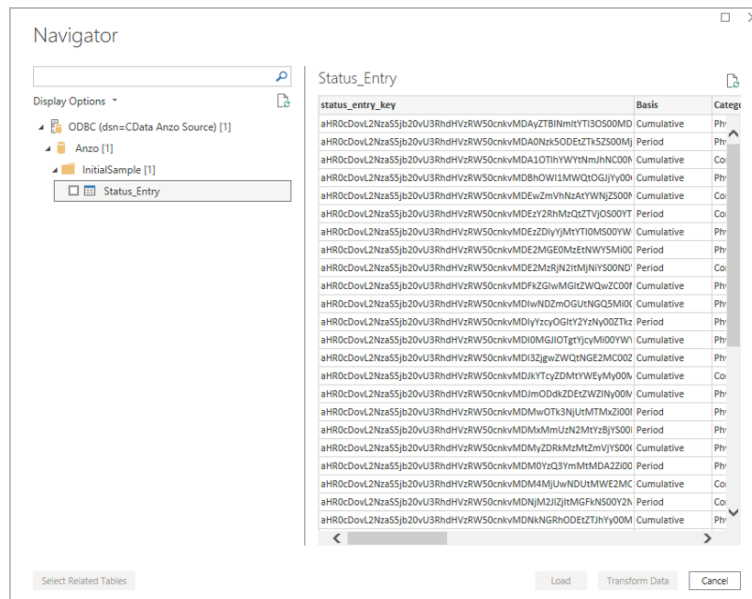
4. Click **OK** to save the configuration changes and close the dialog box.
5. Next, connect to the ODBC data source from Power BI. Open Power BI and click the **Get Data** button in the tool bar. In the Get Data dialog box, search for "ODBC." For example:



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.



7. Click **OK** to create the connection. Power BI opens the Navigator screen. For example:



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.

Note

The Anzo Data on Demand service does not impose limitations on the data that can be retrieved. However, some third-party applications do not support multi-value properties.

- [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.
\$expand	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.
\$filter	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.
\$format	Used to specify the output format for the results. The supported formats are text/CSV, JSON, and XML. For example: \$format=json
\$metadata	Used to return the schema, entity set, and property metadata.
\$orderby	Used to return results in ascending (asc) or descending (desc) order. If asc or desc is not specified, solutions are returned in ascending order.
\$select	Used to specify the subset of properties to include in the result set.
\$skip	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.
\$top	Used to limit the number of solutions that are returned.

Example OData Requests

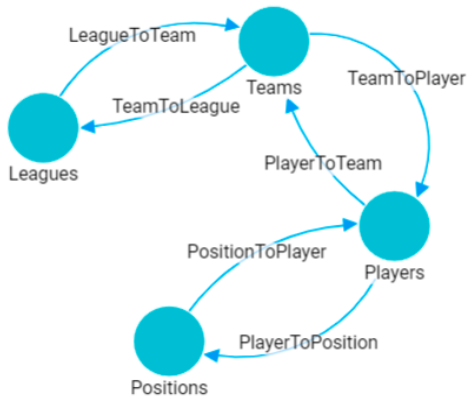
This section demonstrates the use of OData query operators by providing examples of common types of OData requests.

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>
      <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 AI 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": [
        "aHR0cDovL2NzaS5jb20vTGvHb20vV1cy8x"
      ],
      "TeamName": "Black Sox",
      "teamtoplayer_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": [
        "aHR0cDovL2NzaS5jb20vVG9hbmVNA"
      ],
      "PlayerName": "Ted James",
      "DefensiveRating": 92.55
    },
    {
      "players_key": "aHR0cDovL2NzaS5jb20vUGxheWVycy84",
      "playertoposition_key": [
        "aHR0cDovL2NzaS5jb20vUG9zaXRpb25zLzI",
        "aHR0cDovL2NzaS5jb20vUG9zaXRpb25zLzEw"
      ],
      "PlayerId": 8,
      "playertoteam_key": [
        "aHR0cDovL2NzaS5jb20vVG9hbmVMA"
      ],
      "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": [
        "aHR0cDovL2NzaS5jb20vUGVhbXMvMQ"
      ],
      "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 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](#)
- [Searching for Quads in the Query Builder](#)

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 Data Sets
- 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 Query Templates and Best Practices](#).

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 radio button for the target data source:
 - To query data that is in a graphmart, select the **Graphmart** radio button.

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 data layer or layers to target.

- To query data that is in a linked data set, select the **Linked Dataset** radio button.

Click the **Select linked dataset** drop-down list and select the linked data set to query.

- To run queries against the system data source, data metrics volume, Anzo system tables, LDAP server, or AnzoGraph, select the **Datasource** radio button.

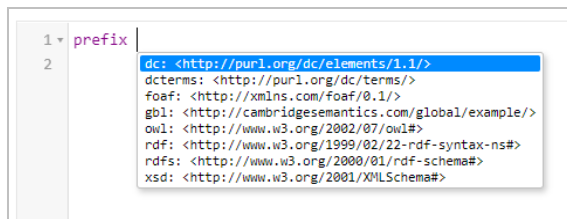
Click the **Datasource** drop-down list and select the target source:

- Select **System Datasource** to search the local Anzo 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.

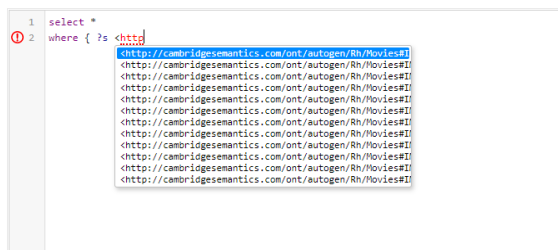
3. In the text box below the target, compose the SPARQL query. For information about the supported SPARQL functions, see [Function and Formula Reference](#).

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:

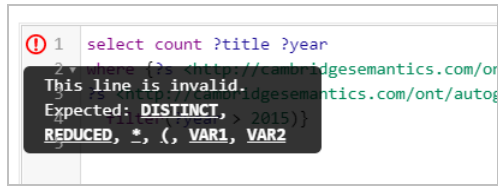


In addition, typing the abbreviation for a global prefix followed by a colon (:) automatically inserts the PREFIX statement into the query without opening the tooltip. For more information about global prefixes, see [Configure Global Prefixes](#).

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:

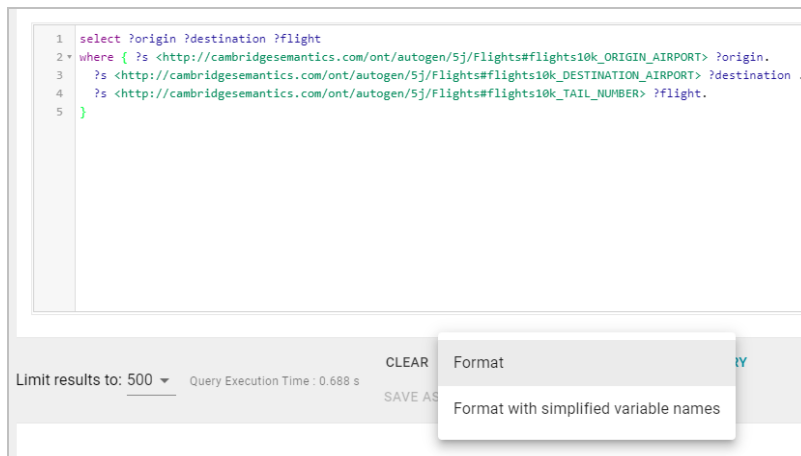


When a red exclamation mark icon (❗) is displayed next to a line number, you can hover the pointer over the icon to view guidance on how to resolve the issue. For example:



4. If you want to format the query for readability, click **Format Query** and select one of 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.

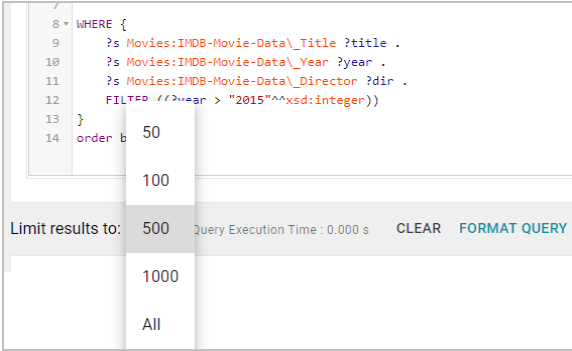
For example, the image below shows a query before it is formatted.



After the query is formatted, prefixes and URI abbreviations are added. For example:



5. If the query is an INSERT or DELETE query, the **Dry Run Query** button becomes active. You can click **Dry Run Query** 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 **Run Query** and committing the updates.
6. 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:



7. To run the query, click **Run Query**. The results appear at the bottom of the screen. For example:

The screenshot shows the results of the query in a table. The table has three columns: title, year, and dir. The results are limited to 100. The table is scrollable.

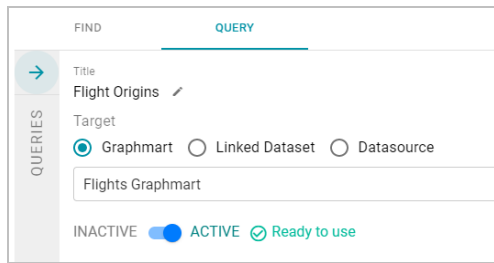
title	year	dir
"Tramps"	"2016"^^<http://www.w3.org/2001/XMLSchema#int>	"Adam Leon"
"Blair Witch"	"2016"^^<http://www.w3.org/2001/XMLSchema#int>	"Adam Wingard"
"Maudie"	"2016"^^<http://www.w3.org/2001/XMLSchema#int>	"Aisling Walsh"
"Hacker"	"2016"^^<http://www.w3.org/2001/XMLSchema#int>	"Akan Satayev"
"Popstar: Never Stop Never Stopping"	"2016"^^<http://www.w3.org/2001/XMLSchema#int>	"Akiva Schaffer"
"Kung Fu Panda 3"	"2016"^^<http://www.w3.org/2001/XMLSchema#int>	"Alessandro Carloni"
"Gods of Egypt"	"2016"^^<http://www.w3.org/2001/XMLSchema#int>	"Alex Proyas"
"American Wrestler: The Wizard"	"2016"^^<http://www.w3.org/2001/XMLSchema#int>	"Alex Ranarivelo"
"The Life of a Fish"	"2016"^^<http://www.w3.org/2001/XMLSchema#int>	"Alexander Aja"

Tip You can click any value in the result list to copy that value to the clipboard.

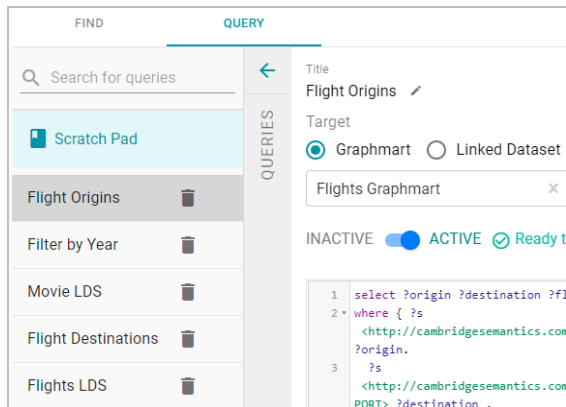
8. To save the query for later use, click **Save Query**. Anzo displays the Save Query dialog box.

The screenshot shows the "Save Query" dialog box. It has two fields: "Title *" and "Description". Below the "Title *" field, there is a hint text: "The title of the bookmark query". Below the "Description" field, there is a hint text: "Description of the bookmark query". At the bottom right, there are two buttons: "CANCEL" and "SAVE".

9. In the Save Query dialog box, specify a name for the query in the **Title** field and an optional description in the **Description** field. Then click **Save**. The query is saved in the gray **Queries** list on the left side of the screen and is collapsed by default.



Click the arrow or anywhere on the gray tab to expand the list. For example:



Select a query to open it in the query editor. You can delete a query by clicking the trashcan icon next to the query name. If you change a query and want to save it as a new query, click **Save as New Query**.

Related Topics

[Searching for Quads in the Query Builder](#)

[Accessing Data with Hi-Res Analytics Dashboards](#)

[Accessing Data on Demand Endpoints](#)

[Accessing Data from the SPARQL Endpoint](#)

[Accessing Data from the HTTP Client Interface](#)

[SPARQL Query Templates and Best Practices](#)

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.

3. Click the **Source** drop-down list and select the data source that you want to search.
 - Select **System Datasource** to search the local Anzo 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.
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:

The screenshot shows the 'Find' tab in the Anzo interface. The 'Source' is set to 'AnzoGraph'. The search criteria are: Subject (blank), Predicate (blank), Object ('BOS'), and Graph (blank). Below the search bar, there are buttons for 'CLEAR', 'ADD STATEMENT', and 'FIND'. The results section shows 'Result(50)' and a 'Quick Filter' section with checkboxes for 'Subject', 'Predicate', 'Object', and 'Named Graph', all of which are checked. The results are displayed in a table with four columns: Subject, Predicate, Object, and Named Graph. The table contains five rows of results, each representing a statement with a unique ID, a predicate, an object, and a named graph.

Subject	Predicate	Object	Named Graph
<http://csi.com/flights10k/e45c0034453b42e2a6bb7b33f253281d>	<http://cambridgesemantics.com/ont/autogen/5j/Flights#flights10k_DESTINATION_AIRPORT>	"BOS"	<http://cambridgesemantics.com/Layer/0a3c1ad4590c47d48a1ff28802ba62e0>
<http://csi.com/flights10k/e5268ce6d97344969e2d08c3b620559a>	<http://cambridgesemantics.com/ont/autogen/5j/Flights#flights10k_DESTINATION_AIRPORT>	"BOS"	<http://cambridgesemantics.com/Layer/0a3c1ad4590c47d48a1ff28802ba62e0>
<http://csi.com/flights10k/e598a20305c845b28a9d8a5d8d1593d7>	<http://cambridgesemantics.com/ont/autogen/5j/Flights#flights10k_ORIGIN_AIRPORT>	"BOS"	<http://cambridgesemantics.com/Layer/0a3c1ad4590c47d48a1ff28802ba62e0>
<http://csi.com/flights10k/e695bb8edf4c430c-aa3f-ef55cc2a85c7>	<http://cambridgesemantics.com/ont/autogen/5j/Flights#flights10k_ORIGIN_AIRPORT>	"BOS"	<http://cambridgesemantics.com/Layer/0a3c1ad4590c47d48a1ff28802ba62e0>
<http://csi.com/flights10k/e726b7527aa049579134062bd53cd7e>	<http://cambridgesemantics.com/ont/autogen/5j/Flights#flights10k_DESTINATION_AIRPORT>	"BOS"	<http://cambridgesemantics.com/Layer/0a3c1ad4590c47d48a1ff28802ba62e0>

Tip You can click a value in the result list to copy that value to the search criteria.

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 Quick Filter checkboxes above the results.

Quick Filter : ☒ 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**.

Result			Quick Filter : <input checked="" type="checkbox"/> Subject <input checked="" type="checkbox"/> Predicate <input checked="" type="checkbox"/> Object <input type="checkbox"/> Named Graph	
Subject ↓	Predicate	Object		
« <http://csi.com/flights10k/fff437e6-db95-4024-a083-da685926706f> »	« <http://cambridgesemantics.com/ont/autogen/5j/Flights#flights10k_DEPARTURE_TIME> »	« "1113"^^<http://www.w3.org/2001/XMLSchema#int> »	<div>  Edit  Delete </div>	
« <http://csi.com/flights10k/fff437e6-db95-4024-a083-da685926706f> »	« <http://cambridgesemantics.com/ont/autogen/5j/Flights#flights10k_SCHEDULED_DEPARTURE> »	« "1120"^^<http://www.w3.org/2001/XMLSchema#int> »		
« <http://csi.com/flights10k/fff437e6-db95-4024-a083-da685926706f> »	« <http://cambridgesemantics.com/ont/autogen/5j/Flights#flights10k_TAIL_NUMBER> »	« "N562UW" »		

Anzo displays the Edit Statement dialog box. For example:

Edit Statement

Subject *

<http://csi.com/flights10k/fff437e6-db95-4024-a083-da685926706f>

Predicate *

<http://cambridgesemantics.com/ont/autogen/5j/Flights#flights10k_DEPARTURE_TIME>

Object *

"1113"^^<http://www.w3.org/2001/XMLSchema#int>

Named Graph URI *

<http://cambridgesemantics.com/Layer/4b83c51443ce45469f59f0a22855c8ce>

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:

Confirm

Are you sure you want to delete these statements?

Subject

<http://csi.com/flights10k/fff437e6-db95-4024-a083-da685926706f>

Predicate

<http://cambridgesemantics.com/ont/autogen/5j/Flights#flights10k_DEPARTURE_TIME>

Object

"1113"^^<http://www.w3.org/2001/XMLSchema#int>

Graph

<http://cambridgesemantics.com/Layer/4b83c51443ce45469f59f0a22855c8ce>

CANCEL

OK

Click **OK** to remove the statement from the system data source.

Adding a Statement

To add a quad to the data source, click **Add Statement** at the top of the result list. For example:

FIND

QUERY

Source : AnzoGraph

Subject

Predicate

<http://cambridgesemantics.com/ont/autogen/5j/Flights#flights10k_DEPARTURE_TIME>

Object

Graph

CLEAR

ADD STATEMENT

FIND

Result

Quick Filter : ☒ Subject ☒ Predicate ☒ Object ☐ Named Graph

Subject ↓	Predicate	Object
« <http://csi.com/flights10k/fee8d108-883d-42a2-b7dc-a2d27d7bda05> »	« <http://cambridgesemantics.com/ont/autogen/5j/Flights#flights10k_DEPARTURE_TIME> »	« "1518"^^<http://www.w3.org/2001/XMLSchema#int> »

Anzo displays the Create Statement dialog box.

Create Statement

Subject *

Predicate *

Object *

Named Graph URI *

CANCEL

SAVE

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. For example:

Create Statement

Subject *

<urn:com.cambridgesemantics.lens.container.list.ReorderableListContainerLens>

Predicate *

<http://purl.org/dc/elements/1.1/description>

Object *

"This lens allows users to view lenses in list."

Named Graph URI *

<urn:com.cambridgesemantics.lens.container.list.ReorderableListContainerLens>

CANCEL

SAVE

Click **Save** to add the new quad to the data source.

Related Topics

[Running SPARQL Queries in the Query Builder](#)

[Accessing Data with Hi-Res Analytics Dashboards](#)

[Accessing Data on Demand Endpoints](#)

[Accessing Data from the SPARQL Endpoint](#)

[Accessing Data from the HTTP Client Interface](#)

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: http for HTTP protocol or https 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 80 and the HTTPS port is 443 . To view the ports that are configured for your Anzo instance, see Server Settings in the Administration menu.
sparql	Required keyword for the SPARQL endpoint.
store_type	The type of RDF store for the data. Typically users specify graphmart 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 lds as the store type.
url-encoded_dataset_uri	<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>How do I find the URI for a Graphmart?</p> <p>How do I find the catalog entry URI for a Dataset?</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%252F001e517db4f0eaea9f279427e4e2a828%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
Content-Type	<p>The Content-Type specifies the type of request that is being sent by the client. Anzo supports the following Content-Type values:</p> <ul style="list-style-type: none"> • application/x-www-form-urlencoded: 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 Content-Type: application/x-www-form-urlencoded is specified. • application/sparql-query: Including this value specifies that the HTTP request body includes a SPARQL read (non-update) query. • application/sparql-update: Including this value specifies that the HTTP request body includes a SPARQL update query.
Accept	<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 format HTTP parameter. For details about the supported formats, see Format Options 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
query	<p>Specifies the full read-only query string to run. If you do not specify a url-encoded_dataset_uri, default-graph-uri or named-graph-uri in the request, the query string should contain the appropriate FROM clauses.</p> <p>To run an update query (INSERT or DELETE), use the update parameter.</p>

Parameter	Description
default-graph-uri	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.
named-graph-uri	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.
format	Specifies the serialization format to use for the results of the query. For details about the supported formats, see Format Options below.
includeMetadataGraphs	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 includeMetadataGraphs=false .
delim	Specifies a custom delimiter character to use in CSV output results. Valid only for SELECT queries where the output format is text/csv . This field accepts any character. When delim is not specified the default value is a , (comma).
dedup	A boolean value that specifies whether to deduplicate CONSTRUCT results on the client side. When dedup is not specified, the default value is dedup=true .
serverDedup	A boolean value that specifies whether to deduplicate CONSTRUCT results on the server side. When serverDedup is not specified, the default value is serverDedup=true .
skipCache	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 skipCache=false . Anzo server's query cache should be forcibly skipped/ignored

Parameter	Description
hasHeader	A boolean value that specifies whether to include headers in CSV results. Valid only for SELECT queries where the output format is text/csv . When hasHeader is not specified, the default value is hasHeader=false .
attachResult	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 attachment . When attachResult is not specified, the default value is attachResult=false . When returning results as an attachment, you can specify a file name in filename the parameter.
filename	If attachResult 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 QueryResult .

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	application/sparql-results+xml application/xml xml xml2 srx	SELECT only	Returns results in SPARQL Query Results XML Format .
	application/rdf+xml rdf owl rdfs	CONSTRUCT only	Returns results in RDF 1.1 XML format.
JSON	application/json json	SELECT and CONSTRUCT	<p>For SELECT queries, results are returned in SPARQL Query Results JSON Format.</p> <p>For CONSTRUCT queries, results are returned in Anzo's native JSON RDF serialization format. See Anzo JSON RDF Serialization for details.</p>
	application/sparql-results+json	SELECT only	Returns results in SPARQL Query Results JSON Format .
CSV	text/csv csv	SELECT only	Returns results in SPARQL Query Results CSV Format .
TriG and Gzipped TriG	application/x-trig trig application/x-trigz trigz gz trig.gz	CONSTRUCT only	CONSTRUCT queries with a GRAPH clause return RDF 1.1 TriG by default if no format is specified.

Format	Accepted Values	Query Type	Description
Turtle and Gzipped Turtle	application/x-turtle ttl application/x-turtlez ttlz ttl.gz	CONSTRUCT only	Returns RDF 1.1 Turtle . CONSTRUCT queries without a GRAPH clause return Turtle by default if no format is specified.
N-Triples	text/plain nt	CONSTRUCT only	Returns results in RDF 1.1 N-Triples format.
Notation3 and Gzipped Notation3	text/rdf+n3 n3 text/rdf+n3z n3z n3z.gz	CONSTRUCT only	Returns results in RDF Notation3 format.
N-Quads	text/x-nquads nq nquad nquads	CONSTRUCT only	Returns results in RDF 1.1 N-Quads format.
TriX	application/trix trix	CONSTRUCT only	Returns results in RDF Triples in XML format.

Update Parameters

Parameter	Description
update	<p>Specifies the full update string to run. If you do not specify a url-encoded_dataset_uri, using-graph-uri or using-named-graph-uri in the request, the update query should contain the appropriate USING clauses.</p> <p>To run a non-update query (SELECT or CONSTRUCT), use the query parameter.</p>

Parameter	Description
using-graph-uri	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.
using-named-graph-uri	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.
includeMetadataGraphs	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 false .

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%2FGraphmart%2F2dc
579b101654ae29eb91b0c7d046ca1
--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%2Fbb2d4b4c138403bab1c671ebd9763...

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 { ?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#DatatypeProperty"
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#FunctionalProperty"
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#DatatypeProperty"
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#FunctionalProperty"
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#DatatypeProperty"
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#FunctionalProperty"

```

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%2Fcambridgesemantics.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
<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://cambridgesemantics.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%2Fcambridgesemantics.com%2FGraphmart%2Fbbf2d4b4c138403bab1c671eb6d9763... Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION
<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    "namedGraphUri": "http://csi.com/test",
3    "subject": {
4      "objectType": "uri",
5      "value": "http://cambridgesemantics.com/ont/autogen/Fu/DB/northwind#Customers"
6    },
7    "predicate": "http://www.w3.org/1999/02/22-rdf-syntax-ns#type",
8    "object": {
9      "objectType": "uri",
10     "value": "http://www.w3.org/2000/01/rdf-schema#Resource"
11   }
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 with Hi-Res Analytics Dashboards](#)

[Accessing Data with the Query Builder](#)

[Accessing Data on Demand Endpoints](#)

[Accessing Data from the HTTP Client Interface](#)

[SPARQL Query Templates and Best Practices](#)

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: http for HTTP protocol or https 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 80 and the HTTPS port is 443 . To view the ports that are configured for your Anzo instance, see Server Settings in the Administration menu.
anzoclient	Required keyword for the client servlet.
client_operation	<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 Client Operations below.</p> <ul style="list-style-type: none"> • call: Invokes the semantic service operation identified by the URI provided in the request (analogous to the <code>anzo call</code> CLI command) • add: Imports the specified statements to Anzo (analogous to the <code>anzo import</code> CLI command) • remove: Removes the specified statements from Anzo (analogous to the <code>anzo update -r</code> CLI command) • get: Gets the specified named graph from Anzo (analogous to the <code>anzo get</code> CLI command) • find: Finds the statements in Anzo that match the specified pattern (analogous to the <code>anzo find</code> CLI command)

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.
- **contentType**, **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 mimeType, Content-type, or 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. Header options are not applicable, and the request body includes the statements to add. **The statements to add must be**

specified in Anzo JSON RDF serialization format. See [Anzo JSON RDF Serialization](#) below for details.

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. Header options are not applicable, and the request body specifies the statements to remove. **The statements to remove must be specified in Anzo JSON RDF serialization format.** See [Anzo JSON RDF Serialization](#) below for details.

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. Header options are not applicable. 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. Graphs are returned in Anzo JSON RDF serialization format. See [Anzo JSON RDF Serialization](#) below for details.

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%23LinkedDataSet' \
--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%2FLinkedData%23
LinkedDataSet'
```

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 with Hi-Res Analytics Dashboards](#)

[Accessing Data with the Query Builder](#)

[Accessing Data on Demand Endpoints](#)

[Accessing Data from the SPARQL Endpoint](#)

[Anzo Admin CLI](#)

[SPARQL Query Templates and Best Practices](#)

SPARQL Query Templates and Best Practices

To provide guidance on developing performant SPARQL queries and avoiding unexpected results, this topic 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 Query Templates](#)
- [SPARQL Best Practices](#)

SPARQL Query Templates

This section provides templates that you can use as a starting point for writing SPARQL queries.

- [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/rootOfUris#>

# 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/rootOfUris#varName2> ?var2 .
}
```

Example Query – Get 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 string values 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 Data Layer 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 when Possible](#) below 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)
}
```

SPARQL Best Practices

To ensure that your SPARQL queries perform well and do not overtax Anzo, Cambridge Semantics recommends that you follow these guidelines when writing and testing your queries:

- [Limit Results when Developing and Testing Queries](#)
- [Replace FILTER with VALUES or Triple Patterns when Possible](#)
- [Beware of Cross-Product Joins](#)
- [Use Subqueries when Querying Large Amounts of Data](#)

Limit Results when Developing and Testing Queries

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 Solution – Get Sample ID and the Binding Density for the top 10 most dense Samples

```
PREFIX bm: <http://identifiers.csi.com/pharmakg/def/biomarker#>

SELECT ?sampleId ?bindingDensity
WHERE {
    ?sample a bm:Sample ;
        bm:sampleId ?sampleId ;
        bm:bindingDensity ?bindingDensity .
}
ORDER BY DESC(?bindingDensity)
LIMIT 10
```

Replace FILTER with VALUES or Triple Patterns when Possible

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 – Inappropriate use of FILTER for value-driven SELECT

```
PREFIX uriRoot: <http://example.com/rootOfUris#>

SELECT ?var1 ?var2
WHERE {
    ?instanceOfClass a uriRoot:ClassName ;
        uriRoot:varName1 ?var1 ;
        uriRoot:varName2 ?var2 ;
        uriRoot:filteredVar ?filteredVar .
    FILTER(?filteredVar = 'COMPAREDDATA1' || ?filteredVar = 'COMPAREDDATA2' || ?filteredVar
= 'COMPAREDDATA3')
    # filteredVar is first retrieved, then run through several comparisons
}
```

Solution – VALUES used to select data of certain values

```
PREFIX uriRoot: <http://example.com/rootOfUris#>

SELECT ?var1 ?var2
WHERE {
    ?instanceOfClass a uriRoot:ClassName ;
        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 – Inappropriate use of FILTER for value-driven SELECT

```
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, then compared
}
```

Solution – Triple literal used to select data of a certain value

```
PREFIX uriRoot: <http://example.com/rootOfUris#>

SELECT ?var1 ?filteredVar
WHERE {
  ?instanceOfClass a uriRoot:ClassName ;
    uriRoot:varName1 ?var1 ;
    uriRoot:filteredVar 'COMPAREDDATA' .
  # data is only retrieved if filteredVar matches desired compared data upon initial
  retrieval
}
```

Beware of 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:ClassName1 ;
    uriRoot:varName1 ?var1 .
  ?instanceOfClass2 a uriRoot:ClassName2 ;
    uriRoot:varName2 ?var2 .
}
```

In the above example, the goal may have been to retrieve IDs from all instances of ClassName1 and all instances of ClassName2, 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:ClassName1 ;
    uriRoot:varName1 ?var1 .
  BIND(?var1 as ?commonVar)
}
UNION

{
  ?instanceOfClass2 a uriRoot:ClassName2 ;
    uriRoot:varName2 ?var2 .
  BIND(?var2 as ?commonVar)
}
# this creates an ephemeral 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:ClassName1 uriRoot:varName1)
    (uriRoot:ClassName2 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 that data were immediately available in the graph.

Example Solution – 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:ClassName1 ;  
        uriRoot:varName1 ?var1 .  
    ?instanceOfClass2 a uriRoot:ClassName2 ;  
        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

[Adding Steps and Views to Data Layers](#)

[Accessing Data with the Query Builder](#)

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](#)
- [Exporting an Artifact](#)
- [Making Values Replaceable on Export](#)
- [Importing Exported Versions of Artifacts](#)

Creating and Restoring Versions of Artifacts

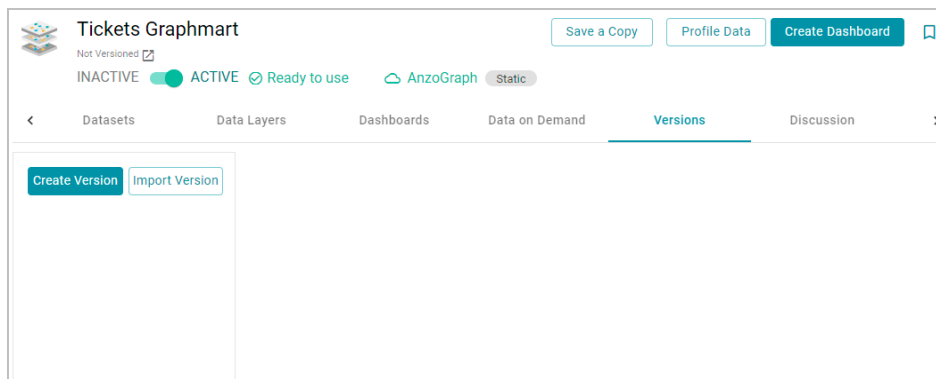
Before making changes to Data Sources, Schemas, Mappings, Pipelines, Models, Graphmarts, etc., users can take a snapshot of the current version of that artifact. When a backup is created, Anzo automatically creates a backup version of each entity that is related to that artifact. For example, backing 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](#)
- [Restoring a Backup Version of an Artifact](#)

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.

2. Click **Create Version**. Anzo displays the Create New Version screen.

Create New Version

Name for New Version *

Comment for New Version

CANCEL

SAVE

3. 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.
4. 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:

<

Datasets

Data Layers

Dashboards

Data on Demand

Versions

Discussion

>

Create Version

Import Version

Tickets Backup

DEV 1

Derived from:

Jun 4, 2021 | System Administrator

Tickets Backup

Description

None

Derived from

Created Jun 4, 2021

Creator System Administrator

Version IDs

Current Env: DEV 1

Related Entities (21)

Type	Title	Version ID	Actions
CSV Data Sour...	Tickets	DEV 1	
Dataset Pipeline	Load Tickets to Store	DEV 1	
File Backed Lin...	Tickets to Store	DEV 1	
File Based Dat...	Tickets to Store	DEV 1	
File Graph Dat...	Store	DEV 1	
Formula Define...	Venues	DEV 1	
Graphmart	Tickets Graphmart	DEV 1	












Export

Restore

5. If necessary, select the new version in the list to view details on the right side of the screen. 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"

Ver: Backupv2

DEV 2

Derived from: BackupV1

Jan 29, 2020 | System Administrator

```

1 <http://cambridge.semantics.com/Graphmart/bf8ad6f71cde47b4bfe8b48a209caf7d> {
2   <http://cambridge.semantics.com/Graphmart/bf8ad6f71cde47b4bfe8b48a209caf7d> a <http://
3     <http://cambridge.semantics.com/ontologies/Graphmarts#graphQueryEngineUri> <http
4       <http://cambridge.semantics.com/ontologies/Graphmarts#ignoreErrors> "true"^^<ht
5         <http://cambridge.semantics.com/ontologies/Graphmarts#layer> <http://cambridge
6           <http://cambridge.semantics.com/ontologies/Graphmarts#manualRefresh> "true"^^<ht
7             <http://purl.org/dc/elements/1.1/title> "Flights Graphmart" ;
8               <http://purl.org/dc/terms/created> "2020-01-29T18:14:00.000Z"^^<http://www.w3.o
9                 <http://purl.org/dc/terms/creator> <http://openanzo.org/system/internal/sysadm
10                   <http://purl.org/dc/terms/modified> "2020-01-30T03:34:34.000Z"^^<http://www.w3.o
11                     <http://purl.org/dc/terms/modified> <http://openanzo.org/system/internal/sysadm
12                       <http://cambridge.semantics.com/OrderedItem/520d3244d45143e5831ef10b059ba34e> a <http
13                         <http://openanzo.org/ontologies/2008/07/anzoindex> "1"^^<http://www.w3.org/200
14                           <http://openanzo.org/ontologies/2008/07/anzoorderid> <http://cambridge.sem
15                             <http://purl.org/dc/terms/created> "2020-01-30T03:34:34.000Z"^^<http://www.w3.o
16                               <http://purl.org/dc/terms/creator> <http://openanzo.org/system/internal/sysadm
17                                 <http://purl.org/dc/terms/modified> "2020-01-30T03:34:34.000Z"^^<http://www.w3.o
18                                   <http://purl.org/dc/terms/modified> <http://openanzo.org/system/internal/sysadm
19                                     <http://purl.org/dc/terms/modified> <http://openanzo.org/system/internal/sysadm
20                                       <http://purl.org/dc/terms/modified> <http://openanzo.org/system/internal/sysadm
21                                         <http://cambridge.semantics.com/OrderedItem/647719c11678494f9032b6288203cb51> a <http
22                                           <http://cambridge.semantics.com/OrderedItem/647719c11678494f9032b6288203cb51> a <http
23                                             <http://cambridge.semantics.com/OrderedItem/647719c11678494f9032b6288203cb51> a <http
24                                             <http://cambridge.semantics.com/OrderedItem/647719c11678494f9032b6288203cb51> a <http

```

Ver: BackupV1

DEV 1

Derived from:

Jan 29, 2020 | System Administrator

```

1 <http://cambridge.semantics.com/Graphmart/bf8ad6f71cde47b4bfe8b48a209caf7d> {
2   <http://cambridge.semantics.com/Graphmart/bf8ad6f71cde47b4bfe8b48a209caf7d> a <http://
3     <http://cambridge.semantics.com/ontologies/Graphmarts#graphQueryEngineUri> <http
4       <http://cambridge.semantics.com/ontologies/Graphmarts#ignoreErrors> "true"^^<ht
5         <http://cambridge.semantics.com/ontologies/Graphmarts#layer> <http://cambridge
6           <http://cambridge.semantics.com/ontologies/Graphmarts#manualRefresh> "true"^^<ht
7             <http://purl.org/dc/elements/1.1/title> "Flights Graphmart" ;
8               <http://purl.org/dc/terms/created> "2020-01-29T18:14:00.000Z"^^<http://www.w3.o
9                 <http://purl.org/dc/terms/creator> <http://openanzo.org/system/internal/sysadm
10                   <http://purl.org/dc/terms/modified> "2020-01-29T18:14:00.000Z"^^<http://www.w3.o
11                     <http://purl.org/dc/terms/modified> <http://openanzo.org/system/internal/sysadm
12                       <http://cambridge.semantics.com/OrderedItem/520d3244d45143e5831ef10b059ba34e> a <http
13                         <http://openanzo.org/ontologies/2008/07/anzoindex> "1"^^<http://www.w3.org/200
14                           <http://openanzo.org/ontologies/2008/07/anzoorderid> <http://cambridge.sem
15                             <http://purl.org/dc/terms/created> "2020-01-30T03:34:34.000Z"^^<http://www.w3.o
16                               <http://purl.org/dc/terms/creator> <http://openanzo.org/system/internal/sysadm
17                                 <http://purl.org/dc/terms/modified> "2020-01-30T03:34:34.000Z"^^<http://www.w3.o
18                                   <http://purl.org/dc/terms/modified> <http://openanzo.org/system/internal/sysadm
19                                     <http://purl.org/dc/terms/modified> <http://openanzo.org/system/internal/sysadm
20                                       <http://purl.org/dc/terms/modified> <http://openanzo.org/system/internal/sysadm
21                                         <http://cambridge.semantics.com/OrderedItem/647719c11678494f9032b6288203cb51> a <http
22                                           <http://cambridge.semantics.com/OrderedItem/647719c11678494f9032b6288203cb51> a <http
23                                             <http://cambridge.semantics.com/OrderedItem/647719c11678494f9032b6288203cb51> a <http
24                                             <http://cambridge.semantics.com/OrderedItem/647719c11678494f9032b6288203cb51> a <http

```

OK

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.

Related Topics

[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:

The screenshot shows the 'Versions' tab for an artifact named 'Tickets Backup'. The interface includes tabs for Datasets, Data Layers, Dashboards, Data on Demand, Versions (active), and Discussion. On the left, there are buttons for 'Create Version' and 'Import Version', and a summary card for 'Tickets Backup' showing it was derived from a previous state on Jun 4, 2021. The main area displays the artifact's description (None), creation details (Created: Jun 4, 2021, Creator: System Administrator), and a list of related entities (21). A table lists these entities with columns for Type, Title, Version ID, and Actions. The 'Current Env' is set to 'DEV 1'. At the top right, there are 'Export' and 'Restore' buttons.

Type	Title	Version ID	Actions
CSV Data Sour...	Tickets	DEV 1	
Dataset Pipeline	Load Tickets to Store	DEV 1	
File Backed Lin...	Tickets to Store	DEV 1	
File Based Dat...	Tickets to Store	DEV 1	
File Graph Dat...	Store	DEV 1	
Formula Define...	Venues	DEV 1	
Graphmart	Tickets Graphmart	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.

Related Topics

[Creating a Backup Version of an Artifact](#)

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](#).

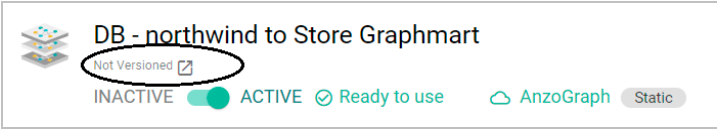
Note Pipeline exports do not contain Dataset Editions. Dataset exports do include Editions.

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.

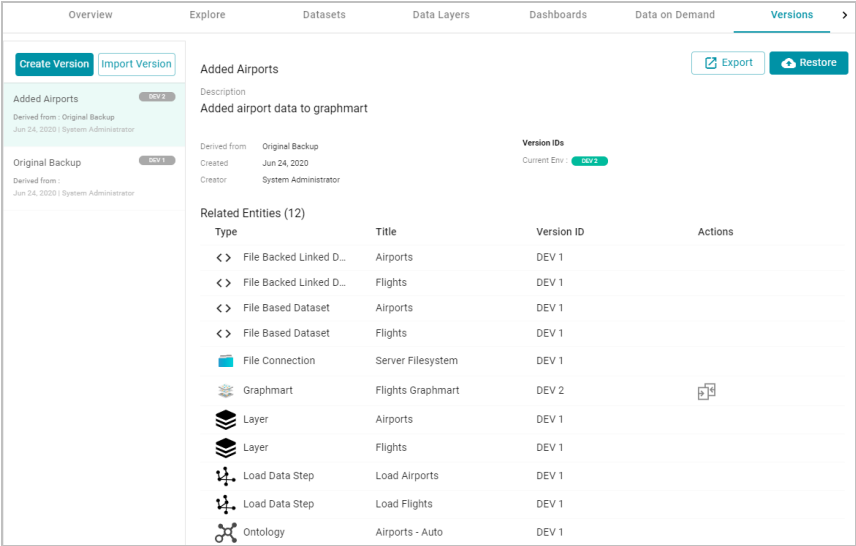
2. Follow one of the options below, depending on whether you want to export the current version of the artifact or a backup version:
 - If you want to export the current, working version of the artifact, click the Export icon () under the artifact name.



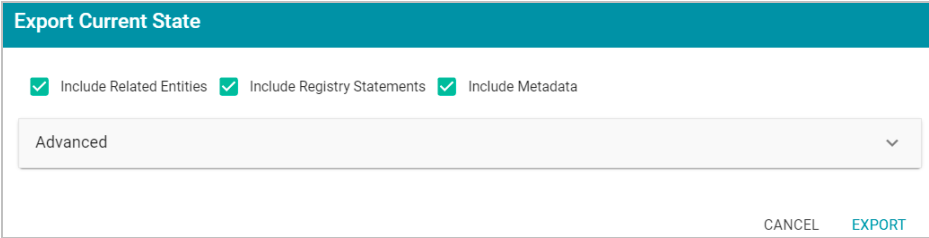
Tip

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:



The Export dialog box is displayed. For example:



3. 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.
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 connection, or the file path for an Anzo Data Store, expand the **Advanced** option to view the Included Entities list. For example:

Advanced			
Included Entities		Sharing	
<input checked="" type="checkbox"/>	Type	Title	
<input checked="" type="checkbox"/>	CSV Data Source	Flights	^
<input checked="" type="checkbox"/>	Dataset Project	Load Flights	
<input checked="" type="checkbox"/>	File Connection	Server Filesystem	^
<input checked="" type="checkbox"/>	File Connection	sysadmin User Folder	^
<input checked="" type="checkbox"/>	File Graph Data Source	Store	^
<input checked="" type="checkbox"/>	Mapping Definition	Flights - flights10k	
<input checked="" type="checkbox"/>	Ontology	Flights - Auto	
<input checked="" type="checkbox"/>	Schema	Flights	
<input checked="" type="checkbox"/>	Schema	Flights-instance	

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		/flights10k.csv	
<input checked="" type="checkbox"/>	 Dataset Project	Load Flights	
<input checked="" type="checkbox"/>	 File Connection	Server Filesystem	
Variables		Replace	
Base Folder		/	
<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:

Compare Versions of Graphmart "Flights Graphmart"

Ver: Added Airports

DEV 2

Derived from: Original Backup

Jun 24, 2020 | System Administrator

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Ver: Original Backup

DEV 1

Derived from:

Jun 24, 2020 | System Administrator

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

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

[Making Values Replaceable on Export](#)

[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 graph data sources. 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.

Note

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

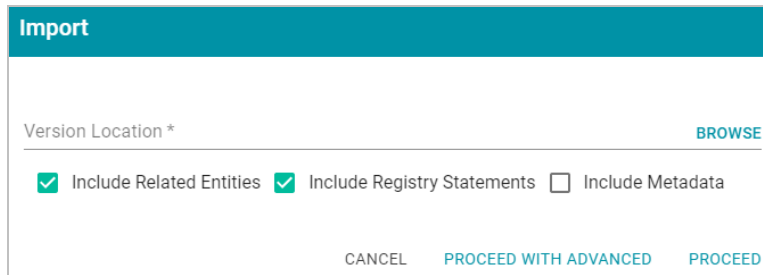
To import a Model that was created outside of Anzo or was downloaded from Anzo (as described in [Downloading a Model](#)), see [Uploading a Model to Anzo](#) for instructions on uploading the Model.

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:

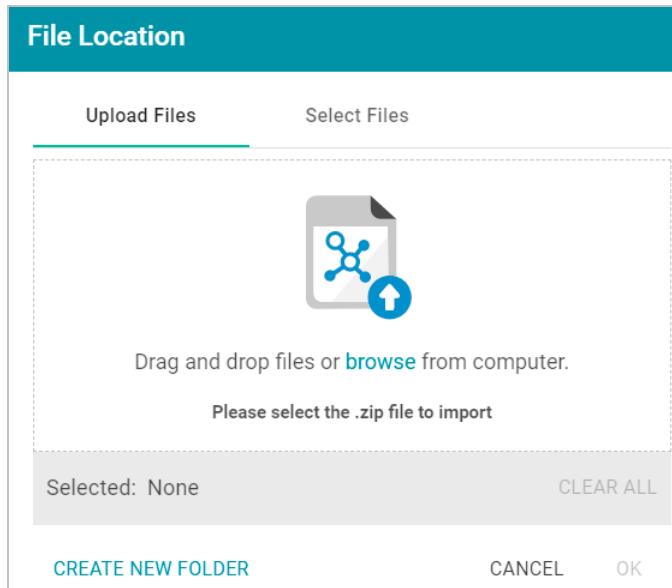
<div>  <input type="text" value="Search"/> <div>Sort By: Title</div> <div>View:  </div> <div>Add Graphmart</div> </div>								
<input type="checkbox"/>	Title	Creator	Status	Statements	Last Accessed	Last Updated	Actions	
	DB - northwind to Store Graphmart	System Admin...	 Ready to us	140,411	7 hours ago	7 hours ago		
	Sample Movie Data to Store Graphm	System Admin...	 Ready to us	13,096	8 hours ago	8 hours ago		
	Tickets Graphmart	System Admin...	 Ready to us	4,433,372	4 hours ago	4 hours ago		

- Click the **Add ...** button on the top of the screen and select **Import ...**. Anzo opens the Import dialog box.



The Import dialog box has a teal header with the title "Import". Below the header, there is a "Version Location *" field with a "BROWSE" link to its right. Underneath, there are three checkboxes: "Include Related Entities" (checked), "Include Registry Statements" (checked), and "Include Metadata" (unchecked). At the bottom, there are three buttons: "CANCEL", "PROCEED WITH ADVANCED", and "PROCEED".

- On the Import screen, click the **Version Location** field to open the File Location dialog box.



The File Location dialog box has a teal header with the title "File Location". Below the header, there are two tabs: "Upload Files" (active) and "Select Files". The "Upload Files" tab contains a large dashed box with a file icon and an upload arrow. Inside the box, it says "Drag and drop files or [browse](#) from computer." and "Please select the .zip file to import". Below the dashed box, there is a grey bar with "Selected: None" and a "CLEAR ALL" link. At the bottom, there are three buttons: "CREATE NEW FOLDER", "CANCEL", and "OK".

If the exported .zip file is on your computer, drag and drop the file onto the Upload Files tab or click **browse** to navigate to the file and select it. If the .zip file is on a file store, click the **Select Files** tab 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.

- **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.
6. 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	
<input checked="" type="checkbox"/>	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.

Related Topics

[Exporting an Artifact](#)

[Making Values Replaceable on Export](#)

Graph Data Storage Reference

This topic describes the way onboarded graph data is shared between and stored in the Anzo and AnzoGraph graph stores.

The onboarding process generates different types of graph data artifacts. Storage of the artifacts differs based on the type of data that is being stored and the purpose of the data. The list below describes the artifacts and storage methods:

- The metadata, such as Data Models, Data Source configuration details, Dataset catalog entries, Registries, Mappings, and access control definitions, are stored in Anzo's embedded graph store. The Anzo graph store is a transaction-oriented store that is built for processing many updates to small amounts of data. Data is persisted to disk in a journal, also known as a volume. The system volume (or system data source) is the default, required volume where Anzo stores Models/Ontologies as well as system configuration, Dataset, registry, and access control metadata. Users can create secondary local volumes that are used for more compartmentalized data and can be created and deleted without affecting the core system.
- The instance data and copies of the data models are written to a File-Based Linked Data Set (FLDS) on the shared file store. Each FLDS is represented as a Dataset in Anzo's Dataset catalog. The Dataset catalog entry includes a pointer to the Anzo Data Store location for the RDF files generated by an ETL pipeline. The Dataset and the files on disk comprise the FLDS.
- When a Dataset from the catalog is added to a Graphmart and the Graphmart is activated, Anzo loads the data from the FLDS into the AnzoGraph graph store. AnzoGraph is an in-memory graph OLAP store that is built for processing complex analytics on large amounts of data. Once the instance data is in memory, the rest of the Graphmart's Data Layer steps are executed by AnzoGraph (known as the ELT process). Each Data Layer becomes a graph in AnzoGraph, and each layer graph includes the instance data created by that layer as well as the related Data Models.
- Anzo system ontologies and metadata remain in Anzo's graph store, the system data source, and are not loaded to AnzoGraph unless the system data is added to a Graphmart and the Graphmart is activated.

Example

An Anzo instance has two active Graphmarts. Each Graphmart has two Data Layers, one for loading Datasets into memory and another for creating views and running ELT queries. When the following query is run against AnzoGraph to return a list of all distinct graphs, the results show that there are five graphs:

```
SELECT DISTINCT ?graph
WHERE {
  GRAPH ?graph {
    ?s ?p ?o
```

```
}
}
```

```
graph
```

```
-----
http://cambridgesemantics.com/Layer/546fb89ac6d245f8bea2777a52077bc9
http://cambridgesemantics.com/Layer/1162fb0d0b724a18b4133c10d69f16b7
http://cambridgesemantics.com/Layer/12c7eedddff9449ab4b133373b56e65c
http://cambridgesemantics.com/Layer/b69bb3295ba3434e846b1ed372039416
http://cambridgesemantics.com/GqeDatasource/guid_10492203b5aa4a54f217ababb3dc6dee
5 rows
```

The first four graphs are the Data Layers for the two Graphmarts. The graph URIs match the Data Layer URIs in Anzo. [How do I find the graph URI for a Data Layer in a Graphmart?](#)

The last graph in the results above is the AnzoGraph data source graph. This graph contains one triple that records a timestamp for the last time the data source was updated. If Anzo loses the connection to AnzoGraph, it checks this timestamp when it reconnects. The last updated time is used to determine whether the Anzo and AnzoGraph graph stores are in sync or if the Graphmarts need to be reloaded to AnzoGraph.

Note

AnzoGraph does not have a "Graphmart" construct, and Graphmart URIs do not exist in the database. Though a Graphmart acts as a container for Data Layers and its metadata can be queried in Anzo's embedded graph store, it does not include instance data that is needed by AnzoGraph.

Typically organizations manage all data with Anzo, i.e., data is onboarded to Anzo through pipelines or it is dynamically blended into Data Layers with the Graph Data Interface. Anzo then loads the data to AnzoGraph for analytics. When data is loaded to AnzoGraph through Anzo, Anzo manages the reloading of Graphmarts if AnzoGraph is restarted. Though users can load data and create named graphs directly in AnzoGraph, AnzoGraph is not configured by default to persist the data in memory to disk. Graphs that do not originate in Anzo must be reloaded manually any time AnzoGraph is restarted. If you want to work with named graphs directly in AnzoGraph, consider configuring AnzoGraph to save data to disk. For more information, see [Using AnzoGraph Persistence \(Preview\)](#).


Related Topics

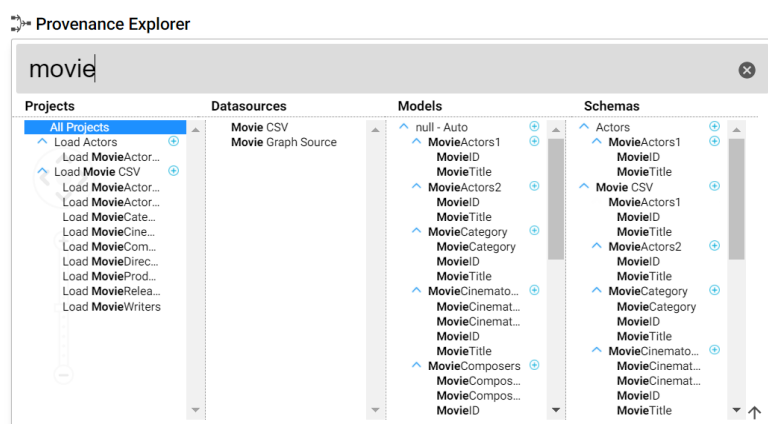
[Onboarding Structured Data](#)

[Onboarding Unstructured Data](#)

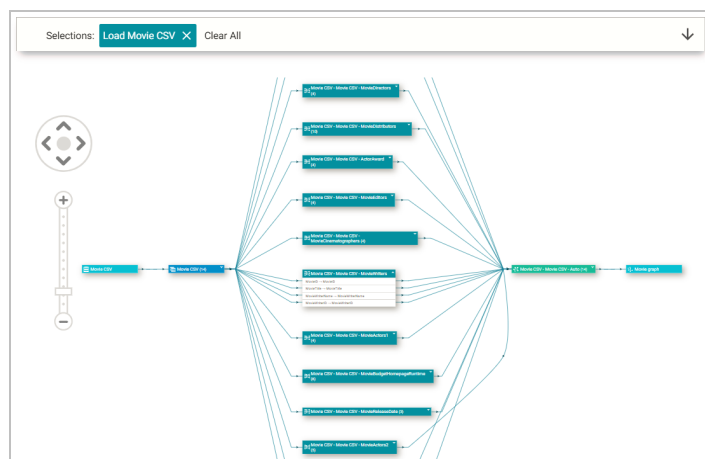
Exploring Data Provenance

The Anzo provenance explorer enables you to trace the lineage of your structured data. You can search for data entities and view associated projects, data sources, models, and schemas. This topic provides information about using the provenance explorer.

1. To open the provenance explorer, click **Provenance** in the Anzo application.
2. Click in the gray **Search** box and type a value to search for. Anzo populates the table in the search drop-down box with any projects, data sources, models, and schemas that include the search value. For example, searching for "movie" displays the pipelines, sources, classes, properties, and schemas that include "movie" and its relationships. Click the plus icon () to view any related elements.



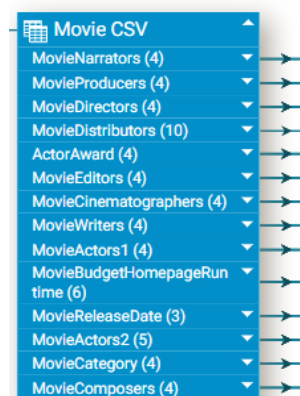
3. To view the provenance for any of the pipeline, data source, model, or schema objects, click the object to highlight it, and then click outside of the search drop-down. Anzo displays the provenance graph for the item that you selected. For example, the image below shows the provenance for the sample Movie data set:



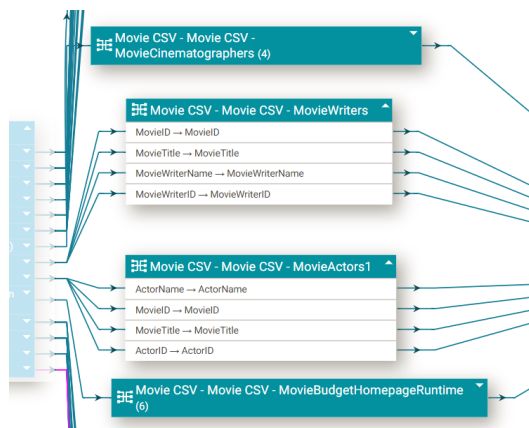
Viewing the provenance graph from left to right, the first object shows the data source, **Movie CSV** in the example:



The second object shows the data that came from the data source, 14 Movie CSV files as shown in the example. To expand the rectangle to view the files and additional details such as the schema information or list of properties from each file, click the triangle icon on the top right of the rectangle:



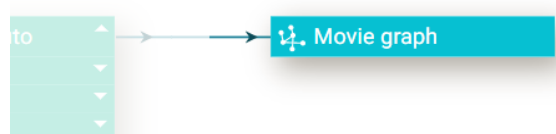
The third group of objects in the graph show the mappings that Anzo generated from the source schemas:



The fourth object shows the model or ontology that Anzo generated:



And the final object shows the graph data source where Anzo generated the load files after ingestion.



- To view the provenance graph for a different object, click in the search box and then choose an item from the drop-down. Click out of the search box to view the graph.

The provenance explorer includes navigation tools that you can use to zoom in or out of the graph or move the graph on the screen.



To view different regions of the graph, click the > characters to move the graph vertically or horizontally. You can also click and drag the graph on the screen. Click the plus and minus icons to zoom in and out. You can click and drag the individual tables to rearrange them in the graph or collapse and expand tables or columns using the triangle icons to the right of the table or column name.

Administration Guide

The Administration Guide provides guidance for Anzo administrators. The topics in this section provide information about managing the initial set up and administration of Anzo components.


- [Accessing the Administration Application](#)
- [Anzo Server Administration](#)
- [Connection Administration](#)
- [User Management](#)
- [Monitoring and Diagnostics](#)
- [AnzoGraph Server Administration](#)
- [Anzo Admin CLI](#)

Accessing the Administration Application

By default, go to the following URL to open the Administration application:

```
https://<hostname>/sdl/index.html#/admin
```

Where <hostname> is the Anzo server DNS name or IP address. You can change the URL for the Administration application by configuring the **Admin Home Page** value in server settings. For more information, see [Configure the Default Root Pages](#).

To access the Administration application from the Anzo application, click the administration icon () on the right side of the top menu bar. Clicking the icon opens the Administration menu, and selecting a menu item opens the application.

Related Topics

[Anzo Server Administration](#)

[Connection Administration](#)

[User Management](#)

[Monitoring and Diagnostics](#)

[AnzoGraph Server Administration](#)

[Anzo Admin CLI](#)

Anzo Server Administration

The topics in this section provide information about managing the Anzo server configuration.

- [Starting and Stopping Anzo](#)
- [Changing Anzo Server Settings](#)
- [Managing Certificates](#)
- [Updating the Server License](#)
- [Managing Volumes](#)
- [Uploading a Plugin](#)
- [Advanced Configuration of Semantic Services](#)

Starting and Stopping Anzo

If Anzo is run via a systemd service, as described in [Configure and Start the Anzo Service](#), use `systemctl` to start and stop Anzo. To start Anzo, run the following command:

```
sudo systemctl start <service_name>
```

For example: `sudo systemctl start anzo-server`

To stop Anzo, run the following command:

```
sudo systemctl stop <service_name>
```

For example: `sudo systemctl stop anzo-server`

To start Anzo using the AnzoServer utility, run the following command. Make sure that you are logged in as the Anzo service user before stopping or starting Anzo:

```
<install_path>/Server/AnzoServer start
```

To stop Anzo, run the following command:

```
/<install_path>/Server/AnzoServer stop
```

You can also start and stop Anzo from the symbolic links if they were created for your installation. For example, `/etc/init.d/AnzoServer start` or `/etc/init.d/AnzoServer stop`.

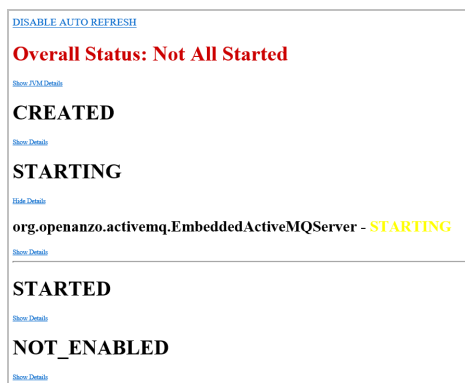
Monitoring Startup Status

It can take a few minutes for Anzo to complete the startup process. You can monitor the status by viewing the Anzo Status page. To see the Status page, go to the following URL in your browser:

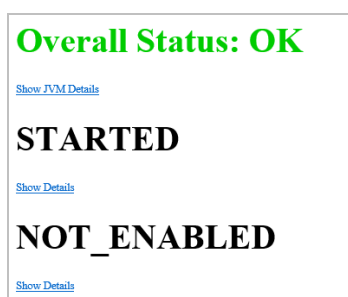
```
http://<server_name_or_IP_address>:8945/status
```

Where `<server_name_or_IP_address>` is the name or IP address of the server that hosts Anzo.

For example, the following image shows the Status page message displayed while Anzo is starting:



The image below shows the Status page message when Anzo startup is complete:



Related Topics

[Changing Anzo Server Settings](#)

[Updating the Server License](#)

Changing Anzo Server Settings

This topic provides instructions for changing Anzo server settings as well as reference information for each of the options.

Changing Settings

1. In the Administration application, expand the **Servers** menu and click **Server Settings**. The Server Settings screen is displayed. The options that you can configure are described on the screen:

2. To change the configuration, expand an option to display the related settings. Then click the **Edit** button and specify the desired value for each setting. For specifics about each option, see [Settings Reference](#) below.

Note

You can have one option open for editing at a time. If you are in the process of modifying an option and have not saved the changes, all other **Edit** buttons are disabled until you save or cancel the changes.

3. Click **Save** to save the changes, and then restart Anzo to complete the configuration change.

Important

After changing any of the server configuration settings, you must restart Anzo to apply the change.

Settings Reference

This section provides reference information for each configuration option.

- [Set the System Administrator Password](#)
- [Configure the Ports to be Used by the System](#)
- [Configure the Binary Store Server Options](#)
- [Configure the SMTP Server Used to Send Email](#)
- [Configure the Default Root Pages](#)
- [Configure HTTP Session Options](#)
- [Configure Anonymous User Access](#)
- [Configure URI Prefix and SPARQL Options](#)

- [Configure Global Prefixes](#)
- [Configure the Versioning Environment](#)
- [Configure Network Connections to an Anzo Distributed Unstructured Cluster](#)
- [Configure the Default ETL Engine](#)

Set the System Administrator Password

To change the system administrator (**sysadmin**) password, expand the **Administrator** option and click **Edit**.

The screenshot shows a dialog box titled "Administrator" with the subtitle "Set the System Administrator password." It contains three input fields: "Current Password" (masked with asterisks), "Password *" (with a toggle icon), and "Confirm Password *" (with a toggle icon). At the bottom right, there are "CANCEL" and "SAVE" buttons.

Type the new password in the **Password** and **Confirm Password** fields. Then click **Save**.

Configure the Ports to be Used by the System

To change, enable, or disable the Anzo server ports, expand the **Ports** option and click **Edit**.

The screenshot shows a dialog box titled "Ports" with the subtitle "Configure the ports to be used by the system." It features a table with columns: "Enabled", "Port", "SSL Port", and "Certificates". The table lists four rows: "Anzo Port and Anzo SSL Port", "Application", "Application SSL", and "Auxiliary". Each row has a toggle switch, a "Port" field, an "SSL Port" field, and a "Certificates" dropdown menu. At the bottom right, there are "CANCEL" and "SAVE" buttons.

Enabled	Port	SSL Port	Certificates
Anzo Port and Anzo SSL Port	61616	61617	anzo
Application	80		
Application SSL	443		anzo
Auxiliary	8945		
Auxiliary SSL	8946		anzo

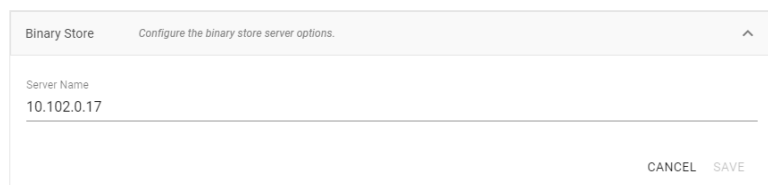
Change the values in the Port fields to specify alternate port numbers. To enable or disable a port, move slider next to the application name to the left or right. The list below describes the settings:

- The fields at the top of the screen specify the Anzo server ports. By default, the Anzo and Anzo SSL ports are enabled. If you want to disable one of the ports, click the **Enabled** drop down list and select the option that you want to leave enabled. To change port numbers, click in the Port field and specify the port.
- The **Application** and **Application SSL** ports are the HTTP and HTTPS client application ports.
- The **Auxiliary** and **Auxiliary SSL** ports are the HTTP and HTTPS Administration client ports.

For information about managing the certificates to use for the SSL ports, see [Using a Signed Certificate](#).

Configure the Binary Store Server Options

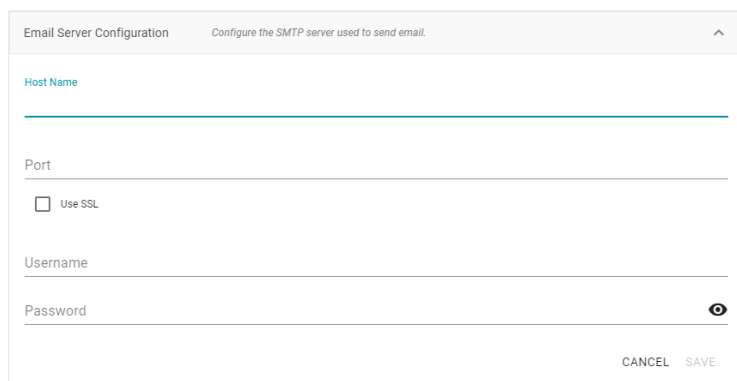
To change the host server for the binary (blob) store, expand **Binary Store** and click **Edit**.

A screenshot of a configuration dialog titled "Binary Store" with the subtitle "Configure the binary store server options." The dialog has a header bar with a close button (X) on the right. Below the header, there is a label "Server Name" followed by a text input field containing "10.102.0.17". At the bottom right of the dialog, there are two buttons: "CANCEL" and "SAVE".

The Server Name defaults to the host name or IP address for the Anzo server. To specify a different host for the binary store, type the new host name or IP address in the **Server Name** field, and then click **Save**.

Configure the SMTP Server Used to Send Email

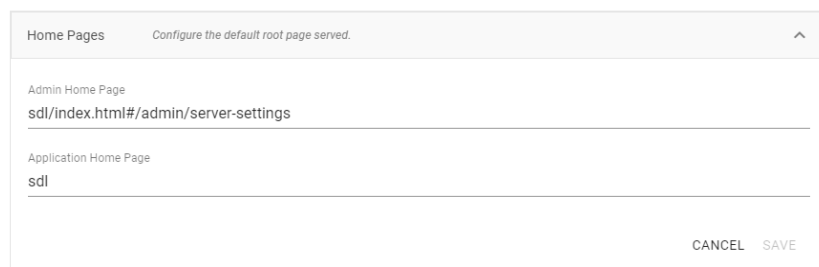
To configure an SMTP server for sending email, expand **Email Server Configuration** and click **Edit**.

A screenshot of a configuration dialog titled "Email Server Configuration" with the subtitle "Configure the SMTP server used to send email." The dialog has a header bar with a close button (X) on the right. Below the header, there are four text input fields: "Host Name", "Port", "Username", and "Password". The "Host Name" field is highlighted with a blue border. Below the "Port" field, there is a checkbox labeled "Use SSL". To the right of the "Password" field, there is an eye icon for toggling password visibility. At the bottom right of the dialog, there are two buttons: "CANCEL" and "SAVE".

- **Host Name** is the host name or IP address for the SMTP server.
- **Port** is the port for the connection.
- If the email server is configured for SSL authentication, select the **Use SSL** checkbox to enable SSL authentication.
- Specify the **Username** and **Password** to use for authentication.

Configure the Default Root Pages

To change the home page path for the Anzo application and Administration application URLs, expand **Home Pages** and click **Edit**.

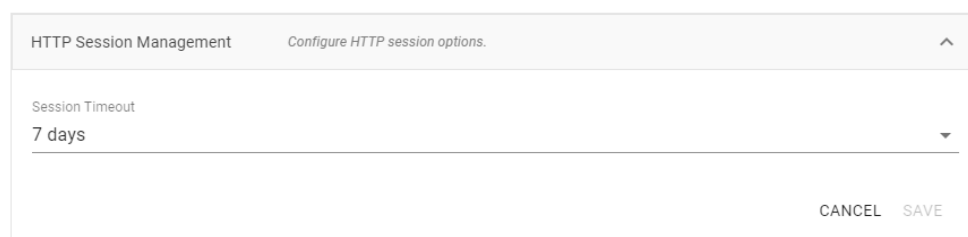


The screenshot shows a dialog box titled "Home Pages" with the subtitle "Configure the default root page served." It contains two input fields: "Admin Home Page" with the value "sdl/index.html#/admin/server-settings" and "Application Home Page" with the value "sdl". At the bottom right are "CANCEL" and "SAVE" buttons.

- The **Admin Home Page** is the home page path for the Administration application.
- The **Application Home Page** is the home page path for the Anzo application.

Configure HTTP Session Options

To configure the HTTP session timeout value, expand **HTTP Session Management** and click **Edit**.



The screenshot shows a dialog box titled "HTTP Session Management" with the subtitle "Configure HTTP session options." It contains a "Session Timeout" dropdown menu currently set to "7 days". At the bottom right are "CANCEL" and "SAVE" buttons.

Click the **Session Timeout** drop-down list and select the timeout value.

Configure Anonymous User Access

Before enabling anonymous access, consider the following security implications:

- [Anonymous User Permissions](#)
- [Anonymous User Limitations](#)
- [Important Considerations](#)

Anonymous User Permissions

When anonymous access is enabled:

- The server allows any user to connect to the Hi-Res Analytics application without a username and password. A user can connect to without having an account in Anzo.
- Anonymous users are considered members of the Everyone role. Anonymous users can read data in Anzo that is tagged as readable by Everyone.

Anonymous User Limitations

Anonymous users cannot:

- Add, delete, or modify data. Anonymous users cannot write or delete data even if the Everyone role has write or delete access.
- Change permissions on the artifacts in Anzo. Anonymous users cannot change the Sharing or Security tab settings for any data on the server even if the Everyone role has write or delete access to an artifact's metadata.

Important Considerations

This section lists important ideas to consider before enabling anonymous access.

Consider Existing Access Control

Users might have been permissions without anticipating that users could have anonymous access. Before enabling anonymous access, consider that data that is viewable by the **Everyone** role becomes visible to anonymous users. You might need to change the permissions for existing data, such as by granting read access to the **Authenticated Users** role instead of the Everyone role. For more information about permissions, see [Predefined Anzo Roles and Permissions](#).

Consider Server Network Protections

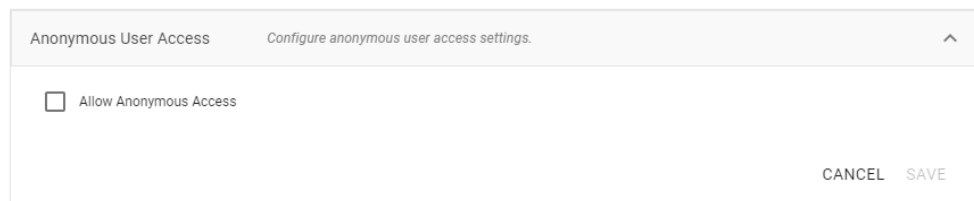
Consider that anyone who can reach the server via the network will be able to use it as an anonymous user. Evaluate firewalls and other network protection mechanisms to limit access to the Anzo server as desired. For example, you might want to allow anonymous access to anyone inside your organization's internal network but disable access to the server from the public internet.

Anonymous Access Can Be Useful

Allowing anonymous access makes it easy to share data and views of data with others. For example, it means that you can share your Hi-Res Analytics dashboards with people who do not have a user account. It also lets you embed read-only interactive Hi-Res Analytic views inside other websites.

Configuring Anonymous Access

To enable or disable anonymous user access, expand **Anonymous User Access** and click **Edit**.

A screenshot of a configuration window titled "Anonymous User Access" with a subtitle "Configure anonymous user access settings." and an upward arrow icon. Inside the window, there is a checkbox labeled "Allow Anonymous Access" which is currently unchecked. At the bottom right of the window, there are two buttons: "CANCEL" and "SAVE".

Anonymous User Access	
Configure anonymous user access settings.	
<input type="checkbox"/>	Allow Anonymous Access
CANCEL SAVE	

To enable anonymous access, select the **Allow Anonymous Access** checkbox. To disable anonymous access if it is enabled, clear the checkbox. Then click **Save**.

Configure URI Prefix and SPARQL Options

To enable or disable the Anzo SPARQL endpoint or customize the URI prefix that Anzo generates for data identifiers, expand **Data Interchange** and click **Edit**.

Data Interchange Configure URI prefix and SPARQL options. Enable [SPARQL](#) endpoint

☒ Enable Sparql Endpoint

URI Prefix

CANCEL SAVE

- If you want to enable or disable the Anzo SPARQL endpoint, select or clear the **Enable SPARQL Endpoint** checkbox.
- To change the prefix that Anzo uses when generating URIs, type the new value in the **URI Prefix** field. The URI Prefix is mostly used for consistency in internal data, but it is also used by default for data model URI prefixes when the model does not define the URI template to use. When changing the URI Prefix, make sure that the value is a valid prefix. See [Relative IRIs](#) in the SPARQL Query Language specification for more information.

Configure Global Prefixes

The Global Prefix Manager stores standard prefixes and any custom prefixes that you want Anzo to recognize globally. Defining global prefixes creates shortcuts for inserting the prefixes in Query Builder and data layer queries.

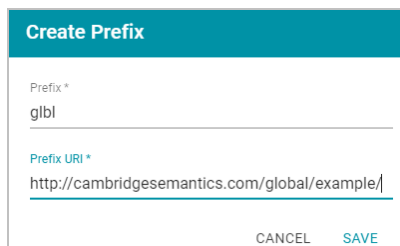
To manage global prefixes, expand **Global Prefix Manager**:

Global Prefix Manager Configure Global Prefixes

[+ ADD PREFIX](#)

Prefix	Uri		
dcterms	http://purl.org/dc/terms/	EDIT	DELETE
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	EDIT	DELETE
gbl	http://cambridgesemantics.com/global/example/	EDIT	DELETE
owl	http://www.w3.org/2002/07/owl#	EDIT	DELETE
dc	http://purl.org/dc/elements/1.1/	EDIT	DELETE
rdfs	http://www.w3.org/2000/01/rdf-schema#	EDIT	DELETE
foaf	http://xmlns.com/foaf/0.1/	EDIT	DELETE
xsd	http://www.w3.org/2001/XMLSchema#	EDIT	DELETE

To add a prefix, click **Add Prefix**. Anzo opens the Create Prefix dialog box. In the **Prefix** field, specify the abbreviation that you want to use to represent the URI. In the **Prefix URI** field, specify the full, valid URI. For example:



Create Prefix

Prefix *

gbl

Prefix URI *

<http://cambridgesemantics.com/global/example/>

CANCEL SAVE

Click **Save** to save the definition. To use global prefix shortcuts in the Anzo application, type "prefix" followed by a space in the Query Builder or a Query Step to open a tooltip that lists the global prefixes. For example:



Clicking a prefix inserts a PREFIX statement into the query. 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. For example, typing **gbl:** inserts a statement for the prefix that was defined in the example above.

Configure the Versioning Environment

To change the variable value for the Version Environment tag that Anzo adds to archived versions of entities, expand **Versioning** and click **Edit**.



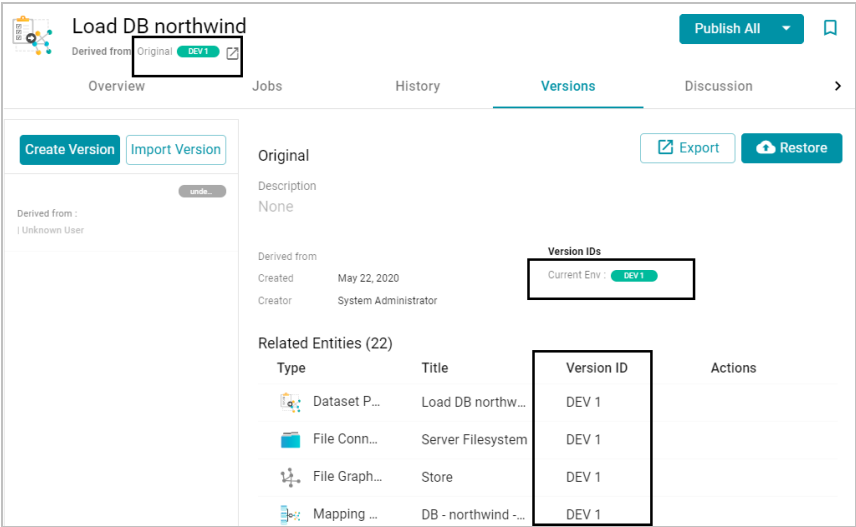
Versioning Configure the versioning environment

Versioning Environment

DEV

CANCEL SAVE

Edit the value in the **Versioning Environment** field and click **Save**. The image below shows an example of the version tags that are controlled by the Versioning Environment setting. The black rectangles highlight the areas where the environment version variable value is displayed:



Configure Network Connections to an Anzo Distributed Unstructured Cluster

To change the network settings for an Anzo Distributed Unstructured cluster, expand **Distributed Pipeline** and click **Edit**.

Note

If the Kubernetes infrastructure is set up to deploy Anzo Unstructured clusters on-demand, you do not need to configure these settings. For information about Kubernetes-based deployments, see [Using K8s for Dynamic Deployments of Anzo Components](#).

Distributed Pipeline

Configuration properties used for Network Connections in the Distributed Pipeline Service.

Distributed Pipeline Client Hostname

10.102.0.17

Distributed Pipeline Primary Seednode

akka.ssl.tcp://AnzoAkkaCluster@10.102.0.17:2551

Distributed Pipeline Callback Hostname

localhost

CANCEL

SAVE

Modify the settings as needed:

- Distributed Pipeline Client Hostname:** The hostname or IP address for the Anzo Unstructured leader instance.

Important

The value must be a routable IP address or hostname. If the leader instance is installed on the Anzo host server, specify the IP address or hostname of the server; do not use 127.0.0.1 or localhost.

- Distributed Pipeline Primary Seednode:** The IP address and port for the leader instance. By default the leader port is **2551**.

- **Distributed Pipeline Callback Hostname:** The hostname or IP address for the Anzo Unstructured leader instance. Typically this is the same value as the **Distributed Pipeline Client Hostname**.

Configure the Default ETL Engine

To set the default ETL engine so that it is automatically selected when users set up ingestion pipelines, expand **Default ETL Engine Config** and click **Edit**.



Click the **ETL Engine Config** drop-down list and select the ETL engine to make the default engine. Then click **Save**.

Related Topics

[Anzo Server Administration](#)

Managing Certificates

The topics in this section provide information about managing server certificates.

- [Using a Signed Certificate](#)
- [Adding a Certificate to the Trust Store](#)

Related Topics

[Changing Anzo Server Settings](#)

Using a Signed Certificate

Follow the instructions below if you want to replace the Anzo self-signed certificate with a signed certificate from a signing authority. The steps guide you through generating an SSL certificate using the OpenSSL utility, creating a signing request, and then uploading the signed certificate to Anzo.

- [Generating the SSL Certificate and Signing Request](#)
- [Uploading a Signed Certificate to Anzo](#)

Generating the SSL Certificate and Signing Request

1. If necessary, install OpenSSL.
2. Create a request configuration file. For example, create a file called **certificate.cnf**. Then add the following contents to the file. These contents include parameters for creating a multi-domain certificate:

```
# certificate.cnf

[req]
default_bits = 2048
prompt = no
default_md = rsa
req_extensions = req_ext
distinguished_name = dn

[ dn ]
C = <country>
ST = <state>
L = <locality>
O = <organization-or-company-name>
OU = <organizational-unit>
emailAddress = <email-address>
CN = <common-name-or-server-fqdn>

[ req_ext ]
subjectAltName = @alt_names

[ alt_names ]
DNS.1 = <domain1-name-or-ip>
DNS.2 = <domain2-name-or-ip>
DNS.3 = <domain3-name-or-ip>
```

3. Replace the placeholders in the file with the appropriate values. For example:

```
# certificate.cnf

[req]
default_bits = 2048
prompt = no
default_md = rsa
req_extensions = req_ext
distinguished_name = dn

[ dn ]
C = US
ST = MA
L = Boston
O = Cambridge Semantics
OU = IT
emailAddress = webmaster@cambridgesemantics.com
CN = sample.cambridgesemantics.com
```

```
[ req_ext ]
subjectAltName = @alt_names

[ alt_names ]
DNS.1 = sample1.domain.com
DNS.2 = 10.0.33.103
DNS.3 = sample3.domain.com
```

4. Run the following command to generate the signing request and private key using the configuration file:

```
openssl req -new -sha256 -nodes -out <csr_file_name>.csr -newkey rsa:2048
-keyout <key_name>.pem -config <config_file_name>.cnf
```

For example:

```
openssl req -new -sha256 -nodes -out anzo-csr.csr -newkey rsa:2048
-keyout anzo-key.pem -config certificate.cnf
```

5. Send the resulting CSR to a certificate authority for signing.

Uploading a Signed Certificate to Anzo

1. When you receive the signed certificate from the certificate authority, rename the certificate to **anzo-crt.crt**.
2. Create a PKCS12 key:
 - a. Run the following command to concatenate the signed certificate and private key file that you generated into an **anzo.pem** file:

```
cat <key_name>.pem anzo-crt.crt > anzo.pem
```

For example:

```
cat anzo-key.pem anzo-crt.crt > anzo.pem
```

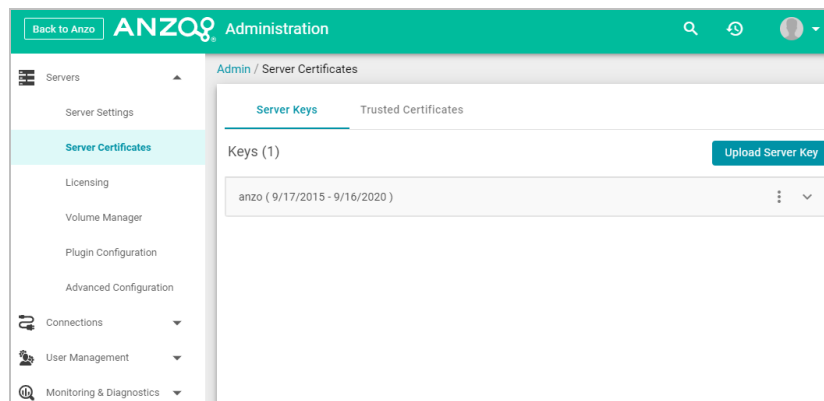
- b. Run the following command to convert the resulting **anzo.pem** file to PKCS12, choose a name for the certificate, and set an export password:

```
openssl pkcs12 -export -in anzo.pem -out anzo.pkcs12 -name "<alias>"
```

```
Enter Export Password:
```

```
Verifying - Enter Export Password:
```

3. Copy the **anzo.pkcs12** certificate to your computer if necessary.
4. In the Administration application, expand the **Servers** menu and click **Server Certificates**. Anzo displays the Server Certificates screen. For example:



5. Click **Upload Server Key**. Anzo displays the Upload Server Key dialog box.

6. Supply the required values:

- In the **Destination Alias** field, specify the alias that you chose when you created the PKCS12 certificate.
- In the **Password** field, specify the Export Password that you set when you created the PKCS12 certificate.
- Click the **Choose File** button and select the **anzo.pkcs12** file.
- Click the **Keystore type** field and select **PKCS12** from the drop-down list.

7. Click **Upload** to upload the certificate.

8. Finally, follow these steps to apply the new certificate to the Anzo server SSL ports:

- a. In the Servers menu, click **Server Settings**.
- b. On the Server Settings screen, expand **Ports** and click **Edit**. For example:

Enabled	Port	SSL Port	Certificates
<input checked="" type="checkbox"/>	61616	61617	anzo
<input checked="" type="checkbox"/>	80		
<input checked="" type="checkbox"/>	443		anzo
<input checked="" type="checkbox"/>	8945		
<input type="checkbox"/>	8946		anzo

- c. Click the **Certificates** drop-down list for each of the enabled SSL ports and select the new certificate. Then click **Save**.

9. Restart Anzo to apply the configuration change.

Related Topics

[Changing Anzo Server Settings](#)

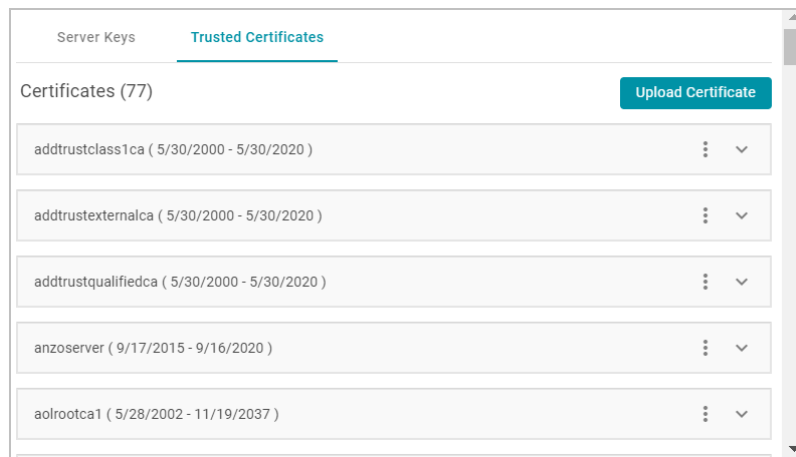
[Adding a Certificate to the Trust Store](#)

Adding a Certificate to the Trust Store

Follow the instructions below to upload a certificate to the Anzo trust store.

1. In the Administration application, expand the **Servers** menu and click **Server Certificates**. Anzo displays the Server Certificates screen. For example:

2. On the Server Certificates screen, click the **Trusted Certificates** tab. Anzo displays the list of existing certificates. For example:



- To upload a new trusted certificate, click the **Upload Certificate** button. Browse to the certificate file, and double-click the file to upload it to Anzo.

Related Topics

[Using a Signed Certificate](#)

Updating the Server License

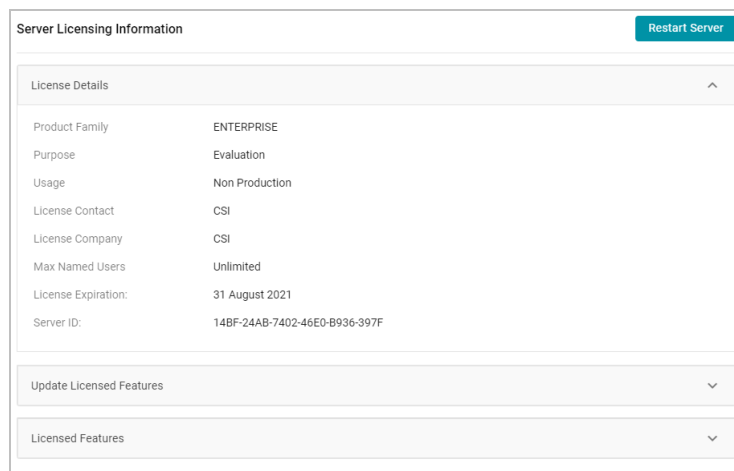
This topic provides important information about licenses and user accounts as well as instructions for updating a license key.

- [Updating the License Key](#)
- [Licensing and User Account Best Practices](#)

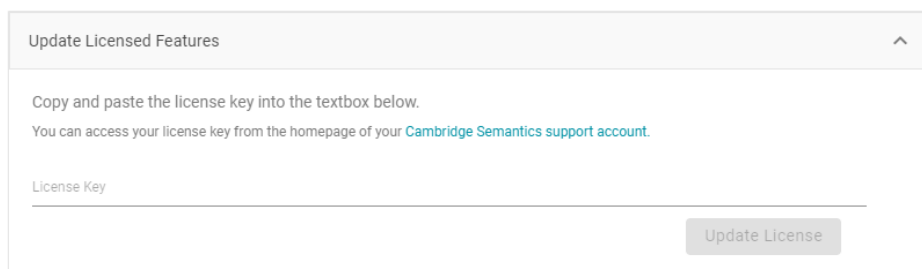
Updating the License Key

Follow the instructions below to update the Anzo server license key.

- In the Administration application, expand the **Servers** menu and click **Licensing**. Anzo displays the Server Licensing Information screen. For example:



2. Click **Update Licensed Features** to expand that section of the screen.



Update Licensed Features

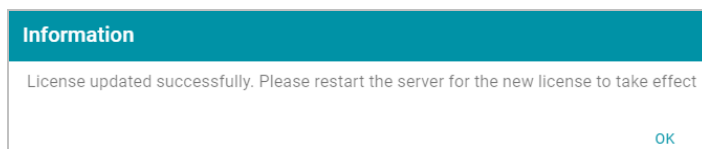
Copy and paste the license key into the textbox below.

You can access your license key from the homepage of your [Cambridge Semantics support account](#).

License Key

Update License

3. Paste the new license key into the **License Key** field, and then click the **Update License** button. The license is updated but does not take effect until Anzo is restarted. The following dialog box is displayed:



Information

License updated successfully. Please restart the server for the new license to take effect

OK

4. Click **OK** to close the dialog box. Then restart Anzo to apply the license updates. You can click the **Restart Server** button at the top of the screen. For information about other ways to stop and start Anzo, see [Starting and Stopping Anzo](#).

Licensing and User Account Best Practices

When Anzo is initially installed, a server ID is generated based on a number of system properties, including the user account that runs the installation script. The Anzo server license is tied to that server ID. If Anzo is re-installed (for instance, during an upgrade) by a different user account, a new server ID is generated and the existing license becomes invalid for the current installation. Whenever you upgrade or re-install Anzo, it is important to use the same user account that was used for the initial installation.

Restoring the Server ID if Anzo is Updated by the Wrong User

If Anzo is updated by a different user, the best way to resolve the issue is to revert the server ID to its original value by rolling back the update:

- If it was a new installation that used the wrong user account, uninstall Anzo. Then change to the correct user and run the installation script again.
- If your backup is a snapshot of the previous application disk, restore the disk. Then change to the correct user and update the installation.
- If it was an upgrade that used the wrong user account, restore Anzo from the backup that was saved before the upgrade:

If your backup is a copy of the Anzo system journal, follow these steps:

- a. Uninstall Anzo.
- b. Change to the correct user account.

- c. Reinstall the previous version of Anzo using the original installation script.
- d. After the installation, replace the **anzo.jnl** file in the `install_path/Server/data/journal` directory with the backup version of the file.

At this point, Anzo is restored to the previous version and has the server ID that is associated with the license.

- e. Now Anzo can be re-upgraded to the later release.

If your backup is a copy of the entire Anzo installation directory, follow these steps:

- a. Uninstall Anzo.
- b. Change to the correct user account.
- c. Move the copy of the previous Anzo installation directory to the original location on the file system.

At this point, Anzo is restored to the previous version and has the server ID that is associated with the license.

- d. Now Anzo can be re-upgraded to the later release.

Important

Cambridge Semantics strongly recommends that you do NOT change the user running Anzo. If it is absolutely necessary, the license can be changed so that it is associated with the new server ID, and Anzo can be restarted once the license is updated. However, using a new server ID resets (or regenerates from non-customer-specific templates) all previously configured OSGI properties to their default values. Changing the Anzo user should only be attempted if there is a complete record of all of the customized OSGI properties and their values as well as a thorough change log so that the configuration can be restored if necessary.

Related Topics

[Upgrading Anzo](#)

[Starting and Stopping Anzo](#)

Managing Volumes

The topics in this section provide information about creating new volumes (also known as journals or database instances) and mounting existing volumes.

- [Creating a New Volume](#)
- [Mounting an Existing Volume](#)

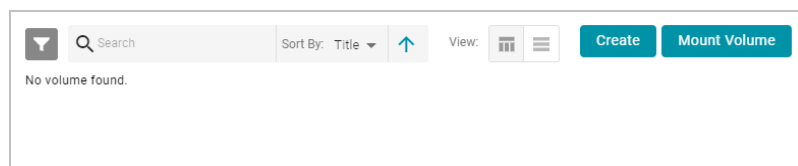
Creating a New Volume

This topic provides instructions for creating new volumes or journals.

Note

The number of volumes that you can create depends on your software license. For more information, contact Cambridge Semantics Support.

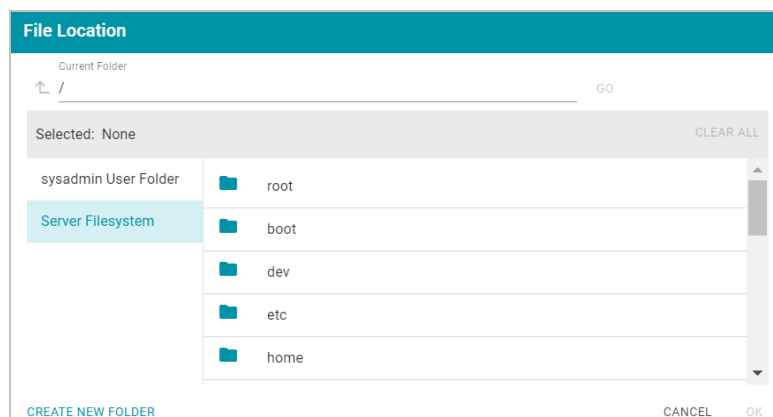
1. In the Administration application, expand the **Servers** menu and click **Volume Manager**. Anzo displays the Volume Manager screen, which lists any existing user-defined volumes (system volumes can be displayed by selecting the system data filter). For example:



2. Click the **Create** button. Anzo displays the Create New Volume dialog box.

The 'Create New Volume' dialog box has a teal header with the title 'Create New Volume'. Below the header, there are four input fields: 'Title *' with a placeholder 'The title of the datasource', 'Description' with a placeholder 'A brief description of the Datasource', 'Path *' with a 'BROWSE' link to its right, and 'Instance URI'. At the bottom left, there is a checkbox labeled 'Reset Enabled'. At the bottom right, there are 'CANCEL' and 'OK' buttons.

3. In the **Title** field, type a name for the new volume, and type an optional description in the **Description** field.
4. Click the **Path** field to open the File Location dialog box. For example:



- On the left side of the screen, select the file store where you want to create this volume. On the right side of the screen, select the directory where you want Anzo to save the volume. Then click **OK** to close the File Location dialog box. For instructions on creating a new file store, see [Connecting to a File Store](#).
- On the Create New Volume screen, complete the remaining fields:
 - Instance URI:** Anzo automatically assigns an instance URI to this volume. If you want to specify a custom URI, type the URI in this field.
 - Reset Enabled:** Specifies whether to enable resets. When reset is enabled, the option to reset the entire contents of the volume becomes available. To enable resets for this volume, select the **Reset Enabled** checkbox. To disable the reset option, leave the checkbox clear.
- Click **Save** to create the new volume in the location that you specified.

Related Topics

[Mounting an Existing Volume](#)

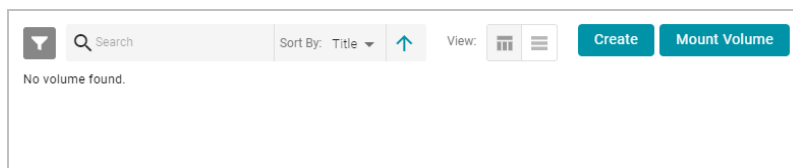
Mounting an Existing Volume

This topic provides instructions for mounting an existing volume or journal.

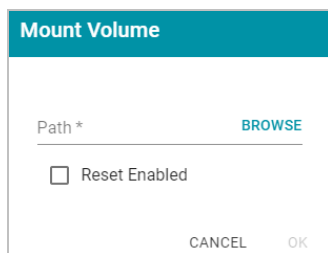
Note

The number of volumes that you can mount depends on your software license. For more information, contact Cambridge Semantics Support.

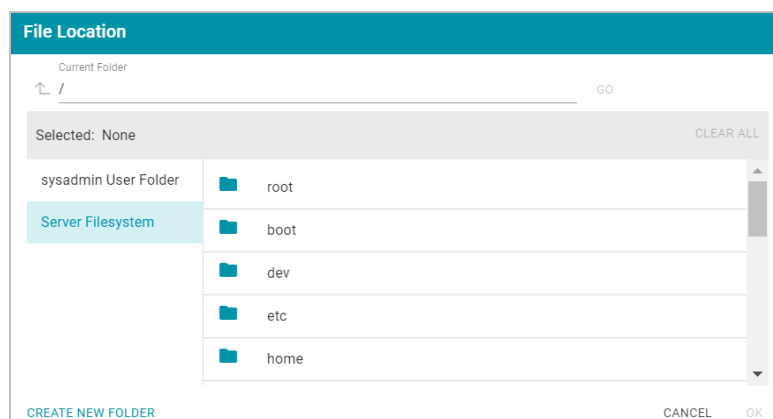
- In the Administration application, expand the **Servers** menu and click **Volume Manager**. Anzo displays the Volume Manager screen, which lists any existing user-defined volumes (system volumes can be displayed by selecting the system data filter). For example:



- Click the **Mount Volume** button. Anzo displays the Mount Volume screen.



- Click the **Path** field to open the File Location dialog box. For example:



- On the left side of the screen, select the file store that hosts the volume (.jnl file) that you want to mount. On the right side of the screen, navigate to the .jnl file and select it. Then click **OK**. Anzo mounts the new volume.

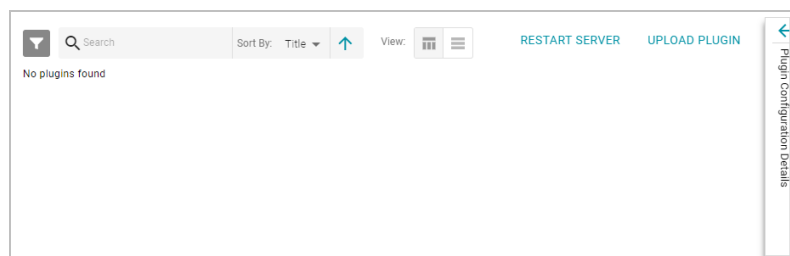
Related Topics

[Creating a New Volume](#)

Uploading a Plugin

When connecting to a relational database to import data, you may need to upload a JDBC driver to Anzo. You may also need to import custom bundles or other bundles received from Cambridge Semantics. This topic provides instructions for uploading executable .jar files from your computer to Anzo.

- In the Administration application, expand the **Servers** menu and click **Plugin Configuration**. Anzo displays the Plugin Configuration screen. For example:



- In the top right corner, click **Upload Plugin**. The application opens the file browser on your computer.
- In the file browser, navigate to the .jar file to upload, and then double-click the file to upload it. Anzo uploads the file and displays a "Completed" message. You do not need to restart Anzo to apply the new executable.

Related Topics

[Creating a Database Data Source](#)

[Connecting to an ETL Engine](#)

Advanced Configuration of Semantic Services

The topics in the section provide instructions for making the types of semantic service or application configuration changes that are commonly desired.

- [Setting a Base File Store Path for File Uploads](#)
- [Enabling and Configuring the System Monitor Service](#)
- [Normalizing LDAP Names](#)
- [Routing Hi-Res Analytics to a Custom URL](#)
- [Separating Audit Logs by Type of Event](#)
- [Limiting the Age \(and Size\) of Audit Logs](#)
- [Configuring a User Inactivity Timeout](#)
- [Reporting on Binary Store Access Events](#)
- [Configuring the Max Page Size for OData Feeds](#)
- [Scanning the Whole CSV File on Import](#)
- [Including Views as Schemas for Database Data Sources](#)
- [Limiting the Number of Anzo Unstructured Status Journals](#)

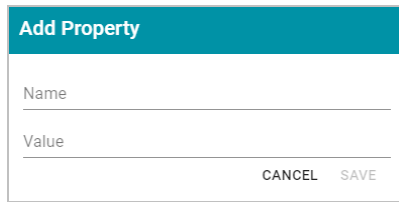
Setting a Base File Store Path for File Uploads

By default, if a user creates a file-based data source and uploads the source file (such as a CSV, XML, or JSON file) to Anzo from their computer, the file is copied to the server's data directory, `<install_path>/Anzo/Server/data/userUploads`. When the file is in the server installation path and not the shared file store it is not accessible by applications like AnzoGraph or Spark. In addition, other users cannot publish pipelines for that data source because they typically do not have access to the file. Source files that are routinely updated and re-ingested should be hosted on the shared file store.

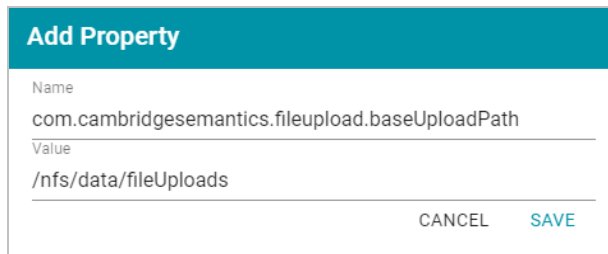
Follow the instructions below to configure Anzo to copy uploaded files to a location on the file store.

1. If necessary, create a directory on the shared file store that you can designate as the base location for saving uploaded files.
2. In the Administration application, expand the **Servers** menu and click **Advanced Configuration**. Click **I understand and accept the risk**.
3. Search for the **Anzo File Upload** bundle and view its details.
4. Click the **Services** tab and expand the `com.cambridgesemantics.anzo.fileupload.FileUploadServlet` service.

5. Click **Add Property** next to the service name. Anzo opens the Add Property dialog box.

A dialog box titled "Add Property" with a teal header. It contains two text input fields: "Name" and "Value". At the bottom right, there are two buttons: "CANCEL" and "SAVE".

6. In the **Name** field, specify **com.cambridgesemantics.fileupload.baseUploadPath**, and then set the **Value** to the location on the file store where uploaded files should be saved. The base directory that you specify must exist on the file store. For example:

A dialog box titled "Add Property" with a teal header. The "Name" field contains the text "com.cambridgesemantics.fileupload.baseUploadPath" and the "Value" field contains the text "/nfs/data/fileUploads". At the bottom right, there are two buttons: "CANCEL" and "SAVE".

7. Click **Save** to add the new property. And restart Anzo to apply the configuration changes.

When the base upload path is configured, source files that are uploaded from a user's computer will be saved to the location that you specified.

Related Topics

[Onboarding Structured Data](#)

Enabling and Configuring the System Monitor Service

The System Monitor service, which monitors state of the Java virtual machine (JVM), is disabled by default. You can enable the service to poll the state of the JVM at a certain interval and capture stack and heap dumps when memory utilization increases beyond a specified threshold. This topic provides instructions for enabling the service and configuring its options.

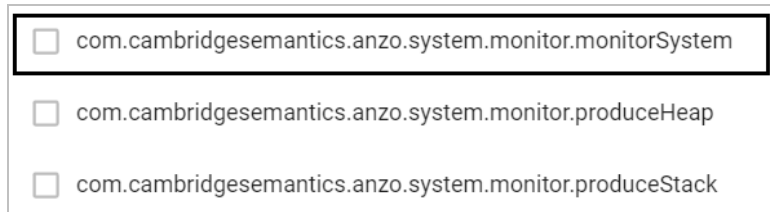
- [Enabling the System Monitor Service](#)
- [Configuring the System Monitor Service](#)

Enabling the System Monitor Service

Follow the steps below to enable the System Monitor.

1. In the Administration application, expand the **Servers** menu and click **Advanced Configuration**. Click **I understand and accept the risk**.
2. Search for the **Anzo System Monitor** bundle and view its details.
3. Click the **Services** tab and expand **System Monitor Activator**.

4. Locate the **com.cambridgesemantics.anzo.system.monitor.monitorSystem** property (shown in the image below).

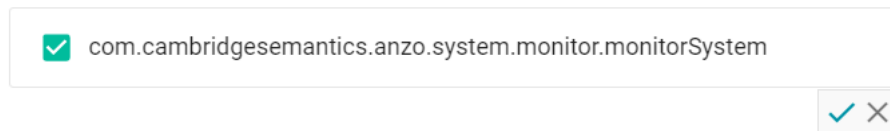


☐ com.cambridgesemantics.anzo.system.monitor.monitorSystem

☐ com.cambridgesemantics.anzo.system.monitor.produceHeap

☐ com.cambridgesemantics.anzo.system.monitor.produceStack

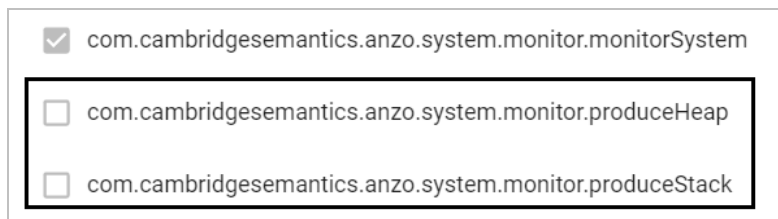
5. Click the property to make it editable, and then select the checkbox to enable it.



☒ com.cambridgesemantics.anzo.system.monitor.monitorSystem

☒ ☐

6. Click the checkmark icon (✓) for that property to save the change.
7. Next, configure the service to dump the stack and/or heap logs to disk by enabling the properties under the **monitorSystem** property:



☒ com.cambridgesemantics.anzo.system.monitor.monitorSystem

☐ com.cambridgesemantics.anzo.system.monitor.produceHeap

☐ com.cambridgesemantics.anzo.system.monitor.produceStack

To create heap dumps, enable **com.cambridgesemantics.anzo.system.monitor.produceHeap**. To create stack dumps, enable **com.cambridgesemantics.anzo.system.monitor.produceStack**.

8. You can restart Anzo to enable the service without performing additional configuration. Or see [Configuring the System Monitor Service](#) below for information about the configuration options.

Configuring the System Monitor Service

By default, the System Monitor Service is configured to monitor memory usage and take the following actions:

- Every **60 seconds** (60000 milliseconds), evaluate whether a stack or thread dump should be written.
- Write stack and/or heap dumps if the memory threshold reaches **85%** (0.85).
- Continue to write stack and/or heap dumps at an interval of every **10 minutes** (600000 milliseconds) as long as memory usage remains at or above the threshold.
- Save heap and stack dumps in the `<install_path>/Server/logs/system_monitor/heap` and `stack` directories.

To modify the characteristics described above, you can change the values for the following properties:

- To change the frequency with which memory usage is evaluated to see if it has reached the threshold, update the **com.cambridgesemantics.anzo.system.monitor.monitorDelay** property. Specify the number of milliseconds to wait between checks.
- To change the memory threshold, update the **com.cambridgesemantics.anzo.system.monitor.memoryThreshold** property. Specify the percent of total memory as a decimal value.
- To change how often stack and/or heap dumps are written when memory usage is above the threshold, update the **com.cambridgesemantics.anzo.system.monitor.dumpFrequency** property. Specify the number of milliseconds to wait between dumps.
- To change the location where heap and/or stack dumps are saved, update the **com.cambridgesemantics.anzo.system.monitor.heapLocation** and/or **com.cambridgesemantics.anzo.system.monitor.stackLocation** property to specify an alternate path and directory.

After changing any of the properties, make sure that you restart Anzo to apply the configuration change.

Related Topics

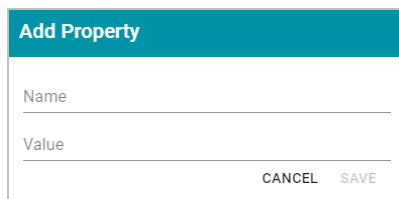
[Advanced Configuration of Semantic Services](#)

[Viewing the Current Stack in a Browser](#)

Normalizing LDAP Names

To ensure that duplicate user accounts are not created in Anzo if an LDAP distinguished name has both a lowercase and uppercase version, you can configure the system to normalize distinguished name strings so that values that differ only in capitalization are treated as the same value. Follow the steps below to normalize distinguished name strings.

1. In the Administration application, expand the **Servers** menu and click **Advanced Configuration**. Click **I understand and accept the risk**.
2. Search for the **Anzo Enterprise Directory Connect** bundle and view its details.
3. Click the **Services** tab and expand the **com.cambridgesemantics.anzo.virtualdirectory.VirtualDirectoryServer** service.
4. Click **Add Property** next to the service name. Anzo opens the Add Property dialog box.



5. In the **Name** field, specify **org.openanzo.security ldap.normalizeDnStrings**, and set the **Value** to **LCASE** if you want distinguished name values to be normalized to lowercase or **UCASE** if you want values to be normalized to uppercase. For example:

Add Property

Name
org.openanzo.security.Idap.normalizeDnStrings

Value
LCASE

CANCEL SAVE

- Click **Save** to add the property to the service. Then restart Anzo to apply the configuration changes.

Important

After making the service configuration change and restarting Anzo, any existing LDAP users or roles must be removed and then added to Anzo again.

Related Topics

[Connecting to a Directory Server](#)

Routing Hi-Res Analytics to a Custom URL

If you have a custom skin or personality for the Hi-Res Analytics application, and you want those customizations to be loaded automatically when users access the application, you can configure the Anzo application to re-route users to the preferred URL. Follow the instructions below to change the entry points to the Hi-Res application in the Anzo application. The instructions use the Find feature in the Query Builder to find and modify the object of the Hi-Res Analytics routing property.

- In the Anzo application, expand the **Access** menu and click **Query Builder**.
- In the Query Builder, click the **Find** tab. The Find screen is displayed with the **System Datasource** selected as the Source.

Query
Find

Source : System Datasource

Subject Predicate Object Graph

CLEAR ADD STATEMENT FIND

- In the **Subject** field, specify the following URI:

```
http://cambridgesemantics.com/Routes/sdi/hi-res-analytics-urn
```

4. In the **Predicate** field, specify this URI:

```
http://cambridgesemantics.com/ontologies/AnzoRoute#link
```

5. Click **Find** to display the quads with the specified subject and predicate. You can clear the **Subject** and **Named Graph Quick Filter** checkboxes to make the results easier to read. For example:

Result(1)		Quick Filter : <input type="checkbox"/> Subject <input checked="" type="checkbox"/> Predicate <input checked="" type="checkbox"/> Object <input type="checkbox"/> Named Graph
Predicate	Object	
« <http://cambridgesemantics.com/ontologies/AnzoRoute#link> »	« "/anzoweb/index.html?lens={value}" »	⋮
		Rows per page: 50 < >

6. Click the menu icon (⋮) for the quad and select **Edit**. Anzo opens the Edit Statement dialog box.

Edit Statement

Subject *

<http://cambridgesemantics.com/Routes/sdi/hi-res-analytics-urn>

Predicate *

<http://cambridgesemantics.com/ontologies/AnzoRoute#link>

Object *

"/anzoweb/index.html?lens={value}"

Named Graph URI *

<http://cambridgesemantics.com/Routes/sdi/hi-res-analytics>

CANCEL

SAVE

7. In the Edit Statement dialog box, replace the **Object** value ("/anzoweb/index.html?lens={value}") with the URL that you want to route users to. For example: "/myplace/index.html?lens={value}".

Edit Statement

Subject *

<http://cambridgesemantics.com/Routes/sdi/hi-res-analytics-urn>

Predicate *

<http://cambridgesemantics.com/ontologies/AnzoRoute#link>

Object *

"/myplace/index.html?lens={value}"

Named Graph URI *

<http://cambridgesemantics.com/Routes/sdi/hi-res-analytics>

CANCEL

SAVE

8. Click **Save** to apply the change and return to the Find screen.

The Anzo application is now configured to route users to the custom URL if they open the Hi-Res Analytics application from the Home page, open a dashboard from the Hi-Res Analytics screen, or click **Create Dashboard** from a Graphmart screen.

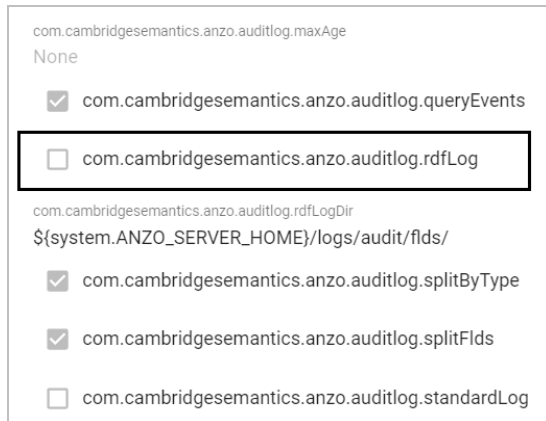
Related Topics

[Accessing Data with Hi-Res Analytics Dashboards](#)

Separating Audit Logs by Type of Event

By default, when Audit Log Packages, such as UserAudit, are enabled and set to Log Level **Info**, all types of audit events are logged to a single file: **anzo_audit_info.log**. You have the option, however, to configure Anzo to create and store smaller audit logs by generating separate files in subdirectories that are sorted by event type, such as **userEvents**, **queryEvents**, **accessEvents**, etc. Follow the instructions below to enable this option:

1. In the Administration application, expand the **Servers** menu and click **Advanced Configuration**. Click **I understand and accept the risk**.
2. Search for the **Anzo Audit Logging Framework** bundle and view its details.
3. Click the **Services** tab and expand **com.cambridgesemantics.anzo.AuditLog**.
4. Find the **com.cambridgesemantics.anzo.auditlog.rdfLog** property (shown below).



com.cambridgesemantics.anzo.auditlog.maxAge
None

☒ com.cambridgesemantics.anzo.auditlog.queryEvents

☐ com.cambridgesemantics.anzo.auditlog.rdfLog

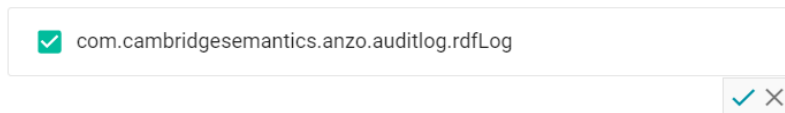
com.cambridgesemantics.anzo.auditlog.rdfLogDir
\${system.ANZO_SERVER_HOME}/logs/audit/flds/

☒ com.cambridgesemantics.anzo.auditlog.splitByType

☒ com.cambridgesemantics.anzo.auditlog.splitFlds

☐ com.cambridgesemantics.anzo.auditlog.standardLog

5. Click the property to make it editable, and then select the checkbox to enable it.



☒ com.cambridgesemantics.anzo.auditlog.rdfLog

✓ ✕

6. Click the checkmark icon (✓) to save the change.
7. Restart Anzo to apply the configuration changes.

Once new audit events are triggered, an **audit/audit-flds** subdirectory is created in the **<install_path>/Server/logs** directory. And audit logs will be created in the **userEvents**, **queryEvents**, **accessEvents**, etc. subdirectories.

Related Topics

[System Query Audit](#)

[Enabling and Viewing Audit Logs](#)

Limiting the Age (and Size) of Audit Logs

If you want to retain all of the audit log data but work with smaller data sets when loading and analyzing the log, you can configure Anzo to add an age limit (in days) to audit log data sets. Once an audit log data set reaches that age, Anzo stops writing to it and a new audit log data set is started. Follow the instructions below to configure the audit log service to add an age limit.

1. In the Administration application, expand the **Servers** menu and click **Advanced Configuration**. Click **I understand and accept the risk**.
2. Search for the **Anzo Audit Logging Framework** bundle and view its details.
3. Click the **Services** tab and expand **com.cambridgesemantics.anzo.AuditLog**.
4. Find the **limitAge** and **maxAge** properties (shown below).

The screenshot shows the configuration page for the **com.cambridgesemantics.anzo.AuditLog** service. The page has a header with the service name and a 'STARTED' status. Below the header is a URL and two buttons: 'ADD PROPERTY' and 'DELETE INSTANCE'. The main content area lists several properties, each with a checkbox:

- ☒ org.openanzo.services.enabled
- ☒ com.cambridgesemantics.anzo.auditlog.accessEvents
- ☒ com.cambridgesemantics.anzo.auditlog.activityEvents
- com.cambridgesemantics.anzo.auditlog.auditQueryCutoffTime: None
- ☐ com.cambridgesemantics.anzo.auditlog.auditSystemtableQueries
- ☐ com.cambridgesemantics.anzo.auditlog.gzipRdf
- ☐ com.cambridgesemantics.anzo.auditlog.includeCallStack
- ☒ com.cambridgesemantics.anzo.auditlog.includeNonSysadmin
- ☒ com.cambridgesemantics.anzo.auditlog.includeSysadmin
- ☐ com.cambridgesemantics.anzo.auditlog.includeTimingStack
- ☐ com.cambridgesemantics.anzo.auditlog.limitAge (highlighted with a red box)
- ☐ com.cambridgesemantics.anzo.auditlog.logFullTransaction
- com.cambridgesemantics.anzo.auditlog.maxAge: None (highlighted with a red box)
- ☒ com.cambridgesemantics.anzo.auditlog.queryEvents

5. Select the **com.cambridgesemantics.anzo.auditlog.limitAge** checkbox to enable the age limit feature.
6. Edit the **com.cambridgesemantics.anzo.auditlog.maxAge** property to specify the maximum number of days to log in each data set. When the current audit log reaches that age, Anzo starts writing to a new data set.
7. Restart Anzo to apply the configuration changes.

Related Topics

[System Query Audit](#)

[Enabling and Viewing Audit Logs](#)

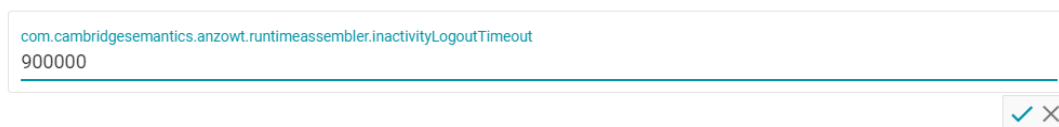
Configuring a User Inactivity Timeout

By default, the user inactivity timeout setting in the Anzo Java Script Runtime Assembler service is set to **unlimited**, meaning Anzo will not automatically log out users who have a session open but remain inactive. If you want to configure Anzo to log users out if they are inactive for a period of time, follow the instructions below.

1. In the Administration application, expand the **Servers** menu and click **Advanced Configuration**. Click **I understand and accept the risk**.
2. Search for the **Anzo Java Script Runtime Assembler** bundle and view its details.
3. Click the **Services** tab and expand the **Anzo Java Script Runtime Assembler** service.
4. Edit the **com.cambridgesemantics.anzowt.runtimeassembler.inactivityLogoutTimeout** property (shown in the image below) to specify the number of **milliseconds** that a user can remain inactive before being logged out.



For example, setting the value to **900000** milliseconds means that a user who is inactive for more than 15 minutes is automatically logged out.



5. After specifying the value, click the checkmark icon (✓) for that property to save the change.
6. Restart Anzo to apply the configuration change.

Note

By default, Anzo is not configured to log an event when the user inactivity value is changed. If you would like this event to be noted in the Audit log when the setting is changed, see [Enabling and Viewing Audit Logs](#) for instructions.

Related Topics

[Enabling and Viewing Audit Logs](#)

Reporting on Binary Store Access Events

By default, binary store access events are not captured in the Audit log. You can configure the audit logging framework to capture information about binary store requests, however. Data such as the time of the request, the user who made the request, and the document that was accessed will be captured. Follow the instructions below to configure the log to report on binary store events.

1. In the Administration application, expand the **Servers** menu and click **Advanced Configuration**. Click **I understand and accept the risk**.
2. Search for the **Anzo Audit Logging Framework** bundle and view its details.
3. Click the **Services** tab and expand **com.cambridgesemantics.anzo.AuditLog**.
4. Select the checkbox next to the **com.cambridgesemantics.anzo.auditlog.rdfLog** property to enable the option.
5. Make sure that the **com.cambridgesemantics.anzo.auditlog.splitByType** property is selected/enabled (it is enabled by default).
6. Restart Anzo to apply the configuration change.

New binary store access audit events will be added to the logs in the subdirectories under `<install_path>/Server/logs/audit/audit-flds`.

Related Topics

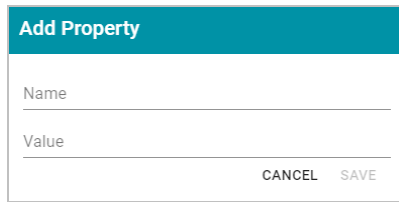
[Advanced Configuration of Semantic Services](#)

Configuring the Max Page Size for OData Feeds

When a user sends a request to an Anzo Data on Demand endpoint, they do not necessarily know the total number of results that will be returned. In some cases, the result set can be hundreds of millions of values, and the request times out before the results can be returned. You can configure the Data on Demand service to specify a maximum limit on the number of results that can be returned for a single OData feed request. If a user sends a request and the result set is larger than the maximum value, Anzo will limit the results to the configured maximum value. Follow the instructions below to configure the Data on Demand service to enforce a maximum page size.

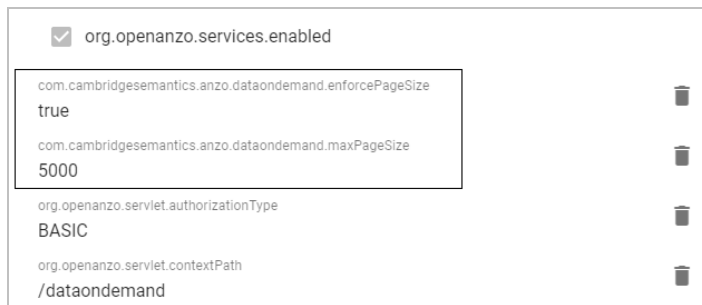
1. In the Administration application, expand the **Servers** menu and click **Advanced Configuration**. Click **I understand and accept the risk**.
2. Search for the **Anzo DataOnDemand** bundle and view its details.
3. Click the **Services** tab and expand **DataOnDemandServiceActivator**.

- Click **Add Property** next to the service name. Anzo opens the Add Property dialog box.



The 'Add Property' dialog box has a teal header. It contains two input fields: 'Name' and 'Value'. At the bottom right, there are two buttons: 'CANCEL' and 'SAVE'.

- In the **Name** field, specify **com.cambridgesemantics.anzo.dataondemand.enforcePageSize**, and set the **Value** to **true**. Then click **Save**.
- Click **Add Property** again. In the **Name** field, specify **com.cambridgesemantics.anzo.dataondemand.maxPageSize**, and set the **Value** to the maximum number of results that to return per request. Then click **Save**. The two settings are displayed on the Services screen. For example:



The Services screen shows a list of properties. At the top, there is a checked checkbox for 'org.openanzo.services.enabled'. Below it, a box highlights two properties: 'com.cambridgesemantics.anzo.dataondemand.enforcePageSize' with value 'true', and 'com.cambridgesemantics.anzo.dataondemand.maxPageSize' with value '5000'. Below this box, there are two more properties: 'org.openanzo.servlet.authorizationType' with value 'BASIC', and 'org.openanzo.servlet.contextPath' with value '/dataondemand'. Each property has a trash icon to its right.

- Restart Anzo to apply the configuration changes.

Related Topics

[Accessing Data on Demand Endpoints](#)

Scanning the Whole CSV File on Import

To help improve accuracy of data type assignment when importing CSV files, you have the option to configure the system so that any time a CSV file is imported, Anzo scans the entire file before inferring the data types for each column. Follow the instructions below if you want to configure the system to scan entire CSV files.

Important

This change affects all CSV file imports. Users cannot opt-out of a complete scan at import time. This configuration is not related to the **Use Extended Sample** setting in file import options. Choosing to scan entire files will significantly increase the time it takes to import files. However, scanning the complete file is the best way to ensure that data type assignments are accurate.

- In the Administration application, expand the **Servers** menu and click **Advanced Configuration**. Click **I understand and accept the risk**.
- Search for the **Anzo Utilityservices VFS** bundle and view its details.
- Click the **Services** tab and expand **UtilityServices VFS Activator**.

- Find the **com.cambridgesemantics.anzo.utilityservices.vfs.isSampleEntireFile** property, and select the checkbox to enable the option.

Note

When **SampleEntireFile** is enabled, the values in the **maxSampleSize** and **sampleSize** properties are ignored and Anzo always scans entire CSV files on import.

- Restart Anzo to apply the configuration changes.

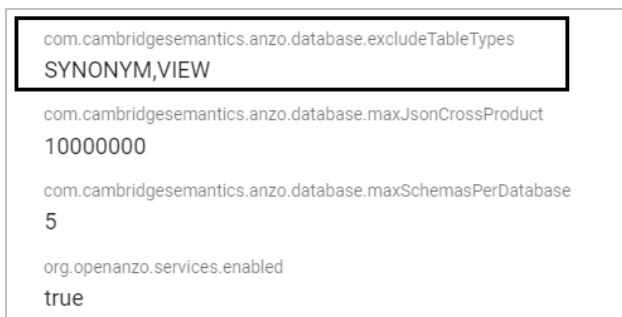
Related Topics

[Creating a CSV Data Source](#)

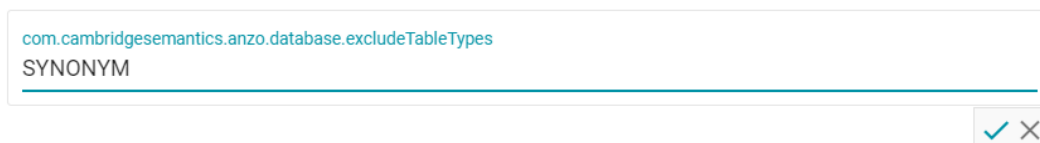
Including Views as Schemas for Database Data Sources

By default, when you create a Database Data Source and import a predefined Schema, Views are excluded from the list of Schemas that are available to import. However, you can configure the Anzo Database DataSource Provider Service to include Views as Schemas. Follow the steps below to remove Views from the list of table types that are excluded from import.

- In the Administration application, expand the **Servers** menu and click **Advanced Configuration**. Click **I understand and accept the risk**.
- Search for the **Anzo Database DataSource Provider** bundle and view its details.
- Click the **Services** tab and expand **com.cambridgesemantics.anzo.database.IDbConnectionService**.
- Locate the **com.cambridgesemantics.anzo.database.excludeTableTypes** property (shown in the image below).



- Click the property to make it editable, and then delete the word **VIEW**.



- Click the checkmark icon (✓) for that property to save the change.
- Restart Anzo to apply the configuration change.

The service is now configured to display Views in the Import Schemas dialog box as described in [Importing a Predefined Schema](#).

Related Topics

[Advanced Configuration of Semantic Services](#)

[Defining a Database Schema](#)

Limiting the Number of Anzo Unstructured Status Journals

To limit the disk space used by Anzo Unstructured pipelines, you have the option to configure the Anzo Unstructured Distributed service to limit the number of status journals that are preserved on disk. When the specified limit is reached and a pipeline generates a new journal, the oldest journal is deleted.

Note

Journals are removed based on their timestamps alone. The pipeline they are associated with is not a factor in determining the journals to delete.

Follow the instructions below to configure the Unstructured Distributed service to limit the number of status journals on disk.

1. In the Administration application, expand the **Servers** menu and click **Advanced Configuration**. Click **I understand and accept the risk**.
2. Search for the **Anzo Unstructured Distributed** bundle and view its details.
3. Click the **Services** tab and expand **Anzo Unstructured Distributed**.
4. Edit the `com.cambridgesemantics.anzo.unstructured.distributed.defaultNumStatusJournalGlobalLimit` property to specify the maximum number of status journals to keep on disk. The default value is `-1`, which is unlimited.
5. After changing the value, click the checkmark icon (✓) for that property to save the change.
6. Restart Anzo to apply the configuration change.

Connection Administration

The topics in this section provide information about managing connections to the Anzo server.

- [Connecting to a File Store](#)
- [Creating an Anzo Data Store](#)
- [Connecting to AnzoGraph](#)
- [Connecting to Elasticsearch](#)
- [Connecting to an ETL Engine](#)
- [Connecting to a Cloud Location](#)

Connecting to a File Store

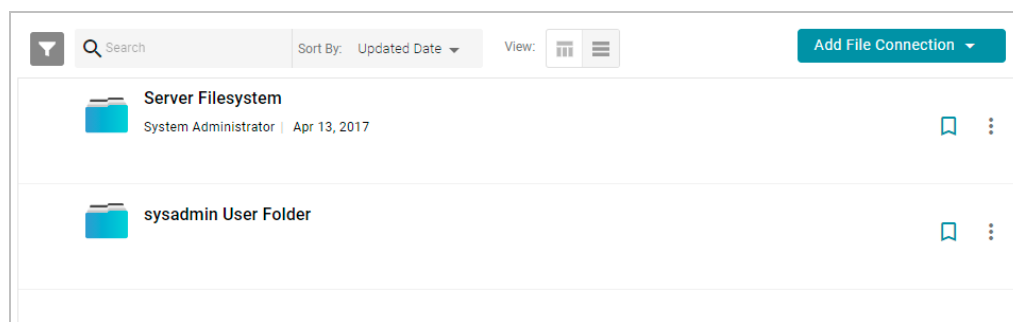
This topic provides instructions for connecting to the file system that all Anzo components will read from and write to during the onboarding processes. At least one file store needs to be shared between Anzo, AnzoGraph, and any Anzo Unstructured, Elasticsearch, or Spark servers. In almost all cases, organizations create an NFS to mount to all of the servers in the Anzo environment. Mounted file systems typically offer the best performance for reading and writing files. For more information, see [Deploying the Shared File System](#).

Note

The Anzo server file system location is configured and accessible by default. If you store files on a storage system that is mounted directly onto the Anzo, AnzoGraph, Elasticsearch, Anzo Unstructured, and Spark servers, you are not required to configure that location.

Anzo supports reading from and writing to local or mounted file systems (such as NFS), Hadoop Distributed File Systems (HDFS), File Transfer Protocol (FTP or FTPS) systems, Google Cloud Platform (GCP) storage, and Amazon Simple Cloud Storage Service (S3).

1. In the Administration application, expand the **Connections** menu and click **File Store**. Anzo displays the File Store screen, which lists existing file store connections. For example:



2. Click the **Add File Connection** button and select the type of file connection that you want to create. For the local disk or mounted NFS, choose **Local File Connection**. Anzo displays the create connection screen for the type of connection you chose.
3. On the connection screen, provide the file system details. The settings that display depend on the type of file connection that you chose. The list below describes the settings for each file connection type.

Local File Connection

Create Local File Connection

Name *

Base Folder

☐ Globally accessible filesystem

CANCEL SAVE

- **Name:** The name to use to describe this file connection within Anzo.
- **Base Folder:** The base or root folder on the file system where you want Anzo to either read or write files. Each time Anzo generates new files it creates a new subdirectory under this base location.
- **Globally accessible filesystem:** Select this option if this file store is accessible by all of the servers in an AnzoGraph cluster. If only the AnzoGraph leader server can access this system, leave this option blank.

HDFS File Connection

Create HDFS File Connection

Name *

Nameservice IP or Name *

Port

Base Folder

HDFS Configuration Path [BROWSE](#)

Keytab Path [BROWSE](#)

Username

Password

Confirm Password

CANCEL SAVE

- **Name:** The name to use to describe this file connection within Anzo.
- **Nameservice IP or Name:** The IP address or host name for the storage system.

- **Port:** The RPC port to access the server on. The default RPC port is 8020.
- **Base Folder:** The base or root folder on the file system where you want Anzo to either read or write files. Each time Anzo generates new files it creates a new subdirectory under this base location.
- **HDFS Configuration Path:** Enter the full path to the configuration files.
- **Keytab Path:** The full path to the keytab file.
- **Username:** The user name for the account used to access the server.
- **Password** and **Confirm Password:** The password for the account used to access the server.
- **Nameservice Rest IP or Name:** The HTTP REST IP address or host name. Typically this value is the same as the Nameservice IP or Name.
- **Nameservice Rest Port:** The HTTP port. AnzoGraph uses this port to access HDFS and load the FLDS. The default HTTP port for the namenode is 9870.
- **Nameservice Rest Protocol:** The protocol to use for requests. Specify one of the following values:
 - **hdfs:** Specify **hdfs** for non-secure HTTP protocol.
 - **shdfs:** Specify **shdfs** for secure HTTPS protocol.
 - **khdfs:** Specify **khdfs** for non-secure HTTP protocol with Kerberos authentication.
 - **kshdfs:** Specify **kshdfs** for secure HTTPS protocol with Kerberos authentication.

Important

If you use Kerberos Authentication with HDFS, you must also configure your AnzoGraph cluster to authenticate with Kerberos. For instructions, see [Configuring AnzoGraph for Kerberos Authentication](#).

- **Globally accessible filesystem:** Select this option if this file store is accessible by all of the servers in an AnzoGraph cluster. If only the AnzoGraph leader server can access this system, leave this option blank.

FTP or FTPS File Connection

Create FTPS File Connection


Name *


Server IP or Name *

Port

Base Folder

Username

Password 

Confirm Password 

Keystore Path

BROWSE

☐ Globally accessible filesystem

CANCEL

SAVE

- **Name:** The name to use to describe this file connection within Anzo.
- **Server IP or Name:** The IP address or host name for the storage system.
- **Port:** The port to access the server on.
- **Base Folder:** The base or root folder on the file system where you want Anzo to either read or write files. Each time Anzo generates new files it creates a new subdirectory under this base location.
- **Username:** The user name for the account used to access the server.
- **Password** and **Confirm Password:** The password for the account used to access the server.
- **Keystore Path:** For FTPS connections, the full path to the keystore file.
- **Globally accessible filesystem:** Select this option if this file store is accessible by all of the servers in an AnzoGraph cluster. If only the AnzoGraph leader server can access this system, leave this option blank.

Google Cloud Platform File Connection

Create Google Cloud Platform File Connection

Name *

Bucket Name *

Base Folder

Account Email

Key File Location

BROWSE

☐ Globally accessible filesystem

CANCEL

SAVE

- **Name:** The name to use to describe this file connection within Anzo.
- **Bucket Name:** The name of the bucket to store files in.

- **Base Folder:** The base or root folder on the file system where you want Anzo to either read or write files. Each time Anzo generates new files it creates a new subdirectory under this base location.
- **Account Email:** The email address for the account used to access the storage.
- **Key File Location:** The full path to the keystore password file.
- **Globally accessible filesystem:** Select this option if this file store is accessible by all of the servers in an AnzoGraph cluster. If only the AnzoGraph leader server can access this system, leave this option blank.

S3 File Connection

Important

When using Amazon S3 for file storage, do not use client-side encryption, where data is encrypted before it is sent to Amazon S3. Anzo cannot read files on S3 if the object store uses client-side encryption.

- **Name:** The name to use to describe this file connection within Anzo.
- **Bucket Name:** The name of the bucket to store files in.
- **Base Folder:** The base or root folder on the file system where you want Anzo to either read or write files. Each time Anzo generates new files it creates a new subdirectory under this base location.
- **Access Key:** The Access Key ID to use for accessing the S3 location.
- **Secret Key** and **Confirm Secret Key:** The Secret Key ID for the Access Key.
- **S3 URI Scheme:** Specifies whether the URI scheme is S3, S3 Native, or S3A.
- **Globally accessible filesystem: Required.** Enable this option for S3 file stores.

4. Click **Save** to save the configuration. The file store connection that you specified becomes available as a choice when you create graph data stores or select source files to onboard.

Related Topics

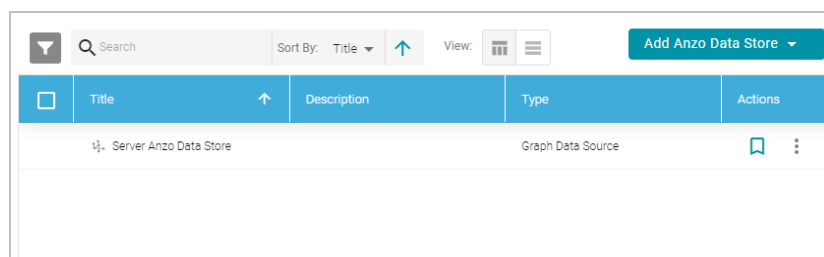
[Creating an Anzo Data Store](#)

Creating an Anzo Data Store

This topic provides instructions for creating an Anzo Data Store (sometimes called a graph data source). Creating a Data Store means that you designate a directory on the File Store where file-based linked data sets and other files can be created and shared during the ETL process. All installations require at least one 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 directory) or you can create multiple Data Stores to use for different Datasets.

For information about setting up a connection to the shared file system that will host the data store, see [Connecting to a File Store](#).

1. In the Administration application, expand the **Connections** menu and click **Anzo Data Store**. Anzo displays the Anzo Data Store screen, which lists any existing data stores. For example:



	Title	Description	Type	Actions
	Server Anzo Data Store		Graph Data Source	

Important

The **Server Anzo Data Store** is a default data store that points to the local Anzo file system. This store exists so that first-time users can quickly test the onboarding process. It is not meant to be used in production. Do not change the Data Location to a shared file store; reconfiguring this data store can cause unexpected consequences when upgrading or migrating the system. It is safe to delete this data store so that it is not presented as an option when users configure ingestion pipelines.

2. On the Anzo Data Store screen, click the **Add Anzo Data Store** button and select **Add Anzo Data Store**. Anzo opens the Create Anzo Data Store screen.

Create Anzo Data Store

Title *

Description

Data Location * [BROWSE](#)

Max File Size Before Compression (Bytes)

☒ Compress output ☐ Dedupe output per executor

[CANCEL](#) [SAVE](#)

3. Type a **Title** and optional **Description** for the Data Store.
4. Click in the **Data Location** field. Anzo opens the File Location dialog box.

File Location

Current Folder
[↑](#) /nfs/data/ [GO](#)

Selected: None [CLEAR ALL](#)

Server Filesystem

- crawl
- csv
- datafox
- dictionary
- docs

[CREATE NEW FOLDER](#) [CANCEL](#) [OK](#)

5. On the left side of the screen, select the File Store on which to create this Data Store. On the right side of the screen, navigate to the directory that you want to designate as the data location. Select a directory, and then click **OK**. Or click **Create New Folder** to create a new directory. Each time a pipeline is run for this Data Store, a new subdirectory is created under the specified data location.

Note

The Data Location needs to be a directory on the file store that is shared between Anzo, AnzoGraph, and any Anzo Unstructured, Elasticsearch, or Spark servers. If you want Anzo to generate files for this data store in one location and then load the files into AnzoGraph from another location, specify the file generation location in this field, and then specify the AnzoGraph load location in the **Alternate Data Location** field that is displayed on the Details screen after you save the data store.

6. If necessary, you can modify the maximum limit for the size of the files that are created by pipelines that write to this data store by specifying the size (in bytes) in the **Max File Size Before Compression (Bytes)** field. The

value applies to files before they are compressed. The Spark ETL engine partitions files on output, and the default maximum file size is 100 MB (uncompressed). The Sparkler ETL engine partitions files on input, and the default maximum file size is 128 MB (uncompressed). Since Sparkler files are partitioned on input, the resulting output FLDS files can be significantly larger than 128 MB since the source is converted to Turtle (TTL) format after it is partitioned.

Note

Cambridge Semantics recommends that you do not set this value unless instructed to do so by Cambridge Semantics Support.

7. Specify whether to compress the generated load files. By default, the **Compress output** checkbox is selected, indicating that Anzo generates .ttl.gz files when writing to this graph data source. If you clear the checkbox, Anzo generates uncompressed .ttl files. To preserve disk space and reduce read times when loading data into memory, Cambridge Semantics recommends that you accept the default configuration and compress load files.
8. The ETL engine does not remove duplicates by default when running Pipelines. If the source contains a significant number of duplicate entities, you have two options for deduplicating the data:
 - **Deduplicate the data during the ETL process:** To deduplicate the data while running the jobs that will generate this graph source, select the **Dedupe output per executor** option. Enabling the dedupe option limits the number of duplicates to one duplicate per executor node. For example, if the Spark configuration has 10 executor nodes, the resulting data set can contain a maximum of 10 duplicate entities.

Important

Deduplication is based on primary keys and URI templates. If the source does not employ templating, do not enable the dedupe option. In addition, enabling this option substantially increases the time it takes to run the jobs for this data store.

- **Deduplicate the data after loading it to AnzoGraph:** AnzoGraph deduplicates data during a "vacuum" process that runs automatically after data is loaded into memory. If you leave the **Dedupe output per executor** option disabled, duplicates will be removed by AnzoGraph.

Note

Deduplicating data with AnzoGraph streamlines the ETL process but can increase load time and temporary memory usage in AnzoGraph during the load.

9. Click **Save** to create the data store. Anzo saves the configuration and displays the details view. For example:

The screenshot shows the 'Store' configuration page in Anzo. The 'Overview' tab is selected. On the left, there are fields for 'Description' (None), 'Data Location' (/nfs/data/store/), 'Alternate Path' (None), and 'Max File Size Before Compression (Bytes)' (None). Below these are two checkboxes: 'Compress output' (checked) and 'Dedupe output per executor' (unchecked). On the right, a 'General' section displays metadata: 'Type' (Graph), 'Creator' (System Administrator), 'Updated' (a few seconds ago), 'Released' (a few seconds ago), and 'Tags' (None). A URL 'http://cambridgesemantics.com/FileGra...' is also visible.

You can click the Edit icon (✎) to modify any of the options. Click the check mark icon (✓) to save changes to an option, or click the X icon (✕) to clear the value for an option.

10. If you plan to load files into AnzoGraph from a location that is different than the **Data Location** that you specified, edit the **Alternate Data Location** field and select the location for AnzoGraph load files.

Related Topics

[Connecting to a File Store](#)

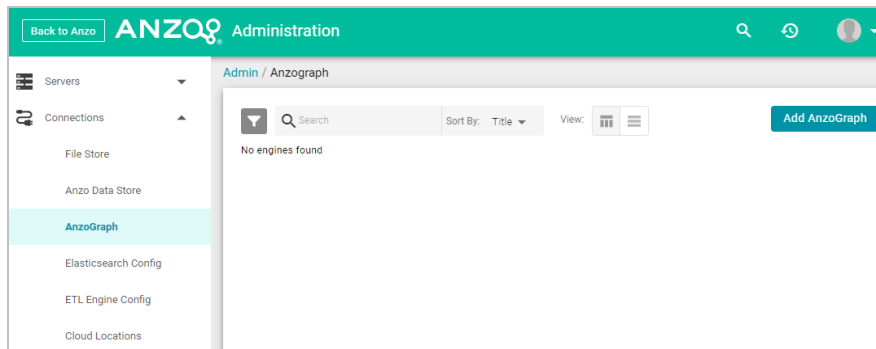
Connecting to AnzoGraph

This topic provides instructions for configuring the connection to AnzoGraph. For information about managing AnzoGraph servers, see [AnzoGraph Server Administration](#).

Important

Do not connect multiple Anzo instances to the same AnzoGraph instance. Since AnzoGraph is stateless and Anzo manages all of the data, connecting more than one Anzo instance to the same AnzoGraph instance causes severe data management conflicts that result in unexpected behavior. This type of configuration is not supported.

1. In the Administration application, expand the **Connections** menu and click **AnzoGraph**. Anzo opens the AnzoGraph connection overview screen, which lists any existing connections. For example:



2. On the AnzoGraph screen, click **Add AnzoGraph** to add a connection. Anzo displays the Create AnzoGraph dialog box.

3. On the **Basic** tab, type a name for the engine in the **Title** field.
4. In the optional **Description** field, type a description for the graph query engine. If you leave this field blank, Anzo creates a description when you save the configuration.
5. In the **Host** field, type the AnzoGraph server host name or IP address. If you have a cluster, type the name or IP address of the leader server.
6. In the **AnzoGraph User** field, type the username that was created when AnzoGraph was installed.
7. Type the password for the AnzoGraph user in the **AnzoGraph Password** and **Confirm Password** fields.
8. If this AnzoGraph instance will host data from unstructured pipelines, click the **Elasticsearch Configuration** drop-down list and select the Elasticsearch instance to associate with this AnzoGraph connection. For information about configuring an Elasticsearch connection, see [Connecting to Elasticsearch](#).
9. Click **Test Connection** to check if Anzo can connect to AnzoGraph. If the connection fails, make sure that AnzoGraph is running and that you typed the correct username and password.

10. **Optional:** Click the **Advanced** tab and configure any of the optional advanced settings. For example:

The screenshot shows the 'Create AnzoGraph' configuration window with the following settings:

- Instance URI:** (Empty text field)
- Trust All TLS Certificates:** ☒
- AnzoGraph Concurrent Queries:** 10
- AnzoGraph connection timeout (seconds):** 60
- Use AnzoGraph persistence if available:** ☒
- Force reload of Graphmart data during Anzo startup or when Datasource enabled:** ☒
- Keep AnzoGraph Datasource enabled on Anzo startup:** ☒
- Port:** 5700
- AnzoGraph Management Port:** 5600
- Callback HostName:** (Empty text field)
- Readonly Replica:** ☐
- Vacuum:** ☒
- Gather Statistics on Load:** ☒
- Use Priority Queue Query Manager:** ☒
- Enable Detailed Query Timing:** ☐
- Max allowed duration for system operations (Minutes):** 2
- Max allowed duration for queries (ex: 3d12h, 1h, 20m):** (Empty text field)

At the bottom, there are buttons for 'TEST CONNECTION', 'CANCEL', and 'SAVE'.

The list below describes each of the advanced settings:

- **Instance URI:** The URI for this AnzoGraph instance. Anzo automatically assigns an instance URI. If you specify a custom URI, make sure that the URI is valid and unique.
- **Trust All TLS Certificates:** Indicates whether Anzo should trust the AnzoGraph certificates for this connection. Cambridge Semantics recommends that you accept the default value of enabled.
- **AnzoGraph Concurrent Queries:** The maximum number of queries that Anzo can send to AnzoGraph concurrently. The default value is **10** queries. Cambridge Semantics recommends that you accept the default value. If you want to increase the number of concurrent queries, Cambridge Semantics recommends that you choose a value between 10 and 20.
- **AnzoGraph Connection Timeout (seconds):** This setting controls how often (in seconds) Anzo checks the status of the connection to this AnzoGraph instance. The connection is tested every N seconds, where N is the value of this setting. The default value is **60**. If the test fails, Anzo re-tests the connection every 15 seconds for 2 minutes to rule out a brief network glitch. If the connection continues to fail after 2 minutes, the status is changed to "Offline." If the connection is re-established within the 2-minute window, Anzo determines whether the connection came back automatically or whether AnzoGraph was restarted.
- **Use AnzoGraph Persistence if Available:** This setting controls how Anzo manages graphmart data if persistence is enabled for this data source and AnzoGraph is restarted.

Note

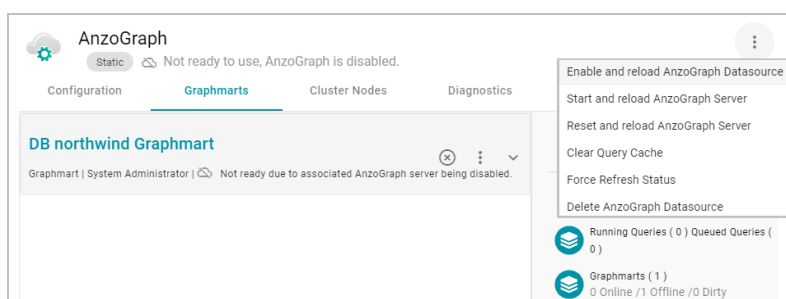
The **Use AnzoGraph Persistence if Available** setting is enabled by default but persistence is disabled for AnzoGraph by default. For information about how Anzo manages the data when persistence is enabled and for instructions on enabling persistence, see [Using AnzoGraph Persistence \(Preview\)](#).

- **Force Reload of Graphmart Data During Anzo Startup or when Datasource Enabled:** This option is enabled by default and means that Anzo forces a reload of active graphmarts when one of the following actions occur: 1. Anzo restarts and reconnects to AnzoGraph, or 2. Anzo restarts and a user manually re-enables this data source by selecting **Enable and reload AnzoGraph Datasource** from the menu on the AnzoGraph administration screen. When this option is disabled and AnzoGraph persistence is also disabled, graphmarts must be reloaded by clicking the **Reset and Reload all Graphmarts** button on the AnzoGraph screen after the connection is re-established due to an AnzoGraph restart.

Note

If AnzoGraph persistence is enabled and **Force reload of Graphmart data...** is disabled, Anzo may force a reload if the last updated timestamp in AnzoGraph does not match the last updated value in Anzo.

- **Keep AnzoGraph Datasource Enabled on Anzo Startup:** This option is enabled by default and means that Anzo leaves the AnzoGraph data source online in a "Ready to use" state if Anzo is restarted (if this data source is online at the time Anzo is restarted). When this option is disabled, Anzo disables this data source when Anzo is restarted. When Anzo comes online, this source must be manually enabled by selecting **Enable and reload AnzoGraph Datasource** from the menu on the AnzoGraph administration screen. For example:



- **Port:** The port to use for communication between AnzoGraph and Anzo. The default value is **5700**, the Anzo protocol (gRPC) port for secure communication. Do not change the value unless instructed by Cambridge Semantics Support.
- **AnzoGraph Management Port:** The SSL system management port for AnzoGraph. The default value is **5600**. Do not change the value unless instructed by Cambridge Semantics Support.

- **Callback Hostname:** The Callback Hostname is the Anzo server to use when AnzoGraph makes service callbacks. If you have multiple Anzo servers and one or more of them are not routable by the AnzoGraph server, the Callback Hostname is the Anzo host that AnzoGraph can target when making service calls.
- **Readonly Replica:** This option is for use if you have multiple Anzo servers, and only one of those servers loads graphmarts to AnzoGraph. When Is Replica is selected, Anzo treats this AnzoGraph as a read-only source so that this Anzo server can view the data in AnzoGraph but cannot change it.
- **Vacuum:** This option controls whether Anzo initiates an AnzoGraph vacuum process after each data load. The vacuum process improves data organization in memory, deduplicates data, and reclaims memory after data is deleted. Completing a vacuum after update operations is extremely important for maintaining overall query performance and memory allocation accuracy.

Note

Do not disable vacuum unless you are instructed to do so by Cambridge Semantics Support.

- **Gather Statistics on Load:** This option controls whether Anzo initiates AnzoGraph's internal statistics gathering queries after loading data. Gathering statistics helps the query planner generate ideal query execution plans when queries are run. When this option is enabled, the AnzoGraph statistics queries are run immediately after a Graphmart is loaded. It increases Graphmart load time but reduces execution time for the first analytic queries, such as when a Hi-Res Analytic dashboard is created. When this option is disabled (the checkbox is clear), AnzoGraph automatically performs statistics gathering when the first queries are run, increasing the execution time for the initial queries.

Note

Cambridge Semantics recommends that you leave Gather Statistics on Load enabled so that AnzoGraph gathers statistics at the end of a load rather than during query execution. Since loads take longer than queries, adding more time to the load is less noticeable than waiting for statistics to be generated during initial query execution.

- **Use Priority Queue Query Manager:** This option controls whether Anzo provides a view of the queries that are in the queue waiting to be run. The queued queries are displayed in the System Query Audit log.

Note

Enabling or disabling this option after saving the initial configuration requires a restart of Anzo.

- **Enable Detailed Query Timing:** When the Priority Queue Query Manager is enabled, this option controls whether Anzo obtains detailed timing statistics for every AnzoGraph query. If this option is enabled, Anzo sends additional statistics gathering queries to AnzoGraph for each user query. The extra query timing details, such as query compilation time, compilation statistics, and a query summary, are displayed in the

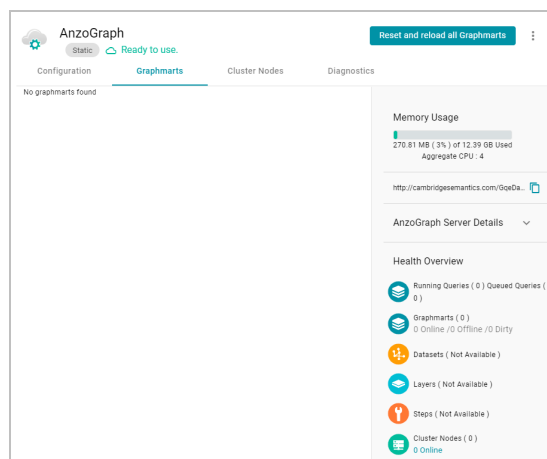
System Query Audit log. For more information about this setting, see [AnzoGraph Detailed Query Timing Reference](#).

Important

Enabling detailed query timing increases the AnzoGraph workload and may decrease overall query performance.

- **Max Allowed Duration for System Operations (Minutes):** This option sets a limit on the number of minutes Anzo waits for AnzoGraph to complete system operation related queries, such as queries for CPU and memory usage statistics. The default value is **2** minutes. If Anzo is waiting on system information from AnzoGraph and AnzoGraph does not respond within the specified time, Anzo cancels the request.
- **Max Allowed Duration for Queries:** This option sets a limit on the amount of time that Anzo waits for AnzoGraph to complete a user query (such as dashboard, data layer, or Query Builder queries). By default, Anzo waits indefinitely. To set a maximum duration, specify the amount of time in any combination of days, hours, and minutes. For example, specifying **1d** sets the maximum duration to one day. Specifying **10h**, sets the maximum duration to 10 hours, and specifying **1d12h30m** sets the duration to 1 day, 12 hours, and 30 minutes. If **Max Allowed Duration for Queries** is set and a query does not complete in the specified time, Anzo cancels the request regardless of whether AnzoGraph has returned partial results.

11. Click **Save** to save the configuration. Anzo connects to AnzoGraph and opens the Graphmarts tab. For example:



To change configuration details, click the **Configuration** tab and adjust values as needed. The right side of the screen shows connection status as well as memory usage details, overall data statistics, and graphmart details. For information about loading data to AnzoGraph, see [Creating a New Graphmart](#).

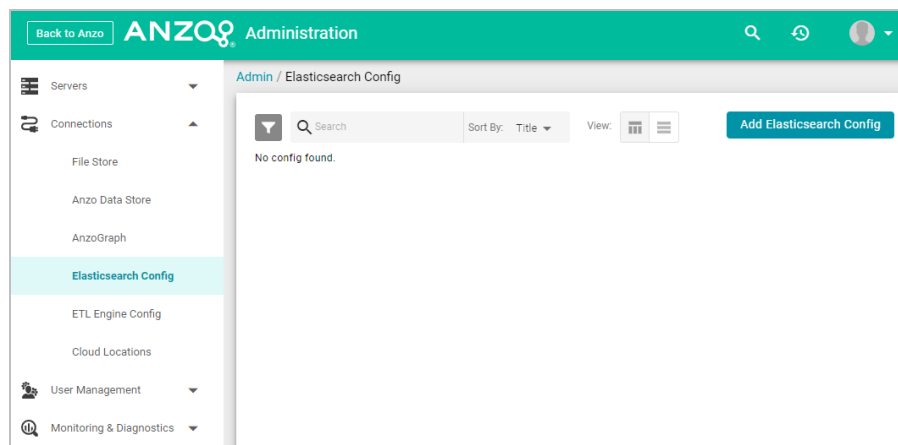
Related Topics

[AnzoGraph Server Administration](#)

Connecting to Elasticsearch

This topic provides instructions for configuring a connection to an Elasticsearch instance in the Administration application. For information about installing Elasticsearch, see [Installing and Configuring Elasticsearch](#).

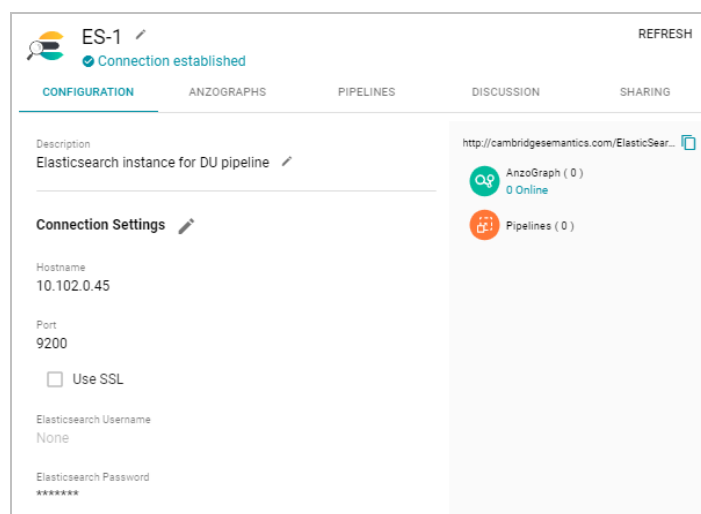
1. In the Administration application, expand the **Connections** menu and click **Elasticsearch Config**. Anzo displays the Elasticsearch Config screen, which lists any existing Elasticsearch connections. For example:



2. On the Elasticsearch Config screen, click the **Add Elasticsearch Config** button. Anzo opens the Create Elasticsearch Config dialog box.

3. On the Create Elasticsearch Config screen, provide the following details about the Elasticsearch instance:
 - **Title:** Type a name for this Elasticsearch connection.
 - **Description:** Optional description for this connection.
 - **Hostname:** Specify the IP address or hostname of the Elasticsearch server.
 - **Port:** Specify the port to use for the Elasticsearch connection. The default Elasticsearch port is **9200**.

- **Trust All Certs:** Indicates whether Anzo should trust the Elasticsearch certificates for this connection. Cambridge Semantics recommends that you accept the default value of enabled.
 - **Use SSL:** If this Elasticsearch instance is configured for SSL authentication, select the **Use SSL** checkbox.
 - **Elasticsearch Username:** If Use SSL is specified, type the user name to use to connect to Elasticsearch.
 - **Elasticsearch Password:** If Use SSL is specified, type the password for the user name that you specified.
4. Click **Test Connection** to check if Anzo can connect to Elasticsearch. If the connection fails, make sure that Elasticsearch is running and that you entered the correct connection details.
 5. Anzo displays a Connection Successful dialog box. Click **OK** to close the dialog, and then click **Save** to save the new connection. Anzo saves the connection and displays the Configuration overview screen. For example:



You can adjust configuration details as needed. For instructions on creating an unstructured pipeline, see [Onboarding Unstructured Data](#).

Related Topics

[Onboarding Unstructured Data](#)

Connecting to an ETL Engine

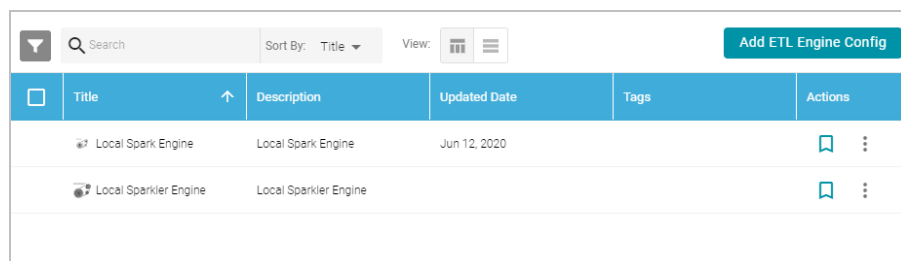
The default Anzo installation includes a pre-configured local Spark ETL engine and Sparkler ETL compiler. Sparkler is Cambridge Semantics' Spark SPARQL interpreter. The Sparkler interpreter expresses Spark ingestion jobs as SPARQL, which adds benefits such as support for ingesting wide CSV files with a large number of columns. The topics in this section provide instructions for changing the configuration of the local engines or connecting to an alternate Spark ETL engine or Sparkler compiler.

- [Configuring a Spark ETL Engine](#)
- [Configuring a Sparkler Engine](#)
- [Limiting Job Concurrency on a Remote Sparkler Engine](#)

Configuring a Spark ETL Engine

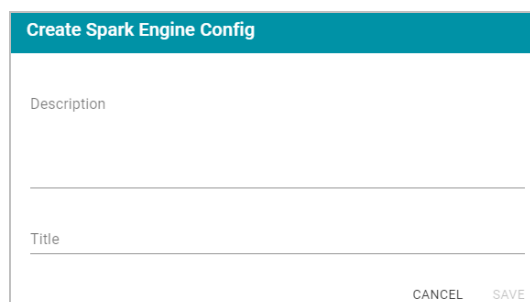
This topic provides instructions for configuring a connection to a Spark ETL engine.

1. In the Administration application, expand the **Connections** menu and click **ETL Engine Config**. Anzo displays the ETL Engine Config screen, which lists existing ETL engine connections. For example:



<input type="checkbox"/>	Title	Description	Updated Date	Tags	Actions
	Local Spark Engine	Local Spark Engine	Jun 12, 2020		
	Local Sparkler Engine	Local Sparkler Engine			

2. On the ETL Engine Config screen, click the **Add ETL Engine Config** button and select **Spark Engine Config**. Anzo displays the Create Spark Engine Config screen.



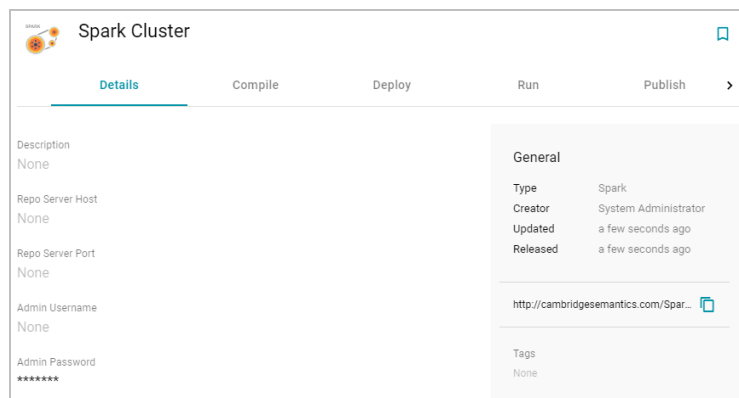
Create Spark Engine Config

Description

Title

CANCEL SAVE

3. On the Create screen, type a **Title** and optional **Description** for the engine. Then click **Save**. Anzo displays the Details view for the new engine. For example:



Spark Cluster

Details | Compile | Deploy | Run | Publish >

Description
None

Repo Server Host
None

Repo Server Port
None

Admin Username
None

Admin Password

General

Type: Spark

Creator: System Administrator

Updated: a few seconds ago

Released: a few seconds ago

URL: <http://cambridgesemantics.com/Spar...>

Tags: None

4. Configure the engine by completing the required fields and adding any optional values on the Details, Compile, Deploy and Run tabs. To edit a field, click a value to make the field editable or click the edit icon (). Click the check mark icon () to save changes to an option, or click the X icon () to clear the value for an option. See the [Spark Settings Reference](#) section below for descriptions of the settings.

Spark Settings Reference

This section provides reference information for the Spark ETL engine settings on each of the tabs.

Details Tab

- **Repo Server Host:** Leave this field blank.
- **Repo Server Port:** Leave this field blank.
- **Admin Username:** Not currently used.
- **Admin Password:** Not currently used.

Compile Tab

The Compile settings control where Anzo saves the compiled Scala .jar files for the Spark job.

- **Remote Server:** The host name or IP address of the server where the compilation will be performed.
- **Target Folder:** The path and directory on the server where Anzo can stage temporary artifacts created during the compilation and upload process. The location must be a valid path on the Anzo server that the user running the ETL job has access to.

Deploy Tab

The Deploy step is performed after the job is compiled locally and before the job is submitted to Spark. The Deploy settings control how and where the job's .jar files will be copied from the Anzo server to a file system that Spark can access.

- **Deployment Working Dir:** The directory that the Anzo server should use when executing the deploy commands.
- **Deploy Command:** The command line script that the deploy step should run.

Run Tab

- **Job Runner Endpoint:** The HTTP endpoint used to reach the Livy server. For example, when using the local Anzo Spark engine, the endpoint is localhost:8998.
- **SDI Jobs Dir:** The file system location where the Spark engine will look for the compiled .jar files. This field is required when working with a remote Spark server. It can be left blank when using the local Spark engine.
- **SDI Dependencies Dir:** The file system location where the Spark engine will look for the dependency .jar files, **sdi-full-deps.jar** and **sdi-deps.jar**. If you are using a remote Spark cluster, sdi-full-deps.jar and sdi-deps.jar can be copied to the Spark master node from the `<install_path>/Server/data/sdiScripts/<Spark_version>/compile/dependencies-lib` directory on the Anzo server.
- **Additional Jars:** For relational database sources, this field lists the file system location for the JDBC driver .jar file or files that are used to connect to the source. All paths must be absolute. For multiple jar files, specify a comma-separated list. Do not include a space after the commas.

For RDBs whose drivers are installed with Anzo, such as MSSQL (com.springsource.net.sourceforge.jtds_1.2.2.jar), Oracle (oracle.jdbc_11.2.0.3.jar), Amazon Redshift (org.postgresql.osgi.redshift_9.3.702.jar), and PostgreSQL (com.springsource.org.postgresql.jdbc3_8.3.603.jar), you can find the driver jar files in the `<install_path>/Server/plugins` directory.

- If you use the local Spark ETL engine, the Additional Jars field should list the path to the jar files in the Anzo plugins directory. For example, `/opt/Anzo/Server/plugins/org.postgresql.osgi.redshift_9.3.702.jar`.
- If you use a remote Spark cluster in **cluster mode**, the driver jar files need to be copied onto the HDFS. If Spark is running in **client mode**, jar files can be copied to the Hadoop/Spark master node file system. Specify the path to the copied jar files in the Additional Jars field.

Note

If a driver is uploaded to Anzo as described in [Uploading a Plugin](#), the driver will be in the `<install_path>/Server/dropins` directory. For example, `/opt/Anzo/Server/dropins/com.springsource.com.mysql.jdbc-5.1.6.jar`

- **Execute Locally:** Select this option for local Spark engines on the Anzo server. Make sure this option is not selected when using a remote Spark server.
- **Do Callback:** Select this option when you want Anzo to create a new data set in the Dataset catalog and generate load files for the graph source.
- **Run with Yarn:** Employs the Spark YARN cluster manager when running ETL jobs.
- **Callback URL:** When **Do Callback** is selected, enter one of the following URLs:

```
http://Anzo_hostname_or_IP:Anzo_app_HTTP_port/anzoclient/call
```

```
https://Anzo_hostname_or_IP:Anzo_app_HTTPS_port/anzoclient/call
```

For example:

```
https://10.100.0.1:8443/anzoclient/call
```

Publish Tab

The Publish tab controls the action of the **Publish All** button when a pipeline is published.

Sharing Tab

The Sharing tab enables you to share or restrict access to this ETL engine.

When the configuration is complete, Anzo provides this ETL engine as a choice to select when ingesting data and configuring pipelines. If you want to specify the default ETL engine to use automatically any time a pipeline is configured, see [Configure the Default ETL Engine](#).

Related Topics

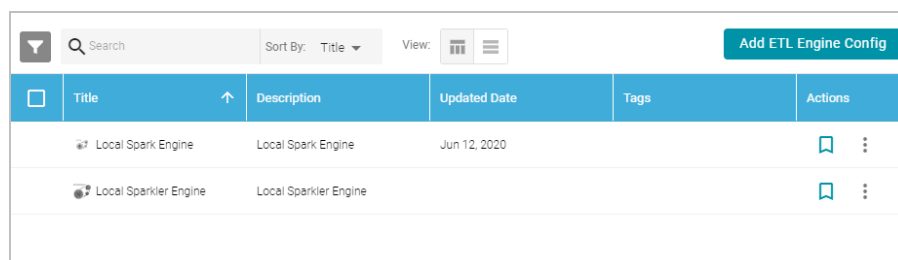
[Configuring a Sparkler Engine](#)

[Configure the Default ETL Engine](#)

Configuring a Sparkler Engine

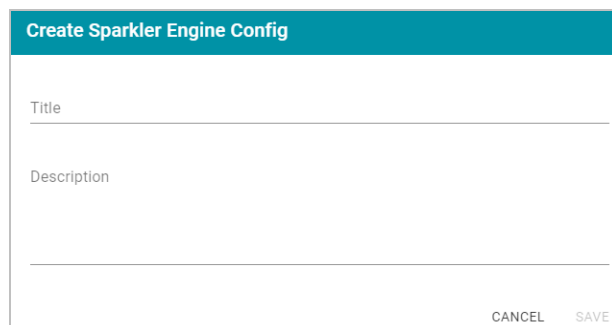
This topic provides instructions for configuring a connection to a Sparkler compiler. Sparkler is Cambridge Semantics' Spark SPARQL interpreter. Sparkler expresses Spark ingestion jobs as SPARQL, and Sparkler jobs are executed by Spark. They are submitted to Spark using Livy interactive sessions.

1. In the Administration application, expand the **Connections** menu and click **ETL Engine Config**. Anzo displays the ETL Engine Config screen, which lists existing ETL engine connections. For example:



<input type="checkbox"/>	Title	Description	Updated Date	Tags	Actions
	Local Spark Engine	Local Spark Engine	Jun 12, 2020		
	Local Sparkler Engine	Local Sparkler Engine			

2. On the ETL Engine Config screen, click the **Add ETL Engine Config** button and select **Sparkler Engine Config**. Anzo displays the Create Sparkler Engine Config screen.



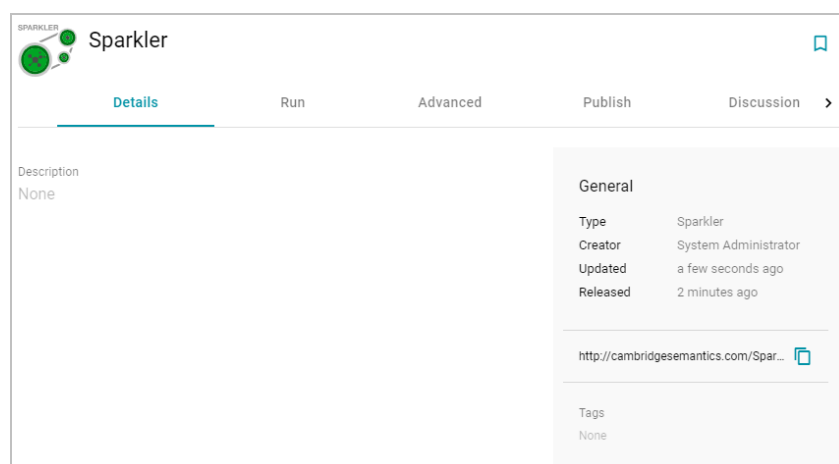
Create Sparkler Engine Config

Title

Description

CANCEL SAVE

3. On the Create screen, type a **Title** and optional **Description** for the engine. Then click **Save**. Anzo displays the Details view for the new engine. For example:



4. Configure the engine by completing the required fields and adding any optional values on the Run, Advanced, and Publish tabs. To edit a field, click a value to make the field editable or click the edit icon (✎). Click the check mark icon (✓) to save changes to an option, or click the X icon (✕) to clear the value for an option. See the [Sparkler Settings Reference](#) section below for descriptions of the settings.

Sparkler Settings Reference

This section provides reference information for the Sparkler ETL engine settings on each of the tabs.

Run Tab

- **Remote Server Name:** The host name or IP address of the server where the compilation will be performed.
- **Job Runner Endpoint:** The HTTP endpoint used to reach the Livy server. For example, when using the local Anzo Sparkler engine, the endpoint is localhost:8998.
- **Target Folder Name:** The path and directory on the host where temporary artifacts can be created during the compilation and upload process. The location must be a valid path on the server that the user running the ETL job has access to.
- **Sparkler Home:** The path and directory where the Sparkler compiler is installed on the host server.
- **SDI Dependencies Dir:** The file system location where the Spark engine will look for the dependency .jar files, **sdi-full-deps.jar** and **sdi-deps.jar**. If you are using a remote Spark cluster, sdi-full-deps.jar and sdi-deps.jar can be copied to the Spark master node from the `<install_path>/Server/data/sdiScripts/<Spark_version>/compile/dependencies-lib` directory on the Anzo server.
- **Additional Jars:** For relational database sources, this field lists the file system location for the JDBC driver .jar file or files that are used to connect to the source. All paths must be absolute. For multiple jar files, specify a comma-separated list. Do not include a space after the commas.

For RDBs whose drivers are installed with Anzo, such as MSSQL (com.springsource.net.sourceforge.jtds_1.2.2.jar), Oracle (oracle.jdbc_11.2.0.3.jar), Amazon Redshift (org.postgresql.osgi.redshift_9.3.702.jar), and PostgreSQL (com.springsource.org.postgresql.jdbc3_8.3.603.jar), you can find the driver jar files in the `<install_path>/Server/plugins` directory.

- If you use the local Sparkler ETL engine, specifying the path to drivers in the Additional Jars field is not mandatory. Anzo will automatically locate the drivers in the plugins directory. If you do list the path to the jar files, specify the path to the Anzo plugins directory. For example, `/opt/Anzo/Server/plugins/org.postgresql.osgi.redshift_9.3.702.jar`.
- If you use a remote Spark cluster in **cluster mode**, the driver jar files need to be copied onto the HDFS. If Spark is running in **client mode**, jar files can be copied to the Hadoop/Spark master node file system. Specify the path to the copied jar files in the Additional Jars field.

Note

If a driver is uploaded to Anzo as described in [Uploading a Plugin](#), the driver will be in the `<install_path>/Server/dropins` directory. For example, `/opt/Anzo/Server/dropins/com.springsource.com.mysql.jdbc-5.1.6.jar`

- **Execute Locally:** Select this option for local Sparkler engines on the Anzo server. Make sure this option is not selected when using a remote Sparkler server.
- **Do Callback:** Select this option when you want Anzo to create a new data set in the Dataset catalog and generate load files for the graph source.
- **Run with Yarn:** Employs the Spark YARN cluster manager when running ETL jobs.
- **Callback URL:** When **Do Callback** is selected, enter one of the following URLs:

```
http://Anzo_hostname_or_IP:Anzo_app_HTTP_port/anzoclient/call
```

```
https://Anzo_hostname_or_IP:Anzo_app_HTTPS_port/anzoclient/call
```

For example:

```
https://10.100.0.1:8443/anzoclient/call
```

Advanced Tab

The options on this tab enable users with advanced Spark expertise to customize the values that are passed to Spark.

- **Enable CSV Error Reporting:** Controls whether detailed CSV errors are displayed in the Anzo user interface.
- **Input Database Partition Default:** By default, Sparkler attempts to partition relational database tables if the table has a primary column with an integer data type and the source data has been profiled as described in [Generating a Source Data Profile](#). When **Input Database Partition Default** is enabled, Sparkler attempts to partition RDBMS tables when they have a primary column with an integer type even if a data source profile has not been generated.
- **Enable Hive Context (Enable in Livy Conf for Spark 2):** Controls Hive context for Spark version 1.6. Selecting this setting enables the Hive context for Spark 1.6.

- **Redirect Graph Output to Hive:** Controls whether the ETL process writes data to Hive or a file-based linked data set (FLDS). When this option is disabled (the default configuration) data is written to an FLDS that can be added to a graphmart and loaded to AnzoGraph. When this option is enabled, the ETL process writes data to Hive rather than creating an FLDS.
- **Run As User:** Specifies the user to impersonate when starting the Livy session.
- **Max Graph Output File Size Default (Bytes):** The maximum number of bytes to limit graph output files to.
- **Max Input File Partition Size (Bytes):** The maximum number of bytes to pack into a partition when reading files. Maps to the `spark.files.maxPartitionBytes` Spark configuration setting.
- **Spark Job Driver Cores:** The number of cores to use for the driver process. Maps to the `spark.driver-cores` Spark configuration setting.
- **Spark Job Driver Memory:** The amount of memory to use for the driver process. Maps to the `spark.driver-memory` Spark configuration setting.
- **Number of Executors Per Spark Job:** The number of executors to request per Spark job. Maps to the `spark.executor.instances` Spark configuration setting.
- **Spark Job Cores Per Executor:** The number of cores to use on each executor. Maps to the `spark.executor.cores` Spark configuration setting.
- **Spark Job Memory Per Executor:** The amount of memory to use per executor process. Maps to the `spark.executor.memory` Spark configuration setting.
- **Off Heap Size (Bytes):** The amount of memory in bytes that can be used for off-heap allocation. Maps to the `spark.memory.offHeap.size` Spark configuration setting.
- **Job Dependencies (Maven Package Coordinate):** The comma-separated list of Maven jar coordinates to include on the driver and executor classpaths. Maps to the `spark.jars.packages` Spark configuration setting.
- **Maven Package Excludes:** To avoid dependency conflicts, this is the comma-separated list of `groupId:artifactId` to exclude while resolving the dependencies listed in `spark.jars.packages`. Maps to the `spark.jars.excludes` Spark configuration setting.
- **Maven Repositories:** A comma-separated list of additional remote repositories to search for the maven coordinates from the Job Dependencies setting. Maps to the `spark.jars.repositories` Spark configuration setting.
- **Spark Job Deploy Mode (Livy Config has Precedence):** The deploy mode of the Spark driver program. If this value is set in the Livy configuration, the Livy value takes precedence. Maps to the `spark.submit.deployMode` Spark configuration setting.

Publish Tab

The Publish tab controls the action of the **Publish All** button when a pipeline is published.

Sharing Tab

The Sharing tab enables you to share or restrict access to this ETL engine.

When the configuration is complete, Anzo provides this ETL engine as a choice to select when ingesting data and configuring pipelines. If you want to specify the default ETL engine to use automatically any time a pipeline is configured, see [Configure the Default ETL Engine](#).

Related Topics

[Limiting Job Concurrency on a Remote Sparkler Engine](#)

[Configuring a Spark ETL Engine](#)

[Configure the Default ETL Engine](#)

Limiting Job Concurrency on a Remote Sparkler Engine

When compiling ETL jobs on a remote Sparkler engine, all jobs are executed simultaneously. For pipelines with more than 110 jobs, running all jobs concurrently can consume all of ports in the default port range and cause the pipeline to fail. To limit the number of jobs that can be executed concurrently on a remote Spark cluster with Sparkler, you can add a configuration file to the cluster and specify the maximum number of jobs that can be executed at the same time. When the number of jobs exceeds the limit, additional jobs are queued and then executed as resources are freed. Follow the instructions below to configure the limit.

1. If necessary, run the following command to stop the remote Sparkler server:

```
./<install_path>/sparkler/bin/sparkler-server stop
```

2. The Anzo embedded Sparkler engine includes a configuration file template, **application.conf.template**, that you can copy to the remote cluster. If needed, you can retrieve **application.conf.template** from the following directory on the Anzo server:

```
<install_path>/Server/data/sdiScripts/spark-2.2/compile/dependencies-lib/sparkler/conf
```

3. Rename **application.conf.template** to **application.conf** and place **application.conf** in the `<install_path>/sparkler/conf/` directory on the remote cluster.
4. Open **application.conf** in an editor. At the top of the file under server options, change the value for **maxActiveJobs** to the maximum number of jobs that you want Sparkler to execute concurrently. The setting and default value are shown in bold below:

```
server {
  actorSystemName = "SparklerServerSystem"
  actorName = "SparklerJobActor"
  retryDelay = "3 seconds"
  maxRetries = 5
  maxActiveJobs = 1
  ...
}
```

5. Save and close application.conf, and then run the following command to restart the Sparkler server:

```
./<install_path>/sparkler/bin/sparkler-server start
```

Related Topics

[Configuring a Sparkler Engine](#)

[Configuring a Spark ETL Engine](#)

Connecting to a Cloud Location

A Cloud Location is a connection between Anzo and the Kubernetes (K8s) cluster that will host the dynamic Anzo Agent and Anzo Unstructured, AnzoGraph, Spark, and Elasticsearch applications. When you create a Cloud Location, Anzo discovers the K8s cluster and any internal container registries, authenticates the K8s API services, obtains the node pool or group specifications and retrieves pricing information from the Cloud Service Provider for the configured compute instances, and maps the node pool specifications to Launch Configurations in Anzo.

Tip

For instructions on deploying the K8s infrastructure to support Cloud Locations, see [Using K8s for Dynamic Deployments of Anzo Components](#).

The topics in this section provide instructions on setting up the NFS configuration for the dynamically deployed applications and creating a Cloud Location.

- [Importing the NFS Configuration](#)
- [Creating a Cloud Location](#)

Importing the NFS Configuration

Before creating a Cloud Location in the Administration application, the configuration details for the NFS server need to be imported into Anzo. This is a one-time procedure; the configuration that you import is used for all Cloud Locations. Anzo will automatically mount this NFS server to any nodes that are provisioned when applications are deployed.

Tip For information about the NFS requirements, see [NFS Guidelines](#).

Create the NFS Configuration File

The NFS configuration details need to be specified in TriG format. The TriG file is imported to Anzo using the Anzo Admin CLI. Use the following contents as a template to create a .trig file on the Anzo server. The objects to supply values for are described below:

```
@prefix : <http://cambridgesemantics.com/ontologies/cloud/deployment/config#> .
@prefix nfsmountconfig:
<http://cambridgesemantics.com/ontologies/CloudDeployment/NFSMountConfiguration/> .
```

```

@prefix deployment: <http://cambridgesemantics.com/ontologies/CloudDeployment/> .
@prefix anzo: <http://openanzo.org/ontologies/2008/07/Anzo#> .
@prefix int: <http://openanzo.org/system/internal/> .
@prefix role: <http://openanzo.org/Role/> .

#Mode:REPLACE
:nfsMountConfig1
{
  :nfsMountConfig1 a deployment:NFSMountConfiguration, deployment:MountConfiguration;
  nfsmountconfig:NFSfqdn "NFSfqdn" ;
  nfsmountconfig:NFSMountDir "NFSMountDir" ;
  nfsmountconfig:NFSMountOptions "NFSMountOptions" ;
  nfsmountconfig:NFSSharedDir "NFSSharedDir" .
}

```

NFSfqdn

The IP address for the NFS server.

NFSMountDir

The NFS mount location on the Anzo server. The same mount location will be used to mount the NFS when dynamic resources are provisioned.

NFSMountOptions

The mount options to use when mounting the NFS.

NFSSharedDir

The NFS directory to share between Anzo and the dynamic resources.

For example:

```

# nfs-config.trig
@prefix : <http://cambridgesemantics.com/ontologies/cloud/deployment/config#> .
@prefix nfsmountconfig:
<http://cambridgesemantics.com/ontologies/CloudDeployment/NFSMountConfiguration/> .
@prefix deployment: <http://cambridgesemantics.com/ontologies/CloudDeployment/> .
@prefix anzo: <http://openanzo.org/ontologies/2008/07/Anzo#> .
@prefix int: <http://openanzo.org/system/internal/> .
@prefix role: <http://openanzo.org/Role/> .

#Mode:REPLACE
:nfsMountConfig1
{
  :nfsMountConfig1 a deployment:NFSMountConfiguration, deployment:MountConfiguration;
  nfsmountconfig:isTransferFiles false ;
  nfsmountconfig:NFSfqdn "10.104.0.6" ;
}

```

```
nfsmountconfig:NFSMountDir "/private/var/nfsshare_dev" ;  
nfsmountconfig:NFSMountOptions "hard,nfsvers=4.1" ;  
nfsmountconfig:NFSSharedDir "/global/nfs/data" .  
}
```

Import the NFS Configuration to Anzo

Once the NFS configuration file is created, run the following command to import the file to Anzo with the Anzo Admin CLI:

```
<install_path>/Client/anzo <file_path>/<filename>.trig -u sysadmin --useModes
```

For example:

```
/opt/Anzo/Client/anzo import nfs-config.trig -u sysadmin --useModes
```

When the NFS configuration details have been imported to Anzo, see [Creating a Cloud Location](#) for instructions on creating a Cloud Location.

Related Topics

[Creating a Cloud Location](#)

Creating a Cloud Location

Follow the instructions below to create a Cloud Location. Note that the steps below are in progress and more details are forthcoming.

1. In the Administration application, expand the **Connections** menu and click **Cloud Locations**.
2. On the Cloud Locations screen, click the **Add Cloud Location** button and select the Cloud Service Provider that hosts your Kubernetes (K8s) cluster. The Create Cloud Location dialog box is displayed. For example, the image below shows the Create Cloud Location screen for Google:

Create Cloud Location

You're creating a **Google** cloud location.

Title *

Description

Please select a Google credentials file (or use the default credentials) and click 'Test' to validate. The credentials will be shown when successfully validated.

Browse Credential File

☐ Use default credentials

Kubernetes Cluster

Kubernetes Namespace

Note: You can choose from the list or key in a custom namespace (Must consist of lower case alphanumeric characters or '-'. Must start and end with an alphanumeric character. Ex. 'my-name' or '123-abc').

Kubernetes Docker Registry

Operators Support

Anzo Operator x AnzoGraph Operator x Elasticsearch Operator x Unstructured Operator x

Spark Engine Operator (Livy Deployment for Spark Config) x

CANCEL

DEPLOY

- At the top of the screen, specify a **Title** for this Cloud Location and type an optional **Description**.
- Next, specify the credentials that have permission to connect to the Cloud Service Provider (CSP) API and deploy resources in the Kubernetes cluster. There are two options, depending on the user that is running Anzo and the Service Account, Principal, or Group that was assigned the K8s Cluster Developer IAM policy when the K8s infrastructure was set up:
 - Since the Anzo Service Account, Principal, or Group is typically running Anzo, and the K8s Cluster Developer IAM policy was assigned to that account when the K8s infrastructure was set up, the appropriate credentials are already applied to this Anzo instance. In this case, select the **Use Default Credentials** checkbox. The dialog box indicates that the default instance credentials will be used and presents a **Test** button (shown in the image below).

Please select a Google credentials file (or use the default credentials) and click 'Test' to validate. The credentials will be shown when successfully validated.

Browse Credential File

☒ Use default credentials

Default

Use default instance credentials

Test

Click **Test** to retrieve the credentials and test that they are valid.

- If another user is running Anzo, and that account does not have the Cluster Developer IAM permissions, retrieve from your CSP the JSON configuration file for the account that is assigned the Cluster Developer IAM policy. Then click the **Browse Credential File** button and upload the JSON credentials file that you downloaded.

User Management

Anzo offers granular artifact and data access control as well as role-based security for controlling access to the Anzo applications and features. This section provides setup and administration information for role-based access control. The topics include instructions for connecting to your central directory server, connecting to an identity provider for SSO access, and configuring users, groups, roles, and permissions in Anzo.

Tip

When planning the user and access management solution for your system, Cambridge Semantics recommends that you refer to [User Management and Access Control Concepts](#) to learn about the fundamental concepts behind Anzo's access control implementation.

- [User Management and Access Control Concepts](#)
- [Connecting to a Directory Server](#)
- [Adding Directory Users and Groups to Anzo](#)
- [Enabling Self-Authorization for Directory Users](#)
- [Connecting to an SSO Provider](#)
- [Creating and Managing Roles](#)
- [Creating an Internal Anzo User](#)
- [Predefined Anzo Roles and Permissions](#)
- [Role Permissions Reference](#)
- [Managing Default Access Policies](#)

User Management and Access Control Concepts

The topics in this section provide an overview of user management and access control in Anzo and introduce the key concepts to consider when planning and implementing user and data access management for your system.

- [User Management Concepts](#)
- [Artifact Access Control Concepts](#)

User Management Concepts

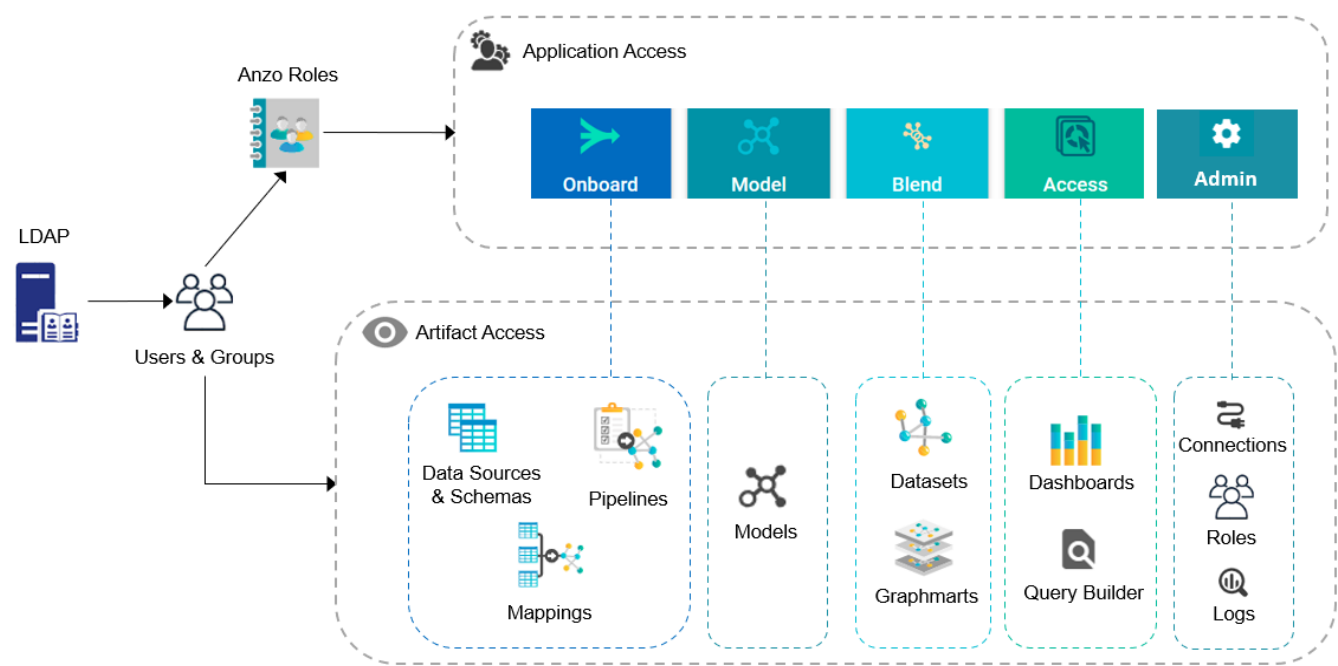
Typically organizations connect Anzo to their central directory server and then add users and groups from the server to Anzo. Once the accounts are added to Anzo, access control is managed in two ways:

1. Groups (or users) are added to **roles** and the roles are configured to grant access to *functionality* in Anzo. Role permissions grant access to menus and screens in the Anzo and Administration applications. Access to functionality cannot be assigned to groups or users, only to roles.
2. Groups and users are used to control access to individual artifacts—Project's Data Sources, Models, Mappings, Pipelines, Graphmarts, etc.—and your data that is stored in Anzo.

Note

Though Anzo is flexible and allows you to assign artifact access to roles, the recommendation is to control access to artifacts with users and groups and reserve roles for granting access to functions in the applications.

The following diagram illustrates the concepts of roles and groups in Anzo:

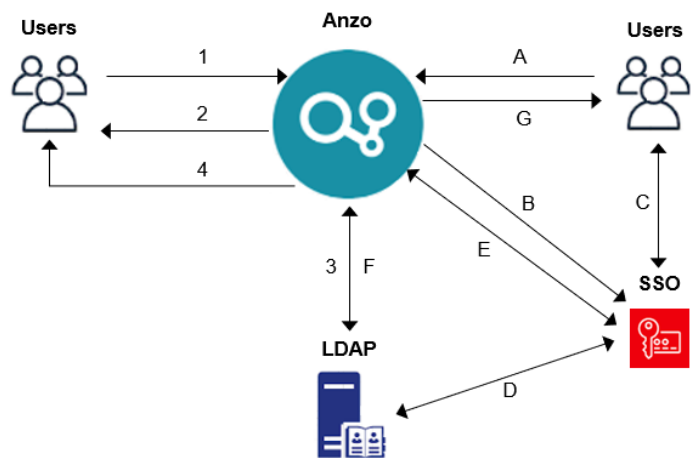


A user's role determines whether they can access the **Onboard** menu and create a new Data Source or see the **Blend** menu and create a new Graphmart. But their group assignment determines whether they can view, modify, or delete Data Source and Graphmart artifacts that are created by other users.

For more information about leveraging a directory server and details about users, groups, and roles see the sections below.

Leveraging a Directory Server (LDAP)

Anzo can be configured to access your directory server via Direct Authorization or Single Sign-On (SSO). The diagram below shows the procedures that are followed for both methods. The left side of the diagram (the numbered steps) shows the direct authorization method. The right side of the diagram (the lettered steps) shows the SSO method. The table below the diagram describes the processes for each method.



Direct Authorization	Single Sign-On
<div>1. A new (unknown) user navigates to the Anzo application.</div> <div>2. Anzo redirects the user to a login form. The user supplies credentials and submits the form.</div> <div>3. Anzo queries the LDAP for the user and group membership.</div> <div>4. Anzo redirects the user to the application with the appropriate roles applied.</div>	<div>A. A new (unknown) user navigates to the Anzo application.</div> <div>B. Anzo redirects the user to the SSO provider. The SSO provider controls authentication validation.</div> <div>C. Depending on the policy, the SSO provider presents a login screen for the user to complete and submit.</div> <div>D. As needed, the SSO provider validates the credentials with the LDAP server.</div> <div>E. The SSO provider authenticates the Anzo session with a callback.</div> <div>F. Anzo fetches group information from the LDAP server.</div> <div><div>Note</div><div>For SSO-configured systems, Anzo currently requires direct access to the LDAP directory (and a bind user) to look up groups.</div></div> <div>G. Anzo redirects the user to the application with the appropriate roles applied.</div>

For more information on connecting to a directory server, see the following topics:

- [Connecting to a Directory Server](#)
- [Connecting to an SSO Provider](#)

Users and Groups

Groups must originate in the directory server and be synced to Anzo. Groups cannot be created in Anzo. Typically users also originate from the directory server, but you can create user accounts in Anzo. Any users that are created in Anzo are stored in Anzo's internal LDAP server.

For information about retrieving user and groups from the directory server or creating internal Anzo users, see the following topics:

- [Adding Directory Users and Groups to Anzo](#)
- [Creating an Internal Anzo User](#)

Roles

Anzo is configured with predefined roles. You can create new roles and disregard the predefined roles, remove the predefined roles, or add your groups to the predefined roles and modify the assigned permissions as needed.

For details about the default roles and instructions on creating new roles, see the following topics:

- [Predefined Anzo Roles and Permissions](#)
- [Creating and Managing Roles](#)

Permissions

The way you give a role access to the Anzo applications and particular functions in those applications is to assign permissions to the role. All permissions are predefined in Anzo. Custom permissions cannot be created, and the predefined permissions cannot be deleted.

For details about all of the permissions, see the following topic:

- [Role Permissions Reference](#)

For an overview of the data access management concepts, see [Artifact Access Control Concepts](#).

Related Topics

[Artifact Access Control Concepts](#)

[Connecting to a Directory Server](#)

[Adding Directory Users and Groups to Anzo](#)

[Connecting to an SSO Provider](#)

[Creating and Managing Roles](#)

[Creating an Internal Anzo User](#)

[Predefined Anzo Roles and Permissions](#)

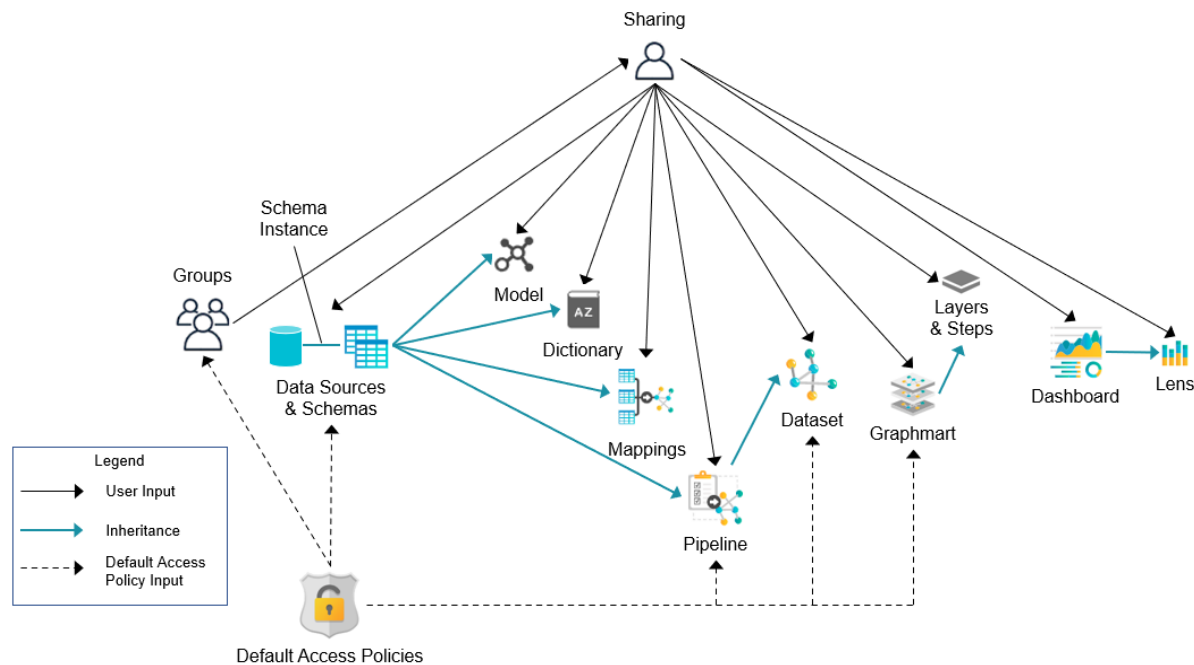
[Role Permissions Reference](#)

Artifact Access Control Concepts

The implementation of artifact and data access control in Anzo is an aggregation of three mechanisms:

1. **Default Access Policies:** These are the base permissions that are applied to artifacts by default when they are created. For most types of artifacts, the access control that is supplied by a Default Access Policy is augmented by the other two access control mechanisms.
2. **Permission Inheritance:** To facilitate common workflows, the Anzo application applies logic so that artifacts in the same workflow inherit the same permissions. For example, when a user creates a Data Source and uses the **Ingest** workflow to onboard the data, the generated Model, Pipeline, and Mapping artifacts inherit their permissions from the Data Source. Once the pipeline is published, the resulting Dataset inherits the permissions from the Pipeline. This permission inheritance is applied in addition to the applicable Default Access Policy.
3. **Sharing:** An artifact's creator can also share access to their artifact with other users or groups. When an artifact is shared, those user-configured permissions are applied in addition to any permissions that were inherited.

The following diagram illustrates the above concepts. Details about the processes and components depicted in the diagram are provided in the sections below.



Default Access Policies

Default Access Policies are the security policies that are applied by default to the artifacts that belong to a particular system **registry** (see [Registries](#) below). Default Access Policies are the base permissions that get assigned when an artifact is created—before any other access control logic (e.g., [Permission Inheritance](#)) is applied. Any artifact-level logic that is applied by Anzo or configured from the **Sharing** tab in the Anzo application augments the permissions that were supplied by the Default Access Policy.

For more information about Default Access Policies, see the following topic:

- [Managing Default Access Policies](#)

Registries

A registry is a system-level graph that stores metadata about artifacts of the same type. For example, a Data Sources Registry stores metadata about all of the Data Source and Schema artifacts, and an Ontology Registry stores metadata about all of the Data Model artifacts. Like onboarded data, registries are stored and managed as RDF named graphs according to system ontologies.

Important

Aside from changing the Default Access Policy for a registry, do not make additional modifications to registries. Changing or removing a registry can irreparably damage your Anzo server.

Permission Inheritance

The concept of inheritance is fundamental to the implementation of access control in Anzo. Inheritance allows related entities to share permissions with each other, making access easier to manage collectively, and ensuring that users have the appropriate access to each of the dependent artifacts that are crucial to their workflow. The following subsections describe the relationships and inheritance rules for each type of artifact.

- [Data Sources & Schemas](#)
- [Ingest Workflow](#)
- [Graphmarts](#)
- [Structured Pipelines](#)
- [Unstructured Pipelines](#)
- [Metadata Dictionaries](#)
- [Users and Roles](#)
- [Role Permissions and Registries](#)

Data Sources & Schemas

Data Sources and Schemas have a fundamental relationship since Schemas are imported from Data Sources and, in a sense, belong to them. Because a Data Source can have more than one Schema and the Schemas can be managed independently, Data Sources and Schemas exist as separate artifacts in Anzo. However, because of their implicit relationship, Anzo uses inheritance to facilitate users' interaction with Data Sources and the Schemas created from them.

If Anzo did not apply inheritance, a user who shares a Data Source would have to remember to add the new user to the data source *and* navigate to each related schema and add the new user there as well. Keeping permissions in sync manually presents a big challenge that is curtailed by applying inheritance.

To summarize the inheritance rules for Data sources and Schemas:

- Schemas inherit from the Data Source from which they were imported.
- Schema instances, which link Schemas to their Data Source, inherit from both the Schema and the Data Source.

Ingest Workflow

A primary workflow in Anzo is to create a new data source and then use the **Ingest** workflow (sometimes referred to as "auto-ingest") to generate all of the artifacts that are needed onboard the data and create the corresponding graph Dataset in Anzo. Artifacts created from the Ingest workflow inherit their permissions from the original Data Source.

If Anzo did not apply this inheritance, a user who wanted to share the Dataset that was derived from a Data Source would need to manually edit permissions for every artifact in the workflow: Model, Mappings, and Pipeline.

To summarize the inheritance rules for the Ingest workflow:

- Models generated by the Ingest workflow inherit permissions from the Data Source.
- Mappings generated by the Ingest workflow inherit permissions from the Data Source.
- Pipelines generated by the Ingest workflow inherit permissions from the Data Source.

Note

In rare cases when inheritance rules do not apply to artifacts, such as if a user manually creates a Mapping outside of the Ingest workflow, the [SDI Registry](#) Default Access Policy would supply the permissions for that Mapping until permissions are configured from the Mapping's **Sharing** tab.

Graphmarts

When a user creates a Graphmart, the Graphmart is assigned permissions according to the [Graphmarts Registry](#) Default Access Policy. Graphmarts contain Data Layers that describe and group the transformations that take place as the knowledge graph is generated. Since Data Layers are created in the context of a Graphmart, they inherit their permissions from the Graphmart by default.

If Anzo did not apply this inheritance, a user who wanted to share a Graphmart would have to remember to configure each newly created Data Layer to assign permissions that match the Graphmart's permissions. Otherwise someone who had access to the Graphmart would not be able to view or edit its layers and steps.

To summarize the inheritance rules for Graphmarts:

- Graphmarts inherit permissions from the Graphmarts Registry Default Access Policy.
- Data Layers and Steps created in a Graphmart inherit from the Graphmart.

For more information about graphmart permissions, see [Sharing Access to Graphmarts](#).

Structured Pipelines

When a Structured Pipeline is published, it creates a Dataset. Since the most common data ingestion workflow is for a user to introduce a Data Source and then ingest the data into a Dataset by running a Pipeline, Datasets created from a Pipeline inherit their permissions from the Pipeline. If Anzo did not apply this inheritance, a user who has access to a

Pipeline might lose the ability to see its output if the Pipeline happened to have been run by someone else first, for example.

To summarize the inheritance rules for Structured Pipelines:

- Datasets created from Structured Pipeline runs inherit from the Pipeline.
- Datasets created from auto-generated structured pipelines inherit from the original Data Source that was used to generate the Structured Pipeline.

Unstructured Pipelines

As with structured pipelines, running an Unstructured Pipeline produces a Dataset. For similar reasons, the output unstructured Dataset inherits from the Unstructured Pipeline. Additionally, each Unstructured Pipeline run produces a status dataset that is specific to the pipeline's execution. Since these status datasets are implicitly related to the Unstructured Pipeline, they inherit permissions from the pipeline.

To summarize the inheritance rules for Unstructured Pipelines:

- Datasets created from Unstructured Pipeline runs inherit from the corresponding Unstructured Pipeline.
- Pipeline status datasets inherit from the related unstructured pipeline. From an end user's perspective, this relates to the status information that is displayed in the Unstructured Pipeline user interface.

Metadata Dictionaries

Users can create Metadata Dictionaries from specific Data Sources. Because the dictionary is directly related to the origin Data Source, Metadata Dictionaries inherit their permissions from the corresponding Data Source. If one dictionary is used for multiple sources, the dictionary inherits the superset of permissions from the origin Data Sources.

To summarize the inheritance rules for Metadata Dictionaries:

- Dictionaries generated from Data Sources inherit permissions from the Data Source.
- Dictionaries that link concepts from multiple Data Sources inherit from all corresponding Data Sources.

Users and Roles

Users and roles are typically managed by administrators as a collective group. There are not clear use cases for a given user to manage some user and role accounts but not others. The expectation is that users who have the **Manage Users, Groups, and Roles** permission should be able to manage all users and roles, not just a subset of them.

To accomplish the above expectation, all users inherit permissions from one system registry, the **Role and Permissions Registry**. If user and role permissions were not centralized, there could be circumstances where one user creates a new user or role in Anzo and other users cannot see or edit that account even if they belong to a role that has the Manage Users, Groups, and Roles permission. Also if the original user or role creator had the Manage Users, Groups, and Roles permission revoked, they may retain control over the accounts they created when they had the ability to do so.

To summarize the inheritance rules for users and roles:

- Anyone who has the **Manage Users, Groups, and Roles** permission has the **Admin** level of access to all users, groups, and roles.
- The **Everyone** role has **View** access to all users, groups, and roles so that they can share artifacts with other users and groups.

Role Permissions and Registries

Access to certain registries is mapped to specific Anzo permissions. This is helpful when artifacts that are added to a registry inherit their permissions from the registry itself rather than another artifact, such as with [Users and Roles](#).

When users have a permission that grants them access to a registry, that means they can see all artifacts that belong to that registry.

The list below describes the registry access that is controlled by a permission.

- Access to the Role and Permissions Registry is granted by the **Manage Users, Groups, and Role** permission.

For more information about the Anzo permissions, see [Role Permissions Reference](#).

Sharing

Artifacts can be shared with other users and groups from the artifact's **Sharing** tab in the Anzo application. When an artifact is shared, those user-defined permissions are added to the set of permissions that came from the Default Access Policy for the related registry as well as the permission inheritance that is applied by Anzo.

For details about artifact sharing, see the following topic:

- [Sharing Access to Artifacts](#)

Related Topics

[User Management Concepts](#)

[Sharing Access to Artifacts](#)

[Managing Default Access Policies](#)

[Role Permissions Reference](#)

Connecting to a Directory Server

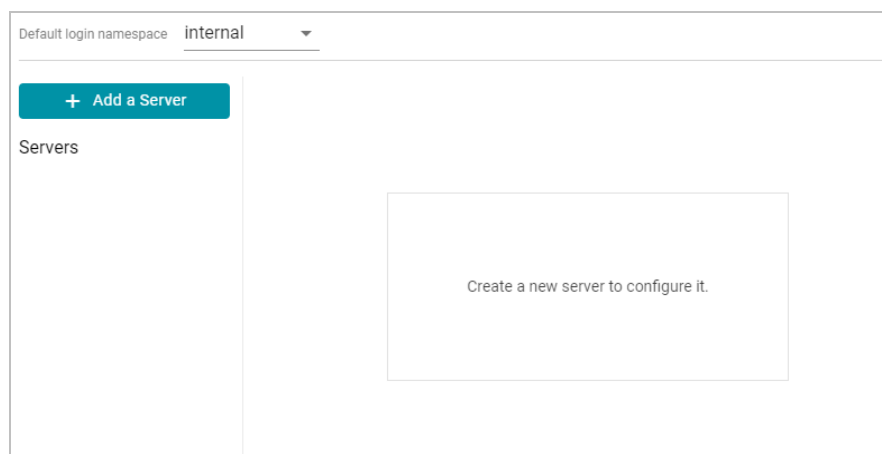
This section provides instructions for connecting to a directory server and mapping the user and group configuration to Anzo so that Anzo can leverage the users and groups from the server.

- [Connect to the Directory Server](#)
- [Map Users to Anzo](#)
- [Map Groups to Anzo](#)

Connect to the Directory Server

Follow the steps below to create a connection between Anzo and your directory server.

1. In the Administration application, expand the **User Management** menu and click **Directory**. Anzo displays the **Directory** screen. For example:



2. On the **Directory** screen, click the **Add a Server** button. Anzo displays the **Create New Server Configuration** screen.

3. Enter the connection details for the server:
 - **Host:** The host name or IP address for the directory server.
 - **Port:** The port to use to connect to the directory server.
 - **SSL Connection:** Indicates whether the directory server uses an SSL connection. Select the **SSL Connection** checkbox to enable the SSL connection. If you use SSL, make sure that you load the directory server's certificate to the Anzo trust store. See [Adding a Certificate to the Trust Store](#) for instructions.
 - **Anonymous Bind:** This option indicates whether you want Anzo to connect to the directory server anonymously. To avoid Anzo login problems when enabling this option, make sure the directory server allows anonymous binding and searches when bound anonymously. Select the **Anonymous Bind** checkbox to enable anonymous binding.

- **User DN:** The full distinguished name of the account that Anzo will bind against to perform searches on the directory server.
 - **Password and Confirm Password:** The password for the User DN.
4. Anzo attempts to connect to the server automatically. If the connection fails, make sure that you entered the correct connection details. You can also click **Test Connection** to check if Anzo can connect to the server.
 5. Click **Save** to save the server configuration and return to the Directory screen. The new server configuration is selected on the screen. For example:

Default login namespace: internal

+ Add a Server

Servers

Server Configs	User Configs	Role Configs
Host: 10.0.1.9 Port: 389 User*: cn=admin,dc=acme,dc=com Password*: ***** <input type="button" value="Test Connection"/>	<input checked="" type="checkbox"/> Directory access successful.	<input type="button" value="EDIT"/>

Once the connection to the server is established, create a user configuration for mapping directory users to Anzo. See [Map Users to Anzo](#) below for instructions.

Map Users to Anzo

Follow the steps below to create a user configuration by supplying the mapping the attributes to use to sync users with Anzo.

1. On the Directory screen, click the **User Configs** tab. Then click the **Create New User Config** button. Anzo displays the Create New Config dialog box.

Create New Config

ID *

User Base DN *

Ldap Filter

http://www.w3.org/1999/02/22-rdf-syntax-ns#type *
person

http://openanzo.org/ontologies/2008/07/System#user *

...

http://xmlns.com/foaf/0.1/surname *

...

http://xmlns.com/foaf/0.1/name *

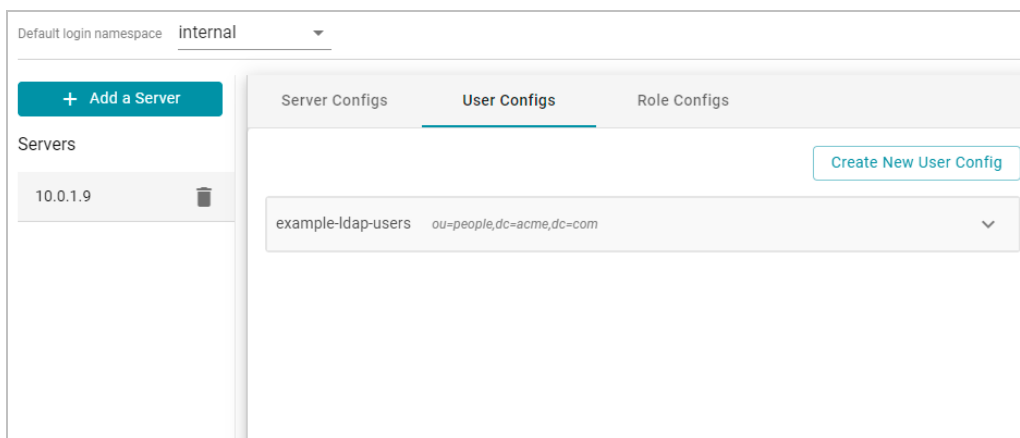
CANCEL SAVE

2. Complete the following required fields and specify the optional values as desired. Each time you map an attribute, Anzo displays some samples of the values it retrieves for that attribute. If the specified attribute does not match an attribute in the system, Anzo displays an "LDAP Attribute unavailable" message.
 - **ID: Required** setting that defines the unique name for this user configuration. Anzo uses this value as a namespace for usernames in case you connect to multiple directories with conflicting names.
 - **User Base DN: Required** setting that specifies the LDAP distinguished name.
 - **LDAP Filter:** The optional LDAP filter to apply when searching for users (usually left blank).
 - **http://www.w3.org/1999/02/22-rdf-syntax-ns#type: Required** setting that specifies the LDAP class of the type of accounts that should be logged on. Typically **person**.
 - **http://openanzo.org/ontologies/2008/07/System#user: Required** setting that specifies the LDAP attribute that contains user login information. Typically **uid**.
 - **http://xmlns.com/foaf/0.1/surname: Required** setting that specifies the LDAP attribute that contains users' surnames. Typically **sn**.
 - **http://xmlns.com/foaf/0.1/name: Required** setting that specifies the LDAP attribute that contains users' full names. Typically **cn**.
 - **http://xmlns.com/foaf/0.1/givenname: Required** setting that specifies the LDAP attribute that contains users' given (first) names. Typically **givenName**.
 - **http://xmlns.com/foaf/0.1/title:** Optional value that specifies the LDAP attribute that contains users' job titles. Typically **title**.
 - **http://xmlns.com/foaf/0.1/phone:** Optional value that specifies the LDAP attribute that contains user phone numbers. Typically **telephoneNumber**.

- **http://xmlns.com/foaf/0.1/mbox**: Optional value that specifies the LDAP attribute that contains users' email addresses. Typically **mail**.
- **http://openanzo.org/ontologies/2008/07/Anzo#location**: Optional value that specifies the LDAP attribute that contains user location information.
- **http://openanzo.org/ontologies/2008/07/Anzo#isInternalUser**: Optional boolean value that indicates whether users are Anzo internally managed users.
- **http://xmlns.com/foaf/0.1/img**: Optional value that specifies the LDAP attribute that contains images for users.
- **http://purl.org/dc/elements/1.1/description**: Optional value that specifies the LDAP attribute that contains user descriptions. Typically **description**.
- **http://openanzo.org/ontologies/2008/07/Anzo#companyDepartment**: Optional value that specifies the LDAP attribute that contains user department information. Typically **department**.

3. When you have finished mapping attributes, click **Save** to save the user configuration.

The new user configuration is added to the system and Anzo returns to the Directory screen, which shows the newly created configuration. For example:



Once the user configuration is complete, create a role configuration for mapping directory groups to Anzo. See [Map Groups to Anzo](#) below for instructions.

Map Groups to Anzo

Follow the steps below to create a role configuration by supplying the mapping the attributes to use to sync groups with Anzo.

1. On the Directory screen, click the **Role Configs** tab. Then click the **Create New Role Config** button. Anzo displays the Create New Config dialog box.

Create New Config

ID *

Base DN *

Ldap Filter

http://www.w3.org/1999/02/22-rdf-syntax-ns#type *
groupOfNames

http://xmlns.com/foaf/0.1/name *

...

http://xmlns.com/foaf/0.1/member *

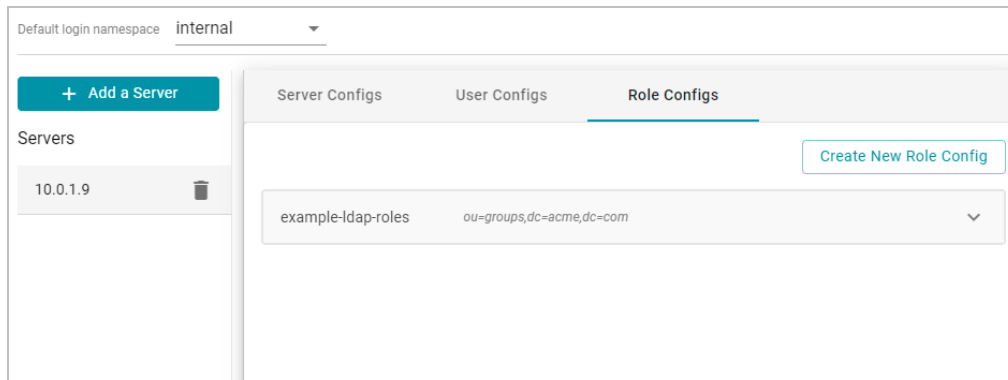
...

http://openanzo.org/ontologies/2008/07/Anzo#usedBy

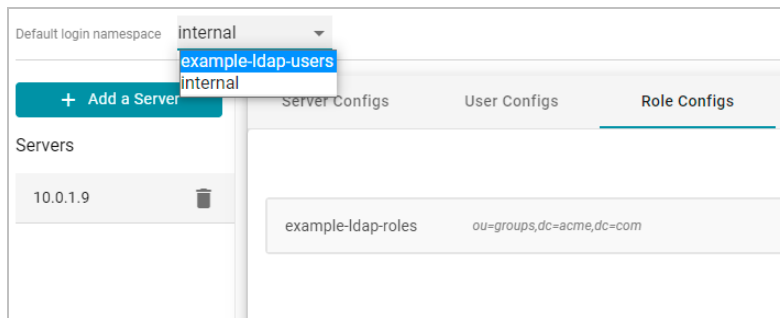
...

CANCEL SAVE

2. Complete the following required fields and specify the optional values as desired. Each time you map an attribute, Anzo displays some samples of the values it retrieves for that attribute. If the specified attribute does not match an attribute in the system, Anzo displays an "LDAP Attribute unavailable" message.
 - **ID: Required** setting that defines the unique name for this role configuration.
 - **Base DN: Required** setting that specifies the LDAP distinguished name that contains all of the system roles.
 - **LDAP Filter:** The optional LDAP filter to apply when searching for roles (usually left blank).
 - **http://www.w3.org/1999/02/22-rdf-syntax-ns#type: Required** setting that specifies the group object class of the type of roles. Typically **groupOfNames**.
 - **http://xmlns.com/foaf/0.1/name: Required** setting that specifies the LDAP attribute that contains the names of the roles.
 - **http://xmlns.com/foaf/0.1/member: Required** setting that specifies the LDAP attribute that contains common member attributes. Typically **member** or **uniqueMember**.
 - **http://openanzo.org/ontologies/2008/07/Anzo#permission:** Optional value that specifies the LDAP attribute that contains the permissions for the roles.
 - **http://purl.org/dc/elements/1.1/description:** Optional value that specifies the LDAP attribute that contains role descriptions.
3. Click **Save** to save the role configuration. The new role configuration is added to the system and Anzo returns to the Directory screen, which shows the newly created configuration. For example:



4. The last step in configuring the server is to designate the default login namespace to use if users do not fully qualify their username with the @suffix when they log in to Anzo. To set the namespace, click the **Default login namespace** drop-down list at the top of the screen and select the namespace for the directory server. It will be displayed as the ID that was specified when you set up the user configuration. The "Internal" namespace that is also listed is the internal Anzo LDAP server for local users. For example:



Once you have connected the directory server to Anzo and created user and role configurations, the next step is to add the directory users and groups to Anzo. See [Adding Directory Users and Groups to Anzo](#) for instructions.

You can also set up single-sign on access to Anzo. See [Connecting to an SSO Provider](#) for instructions.

Related Topics

[Normalizing LDAP Names](#)

[User Management and Access Control Concepts](#)

[Adding Directory Users and Groups to Anzo](#)

[Connecting to an SSO Provider](#)

Adding Directory Users and Groups to Anzo

After you connect to a central directory server, you have multiple options for how LDAP users gain access to Anzo. Some organizations retrieve the LDAP users and groups from the server and add them to Anzo. An Anzo administrator then manages role and license assignment in Anzo. Other organizations pre-define LDAP-to-Anzo role configurations and mappings so that users are automatically assigned an Anzo license and can log in to Anzo as soon

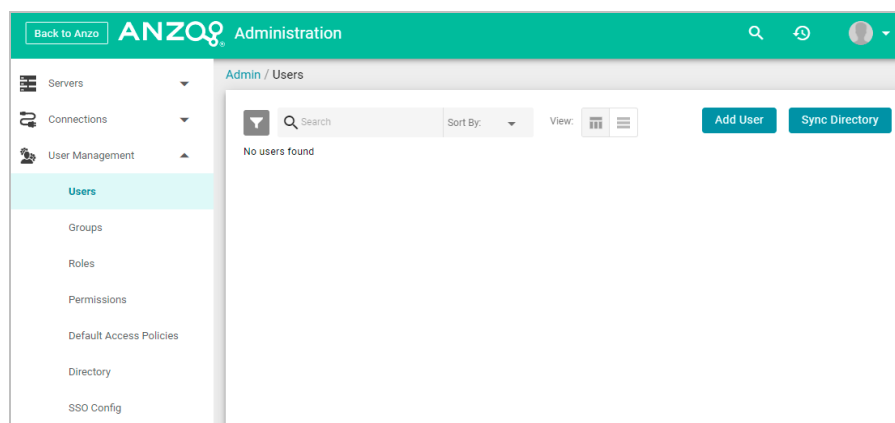
as the LDAP administrator adds them to the appropriate LDAP role. With this option, no action needs to be taken in Anzo once the directory server is connected and user and role mappings are configured.

This topic provides instructions for adding directory users and groups to Anzo. For instructions on setting up self-authorization for directory users so that they can log into Anzo and automatically become licensed after being added to the appropriate LDAP group, see [Enabling Self-Authorization for Directory Users](#).

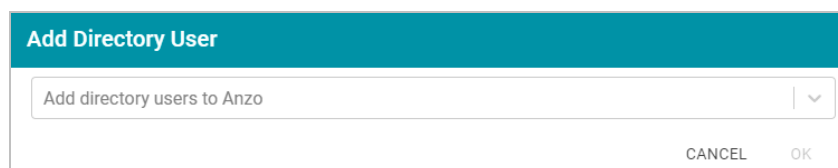
- [Adding Directory Users to Anzo](#)
- [Adding Directory Groups to Anzo](#)

Adding Directory Users to Anzo

1. To add directory users to Anzo, select **Users** from the **User Management** menu in the Administration application. The Users screen is displayed. For example:







2. Click the **Add User** button and select **Add Directory Users**. The Add Directory User dialog box is displayed:



3. Click the **Add directory users to Anzo** drop-down list, and select each user to add to Anzo. Repeat this step for all of the users that you want to add.

4. When you have finished adding users, click **OK** to return to the Users screen. For example:

<div> <div></div> <div>Search</div> <div>Sort By: </div> <div>View:  </div> <div>Add User</div> <div>Sync Directory</div> </div>									
<input type="checkbox"/>	Name	Title	Licensed	Email	Roles	Internal	Login Count	Updated Date	Tags
<input type="checkbox"/>	Aaron H...	Supreme Acc...	No	aaron.heart...		No	0		
<input type="checkbox"/>	Adele Tr...	Supreme Acc...	No	adele.tremba...		No	0		
<input type="checkbox"/>	Amberly ...	Junior Mana...	No	amberly.just...		No	0		
<input type="checkbox"/>	Anton M...	Junior Peons...	No	anton.maday...		No	0		
<input type="checkbox"/>	Ashleigh ...	Supreme Pro...	No	ashleigh.cou...		No	0		
<input type="checkbox"/>	Barton ...	Elite Account...	No	barton.melsn...		No	0		
<input type="checkbox"/>	Beatrice ...	Junior Produ...	No	beatrice.hass...		No	0		
<input type="checkbox"/>	Beryl Mil...	Chief Produc...	No	beryl.milholla...		No	0		
<input type="checkbox"/>	Brent Fel...	Chief Produc...	No	brent.felice@...		No	0		
<input type="checkbox"/>	Bret Hau...	Supreme Pro...	No	bret.hauge@...		No	0		

Rows per page: 20 1-20 of 33

Note

In order for the new users to be able to log in to Anzo, they must be **Licensed** users. Complete the next step to designate licensed users.

5. The last step in the process is to configure the **Licensed** users. If you want a user to be able to log in to Anzo, they must be specified as a licensed user. To designate a user as licensed, open the Edit User dialog box by clicking a user's name in the Users list. In the dialog box, select the **Licensed** checkbox and click **Save**. For example:

Edit User

Username

fredericka.mandel

First Name

Fredericka

Last Name

Mandel

☒ Licensed

Position / Title

Chief Administrative Developer

Email

fredericka.mandel@acme.com

Phone

None

CANCEL

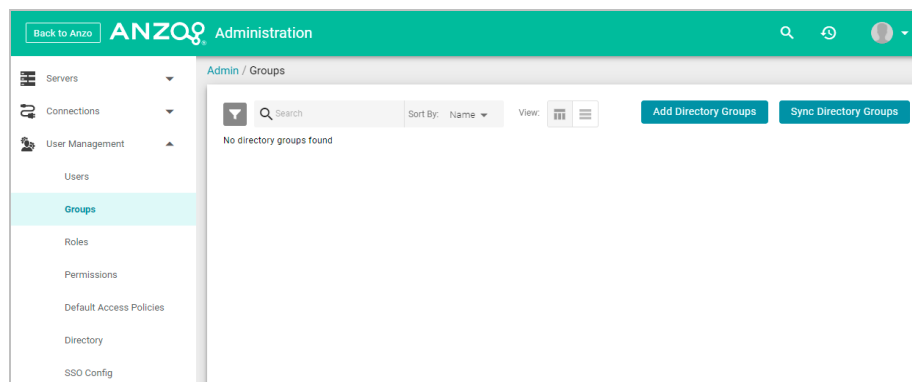
SAVE

Repeat this step for all of the users who should be licensed.

For instructions on adding groups to Anzo, proceed to [Adding Directory Groups to Anzo](#) below.

Adding Directory Groups to Anzo

1. To add directory groups to Anzo, select **Groups** from the **User Management** menu in the Administration application. The Groups screen is displayed. For example:



2. Click the **Add Directory Groups** button. The Add Directory User dialog box is displayed:



3. Click the **Add directory groups to Anzo** drop-down list, and select each group to add to Anzo. Repeat this step for all of the groups that you want to add.
4. When you have finished adding groups, click **OK** to return to the Groups screen. For example:

<div> <div> <div> <div> <div></div> <div>Search</div> </div> <div>Sort By: Name</div> <div>View: <div></div></div> </div> <div> <div>Add Directory Groups</div> <div>Sync Directory Groups</div> </div> </div> </div>						
<input type="checkbox"/>	Name	Members	Updated Date	Tags	Actions	
<input type="checkbox"/>	Accounts	Bret Hauge				
<input type="checkbox"/>	Business Development	Adele Trembath, Diann Bostic				
<input type="checkbox"/>	Customer Services	Amberly Just, Roscoe Dow				
<input type="checkbox"/>	Human Resources (HR)	Carey Breck				
<input type="checkbox"/>	Information Technology...	Beatrice Hass, Carla Goto, L...				
<input type="checkbox"/>	Marketing	Anton Maday, Delana Stigall				
<input type="checkbox"/>	Operations	Derrick Scaglione, Garry Car...				
<input type="checkbox"/>	Purchasing and Quality ...	Brent Felice, Janey Peralta, ...				
<input type="checkbox"/>	Research and Develop...					
<input type="checkbox"/>	Sales	Tobias Movickier				
<input type="checkbox"/>	Technical Support	Chadwick Gallien, Christena...				

Rows per page: 20 1-11 of 11 < >

Now that the users and groups from the directory server are available in Anzo, the next step is to associate the groups with Anzo roles. Roles are used to grant access to the Anzo applications and the functionality in those applications. See [Creating and Managing Roles](#) for instructions.

Related Topics

[Enabling Self-Authorization for Directory Users](#)

[Normalizing LDAP Names](#)

[User Management and Access Control Concepts](#)

[Connecting to a Directory Server](#)

[Creating and Managing Roles](#)

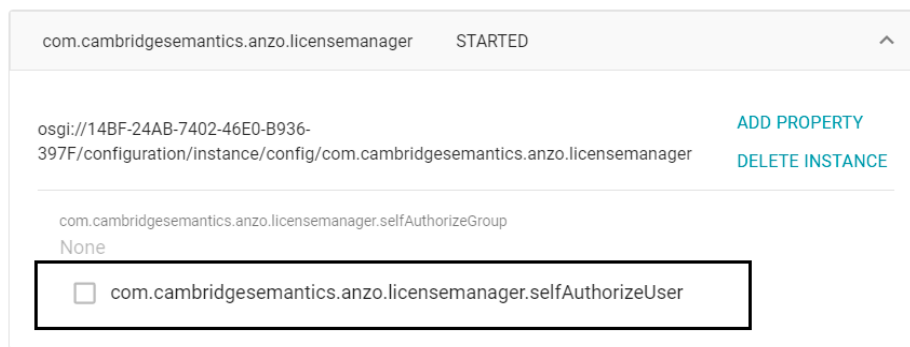
Enabling Self-Authorization for Directory Users

In order to log in to Anzo, a user must be a **Licensed** user. If you defined LDAP-to-Anzo role configurations and mappings so that you can manage all permissions in the directory server without retrieving the user and group accounts and adding them to Anzo, you can configure Anzo to automatically license those users as they log in. Follow the instructions below to enable self-authorization.

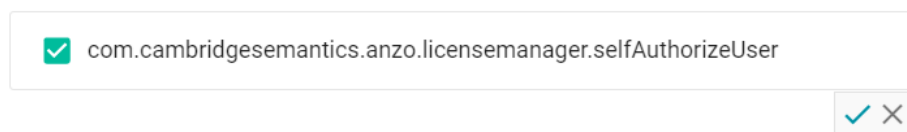
Note

When deciding whether to enable self-authorization at all or whether to limit it to certain LDAP groups, consider the number of users who will access Anzo and the number of users allowed by your Anzo license. Your Cambridge Semantics Customer Success manager can help determine whether to enable the feature if you have questions.

1. In the Administration application, expand the **Servers** menu and click **Advanced Configuration**. Click **I understand and accept the risk**.
2. Search for the **Anzo License and Entitlement Manager** bundle and view its details.
3. Click the **Services** tab and expand **com.cambridgesemantics.anzo.licensemanager**.
4. Find the **com.cambridgesemantics.anzo.licensemanager.selfAuthorizeUser** property (shown below).



- Click the property to make it editable, and then select the checkbox to enable it.



- Click the checkmark icon (✓) to save the change.
- If you want to limit the ability to self-authorize to a certain LDAP group, click the **com.cambridgesemantics.anzo.licensemanager.selfAuthorizeGroup** property to make it editable, and then specify the group name to include.
- Click the checkmark icon (✓) to save the change.

Changes to the Anzo License and Entitlement Manager service take effect immediately. You do not need to restart Anzo or the service to apply the change.

Related Topics

[User Management and Access Control Concepts](#)

[Connecting to a Directory Server](#)

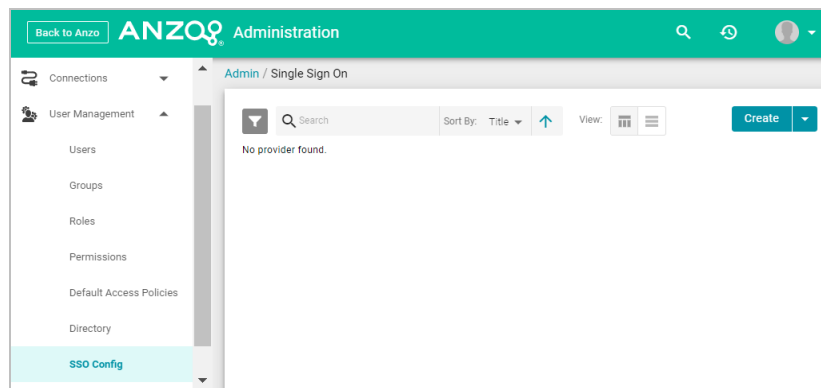
Connecting to an SSO Provider

This topic provides instructions for configuring Anzo to enable single sign-on (SSO) access using one of the following SSO providers:

- Direct and Indirect Basic
- Direct and Indirect Kerberos
- Facebook
- JSON Web Tokens (JWT) Header and Parameter
- OpenID Connect (OIDC)
- Security Assertion Markup Language (SAML)
- Google OpenID Connect (OIDC)

Follow the instructions below to add a provider.

- In the Administration application, expand the **User Management** menu and click **SSO Config**. Anzo displays the Single Sign On screen, which lists any existing SSO providers. For example:



- Click the **Create** button and select the type of provider to configure. Anzo opens the Create dialog box for that provider. Complete the required fields and supply any of the relevant optional information. The list below provides details about the properties for each provider.

Direct Basic Provider

This section describes the settings that are available on the Create Direct Basic Provider screen:

Create Direct Basic Provider

Title *

Description

Enable on matched container ID *

This provider will be active if the request container ID matches one of the supplied container IDs.

Realm Name

authentication required

The text that is displayed in the dialog box that appears when the browser prompts the user for login data.

Enable on match regex

ADD

This provider will be active if the request url matches the supplied regex. It will be active by default if no value is supplied.

Disable on match regex

ADD

This provider will be inactive if the request url matches the supplied regex. It will be active by default if no value is supplied.

CANCEL

SAVE

- Title:** Required field that specifies the name for this provider configuration.
- Description:** Optional field that provides a description for this provider configuration.
- Enable on matched container ID:** Required field that lists the container ID(s) to match. This provider will be active if the request container ID matches one of the container IDs specified in this property. Click the

field and select a container ID from the drop-down list. To specify multiple IDs, click the field again and select another value. To remove a container from the list, click the X on the right of the container name.

- **Realm Name:** Optional field that specifies the name of the security realm.
- **Enable on match regex:** Optional field that defines regular expression rules for matching request URLs to enable. To add a rule, type an expression in the field and click **Add**. This provider will be active if the request URL matches any of the supplied expressions. If Enable on match regex is blank, the provider will be active by default.
- **Disable on match regex:** Optional field that defines regular expression rules for matching request URLs to disable. To add a rule, type an expression in the field and click **Add**. This provider will be inactive if the request URL matches any of the supplied expressions. If Disable on match regex is blank, the provider will be active by default.
- **Email Template regex:** If email was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between email addresses stored by the SSO provider and email addresses returned by the directory server.
- **Email Template Replacement:** Optional field that specifies a replacement email template to use if there are variations found by Email Template regex.
- **User Template regex:** If user was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between user names stored by the SSO provider and user names returned by the directory server.
- **User Template Replacement:** Optional field that specifies a replacement user template to use if there are variations found by User Template regex.
- **Use username directly:**
- **Skip CSRF check:** Optional property that specifies whether to perform a cross-site request forgery (CSRF) check.
- **LDAP domain:** Optional field that specifies the LDAP domain to use for user lookup.
- **LDAP email property:** Optional field that specifies the LDAP email property to use to find the associated user. For example, `http://openanzo.org/ontologies/2008/07/Anzo#ldapEmailInfo`.

Direct Kerberos Provider

This section describes the settings that are available on the Create Direct Kerberos Provider screen:

- **Title:** **Required** field that specifies the name for this provider configuration.
- **Description:** Optional field that provides a description for this provider configuration.
- **Enable on matched container ID:** **Required** field that lists the container ID(s) to match. This provider will be active if the request container ID matches one of the container IDs specified in this property. Click the field and select a container ID from the drop-down list. To specify multiple IDs, click the field again and select another value. To remove a container from the list, click the X on the right of the container name.
- **Service Principal:** **Required** field that specifies the service and DNS name for the application. For authentication through the web browser, specify the service principal value in the following format:

```
HTTP/fully_qualified_domain_name@domain
```

For example, `HTTP/server.example.com@example.com`.

Note The keytab file must contain the key for this principal.

- **Keytab:** **Required** field that specifies the .keytab file that lists the Kerberos principals and encrypted keys. Click the **Keytab** field to open the File Location dialog box and select the keytab file.
- **Realm:** Optional field that specifies the Kerberos realm that the service principal maps to.
- **KRB Configuration:** Optional field that specifies the path and file name for the krb5.conf file on the Kerberos instance. The default location is `/etc/krb5.conf`.

- **KDC:** Optional field that specifies the domain name for the Key Distribution Center.
- **Debug mode:** Optional field that specifies whether Kerberos debug logging is enabled for your provider.
- **Enable on match regex:** Optional field that defines regular expression rules for matching request URLs to enable. To add a rule, type an expression in the field and click **Add**. This provider will be active if the request URL matches any of the supplied expressions. If Enable on match regex is blank, the provider will be active by default.
- **Disable on match regex:** Optional field that defines regular expression rules for matching request URLs to disable. To add a rule, type an expression in the field and click **Add**. This provider will be inactive if the request URL matches any of the supplied expressions. If Disable on match regex is blank, the provider will be active by default.
- **Email Template regex:** If email was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between email addresses stored by the SSO provider and email addresses returned by the directory server.
- **Email Template Replacement:** Optional field that specifies a replacement email template to use if there are variations found by Email Template regex.
- **User Template regex:** If user was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between user names stored by the SSO provider and user names returned by the directory server.
- **User Template Replacement:** Optional field that specifies a replacement user template to use if there are variations found by User Template regex.
- **Use username directly:**
- **Skip CSRF check:** Optional property that specifies whether to perform a cross-site request forgery (CSRF) check.
- **LDAP domain:** Optional field that specifies the LDAP domain to use for user lookup.
- **LDAP email property:** Optional field that specifies the LDAP email property to use to find the associated user. For example, `http://openanzo.org/ontologies/2008/07/Anzo#ldapEmailInfo`.

Facebook Provider

This section describes the settings that are available on the Create Facebook Provider screen:

Create Facebook Provider

Title *

Description

Enable on matched container ID *

This provider will be active if the request container ID matches one of the supplied container IDs.

Client ID *

Client Identifier

Secret *

Confirm Password *

OAuth secret

☒ Enable on login page

Callback URL *

After a successful login, the identity provider will redirect the user back to the application on the callback URL.

CANCEL SAVE

- **Title:** **Required** field that specifies the name for this provider configuration.
- **Description:** Optional field that provides a description for this provider configuration.
- **Enable on matched container ID:** **Required** field that lists the container ID(s) to match. This provider will be active if the request container ID matches one of the container IDs specified in this property. Click the field and select a container ID from the drop-down list. To specify multiple IDs, click the field again and select another value. To remove a container from the list, click the X on the right of the container name.
- **Client ID:** **Required** field that specifies the unique App ID for the client application.
- **Secret:** **Required** field that specifies the App Secret for the specified Client ID.
- **Confirm Password:** **Required** field that confirms the specified Secret.
- **Enable on login page:** Optional field that specifies whether to enable a link for this provider on the Anzo login screen.
- **Callback URL:** **Required** field that specifies the URL for the provider to use to redirect users back to the Anzo application after a successful login.
- **Callback URL port replacement:** Optional field that specifies the port to use for the Callback URL.
- **User Identifier:** Optional field that specifies the SSO provider attribute, such as email or username, to use for looking up users in the directory server.
- **Logout of IDP:** Optional field that specifies whether logging out of Anzo should also prompt the user to log out of the identity provider session. When this option is enabled, logging out of the Anzo application presents

a "Perform central logout" dialog box. Selecting the **Perform central logout** checkbox logs the user out of the SSO session.

- **Default to IDP Logout:** When **Logout of IDP** is enabled, users are presented with an option to perform a central log out when they log out of Anzo. When the **Default to IDP Logout** option is enabled, users are not given a choice about logging out of the IDP. The central logout is performed by default.
- **Logout URL Suffix:** When Logout of IDP is enabled, the Logout URL Suffix is used to access the logout URL for the SSO provider. The [urlAfterLogout] placeholder is replaced with the SSO provider server URL.
- **Email Template regex:** If email was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between email addresses stored by the SSO provider and email addresses returned by the directory server.
- **Email Template Replacement:** Optional field that specifies a replacement email template to use if there are variations found by Email Template regex.
- **User Template regex:** If user was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between user names stored by the SSO provider and user names returned by the directory server.
- **User Template Replacement:** Optional field that specifies a replacement user template to use if there are variations found by User Template regex.
- **Use username directly:**
- **Skip CSRF check:** Optional property that specifies whether to perform a cross-site request forgery (CSRF) check.
- **LDAP domain:** Optional field that specifies the LDAP domain to use for user lookup.
- **LDAP email property:** Optional field that specifies the LDAP email property to use to find the associated user. For example, `http://openanzo.org/ontologies/2008/07/Anzo#ldapEmailInfo`.
- **Icon:** Optional property that specifies an SSO icon to use on the Anzo login screen. To select an image file, click the **Icon** field and select **Add File**.

Indirect Basic Provider

This section describes the settings that are available on the Create Indirect Basic Provider screen:

Create Indirect Basic Provider

Title *

Description

Enable on matched container ID *

This provider will be active if the request container ID matches one of the supplied container IDs.

Realm Name

authentication required

The text that is displayed in the dialog box that appears when the browser prompts the user for login data.

☒ Enable on login page

Callback URL *

After a successful login, the identity provider will redirect the user back to the application on the callback URL.

Callback URL port replacement

[PORT] will be replaced with this value in callback URL

CANCEL SAVE

- **Title: Required** field that specifies the name for this provider configuration.
- **Description:** Optional field that provides a description for this provider configuration.
- **Enable on matched container ID: Required** field that lists the container ID(s) to match. This provider will be active if the request container ID matches one of the container IDs specified in this property. Click the field and select a container ID from the drop-down list. To specify multiple IDs, click the field again and select another value. To remove a container from the list, click the X on the right of the container name.
- **Realm Name:** Optional field that specifies the name of the security realm.
- **Enable on login page:** Optional field that specifies whether to enable a link for this provider on the Anzo login screen.
- **Callback URL: Required** field that specifies the URL for the provider to use to redirect users back to the Anzo application after a successful login.
- **Callback URL port replacement:** Optional field that specifies the port to use for the Callback URL.
- **User Identifier:** Optional field that specifies the SSO provider attribute, such as email or username, to use for looking up users in the directory server.
- **Email Template regex:** If email was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between email addresses stored by the SSO provider and email addresses returned by the directory server.
- **Email Template Replacement:** Optional field that specifies a replacement email template to use if there are variations found by Email Template regex.

- **User Template regex:** If user was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between user names stored by the SSO provider and user names returned by the directory server.
- **User Template Replacement:** Optional field that specifies a replacement user template to use if there are variations found by User Template regex.
- **Use username directly:**
- **Skip CSRF check:** Optional property that specifies whether to perform a cross-site request forgery (CSRF) check.
- **LDAP domain:** Optional field that specifies the LDAP domain to use for user lookup.
- **LDAP email property:** Optional field that specifies the LDAP email property to use to find the associated user. For example, `http://openanzo.org/ontologies/2008/07/Anzo#ldapEmailInfo`.
- **Icon:** Optional property that specifies an SSO icon to use on the Anzo login screen. To select an image file, click the **Icon** field and select **Add File**.

Indirect Kerberos Provider

This section describes the settings that are available on the Create Indirect Kerberos Provider screen:

Create Indirect Kerberos Provider

Title *

Description

Enable on matched container ID * ▼

This provider will be active if the request container ID matches one of the supplied container IDs.

Service Principal *

The service principal of the application. For web apps this is HTTP/full-qualified-domain-name@DOMAIN. The keytab must contain the key for this principal.

Keytab * [BROWSE](#)

A keytab is a file containing pairs of Kerberos principals and encrypted keys.

Realm

System property java.security.krb5.realm

KRB Configuration

System property java.security.krb5.conf

CANCEL SAVE

- **Title:** Required field that specifies the name for this provider configuration.
- **Description:** Optional field that provides a description for this provider configuration.

- **Enable on matched container ID:** **Required** field that lists the container ID(s) to match. This provider will be active if the request container ID matches one of the container IDs specified in this property. Click the field and select a container ID from the drop-down list. To specify multiple IDs, click the field again and select another value. To remove a container from the list, click the X on the right of the container name.
- **Service Principal:** **Required** field that specifies the service and DNS name for the application. For authentication through the web browser, specify the service principal value in the following format:

```
HTTP/fully_qualified_domain_name@domain
```

For example, `HTTP/server.example.com@example.com`.

Note The keytab file must contain the key for this principal.

- **Keytab:** **Required** field that specifies the .keytab file that lists the Kerberos principals and encrypted keys. Click the **Keytab** field to open the File Location dialog box and select the keytab file.
- **Realm:** Optional field that specifies the Kerberos realm that the service principal maps to.
- **KRB Configuration:** Optional field that specifies the path and file name for the `krb5.conf` file on the Kerberos instance. The default location is `/etc/krb5.conf`.
- **KDC:** Optional field that specifies the domain name for the Key Distribution Center.
- **Debug mode:** Optional field that specifies whether Kerberos debug logging is enabled for your provider.
- **Enable on login page:** Optional field that specifies whether to enable a link for this provider on the Anzo login screen.
- **Callback URL:** **Required** field that specifies the URL for the provider to use to redirect users back to the Anzo application after a successful login.
- **Callback URL port replacement:** Optional field that specifies the port to use for the Callback URL.
- **User Identifier:** Optional field that specifies the SSO provider attribute, such as email or username, to use for looking up users in the directory server.
- **Logout of IDP:** Optional field that specifies whether logging out of Anzo should also prompt the user to log out of the identity provider session. When this option is enabled, logging out of the Anzo application presents a "Perform central logout" dialog box. Selecting the **Perform central logout** checkbox logs the user out of the SSO session.
- **Default to IDP Logout:** When **Logout of IDP** is enabled, users are presented with an option to perform a central log out when they log out of Anzo. When the **Default to IDP Logout** option is enabled, users are not given a choice about logging out of the IDP. The central logout is performed by default.
- **Default to IDP Logout:** When **Logout of IDP** is enabled, users are presented with an option to perform a central log out when they log out of Anzo. When the **Default to IDP Logout** option is enabled, users are not given a choice about logging out of the IDP. The central logout is performed by default.

- **Logout URL Suffix:** When Logout of IDP is enabled, the Logout URL Suffix is used to access the logout URL for the SSO provider. The [urlAfterLogout] placeholder is replaced with the SSO provider server URL.
- **Email Template regex:** If email was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between email addresses stored by the SSO provider and email addresses returned by the directory server.
- **Email Template Replacement:** Optional field that specifies a replacement email template to use if there are variations found by Email Template regex.
- **User Template regex:** If user was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between user names stored by the SSO provider and user names returned by the directory server.
- **User Template Replacement:** Optional field that specifies a replacement user template to use if there are variations found by User Template regex.
- **Use username directly:**
- **Skip CSRF check:** Optional property that specifies whether to perform a cross-site request forgery (CSRF) check.
- **LDAP domain:** Optional field that specifies the LDAP domain to use for user lookup.
- **LDAP email property:** Optional field that specifies the LDAP email property to use to find the associated user. For example, `http://openanzo.org/ontologies/2008/07/Anzo#ldapEmailInfo`.
- **Icon:** Optional property that specifies an SSO icon to use on the Anzo login screen. To select an image file, click the **Icon** field and select **Add File**.

JWT Header Provider

This section describes the settings that are available on the Create JWT Header Provider screen:

Create JWT Header Provider

Title *

Description

Enable on matched container ID *

This provider will be active if the request container ID matches one of the supplied container IDs.

Header Prefix

Header Name

Signing Secret *

Private key, private key passcode and/or shared secret depending on algorithm

CANCEL SAVE

- **Title:** **Required** field that specifies the name for this provider configuration.
- **Description:** Optional field that provides a description for this provider configuration.
- **Enable on matched container ID:** **Required** field that lists the container ID(s) to match. This provider will be active if the request container ID matches one of the container IDs specified in this property. Click the field and select a container ID from the drop-down list. To specify multiple IDs, click the field again and select another value. To remove a container from the list, click the X on the right of the container name.
- **Header Prefix:** Optional field that specifies the header prefix if one is used.
- **Header Name:** Optional field that specifies the header name.
- **Signing Secret:** **Required** field that specifies the secret the token is signed with.
- **Key Algorithm:** Optional field that specifies the signing algorithm that is used.
- **Encryption Method:** Optional field that specifies the encryption method used for encrypted tokens.
- **Encryption Secret:** Optional field that specifies the secret used for encrypted tokens.
- **Enable on match regex:** Optional field that defines regular expression rules for matching request URLs to enable. To add a rule, type an expression in the field and click **Add**. This provider will be active if the request URL matches any of the supplied expressions. If Enable on match regex is blank, the provider will be active by default.
- **Disable on match regex:** Optional field that defines regular expression rules for matching request URLs to disable. To add a rule, type an expression in the field and click **Add**. This provider will be inactive if the

request URL matches any of the supplied expressions. If Disable on match regex is blank, the provider will be active by default.

- **Email Template regex:** If email was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between email addresses stored by the SSO provider and email addresses returned by the directory server.
- **Email Template Replacement:** Optional field that specifies a replacement email template to use if there are variations found by Email Template regex.
- **User Template regex:** If user was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between user names stored by the SSO provider and user names returned by the directory server.
- **User Template Replacement:** Optional field that specifies a replacement user template to use if there are variations found by User Template regex.
- **Use username directly:**
- **Skip CSRF check:** Optional property that specifies whether to perform a cross-site request forgery (CSRF) check.
- **LDAP domain:** Optional field that specifies the LDAP domain to use for user lookup.
- **LDAP email property:** Optional field that specifies the LDAP email property to use to find the associated user. For example, `http://openanzo.org/ontologies/2008/07/Anzo#ldapEmailInfo`.

JWT Parameter Provider

This section describes the settings that are available on the Create JWT Parameter Provider screen:

Create JWT Parameter Provider

Title *

Description

Enable on matched container ID *

This provider will be active if the request container ID matches one of the supplied container IDs.

Parameter Name *

token

Parameter Name

☒ Supports GET request ☒ Supports POST request

Signing Secret *

Private key, private key passcode and/or shared secret depending on algorithm

Key Algorithm

AES

Key Algorithm

CANCEL SAVE

- **Title:** Required field that specifies the name for this provider configuration.
- **Description:** Optional field that provides a description for this provider configuration.
- **Enable on matched container ID:** Required field that lists the container ID(s) to match. This provider will be active if the request container ID matches one of the container IDs specified in this property. Click the field and select a container ID from the drop-down list. To specify multiple IDs, click the field again and select another value. To remove a container from the list, click the X on the right of the container name.
- **Parameter Name:** Required field that specifies the header parameter name.
- **Supports GET request:** Optional field that indicates whether GET requests are supported using the token.
- **Supports POST request:** Optional field that indicates whether POST requests are supported using the token.
- **Signing Secret:** Required field that specifies the secret the token is signed with.
- **Key Algorithm:** Optional field that specifies the signing algorithm that is used.
- **Encryption Algorithm:**
- **Encryption Method:** Optional field that specifies the encryption method used for encrypted tokens.
- **Encryption Secret:** Optional field that specifies the secret used for encrypted tokens.
- **Enable on match regex:** Optional field that defines regular expression rules for matching request URLs to enable. To add a rule, type an expression in the field and click **Add**. This provider will be active if the request

URL matches any of the supplied expressions. If Enable on match regex is blank, the provider will be active by default.

- **Disable on match regex:** Optional field that defines regular expression rules for matching request URLs to disable. To add a rule, type an expression in the field and click **Add**. This provider will be inactive if the request URL matches any of the supplied expressions. If Disable on match regex is blank, the provider will be active by default.
- **Email Template regex:** If email was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between email addresses stored by the SSO provider and email addresses returned by the directory server.
- **Email Template Replacement:** Optional field that specifies a replacement email template to use if there are variations found by Email Template regex.
- **User Template regex:** If user was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between user names stored by the SSO provider and user names returned by the directory server.
- **User Template Replacement:** Optional field that specifies a replacement user template to use if there are variations found by User Template regex.
- **Use username directly:**
- **Skip CSRF check:** Optional property that specifies whether to perform a cross-site request forgery (CSRF) check.
- **LDAP domain:** Optional field that specifies the LDAP domain to use for user lookup.
- **LDAP email property:** Optional field that specifies the LDAP email property to use to find the associated user. For example, `http://openanzo.org/ontologies/2008/07/Anzo#ldapEmailInfo`.

Open ID Connect Provider

This section describes the settings that are available on the Create Open ID Connect Provider screen:

Create Open ID Connect Provider

Title *

Description

Enable on matched container ID *

This provider will be active if the request container ID matches one of the supplied container IDs.

Client ID *

Client Identifier

Secret *

Confirm Password *

OAuth secret

Discovery URI *

discovery URI for fetching OP metadata (http://openid.net/specs/openid-connect-discovery-1_0.html)

Scope

openid profile email

CANCEL SAVE

- **Title: Required** field that specifies the name for this provider configuration.
- **Description:** Optional field that provides a description for this provider configuration.
- **Enable on matched container ID: Required** field that lists the container ID(s) to match. This provider will be active if the request container ID matches one of the container IDs specified in this property. Click the field and select a container ID from the drop-down list. To specify multiple IDs, click the field again and select another value. To remove a container from the list, click the X on the right of the container name.
- **Client ID: Required** field that specifies client ID or consumer key value from the provider application.
- **Secret: Required** field that specifies the client secret from the provider application.
- **Confirm Secret: Required** field to confirm the specified Secret.
- **Discovery URI: Required** field that specifies the discovery URI to use for fetching OP Metadata.
- **Scope:** Optional field that specifies the scope to send to the authorization endpoint with the request.
- **Preferred JWS Algorithm:** Optional field that lists the preferred signing algorithm.
- **Enable on login page:** Optional field that specifies whether to enable a link for this provider on the Anzo login screen.
- **Callback URL: Required** field that specifies the URL for the provider to use to redirect users back to the Anzo application after a successful login.
- **Callback URL port replacement:** Optional field that specifies the port to use for the Callback URL.

- **User Identifier:** Optional field that specifies the SSO provider attribute, such as email or username, to use for looking up users in the directory server.
- **Logout of IDP:** Optional field that specifies whether logging out of Anzo should also prompt the user to log out of the identity provider session. When this option is enabled, logging out of the Anzo application presents a "Perform central logout" dialog box. Selecting the **Perform central logout** checkbox logs the user out of the SSO session.
- **Default to IDP Logout:** When **Logout of IDP** is enabled, users are presented with an option to perform a central log out when they log out of Anzo. When the **Default to IDP Logout** option is enabled, users are not given a choice about logging out of the IDP. The central logout is performed by default.
- **Logout URL Suffix:** When Logout of IDP is enabled, the Logout URL Suffix is used to access the logout URL for the SSO provider. The [urlAfterLogout] placeholder is replaced with the SSO provider server URL.
- **Email Template regex:** If email was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between email addresses stored by the SSO provider and email addresses returned by the directory server.
- **Email Template Replacement:** Optional field that specifies a replacement email template to use if there are variations found by Email Template regex.
- **User Template regex:** If user was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between user names stored by the SSO provider and user names returned by the directory server.
- **User Template Replacement:** Optional field that specifies a replacement user template to use if there are variations found by User Template regex.
- **Use username directly:**
- **Skip CSRF check:** Optional property that specifies whether to perform a cross-site request forgery (CSRF) check.
- **LDAP domain:** Optional field that specifies the LDAP domain to use for user lookup.
- **LDAP email property:** Optional field that specifies the LDAP email property to use to find the associated user. For example, `http://openanzo.org/ontologies/2008/07/Anzo#ldapEmailInfo`.
- **Icon:** Optional property that specifies an SSO icon to use on the Anzo login screen. To select an image file, click the **Icon** field and select **Add File**.

SAML Provider

This section describes the settings that are available on the Create SAML Provider screen:

Create SAML Provider

Title *

Description

Enable on matched container ID *

This provider will be active if the request container ID matches one of the supplied container IDs.

Identity Provider Metadata

Identity Provider Metadata

Service Provider Entity ID

Service Provider Entity ID

Service Provider Metadata

Service Provider Metadata

Maximum Authentication Lifetime (s)

3600

By default, the SAML client will accept assertions based on a previous authentication for one hour. If you want to change this behavior, set this to number of seconds you prefer.

CANCEL SAVE

- **Title:** Required field that specifies the name for this provider configuration.
- **Description:** Optional field that provides a description for this provider configuration.
- **Enable on matched container ID:** Required field that lists the container ID(s) to match. This provider will be active if the request container ID matches one of the container IDs specified in this property. Click the field and select a container ID from the drop-down list. To specify multiple IDs, click the field again and select another value. To remove a container from the list, click the X on the right of the container name.
- **Identity Provider Metadata:** Required field that specifies the identity provider metadata .xml file. To add the file, click the **Identity Provider Metadata** field, click **Add File**, and select the file.
- **Service Provider Entity ID:**
- **Service Provider Metadata:** Optional field that specifies the server provider metadata .xml file. To add the file, click the **Server Provider Metadata** field, click **Add File**, and select the file.
- **Maximum Authentication Lifetime (s):**
- **Enable on login page:** Optional field that specifies whether to enable a link for this provider on the Anzo login screen.
- **Callback URL:** Required field that specifies the URL for the provider to use to redirect users back to the Anzo application after a successful login.
- **Callback URL port replacement:** Optional field that specifies the port to use for the Callback URL.

- **User Identifier:** Optional field that specifies the SSO provider attribute, such as email or username, to use for looking up users in the directory server.
- **Logout of IDP:** Optional field that specifies whether logging out of Anzo should also prompt the user to log out of the identity provider session. When this option is enabled, logging out of the Anzo application presents a "Perform central logout" dialog box. Selecting the **Perform central logout** checkbox logs the user out of the SSO session.
- **Default to IDP Logout:** When **Logout of IDP** is enabled, users are presented with an option to perform a central log out when they log out of Anzo. When the **Default to IDP Logout** option is enabled, users are not given a choice about logging out of the IDP. The central logout is performed by default.
- **Logout URL Suffix:** When Logout of IDP is enabled, the Logout URL Suffix is used to access the logout URL for the SSO provider. The [urlAfterLogout] placeholder is replaced with the SSO provider server URL.
- **Email Template regex:** If email was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between email addresses stored by the SSO provider and email addresses returned by the directory server.
- **Email Template Replacement:** Optional field that specifies a replacement email template to use if there are variations found by Email Template regex.
- **User Template regex:** If user was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between user names stored by the SSO provider and user names returned by the directory server.
- **User Template Replacement:** Optional field that specifies a replacement user template to use if there are variations found by User Template regex.
- **Use username directly:**
- **Skip CSRF check:** Optional property that specifies whether to perform a cross-site request forgery (CSRF) check.
- **LDAP domain:** Optional field that specifies the LDAP domain to use for user lookup.
- **LDAP email property:** Optional field that specifies the LDAP email property to use to find the associated user. For example, `http://openanzo.org/ontologies/2008/07/Anzo#ldapEmailInfo`.
- **Icon:** Optional property that specifies an SSO icon to use on the Anzo login screen. To select an image file, click the **Icon** field and select **Add File**.

Google OIDC Provider

This section describes the settings that are available on the Create Google OIDC Provider screen:

Create Google OIDC Provider

Title *

Description

Enable on matched container ID * ▼

This provider will be active if the request container ID matches one of the supplied container IDs.

Client ID *

Client Identifier

Secret * 👁

Confirm Secret * 👁

OAuth secret

Scope

openid profile email

Open ID scope

Preferred JWS Algorithm

Preferred JWS Algorithm

CANCEL SAVE

- **Title:** Required field that specifies the name for this provider configuration.
- **Description:** Optional field that provides a description for this provider configuration.
- **Enable on matched container ID:** Required field that lists the container ID(s) to match. This provider will be active if the request container ID matches one of the container IDs specified in this property. Click the field and select a container ID from the drop-down list. To specify multiple IDs, click the field again and select another value. To remove a container from the list, click the X on the right of the container name.
- **Client ID:** Required field that specifies client ID or consumer key value from the provider application.
- **Secret:** Required field that specifies the client secret from the provider application.
- **Confirm Secret:** Required field to confirm the specified Secret.
- **Scope:** Optional field that specifies the scope to send to the authorization endpoint with the request.
- **Preferred JWS Algorithm:** Optional field that lists the preferred signing algorithm.
- **Enable on login page:** Optional field that specifies whether to enable a link for this provider on the Anzo login screen.
- **Callback URL:** Required field that specifies the URL for the provider to use to redirect users back to the Anzo application after a successful login.
- **Callback URL port replacement:** Optional field that specifies the port to use for the Callback URL.
- **User Identifier:** Optional field that specifies the SSO provider attribute, such as email or username, to use for looking up users in the directory server.
- **Logout of IDP:** Optional field that specifies whether logging out of Anzo should also prompt the user to log out of the identity provider session. When this option is enabled, logging out of the Anzo application presents

a "Perform central logout" dialog box. Selecting the **Perform central logout** checkbox logs the user out of the SSO session.

- **Default to IDP Logout:** When **Logout of IDP** is enabled, users are presented with an option to perform a central log out when they log out of Anzo. When the **Default to IDP Logout** option is enabled, users are not given a choice about logging out of the IDP. The central logout is performed by default.
- **Logout URL Suffix:** When Logout of IDP is enabled, the Logout URL Suffix is used to access the logout URL for the SSO provider. The [urlAfterLogout] placeholder is replaced with the SSO provider server URL.
- **Email Template regex:** If email was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between email addresses stored by the SSO provider and email addresses returned by the directory server.
- **Email Template Replacement:** Optional field that specifies a replacement email template to use if there are variations found by Email Template regex.
- **User Template regex:** If user was specified as the User Identifier, you can use this optional field to include a regular expression to use for identifying variations between user names stored by the SSO provider and user names returned by the directory server.
- **User Template Replacement:** Optional field that specifies a replacement user template to use if there are variations found by User Template regex.
- **Use username directly:**
- **Skip CSRF check:** Optional property that specifies whether to perform a cross-site request forgery (CSRF) check.
- **LDAP domain:** Optional field that specifies the LDAP domain to use for user lookup.
- **LDAP email property:** Optional field that specifies the LDAP email property to use to find the associated user. For example, `http://openanzo.org/ontologies/2008/07/Anzo#ldapEmailInfo`.
- **Icon:** Optional property that specifies an SSO icon to use on the Anzo login screen. To select an image file, click the **Icon** field and select **Add File**.

3. Click **Save** to save the provider configuration.

Related Topics

[User Management and Access Control Concepts](#)

[Connecting to a Directory Server](#)

Creating and Managing Roles

In Anzo, Groups (or Users) are added to **Roles** and the Roles are configured to grant access to *functionality*. Role permissions control access to menus and screens in the Anzo and Administration applications. Access to functionality cannot be assigned to Groups or Users, only to Roles.

Tip

For more information about Role, User, and Group management, see [User Management Concepts](#).

This topic provides instructions for creating or changing the Roles to use for controlling access to Anzo functionality. For information about the predefined Anzo roles, see [Predefined Anzo Roles and Permissions](#).

- [Creating a New Role](#)
- [Adding Users or Groups to a Role](#)
- [Configuring Role Permissions](#)

Creating a New Role

1. In the Administration application, expand the **User Management** menu and click **Roles**. Anzo displays the Roles screen, which lists the existing roles. For example:

	<input type="text" value="Search"/>	Sort By: Name		View:	Create Role
<input type="checkbox"/>	Name	Members	Actions		
	Anzo Administrator	Anzo Admin			
	Data Analyst	Data Analyst			
	Data Citizen	Data Citizen			
	Data Curator	Data Curator			
	Data Governor	Data Governor			
	Data Scientist	Data Scientist			
Rows per page: 20 1-6 of 6 < >					

2. On the Roles screen, click the **Create Role** button. Anzo displays the Add Role dialog box.

Add Role

Name *

Description

Members

The members of the role

Permissions

The roles permissions

CANCEL

SAVE

3. Complete the required fields and enter any optional group details:
 - **Name:** The name for the new role.
 - **Description:** An optional description of the role.
 - **Members:** The users or groups who are members of the role. Click the **Members** field to select a member. Click the field again to select additional members.

- **Permissions:** The list of Anzo features that this role has permission to access. Click the **Permissions** field and select a permission to add it to the list. Click the field again to select additional permissions. For details about each of the permissions, see the [Role Permissions Reference](#).

4. Click **Save** to add the role to the system. Anzo adds the new role to the list of roles on the Roles screen.

Adding Users or Groups to a Role

Follow the instructions below to add users and/or groups to a role.

1. In the Administration application, expand the **User Management** menu and click **Roles**. Anzo displays the Roles screen, which lists the existing roles. For example:

<input type="checkbox"/>	Name	Members	Actions
	Anzo Administrator	Anzo Admin	
	Data Analyst	Data Analyst	
	Data Citizen	Data Citizen	
	Data Curator	Data Curator	
	Data Governor	Data Governor	
	Data Scientist	Data Scientist	

Rows per page: 20 1-6 of 6

2. Click the name of the role that you want to add users or groups to. Anzo opens the Edit Role dialog box. For example:

Edit Role

Name*

Data Curator

Description

Members of this role can create graphmarts and view dashboard analytics.

Data Curator

X

The members of the role

Show Query Builder

Activate Graphmarts

View Graphmarts

Manage Models

14f3dee7-f7c2-4919-ba50-ae7110e7369c

Onboard Unstructured Data

Browse Models

View Activity Logs

Create Dashboards

Manage Graphmarts

View Datasets

Create Anzo Data Stores

Manage Query Blacklists

Data On Demand

Anzo for Excel

Anzo Application

Create Graphmarts

Browse Dashboards

View Provenance

Hi-Res Analytics

Manage Dictionaries

Onboard Structured Data

Create Data Sources

The roles permissions

CANCEL

SAVE

3. Click the **Members** drop-down list to display the list of all available users and groups. You can also search for a user or group by typing a name in the Members field. Click a name to add that user or group to the role. Click the field again to select additional members. To remove a member from the role, click the X to the right of the name.

Note

If you do not see users or groups that you expect to see, it is possible that Anzo is out of sync with the directory server. If groups or users have been modified on the directory server, and a user has not logged in to Anzo for an extended time, the data may need to be refreshed in Anzo. The **Users** and

Groups screens in the User Management menu have **Sync Directory** buttons that you can click to synchronize with the directory server and update the data in Anzo.

- When you have finished adding members, click **Save** to save the changes to the role.

Note

When modifying an existing user's access by adding or removing roles from their account, Cambridge Semantics recommends that the user logs out of Anzo and clears their browser cache to ensure that the access changes are reflected in the user interface.

Configuring Role Permissions

Follow the instructions below to add or remove permissions from a role. For details about each of the permissions, see the [Role Permissions Reference](#).

- In the Administration application, expand the **User Management** menu and click **Roles**. Anzo displays the Roles screen, which lists the existing roles. For example:

	Search	Sort By: Name	View:	Create Role
<input type="checkbox"/>	Name	Members	Actions	
	Anzo Administrator	Anzo Admin		
	Data Analyst	Data Analyst		
	Data Citizen	Data Citizen		
	Data Curator	Data Curator		
	Data Governor	Data Governor		
	Data Scientist	Data Scientist		
Rows per page: 20 1-6 of 6 < >				

- Click the name of the role for which you want to configure permissions. Anzo opens the Edit Role dialog box. For example:

Edit Role

Name *

Data Curator

Description

Members of this role can create graphmarts and view dashboard analytics.

Data Curator

The members of the role

Permissions

Show Query Builder

Activate Graphmarts

View Graphmarts

Manage Models

14f3dee7-f7c2-4919-ba50-ae7110e7369c

Onboard Unstructured Data

Browse Models

View Activity Logs

Create Dashboards

Manage Graphmarts

View Datasets

Create Anzo Data Stores

Manage Query Blacklists

Data On Demand

Anzo for Excel

Anzo Application

Create Graphmarts

Browse Dashboards

View Provenance

Hi-Res Analytics

Manage Dictionaries

Onboard Structured Data

Create Data Sources

The roles permissions

CANCEL

SAVE

3. The **Permissions** field lists all of the permissions that are applied to the role. To remove a permission, click the X to the right of the permission name. To add a permission click the field to open the Permissions drop-down list. Click a name to add that permission to the role. Click the field again to select additional permissions.
4. When you have finished changing permissions, click **Save** to save the changes to the role.

Related Topics

[User Management and Access Control Concepts](#)

[Predefined Anzo Roles and Permissions](#)

[Role Permissions Reference](#)

[Creating an Internal Anzo User](#)

Creating an Internal Anzo User

User and group accounts are typically managed in a central directory server that is connected to Anzo. The groups from the directory server are added to Anzo roles, and access to Anzo applications and features is configured for the roles. However, you can create a user account directly in Anzo. Accounts that are created in Anzo are stored in Anzo's internal LDAP server. Follow the instructions below to create a new internal Anzo user account.

Tip

For instructions on adding directory users to Anzo, see [Adding Directory Users and Groups to Anzo](#).

1. In the Administration application, expand the **User Management** menu and click **Users**. Anzo displays the Users screen, which lists the existing users. For example:

<div> <div></div> <div>Search</div> <div>Sort By: </div> <div>View: </div> <div>Add User</div> <div>Sync Directory</div> </div>									
<input type="checkbox"/>	Name	Title	Licensed	Email	Roles	Internal	Login Count	Updated Date	Tags
	Aaron H...	Supreme Acc...	No	aaron.heart...		No	0		
	Adele Tr...	Supreme Acc...	No	adele.tremba...		No	0		
	Amberly ...	Junior Mana...	No	amberlyjust...		No	0		
	Anton M...	Junior Peons...	No	anton.maday...		No	0		
	Ashleigh ...	Supreme Pro...	No	ashleigh.cou...		No	0		
	Barton ...	Elite Account...	No	barton.meisn...		No	0		
	Beatrice ...	Junior Produ...	No	beatrice.hass...		No	0		
	Beryl Mil...	Chief Produc...	No	beryl.milholla...		No	0		
	Brent Fel...	Chief Produc...	No	brent.felice@...		No	0		
	Bret Hau...	Supreme Pro...	No	bret.hauge@...		No	0		
<div> <div>Rows per page: 20</div> <div>1-20 of 33</div> </div>									

2. On the Users screen, click the **Add User** button and select **Add User**. Anzo opens the Add User dialog box.

3. Complete the required fields and enter any optional user details:
 - **Username:** The user name that the user will use to log in to Anzo.
 - **First Name:** The user's first name.
 - **Last Name:** The user's last name.
 - **Password** and **Confirm Password:** Type a password for the user.
 - **Licensed:** Select the **Licensed** checkbox if you want this user to be able to log in to the Anzo applications. If you want to add this user to the system but do not want to give him or her access to Anzo applications at this time, clear the Licensed checkbox.
 - **Position/Title:** The user's job title or position.
 - **Email:** The user's email address.
 - **Phone:** The user's phone number.
 - **Roles:** The role or roles that the user is a member of. Roles define the user's level of access to Anzo applications and features. Click the **Roles** field and select a role from the drop-down list. Click the field again to select additional roles.
4. When you have finished configuring the user account, click **Save** to add the user to the system.

For more information about roles, see [Creating and Managing Roles](#). For a description of the default Anzo roles, see [Predefined Anzo Roles and Permissions](#).

Related Topics

[User Management and Access Control Concepts](#)

[Creating and Managing Roles](#)

[Predefined Anzo Roles and Permissions](#)

[Adding Directory Users and Groups to Anzo](#)

Predefined Anzo Roles and Permissions

This topic describes the roles that are predefined in Anzo and lists the permissions that are assigned to each role by default. The predefined roles can be removed or modified as desired. For instructions on changing roles, see [Creating and Managing Roles](#).

- [System Administrator](#)
- [Base Permissions \(Everyone and Authenticated User Roles\)](#)
- [Anzo Administrator](#)
- [Data Analyst](#)
- [Data Citizen](#)
- [Data Curator](#)
- [Data Governor](#)
- [Data Scientist](#)

System Administrator

The System Administrator account, usually named **sysadmin**, is created during the Anzo installation. This account has permission to access all Anzo features in the main Anzo application as well as administrative features in the Administration application. In addition, the sysadmin user has read and write access to all of the artifacts (such as data sources, models, and pipelines) that are created by all Anzo users. The sysadmin user permissions cannot be changed, and the account cannot be deleted. In addition, artifacts cannot be configured to restrict sysadmin access. For information about changing the system administrator password, see [Set the System Administrator Password](#).

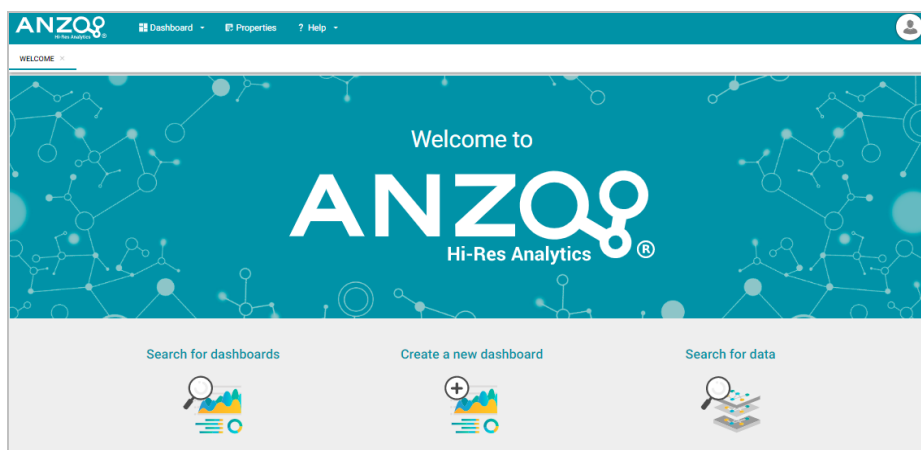
Base Permissions (Everyone and Authenticated User Roles)

There is a set of base permissions that are applied to all user accounts by default. If a user account is created in Anzo but no roles are assigned, that user has the permissions of the **Authenticated User** role. By default, authenticated users cannot access the Anzo application but can access the Hi-Res Analytics application where they can browse for and create dashboards. They can also view data that is shared from Data on Demand endpoints.

Note

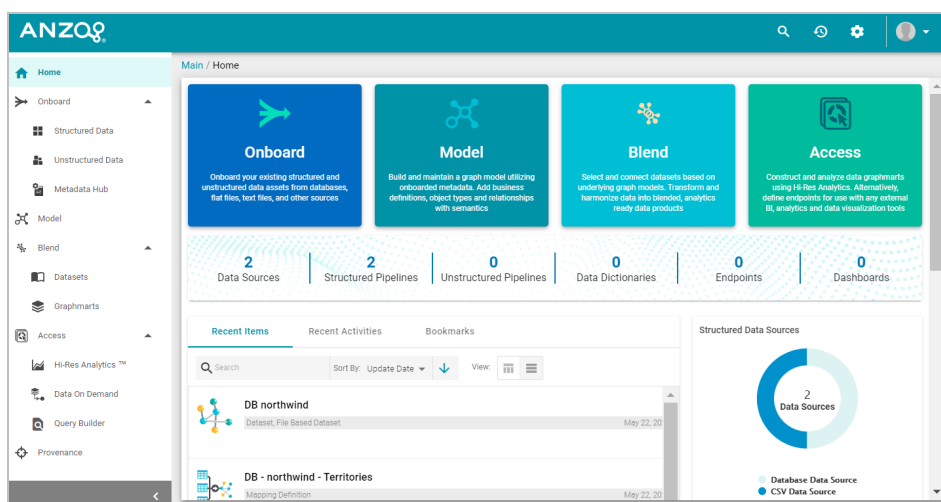
If **Anonymous User Access** is enabled on the system, unauthenticated users (users who do not have a user account in Anzo) have the permissions that are included in the **Everyone** role. The Everyone role is only used to apply permissions for unauthenticated users when anonymous access is allowed. For information about anonymous access, see [Configure Anonymous User Access](#).

The image below shows an example of the view an authenticated user has in the Hi-Res Analytics application.



Anzo Administrator

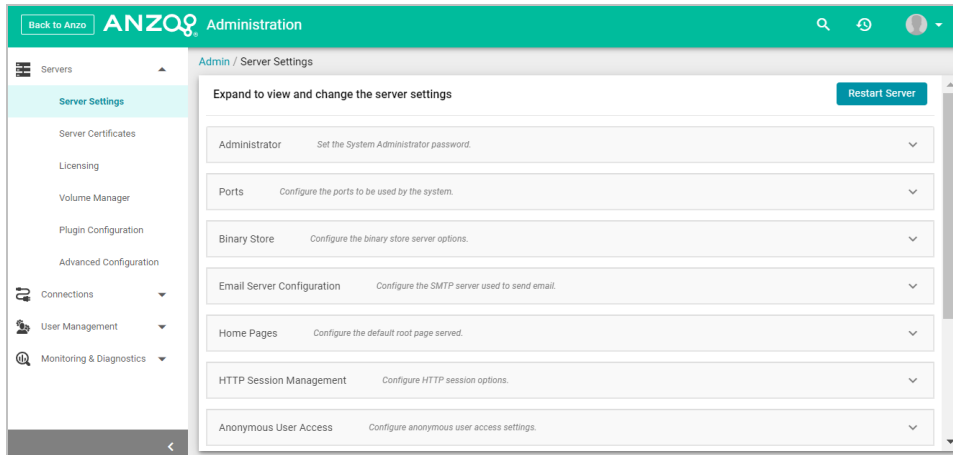
By default the Anzo Administrator role has access to all menus and features in the Anzo application as well as the Administration application. The image below shows an example of the view a user with the Anzo Administrator role has in the Anzo application.



Note

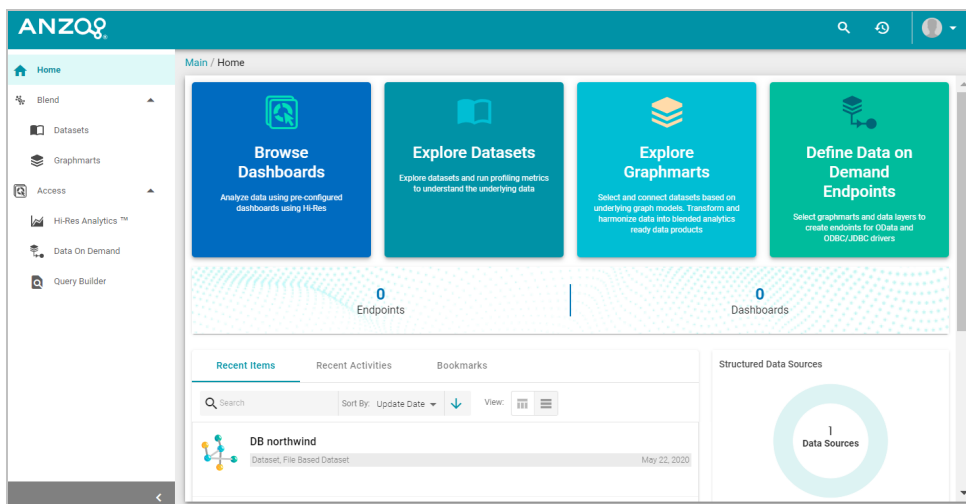
Having full access to all features does not mean the Anzo Administrator has full access to all of the data in the system. Unlike the System Administrator (the **sysadmin** user), Anzo Administrators must still be granted access to specific entities.

The following image shows an example of the Anzo Administrator view of the Administration application.



Data Analyst

By default the Data Analyst role has access to the Blend menu, Access menu, and Activity Log in the Anzo application. The image below shows an example of the view a user with the Data Analyst role has in the Anzo application.

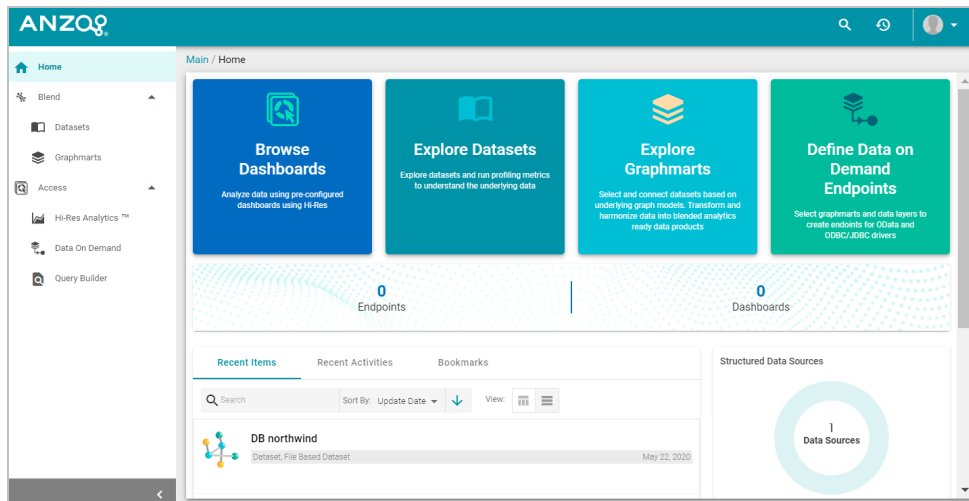


Members of the Data Analyst role can:

- View the Dataset catalog
- View and create graphmarts
- View and create Hi-Res Analytics dashboards
- View the Activity Log
- Access data with the Query Builder
- Create and access Data on Demand endpoints

Data Citizen

By default the Data Citizen role has access to the Blend menu, Access menu, and Activity Log in the Anzo application. The image below shows an example of the view a user with the Data Citizen role has in the Anzo application.

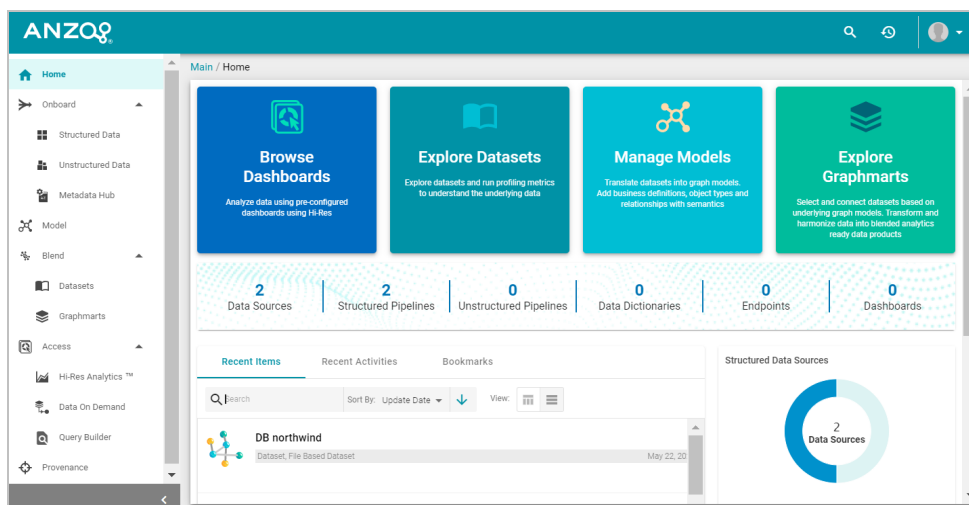


Members of the Data Citizen role can:

- View the Dataset catalog
- View graphmarts
- View and create Hi-Res Analytics dashboards
- View the Activity Log
- Access data with the Query Builder
- Create and access Data on Demand endpoints

Data Curator

By default the Data Curator role has access to the Onboard menu, Model manager, Blend menu, Access menu, Provenance, and Activity Log in the Anzo application. The image below shows an example of the view a user with the Data Curator role has in the Anzo application.

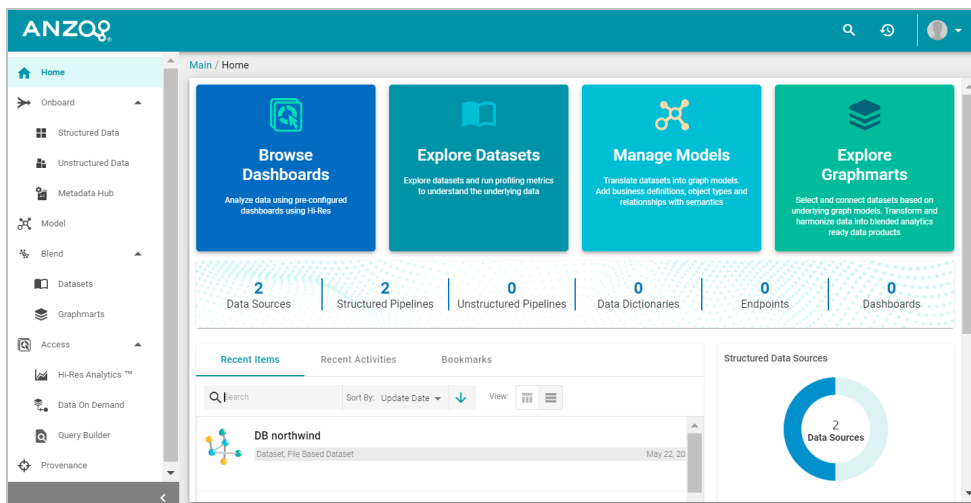


Members of the Data Curator role can:

- Connect to data sources and onboard structured and unstructured data
- View and create data models, mappings, and pipelines
- View and create metadata dictionaries
- View the Dataset catalog
- View and create graphmarts
- View and create Hi-Res Analytics dashboards
- Manage the Query Blacklist Editor in the Hi-Res Analytics application
- View the Activity Log
- Access data with the Query Builder
- Create and access Data on Demand endpoints
- View data provenance

Data Governor

By default the Data Governor role has access to the Onboard menu, Model manager, Blend menu, Access menu, Provenance, and Activity Log in the Anzo application. The image below shows an example of the view a user with the Data Governor role has in the Anzo application.



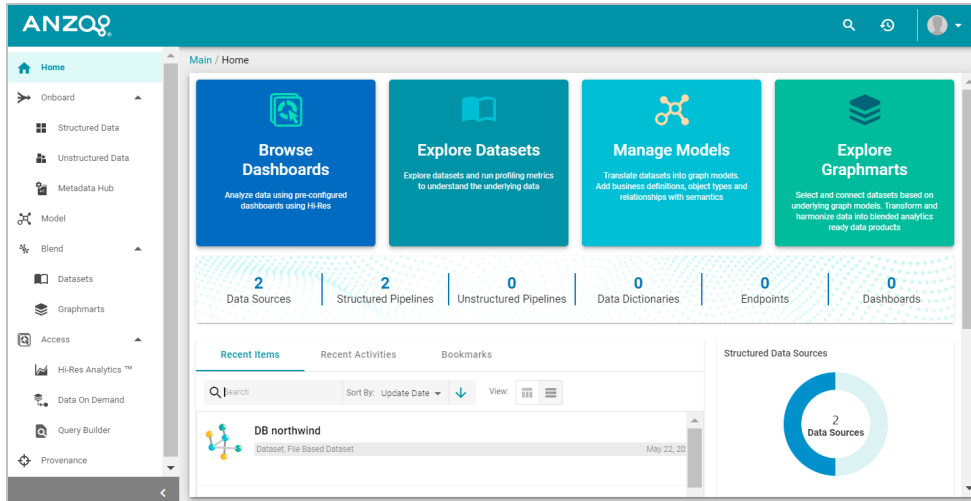
Members of the Data Governor role can:

- Connect to data sources and onboard structured and unstructured data
- View and create data models, mappings, and pipelines
- View and create metadata dictionaries
- View the Dataset catalog
- View and create graphmarts
- View and create Hi-Res Analytics dashboards
- Manage the Query Blacklist Editor in the Hi-Res Analytics application
- View the Activity Log

- Access data with the Query Builder
- Create and access Data on Demand endpoints
- View data provenance

Data Scientist

By default the Data Scientist role has access to the Onboard menu, Model manager, Blend menu, Access menu, Provenance, and Activity Log in the Anzo application. The image below shows an example of the view a user with the Data Scientist role has in the Anzo application.



Members of the Data Scientist role can:

- Connect to data sources and onboard structured and unstructured data
- View and create data models, mappings, and pipelines
- View and create metadata dictionaries
- View the Dataset catalog
- View and create graphmarts
- View and create Hi-Res Analytics dashboards
- View the Activity Log
- Access data with the Query Builder
- Create and access Data on Demand endpoints
- View data provenance

To review the specific permissions for each role, select **Roles** in the **User Management** menu in the Admin application. Click a role to open the Edit dialog box and review the permissions. For more information about the permissions, see [Role Permissions Reference](#).

Related Topics

[User Management and Access Control Concepts](#)

[Creating and Managing Roles](#)

[Role Permissions Reference](#)

Role Permissions Reference

This topic provides details about each of the permissions that can be applied to roles. These permissions grant access to functionality, i.e., the menus and screens in the Anzo and Administration applications. For example, role permissions determine whether a member of a role can access the **Onboard** menu and create a new data source or see the **Blend** menu and create a new graphmart. Whether a member can view, modify, or delete a data source or graphmart artifact that is created by someone else, however, is controlled by the user or group permissions that are applied at the artifact level.

Tip

For more information about artifact-level permissions, see [Artifact Access Control Concepts](#). And for more information about roles versus users and groups, see [User Management Concepts](#).

Permissions Overview Screen

To view an overview of the configured permissions for all Anzo roles, you can view the **Permissions** page under the **User Management** menu in the Administration application. The screen displays a table; the heading row lists each role, and the first column lists each permission. The permissions are grouped into categories, such as Application or Onboarding. For example:

	Everyone	Authenticated Users	Anzo Administrator	Data Analyst	Data Citizen	Data Curator	Data Governor	Data Scientist
Default								
Activate Graphmarts	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Browse Dashboards	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Browse Models	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Create Dashboards	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Create Graphmarts	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Data On Demand	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Manage Graphmarts	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Manage Models	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Show Query Builder	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
View Datasets	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
View Graphmarts	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
View Provenance	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Data Onboarding								
Create Anzo Data Stores	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Create Data Sources	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

The rows for each role column include checkboxes that control permissions. You can select or clear checkboxes to enable or disable permissions for a role.

Permission Descriptions

The tables below list the permissions in each category and describe the pages and menus that are enabled for members of a role where that permission is applied.

Note

The permissions described below give access to functionality in the Anzo and Administration applications. Whether members of the role have view or edit access to certain Datasets, Models, Dashboards, Graphmarts, etc. depends on the permissions that are granted at the artifact level.

Default

Permission	Description
Activate Graphmarts	<p>If the user has the appropriate permissions at the Graphmart level, this permission allows them to activate and deactivate the Graphmarts and import Graphmarts into Anzo. Does not give permission to create new Graphmarts or delete Graphmarts.</p> <p>To be able to access a Graphmart screen in the Anzo application and move the Inactive → Active slider, the Anzo Application permission also needs to be applied.</p>
Browse Dashboards	Gives permission to view existing Dashboards in the Hi-Res Analytics application. Does not give permission to create new Dashboards.
Browse Models	Gives permission to view existing data Models. Applying this permission exposes the Models menu item in the Anzo application. Must also have the Anzo Application permission to access the Anzo application.
Create Dashboards	Gives permission to create Dashboards in the Hi-Res Analytics application. Applying this permission also exposes the Create Dashboard button on the Graphmart screens in the Anzo application when the user has the Anzo Application permission.
Create Graphmarts	Gives permission to create new Graphmarts. Applying this permission exposes the Add Graphmart button on the Graphmarts screen. Must also have the Anzo Application permission to create Graphmarts in the application.

Permission	Description
Data on Demand	If the user has the appropriate permissions at the Graphmart level, this permission enables the user to create Data on Demand endpoints. Applying this permission enables the Create New Endpoint button on the Data on Demand tab for Graphmarts. Must also have the Anzo Application permission to access the application.
Import Artifacts	Gives permission to perform Import operations from the Anzo application. If a user is a member of a role that has Import Artifact assigned, they will see the Import option in the menu when they click the Add button to add a Data Source, Dataset, Model, etc. Must also have the Anzo Application permission.
Manage Graphmarts	Gives permission to manage permissions for Graphmarts. Must also have the Anzo Application permission to access the Graphmart screens.
Manage Models	Gives permission to create and import Models. Must also have the Anzo Application permission to access the Model screen.
Show Query Builder	Gives permission to find data and run SPARQL queries using the Query Builder. Applying this permission exposes the Query Builder option in the Access menu. Must also have the Anzo Application permission.
View Datasets	Gives permission to view the Dataset catalog. Applying this permission exposes the Datasets option in the Blend menu in the Anzo application. Must also have the Anzo Application permission.
View Graphmarts	Gives permission to view the list of existing Graphmarts. Must also have the Anzo Application permission to view the Graphmarts screen in the Anzo application.
View Provenance	Gives permission to view Provenance. Applying this permission exposes the Provenance option in the Anzo application. Must also have the Anzo Application permission to access the application.

Data Onboarding

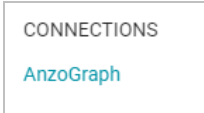
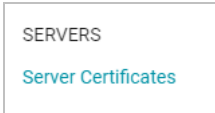
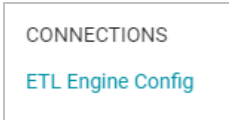
Permission	Description
Create Anzo Data Stores	Gives permission to create Anzo Data Stores. Must also have the Administer System Setup permission to make the Anzo Data Store option available in the Administration application.
Create Data Sources	Gives permission to add new Data Sources. Does not give permission to delete existing Data Sources. Must also have the Anzo Application and Onboard Structured Data permissions to access the Data Sources screen and add new sources.
Manage Dictionaries	Gives permission to view, edit, and create Metadata Dictionaries. Applying this permission exposes the Metadata Hub option in the Onboard menu. Must also have the Anzo Application permission to access the application.
Onboard Structured Data	Gives permission to access the Onboard > Structured Data menu. Must also have the Anzo Application permission.
Onboard Unstructured Data	Gives permission to create Pipelines to onboard Unstructured data. Applying this permission exposes the Onboard > Unstructured Data menu. Must also have the Anzo Application permission.

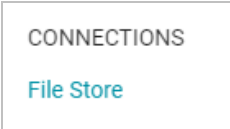
Application

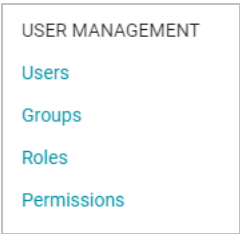
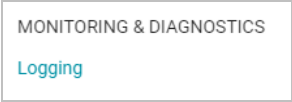
Permission	Description
Anzo Application	Grants access to the main Anzo application.
Anzo CLI	Gives permission to use the administration command line interface.
Anzo for Excel	Gives permission to open, edit, and create Mappings using the Anzo for Office Excel plugin.
Hi-Res Analytics	Grants access to the Hi-Res Analytics application.

Administration

Permission	Description																								
Administer System Setup	<p>Gives permission to access the options in the Administration application that are related to system setup, such as Server Settings, Licensing, Anzo Data Store, and Directory server configuration.</p> <p>The image below shows the view of the Administration menu that users have if Administer System Setup and Anzo Application are the only two applied permissions:</p> <table><tr><td>SERVERS</td><td>CONNECTIONS</td><td>USER MANAGEMENT</td><td>MONITORING & DIAGNOSTICS</td></tr><tr><td>Server Settings</td><td>Anzo Data Store</td><td>Default Access Policies</td><td>System Query Audit</td></tr><tr><td>Licensing</td><td>Elasticsearch Config</td><td>Directory</td><td>Semantic Services</td></tr><tr><td>Volume Manager</td><td>Cloud Locations</td><td>SSO Config</td><td>System Information</td></tr><tr><td>Plugin Configuration</td><td></td><td></td><td></td></tr><tr><td>Advanced Configuration</td><td></td><td></td><td></td></tr></table> <div><p>Note</p><p>Some menu items in the above image, such as Semantic Services, AnzoGraph, and Anzo Data Store, are also controlled by more granular permissions: Manage Semantic Services, Manage AnzoGraph, and Create Anzo Data Stores. To give an administrator full create, modify, and delete access to those functions, the granular permissions need to be enabled in addition to Administer System Setup.</p></div>	SERVERS	CONNECTIONS	USER MANAGEMENT	MONITORING & DIAGNOSTICS	Server Settings	Anzo Data Store	Default Access Policies	System Query Audit	Licensing	Elasticsearch Config	Directory	Semantic Services	Volume Manager	Cloud Locations	SSO Config	System Information	Plugin Configuration				Advanced Configuration			
SERVERS	CONNECTIONS	USER MANAGEMENT	MONITORING & DIAGNOSTICS																						
Server Settings	Anzo Data Store	Default Access Policies	System Query Audit																						
Licensing	Elasticsearch Config	Directory	Semantic Services																						
Volume Manager	Cloud Locations	SSO Config	System Information																						
Plugin Configuration																									
Advanced Configuration																									
Anzo Admin	<p>The Anzo Admin permission is a legacy permission that granted access to the Admin application that existed in pre-5.1 versions of Anzo. This permission no longer controls access to administrative functions and will be removed in an upcoming release.</p>																								

Permission	Description
Manage AnzoGraph	<p>Gives permission to view and create AnzoGraph connections. The image below shows the view of the Administration menu that users have if Manage AnzoGraph and Anzo Application are the only two applied permissions:</p>  <p>Note Manage AnzoGraph does not give permission to delete connections or change the configuration of an existing connection. Administer System Setup is required to grant permission to delete and change existing AnzoGraph connections.</p>
Manage Certificates	<p>Gives permission to upload and delete server certificates. The image below shows the view of the Administration menu that users have if Manage Certificates and Anzo Application are the only two applied permissions:</p> 
Manage ETL Engines	<p>Gives permission to add new ETL engine connections and delete or change the configuration of existing connections.</p> <p>The image below shows the view of the Administration menu that users have if Manage ETL Engines and Anzo Application are the only two applied permissions:</p> 

Permission	Description
Manage File Stores	<p>Gives permission to create new File Store connections and view existing connections.</p> <p>The image below shows the view of the Administration menu that users have if Manage File Stores and Anzo Application are the only two applied permissions:</p>  <p>Note</p> <p>Manage File Stores does not grant permission to delete or change existing file store connections. The Administer System Setup permission is required in conjunction with Manage File Stores to be able to delete or edit existing file stores.</p>
Manage Query Blocklists	<p>Gives permission to create and remove queries from the Query Blocklist tab in the System Query Audit Log.</p> <p>Note</p> <p>If a user only has the Manage Query Blocklist permission, the Administration menu is not available. Use this permission in conjunction with Administer System Setup to grant access to System Query Audit and the Query Blocklist.</p>
Manage Semantic Services	<p>Gives permission to stop and start Semantic Services from the Semantic Services screen as well as view details about the services and use the Service Builder to generate and run semantic service requests.</p> <p>Note</p> <p>If a user only has the Manage Semantic Services permission, the Administration menu is not available. Use this permission in conjunction with Administer System Setup to grant access to the Semantic Services screen.</p>

Permission	Description
Manage Users, Groups, and Roles	<p>Gives permission to create, change, and delete Users, Groups, and Roles. A user who has this permission has Admin level access to all Users, Groups, and Roles. The image below shows the view of the Administration menu users have if Manage Users, Groups, and Roles and Anzo Application are the only two applied permissions:</p> 
Profile Data	Gives permission to Profile data sources, Datasets, and Graphmarts. Applying this permission exposes the Profile Data button on the Data Source, Dataset, and Graphmart screens.
Use Experimental Anzo Features	Grants permission use experimental Anzo features. Experimental features are recently implemented and may not be reliable for production use.
View Activity Logs	<p>Gives permission to view the Activity Log. Applying this permission exposes the Activity Log icon (🔄) in the top menu bar of the Anzo and Administration applications. The Anzo Application permission is needed to give access to the Anzo application.</p>
View Log Files	<p>Gives permission to view and download log files from the Log Files tab. Does not grant permission to change logging levels or add new log packages. Use this permission in conjunction with Administer System Setup to grant access to configure log levels and packages.</p> <p>The image below shows the view of the Administration menu that users have if View Log Files and Anzo Application are the only two applied permissions:</p> 

Related Topics

[User Management and Access Control Concepts](#)

[Creating and Managing Roles](#)

[Predefined Anzo Roles and Permissions](#)

[Sharing Access to Artifacts](#)

Managing Default Access Policies

Default Access Policies are the security policies that are applied by default to the artifacts that are stored in a particular **registry**. A registry is a system-level graph that stores metadata about artifacts of the same type. For example, metadata about all of your Data Source artifacts is stored in a Data Sources Registry, and metadata about all of your data Model artifacts is stored in an Ontology Registry. A Default Access Policy defines the base permissions to assign to a type of artifact when it is created—before permission inheritance and user-configured sharing is applied.

Note

Any **Permission Inheritance** that is applied by Anzo and artifact-level **Sharing** that is configured by users is applied to artifacts in addition to the permissions supplied by the Default Access Policy. For more information about permission inheritance and artifact sharing, see [Artifact Access Control Concepts](#).

This topic provides information about the permission sets that can be assigned to users and groups and describes the default access policies for each registry. This topic also includes instructions for changing access policies.

- [Default Access Policy Permissions Reference](#)
- [Default Access Policy Reference](#)
- [Configuring Default Access Policies](#)

Default Access Policy Permissions Reference

Default access policies use the same predefined permission sets and mechanism for assigning permissions as other artifacts in the Anzo application (see [Sharing Access to Artifacts](#) for more information).

There are three predefined permission sets that include a combination of six permissions that can be assigned to the creator of an artifact and other users and groups.

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"> • See an artifact in the Anzo application. • Create versions of the artifact.
	Meta View	<ul style="list-style-type: none"> • Relates only to an artifact's permissions. A user with Meta View can see the permissions on the artifact's Sharing tab but they cannot change permissions.
Modify	In addition to the View and Meta View permissions described above, the Modify set includes the Add/Edit and Delete permissions described below.	
	Add/Edit	<ul style="list-style-type: none"> • Change an artifact, such as to rename it or edit its description. • Add a related entity to an artifact. For example, add a Schema to a Data Source or a Data Layer to a Graphmart.
	Delete	<ul style="list-style-type: none"> • Remove a related entity from the artifact. For example, delete a Data Layer from a Graphmart or a Schema from a Data Source. • Does not give permission to remove the parent artifact. For example, a user can remove a Schema from a Data Source but cannot delete the Data Source.
Admin	In addition to the View , Meta View , Add/Edit , and Delete permissions described above, the Admin set includes the Meta Add/Edit and Meta Delete permissions described below.	
	Meta Add/Edit	<ul style="list-style-type: none"> • 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.
	Meta Delete	<ul style="list-style-type: none"> • Remove permissions from a user or group. • Delete the parent artifact and its related entities.

Default Access Policy Reference

There is a configurable Default Access Policy for several of the Anzo registries. To see and manage the Default Access Policies, go to the Administration application, expand the **User Management** menu, and click **Default Access Policies**.

Important

Never modify any of the Anzo registries. Changing or removing a registry can irreparably damage your Anzo server.

The sections below provide details about each of the registries for which you can configure Default Access Policies:

- [Role and Permissions Registry](#)
- [Data Sources Registry](#)
- [Graphmarts Registry](#)
- [SDI Registry](#)
- [Global Linked Data Configuration](#)
- [Linked Data Set Registry](#)
- [Elastic Search Configuration Registry](#)
- [Ontology Registry](#)
- [Query Builder Registry](#)

Role and Permissions Registry

The **Role and Permissions Registry** is a system graph that stores metadata about roles and permissions. Roles are not treated like other artifacts in Anzo. Unlike a Data Source, Model, or Graphmart artifact, for example, the permissions for a single Role or subset of Roles cannot be configured separately. Access to create and edit Roles is controlled by the **Manage Users, Groups, and Roles** permission. For more information, see [Role Permissions and Registries](#).

Default Permissions Configuration

- The **System Administrator** is assigned the [Admin](#) permission set for all role artifacts.
- The **Everyone** role is assigned the [View](#) permission set for all role artifacts.
- A member of a role that is assigned the **Manage Users, Groups, and Roles** permission has the [Admin](#) permission set for all role artifacts.

Data Sources Registry

The **Data Sources Registry** is the system graph that stores metadata about all of the **File Store**, **Anzo Data Store**, **Data Source**, and **Schema** artifacts that have been created in Anzo. Since Data Sources and Schemas have a fundamental relationship in that Schemas are imported from Data Sources, one registry stores metadata about both

types of artifacts. The Data Sources Registry access policy is applied by default when a user creates a Data Source or an Anzo Data Store.

Tip

In a typical onboarding scenario, a user creates a Data Source and then uses the Ingest workflow to generate the Model, Mapping, and Pipeline artifacts that are needed to ingest the source data and create a graph Dataset in the Dataset catalog. Since the artifacts created from the Ingest workflow inherit their permissions from the Data Source, the Data Sources Registry policy gets passed to the generated artifacts.

Default Permissions Configuration

- The **Creator** of a Data Source is assigned the [Admin](#) permission set for that Data Source and the associated Schemas. In addition, the Creator of an Anzo Data Store is also assigned the Admin permission set for that data store.
- The **Everyone** role is assigned the [View](#) permission set for a new Data Source and its Schemas. The Everyone role is also assigned the View permission set for any Anzo Data Stores.
- The **Creator Default Group** is assigned the [Modify](#) permission set for new Data Source, Schema, and Anzo Data Store artifacts.

Graphmarts Registry

The **Graphmarts Registry** is a system graph that stores metadata about all of the **Graphmart** artifacts in Anzo. Unlike the artifacts that are generated by the Ingest workflow, which inherit their permissions from the Data Source, all Graphmarts inherit permissions from the Graphmarts Registry Default Access Policy. In addition, Graphmarts contain Data Layers and Steps that describe and group the transformations that take place as the knowledge graph is generated. Since Data Layers and Steps are created in the context of a Graphmart, they inherit permissions from the Graphmart by default.

Default Permissions Configuration

- The **Creator** of a Graphmart is assigned the [Admin](#) permission set for that Graphmart artifact.
- The **Everyone** role is assigned the [View](#) permission set for that Graphmart.
- The **Creator Default Group** is assigned the [Modify](#) permission set for the Graphmart.

SDI Registry

The **SDI Registry** is a system graph that stores metadata about all of the **Mapping**, **Pipeline**, and **Job** artifacts that have been created in Anzo. This access policy is applied by default if a Mapping, Pipeline, or Job is manually created by a user. When Mappings, Pipelines, and Jobs are generated from the Ingest workflow, however, the artifacts inherit their permissions from the related Data Source.

Default Permissions Configuration

- The **Creator** of a Mapping, Pipeline, or Job is assigned the [Admin](#) permission set for that artifact.
- The **Everyone** role is assigned the [View](#) permission set for the new artifact.
- The **Creator Default Group** is assigned the [Modify](#) permission set for that artifact.

Global Linked Data Configuration

The **Global Linked Data Configuration Registry** is a global policy that applies to all artifacts created in Anzo—unless another Default Access Policy (such as the Data Sources Registry, Graphmarts Registry, or Ontology Registry) applies.

Example

If a user created a Model outside of the Ingest workflow and the Ontology Registry Default Access Policy was removed or unset, the Global Linked Data Configuration access policy would be applied to that Model artifact.

Default Permissions Configuration

- The **Creator** of an artifact that follows this policy is assigned the [Admin](#) permission set for that artifact.
- The **Creator Default Group** is assigned the [Modify](#) permission set for that artifact.

Linked Data Set Registry

The **Linked Data Set Registry** is a system graph that stores metadata about all of the Linked Data Sets, notably the File-Based Linked Data Sets (FLDS) that are listed in the **Datasets** catalog in the Anzo application. This includes Datasets that are generated from Pipelines as well as Datasets that are created by users, such as Empty Datasets for Export Steps and Existing RDF imports directly to the Dataset catalog.

Default Permissions Configuration

When the Ingest workflow is used to onboard data, the resulting FLDS artifact inherits its permissions from the Structured or Unstructured Pipeline that created it. If raw RDF files are imported to the Dataset catalog or an empty Dataset is created, the Linked Data Set Registry Default Access policy is applied to the resulting FLDS artifact.

Elastic Search Configuration Registry

The **Elastic Search Configuration Registry** is the system graph that stores metadata about all of the **Elasticsearch** connection artifacts in Anzo. This access policy is applied by default when an Elasticsearch connection is created.

Default Permissions Configuration

- The **Creator** of an Elasticsearch connection is assigned the [Admin](#) permission set for that artifact.
- The **Everyone** role is assigned the [View](#) permission set for that Elasticsearch connection artifact.
- The **Creator Default Group** is assigned the [Modify](#) permission set for that artifact.

Ontology Registry

The **Ontology Registry** is the system graph that stores metadata about all of the **Model** artifacts in Anzo. This access policy is applied by default if a Model is imported or manually created by a user. When a Model is generated from the Ingest workflow, however, the Model inherits the permissions from the related Data Source.

Default Permissions Configuration

- The **Creator** of a Model is assigned the [Admin](#) permission set for that Model artifact.
- The **Everyone** role is assigned the [View](#) permission set for that Model.
- The **Creator Default Group** is assigned the [Modify](#) permission set for that artifact.

Query Builder Registry

The **Query Builder Registry** is a system graph that stores metadata about Saved Query Builder queries. This access policy is applied by default when a new query is saved.

Default Permissions Configuration

The user who saves a query is assigned the [Admin](#) permission set. By default, saved queries are unique to each creator, and other users do not see the creator's queries.

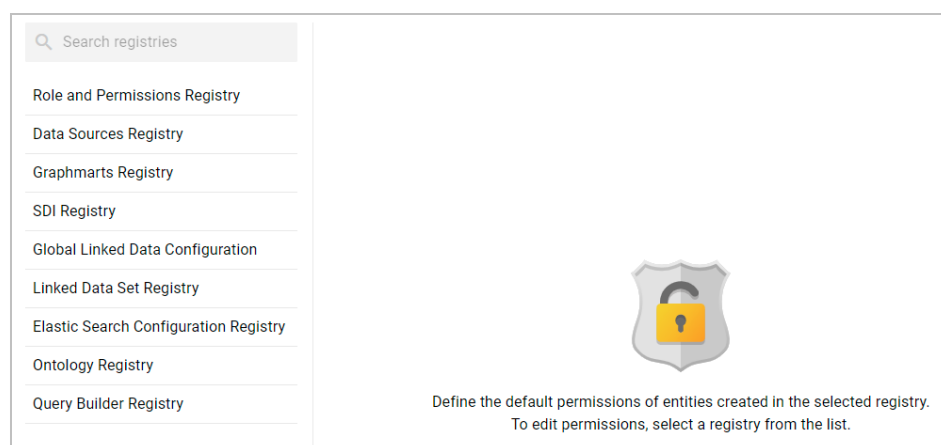
Configuring Default Access Policies

Follow the instructions below to change the default access policy for a registry.

Important

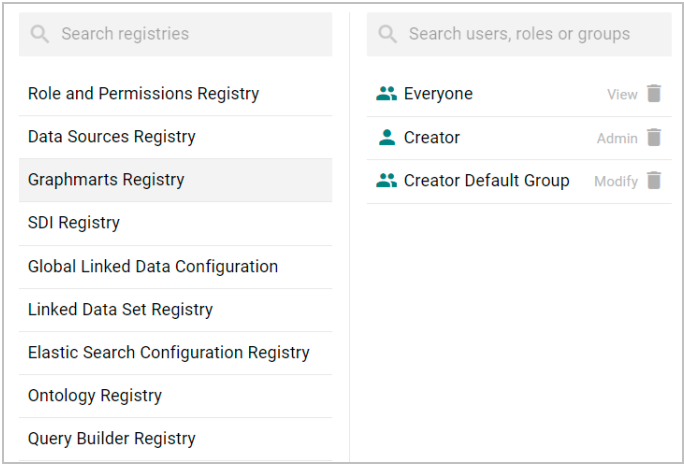
Changing default access control policies does not change permissions on any existing artifacts. The changes affect only new artifacts that are created after the change.

1. In the Administration application, expand the **User Management** menu and click **Default Access Policies**. The Default Access Policies screen is displayed.

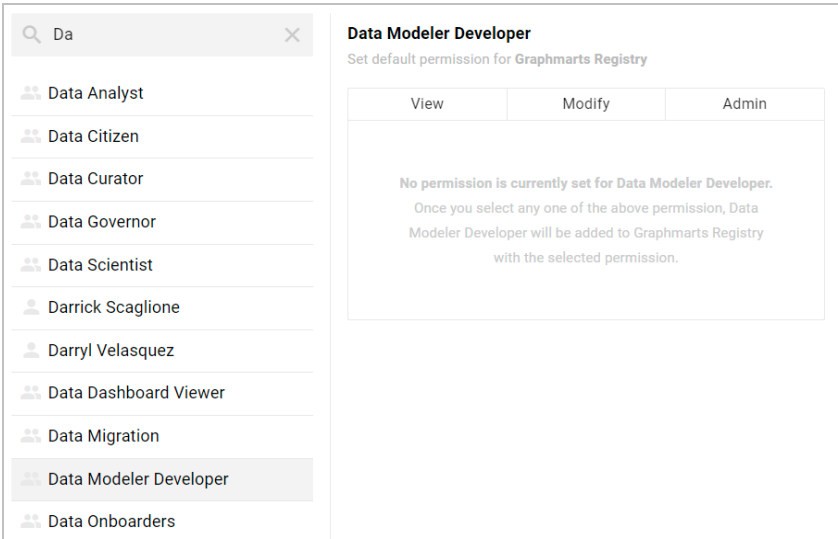


2. On the left side of the screen, select the access policy that you want to configure. The current configuration for that policy is shown on the right side of the screen. For example, the image below shows the Graphmarts

Registry. The Graphmart Creator has **Admin** permissions, the Everyone role has **View** permissions, and the Creator Default Group has **Modify** permissions.



3. To change a configured user or group, select a name in the list to view the permissions on the right side of the screen. To add a user or group, type a term in the **Search** field. Then select a name in the result list to view the permissions details. For example, the image below shows the search results for additional groups and selects the Data Modeler Developer group:



Note

Though Anzo is flexible and allows you to assign default access policies to roles, the recommendation is to control access to artifacts in a registry with users and groups. For more information, see [User Management Concepts](#).

4. On the right side of the screen, click the tab for the predefined permission set that you want to assign to the selected user or group. For information about the permission sets, see [Default Access Policy Permissions Reference](#) above. For example, the image below assigns the **Modify** permission set to the Data Modeler Developer group.

Da

Data Analyst

Data Citizen

Data Curator

Data Governor

Data Scientist

Darrick Scaglione

Darryl Velasquez

Data Dashboard Viewer

Data Migration

Data Modeler Developer

Modify

Data Onboarders

Data Modeler Developer

Set default permission for Graphmarts Registry

View

Modify

Admin

Add/Edit

View


Delete

Meta Add/Edit

Meta View

Meta Delete

Tip

To clear permissions for a user or group, click the trashcan icon () next to the user, role, or group name.

5. To configure additional users or groups, select the name and then repeat the step above to apply a permission set. Changes to access control policies are automatically saved.

Related Topics

[User Management and Access Control Concepts](#)

© 2023 Cambridge Semantics, Inc.

Monitoring and Diagnostics

The topics in this section provide information about monitoring events and managing Anzo and AnzoGraph diagnostic files.

- [Managing Anzo Logging](#)
- [Retrieving AnzoGraph Diagnostic Files](#)
- [Monitoring AnzoGraph](#)
- [System Query Audit](#)

Related Topics

[Enabling and Configuring the System Monitor Service](#)

[Viewing the Current Stack in a Browser](#)

Managing Anzo Logging

The topics in this section provide general information about logging in Anzo, instructions for adding logging for new components, changing the level or type of information that is logged, and reviewing log files. This section also provides guidance on enabling the recommended Log Packages.

- [Introduction to Anzo Logging](#)
- [Adding the Recommended Log Packages](#)

Introduction to Anzo Logging

This topic provides an introduction to Anzo logging concepts, an overview of the Logging user interface, and information about the type of logging that is enabled by default. It also gives a high-level overview about adding new logging, adjusting the level of information that is logged, and reviewing log files.

- [Logging Concepts](#)
- [Default Logging Configuration](#)
- [Adding Log Packages](#)
- [Log Level Definitions](#)
- [Viewing Log Files](#)

Logging Concepts

In order to give users granular control and flexibility over the type and breadth of information that is captured, the concept of **Log Packages** is integral to logging in Anzo. A Log Package is a listener for events that are related to a particular Semantic Service or component, such as core system, LDAP, Anzo Unstructured, or AnzoGraph events. To give users flexibility over the depth of information that is logged, each Log Package can be configured to capture events at a certain **Log Levels**, from all events to fatal events only.

Default Logging Configuration

Logging is managed in the Administration application. To view the Log Packages that are enabled for your server, expand the **Monitoring & Diagnostics** menu in the Administration application and click **Logging**. Then click the **Log Levels** tab to show the enabled Log Packages and their Log Level configuration. For example, the image below shows the default configuration for a new installation:

Log Files	
Log Levels	
Configure the log level of a package or add an additional package to log. Edit	
AccessAudit	INFO
ActivityAudit	INFO
AuditLog	ERROR
com.cambridgesemantics	ERROR
InstallUpdateLog	INFO
org.apache.directory	OFF
org.openanzo	ERROR
org.openanzo.client.registry.RegistryManifestLoader	INFO
org.openanzo.combus.endpoint.BaseServiceListener	ERROR
org.openanzo.osgi.bootstrap.BootstrapActivator	INFO
org.pac4j.http.client.direct.DirectBasicAuthClient	OFF
org.pac4j.http.client.direct.HeaderClient	OFF
QueryAudit	INFO
SystemAudit	INFO
TimingStack	ERROR
UserAudit	INFO

Default Log Packages

The table below describes Log Packages that are enabled by default as well as their default Log Level. Log Levels are defined in [Log Level Definitions](#) below.

Package	Level	Description
AccessAudit	Info	Listener for access audit events such as user login attempts.
ActivityAudit	Info	Listener for activity audit events.

Package	Level	Description
AuditLog	Error	Logger for audit events when the appropriate packages are enabled. For more information, see Enabling and Viewing Audit Logs .
com.cambridgesemantics	Error	Like the org.openanzo package, this base package listens for core system events. Changing the Log Level of this package affects logs across Anzo components and services.
InstallUpdateLog	Info	Listener for installation and upgrade events. Captures information about bundle imports and updates.
org.apache.directory	Off	Listener for events related to the underlying internal LDAP server. Do not modify the Log Level for this package.
org.openanzo	Error	Like the com.cambridgesemantics package, this base package listens for core system events. Changing the Log Level of this package affects logs across Anzo components and services.
org.openanzo.client.registry.RegistryManifestLoader	Info	Listener for installation and upgrade events. Captures information about bundle imports and updates.
org.openanzo.combus.endpoint.BaseServiceListener	Error	Core server listener for requests sent from clients to the server.
org.openanzo.osgi.bootstrap.BootstrapActivator	Info	Listener for installation and upgrade events. Captures information about bundle imports and updates.


Package	Level	Description
org.openanzo.services.PublicLog	Off	Listener for internal Anzo events. Do not modify the Log Level for this package.
org.pac4j.http.client.direct.DirectBasicAuthClient	Off	Low-level listener for user login events.
org.pac4j.http.client.direct.HeaderClient	Off	Low-level listener for user login events.
QueryAudit	Info	Listener for query audit events.
SystemAudit	Info	Listener for system audit events such as changes to bundle properties.
TimingStack	Error	Listener for events related to internal system journal queries.
UserAudit	Info	Listener for user administration related events, such as changes to roles.

Adding Log Packages

Tip

For guidance on adding the recommended Log Packages, see [Adding the Recommended Log Packages](#).

To enable additional Log Packages, click the **Edit** button on the Log Levels screen.

Log Files	
Log Levels	
Configure the log level of a package or add an additional package to log.  Edit	
AccessAudit	INFO
ActivityAudit	INFO
AuditLog	ERROR
com.cambridgesemantics	ERROR
InstallUpdateLog	INFO
org.apache.directory	OFF
org.openanzo	ERROR
org.openanzo.client.registry.RegistryManifestLoader	INFO
org.openanzo.combus.endpoint.BaseServiceListener	ERROR
org.openanzo.osgi.bootstrap.BootstrapActivator	INFO
org.pac4j.http.client.direct.DirectBasicAuthClient	OFF
org.pac4j.http.client.direct.HeaderClient	OFF
QueryAudit	INFO
SystemAudit	INFO
TimingStack	ERROR
UserAudit	INFO

Then click **Add Package** at the bottom of the screen.

Select... 



 Add Package

Clicking the **Select** field opens the package drop-down list. You can browse through the options, or you can start typing a keyword to search for a package. Click a package to add it to the list of packages on the Edit Log Packages screen. Adjust the Log Level as needed and then click **Save** to save the change. See [Log Level Definitions](#) below for more information about Log Levels.

Log Level Definitions

This section defines the Log Levels that are available to apply to a Log Package:

- **Off:** Turns logging off for the Log Package.
- **Debug:** Logs fine-grained error messages that are intended to help debug a problem with an application or the server.
- **Trace:** Logs finer-grained error information than Debug.
- **Info:** The highest level of logging. The Log Package captures all events or queries.
- **Warn:** Logs information about potentially problematic situations.
- **Error:** Logs errors that usually allow the application to continue running.
- **Fatal:** Logs severe errors that prevent the application from running.

To change the Log Level for a package, click the **Edit** button at the top of the screen. On the Edit Log Packages screen, click the **Log Level** field for the Log Package that you want to change and select a level from the drop-down list. Click **Save** when you are finished making changes.

Edit Log Packages

org.openanzo.client.registry.RegistryManifestLoader	Info	
org.pac4j.http.client.direct.HeaderClient	Off	
AuditLog	Error	
InstallUpdateLog	Info	
org.apache.directory	Off	
TimingStack	Error	
org.openanzo	Error	
org.pac4j.http.client.direct.DirectBasicAuthClient	Off	
com.cambridgesemantics	Error	
org.openanzo.osgi.bootstrap.BootstrapActivator	Info	
org.openanzo.combus.endpoint.BaseServiceListener	Error	
org.openanzo.services.PublicLog	Off	

+ Add Package

CANCEL
SAVE

Viewing Log Files

All Anzo log files are generated in the `<install_path>/Server/logs` directory on the server. Files in that directory can be viewed and downloaded from the Administration application on the **Log Files** tab on the Logging screen.

- [Viewing Logs on the Server](#)
- [Viewing Logs in the Administration Application](#)

Viewing Logs on the Server

To avoid generating large log files that are difficult to manage (especially for Log Packages set to **Info**), Anzo starts logging to a new version of a file when any of the following events occur:

- A file size reaches 50 MB.
- Log settings are changed.
- Anzo is restarted.

The current, most recent version of a file is stored directly in the `<install_path>/Server/logs` directory. Earlier versions of the files are saved in `<year>_<month>_<day>_<part>` subdirectories in `Server/logs`. For example:

```
logs
├── 2021_04_27_0
│   ├── anzo_audit_info.log
│   ├── anzo_error.log
│   ├── anzo_full.log
│   ├── anzo_gqe_info.log
│   └── anzo_internal_error.log
├── 2021_04_27_1
│   ├── anzo_audit_info.log
│   ├── anzo_datasource_error.log
│   ├── anzo_error.log
│   ├── anzo_full.log
│   ├── anzo_gqe_error.log
│   ├── anzo_gqe_info.log
│   ├── anzo_install_error.log
│   └── anzo_install_info.log
├── 2021_04_28_0
│   ├── anzo_audit_info.log
│   ├── anzo_error.log
│   ├── anzo_full.log
│   ├── anzo_gqe_info.log
│   ├── anzo_install_error.log
│   └── anzo_install_info.log
├── 2021_04_28_1
│   ├── anzo_error.log
│   └── anzo_full.log
├── 2021_04_28_2
│   ├── anzo_audit_info.log
│   ├── anzo_error.log
│   └── anzo_full.log
├── anzo_audit_info.log
├── anzo_error.log
├── anzo_full.log
├── anzo_gqe_info.log
└── anzo_install_error.log
```

```
└─ anzo_install_info.log
└─ anzo_internal_error.log
```

AnzoGraph query log files are stored in a directory named **gqe** in the `<install_path>/Server/logs` directory. By default all queries that are unsuccessful are captured in the **queriesError** directory. When the AnzoGraph queries Log Package is enabled, successful queries are also captured in the **queriesInfo** directory. For example:

```
logs
└─ gqe
  └─ queriesError
  └─ queriesInfo
    └─ query_1a5548ac-6404-4321-b36b-d5eda4ca45a7_1619540406734.log
    └─ query_1a5548ac-6404-4321-b36b-d5eda4ca45a7.log
    └─ query_292f102e-d222-4261-a069-d7d0c8ceb823_1619469563646.log
    └─ query_292f102e-d222-4261-a069-d7d0c8ceb823.log
    └─ query_2ddc5f96-758d-4133-80d7-21de5f23134f_1619627154151.log
    └─ query_2ddc5f96-758d-4133-80d7-21de5f23134f.log
    └─ query_518ombnsruiyvu8k6pf0a76y4fc-674.log
```

Tip

For instructions on enabling the AnzoGraph query Log Package, see [Enabling and Viewing AnzoGraph Query Logs](#).

Viewing Logs in the Administration Application

Logs in the `<install_path>/Server/logs` directory can be viewed and downloaded from the Administration application on the **Log Files** tab on the Logging screen. The Log Files tab lists the logs that are available to view. For example:

Log Files		Log Levels
<input type="text" value="Search"/> <input type="checkbox"/> Show All File Types <button>Download All Logs</button> <input type="button" value="Download All AnzoGraph Query Errors"/>		
File	Size	Modified
anzo_error.log	4.9 KB	6/3/21 4:09 PM
anzo_full.log	34.6 KB	6/3/21 4:09 PM
anzo_install_info.log	29.7 KB	6/3/21 4:08 PM
error.log	2.0 KB	6/3/21 4:07 PM
output.log	173.0 B	6/3/21 4:09 PM
2021_06_01_0/anzo_anzowt_error.log	334.0 B	5/31/21 4:56 PM
2021_06_01_0/anzo_combus_error.log	904.0 B	5/31/21 9:17 PM
2021_06_01_0/anzo_datasource_error.log	2.2 KB	5/31/21 5:01 PM
2021_06_01_0/anzo_error.log	2.1 KB	5/31/21 9:17 PM
Rows per page: 25 1-25 of 221		

Log Packages that have the Log Level set to **Error** log events to files with the suffix **_error**. Operational information that is logged by packages that are set to **Info** is captured in files with the suffix **_info**. The current versions the log files are shown at the top of the list. Earlier versions of the logs are prefixed with the name of the <date>_<part> subdirectory they are saved in.

Selecting a log from the list displays its contents in the Logging Details section of the screen. For example:

Log Files			Log Levels
<input type="text" value="Search"/> <input type="checkbox"/> Show All File Types <button>Download All Logs</button> <input type="button" value="Download All AnzoGraph Query Errors"/>			
File	Size	Modified	
anzo_error.log	4.9 KB	6/3/21 4:09 PM	<div>Logging Details</div> <div> <input type="button" value="REFRESH"/> <input type="button" value="DOWNLOAD FILE"/> </div> <div>File: anzo_error.log Size: 4.9 KB Modified: 6/3/21 4:09 PM</div> <pre> 2021-06-03 16:09:04,806 ERROR [] [/DataTemplateService] - [OpName=executeService] [OpId=51tyvtdehxl9ib3mitr8gpj-108] [OpUser=http://openanzo.org/sys tem/internal/sysadmin] c.c.a.u.d.DataTemplateService: Error in follow request: org.openanzo.exceptions.AnzoRuntimeException: ErrorCode[4397] No colon in URI: [u ndefined] at org.openanzo.rdf.MemURI.<init>(MemURI.java:94) at org.openanzo.rdf.MemValueFactory.createURI(MemValueFactory.java:348) at org.openanzo.rdf.MemURI.create(MemURI.java:58) at org.openanzo.glitter.query.QueryController.resolveUri(QueryController. java:1282) at org.openanzo.glitter.syntax.concrete.SPARQLParser.IRIref(SPARQLParser. java:4912) at org.openanzo.glitter.syntax.concrete.SPARQLParser.GraphTerm(SPARQLPar ser.java:3418) at org.openanzo.glitter.syntax.concrete.SPARQLParser.VarOrTerm(SPARQLPar ser.java:3528) at org.openanzo.glitter.syntax.concrete.SPARQLParser.TriplesSameSubjectPa th(SPARQLParser.java:2698) at org.openanzo.glitter.syntax.concrete.SPARQLParser.ConstructTriples(SPA RQLParser.java:2587) at org.openanzo.glitter.syntax.concrete.SPARQLParser.ConstructGraphTriple s(SPARQLParser.java:2684) at org.openanzo.glitter.syntax.concrete.SPARQLParser.ConstructTemplateCon tents(SPARQLParser.java:2539) at org.openanzo.glitter.syntax.concrete.SPARQLParser.ConstructTemplate(SPA RQLParser.java:2532) at org.openanzo.glitter.syntax.concrete.SPARQLParser.ConstructQuery(SPARQ LParser.java:427) at org.openanzo.glitter.syntax.concrete.SPARQLParser.Query(SPARQLParser.j ava:88) at org.openanzo.glitter.Engine.prepareQueryInternal(Engine.java:389) at org.openanzo.glitter.Engine.prepareQuery(Engine.java:141) at com.cambridgesemantics.anzo.utilityservices.datatemplate.DataTemplateS ervice.followConstructRequest(DataTemplateService.java:2155) at com.cambridgesemantics.anzo.utilityservices.datatemplate.DataTemplateS ervice.followRequest(DataTemplateService.java:1991) at com.cambridgesemantics.anzo.utilityservices.datatemplate.DataTemplateS ervice.processRequest(DataTemplateService.java:968) at com.cambridgesemantics.anzo.utilityservices.datatemplate.DataTemplateS ervice.getDataInternal(DataTemplateService.java:588) at com.cambridgesemantics.anzo.utilityservices.datatemplate.DataTemplateS ervice.lambda\$(DataTemplateService.java:392) at org.openanzo.execution.v2.ContextTransactionTemplate.lambda\$1(ContextT ransactionTemplate.java:317) at org.openanzo.client.AnzoClientTransactionTemplate.lambda\$0(AnzoClientT ransactionTemplate.java:363) </pre>

The following options are available for viewing and downloading log files:

- To download a .zip file that contains all of the listed logs, click the **Download All Logs** button at the top of the screen.
- To download just the query error logs for AnzoGraph, click the **Download All AnzoGraph Query Errors** button at the top of the screen.
- To re-load the display with the latest version of the selected file, click the **Refresh** button at the top of the details.
- To download the file so you can view it in another editor, click **Download File** at the top of the details.

Related Topics

[Adding the Recommended Log Packages](#)

Adding the Recommended Log Packages

The Log Packages that are enabled by default cover the core Anzo server operations and services to ensure that diagnostics are generated when errors occur. Anzo includes several additional Log Packages, however, that are disabled by default but can be configured to provide valuable information for auditing purposes, such as information about user logins, user administration events, and successful AnzoGraph queries. This section describes the packages that Cambridge Semantics recommends you enable and provides information about reading the resulting log files.

- [Enabling and Viewing AnzoGraph Query Logs](#)
- [Enabling and Viewing Audit Logs](#)

Enabling and Viewing AnzoGraph Query Logs

The GqeQueries Log Package listens for AnzoGraph events like connection errors, restarts, and successful and unsuccessful queries. GqeQueries is Off by default but can be enabled to monitor and log all of the queries that are sent to AnzoGraph by users through Dashboards, the Query Builder, Data Layers, etc., or sent by Anzo, such as when requesting the total number of statements in a graph.

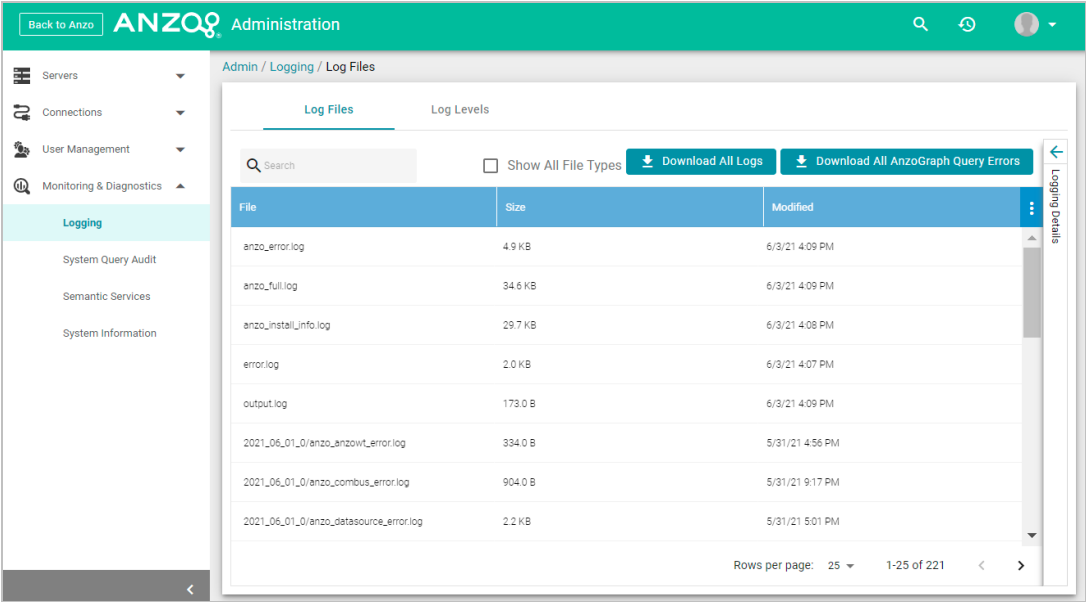
Note

Though GqeQueries is Off by default, AnzoGraph query errors are still captured automatically in the `<install_path>/Server/logs/gqe/queriesError` directory, and connection-related errors are captured in `anzo_gqe_error.log`.

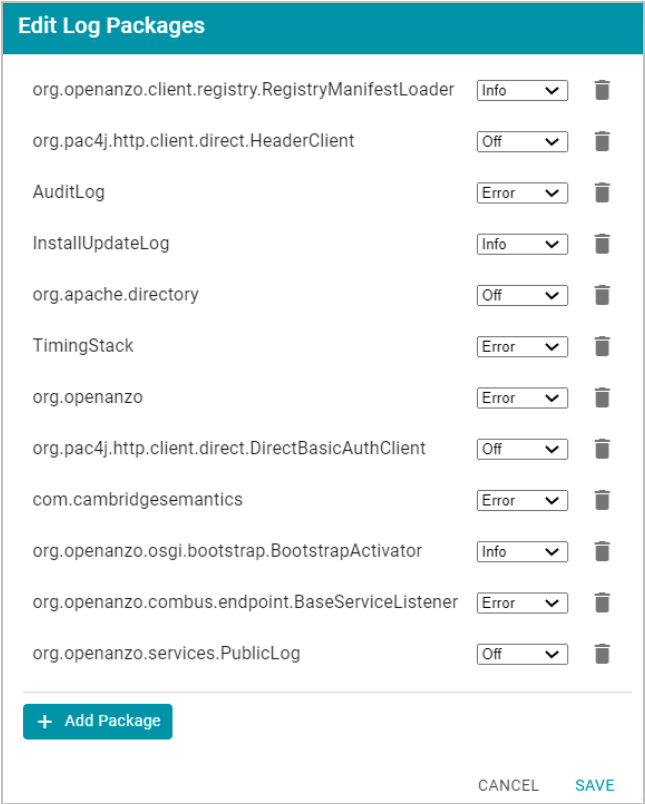
Enabling the GqeQueries Log Package

Follow the steps below to enable the GqeQueries package.

1. In the Administration application, expand the **Monitoring & Diagnostics** menu and select **Logging**. The Log Files tab is displayed on the Logging screen. For example:



2. Click the **Edit** button to open the Edit Log Packages dialog box.



3. Click **Add Package** at the bottom of the screen. The Select field is displayed:



- Click the **Select** field and type **GqeQueries**. Then press **Enter** to add GqeQueries to the list of Log Packages. The package is added to the list with the default Log Level of **Off**.
- Click the Log Level drop-down list and select **Info**. Then click **Save** to save the change.

The screenshot shows a configuration dialog for the 'GqeQueries' log package. At the top, the package name 'GqeQueries' is displayed next to a dropdown menu set to 'Info' and a trash icon. Below this is a teal button labeled '+ Add Package'. At the bottom right, there are 'CANCEL' and 'SAVE' buttons.

The GqeQueries Log Package is now enabled and will start to log the events described above. The log messages for successful queries are captured in a new **anzo_gqe_info.log** file as well as in the `<install_path>/Server/logs/gqe/queriesInfo` directory on the server. Details about each request is logged to a separate file in that directory. The **anzo_gqe_info.log** and the files in `logs/gqe/queriesInfo` can be viewed and downloaded from the Administration application.

Viewing the AnzoGraph Query Logs

Follow the steps below to view the AnzoGraph log files in the application. For information about viewing logs on the server, see [Viewing Logs on the Server](#).

- In the Administration application, expand the **Monitoring & Diagnostics** menu and select **Logging**. The Log Files tab is displayed on the Logging screen. For example:

The screenshot shows the ANZO Administration application interface. The left sidebar has a 'Monitoring & Diagnostics' menu expanded, with 'Logging' selected. The main content area is titled 'Admin / Logging / Log Files' and shows a table of log files. The table has columns for 'File', 'Size', and 'Modified'. Below the table, there are pagination controls showing 'Rows per page: 25' and '1-25 of 221'.

File	Size	Modified
anzo_error.log	4.9 KB	6/3/21 4:09 PM
anzo_full.log	34.6 KB	6/3/21 4:09 PM
anzo_install_info.log	29.7 KB	6/3/21 4:08 PM
error.log	2.0 KB	6/3/21 4:07 PM
output.log	173.0 B	6/3/21 4:09 PM
2021_06_01_0/anzo_anzowl_error.log	334.0 B	5/31/21 4:56 PM
2021_06_01_0/anzo_combus_error.log	904.0 B	5/31/21 9:17 PM
2021_06_01_0/anzo_datasource_error.log	2.2 KB	5/31/21 5:01 PM

- Click the **Log Files** tab to view the list of files. For example:

Log Levels		Log Files	
<input type="text" value="Search"/>		Download All Logs	Download All AnzoGraph Query Errors
File	Size	Modified	
/anzo_audit_info.log	4.4 KB	4/28/21 4:22 PM	
/anzo_error.log	5.1 KB	4/28/21 4:34 PM	
/anzo_full.log	46.0 KB	4/28/21 4:34 PM	
/anzo_gqe_info.log	4.7 KB	4/28/21 4:25 PM	
/anzo_install_error.log	2.0 KB	4/28/21 4:09 PM	
/anzo_install_info.log	29.2 KB	4/28/21 4:08 PM	
/anzo_internal_error.log	604.0 B	4/28/21 4:25 PM	

Log Packages that have the Log Level set to **Error** log events to files with the suffix **_error**. Operational information that is logged by packages that are set to **Info** is captured in files with the suffix **_info**.

Note

The current version of **anzo_gqe_info.log** is shown toward the top of the list. Earlier versions of that log are prefixed with the name of the <date>_<part> subdirectory they are saved in. And individual query files are named as /gqe/queriesInfo/<operation_ID><epoch_timestamp>.

- Select the **anzo_gqe_info.log** file. The contents of the file are displayed in the Logging Details section of the screen. For example:

Log Levels
Log Files

Q Search

Download All Logs

Download All AnzoGraph Query Errors

File	Size	Modified
/anzo_audit_info.log	5.7 KB	4/28/21 9:43 PM
/anzo_error.log	24.6 KB	4/28/21 9:44 PM
/anzo_full.log	68.1 KB	4/28/21 9:44 PM
/anzo_gqe_info.log	5.9 KB	4/28/21 9:44 PM
/anzo_install_error.log	2.0 KB	4/28/21 4:09 PM
/anzo_install_info.log	29.2 KB	4/28/21 4:08 PM
/anzo_internal_error.log	604.0 B	4/28/21 4:25 PM
/output.log	173.0 B	4/28/21 4:10 PM
/2021_04_26_0/anzo_anzowt_error.log	337.0 B	4/26/21 7:02 PM
/2021_04_26_0/anzo_error.log	31.2 KB	4/26/21 7:27 PM
/2021_04_26_0/anzo_execution_error.l...	623.0 B	4/26/21 7:02 PM

Logging Details

REFRESH DOWNLOAD FIL

File: /anzo_gqe_info.log | Size: 5.9 KB | Modified: 4/28/21 9:44 PM

```

2021-04-28 16:09:08,787 INFO [gqe] [PriorityQueue-pool-2] - GqeQuer
es-
http://cambridgesemantics.com/GqeDatasource/guid_e1f38b640fe04bf8fee71bdf518
4bf41
#
# OperationId: 7180f217-b01a-4d7a-8e0f-1b2dc518e1a6
# datasourceUri=[http://cambridgesemantics.com/GqeDatasource/guid_e1f38b640f
e04bf8fee71bdf5184bf41]
# UserURI: http://openanzo.org/system/internal/sysadmin
# Timestamp: Apr 28, 2021 4:09:08 PM
#
# operationId = [7180f217-b01a-4d7a-8e0f-1b2dc518e1a6]
# userUri = [http://openanzo.org/system/internal/sysadmin]
#
SELECT
    ?type
    (COUNT(?s) AS ?count)
FROM <http://cambridgesemantics.com/Layer/f44db5d106ca4186b953a591e873a5f0>
FROM NAMED <http://cambridgesemantics.com/Layer/f44db5d106ca4186b953a591e873
a5f0>
WHERE {
    ?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> ?type .
}
GROUP BY ?type

2021-04-28 16:09:08,854 INFO [gqe] [PriorityQueue-pool-2] - GqeQuer
es- QueryResults:62 [136]: 11
2021-04-28 16:09:08,944 INFO [gqe] [PriorityQueue-pool-3] - GqeQuer
es-
http://cambridgesemantics.com/GqeDatasource/guid_e1f38b640fe04bf8fee71bdf518
4bf41

```

You can expand the details view by clicking the Expand icon (⌵) in the top right corner.

The messages in **anzo_gqe_info.log** vary by the query source, such as whether the query originated in a Dashboard lens or the Query Builder. In general, GqeQueries Info messages contain the following information:

- Date and time the event was logged. For example, 2021-04-28 01:06:48.
- The type of message, i.e., the Log Level, such as INFO.
- The type of log. For example, [gqe].
- The area of the system or service that processed the event. For example, [PriorityQueue-pool-2].
- The Log Package that was listening for the event, i.e., GqeQueries.
- The Data Source URI. For example, `http://cambridgesemantics.com/GqeDatasource/guid_e1f38b640fe04bf8fee71bdf5184bf41`.
- The Operation ID assigned to the query. This value can be used to track the query, such as to find the individual log file in the `logs/gqe/queriesInfo` directory. For example, `OperationId: 7b0op0w-bzqeqls2d482xudkez-83`. The corresponding log file is named `query_7b0op0w-bzqeqls2d482xudkez-83.log`.
- The User URI for the user who submitted the query. For example, `UserURI: ldap:///cn=n=Jay.Blue,ou=groups,dc=com`.
- If the query was submitted from the Hi-Res Analytics application, the message also includes details for identifying the dashboard and lens that submitted the request. For example:

```
# ex_requestDashboard = [http://cambridgesemantics.com/354db630-02b6-46b2-82d0-ef4a7543ebca]
# ex_requestSource = [http://cambridgesemantics.com/4a039bdb-bdcb-4117-830b-cb29190ce18f]
# ex_requestSourceId = [com_cambridgesemantics_application_anzoweb_lens_grid_GridLens_7]
```

- The text of the query that was sent by Anzo. Note that the text is the query as rewritten by Anzo and sent to AnzoGraph. It may not be the exact text that was written by the user.
- When a query returns, a result message is also added to anzo_gqe_info.log below the query text. The QueryResults message includes the Operation ID (which matches the ID from the query that was sent), and it returns the AnzoGraph and Anzo query execution time as well as the number of results returned. In the following example, the QueryResults message is shown in bold. The first value (**2631**) is the number of milliseconds AnzoGraph spent executing the query. The value in brackets (**[13155]**) is the number of milliseconds Anzo spent executing the query. And the last value (**20**) is the number of results that were returned.

```
2021-04-28 22:53:57,134 INFO [gqe] [PriorityQueue-pool-7] - [OpName=query]
[OpId=8ttl1rrc29y31z1ga30srk6t2xx-212]
[OpUser=http://openanzo.org/system/internal/sysadmin]
GqeQueries- QueryResults:2631 [13155]: 20
```

Note A QueryResults message is not logged if the query uses the Anzo cache or returns an error.

A complete example message is shown below:

```
2021-04-27 19:54:25,648 INFO [gqe] [PriorityQueue-pool-2] - GqeQueries-
http://cambridgesemantics.com/GqeDatasource/guid_elf38b640fe04bf8fee71bdf5184bf41
# *****
# OperationId: 66ed1f10-5aae-45b0-861c-3a851022d294
# datasourceUri=[http://cambridgesemantics.com/GqeDatasource/guid_elf38b640fe04bf8fee71bdf5184bf41]
# UserURI: http://openanzo.org/system/internal/sysadmin
# Timestamp:Apr 27, 2021 7:54:25 PM
#
# operationId = [66ed1f10-5aae-45b0-861c-3a851022d294]
# userUri = [http://openanzo.org/system/internal/sysadmin]
# *****
SELECT
    ?type
    (COUNT(?s) AS ?count)
FROM <http://cambridgesemantics.com/Layer/f44db5d106ca4186b953a591e873a5f0>
FROM NAMED <http://cambridgesemantics.com/Layer/f44db5d106ca4186b953a591e873a5f0>
WHERE {
    ?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> ?type .
```

```

}
GROUP BY ?type
2021-04-27 19:54:25,670 INFO [gqe] [PriorityQueue-pool-2] - GqeQueries-
QueryResults:16 [100]: 11

```

Related Topics

[Introduction to Anzo Logging](#)

[Enabling and Viewing Audit Logs](#)

[Retrieving AnzoGraph Diagnostic Files](#)

Enabling and Viewing Audit Logs

The Audit Log Packages listen for user- or security-related events such as access attempts and user administration-related events such as modifications to users, groups, and roles. The Audit Log packages are disabled by default but can be enabled to monitor and log the following types of events:

- The inactivity timeout is changed.
- A bundle's properties are changed.
- A user successfully logs in or out or there are failed login attempts.
- A user account is created or deleted or a user's password is changed.
- A user or group is synchronized with the directory server.
- A role is created or deleted.
- A user is added to or removed from a role or group.
- A permission is added to or removed from a role.
- Data access permissions are changed on artifacts.

Enabling the Audit Log Packages

By default, the Audit Log packages (UserAudit, AccessAudit, QueryAudit, ActivityAudit, and SystemAudit) are set to the Log Level **Info**, which means they are configured to capture all audit events. However, logging the audit events are disabled by default in the Anzo Audit Logging Framework service. Follow the instructions below to configure the service to enable audit logging:

1. In the Administration application, expand the **Servers** menu and click **Advanced Configuration**. Click **I understand and accept the risk**.
2. Search for the **Anzo Audit Logging Framework** bundle and view its details.
3. Click the **Services** tab and expand **com.cambridgesemantics.anzo.AuditLog**.
4. Find the **com.cambridgesemantics.anzo.auditlog.standardLog** property towards the bottom of the list (shown below).

com.cambridgesemantics.anzo.auditlog.rdfLogDir
 \${system.ANZO_SERVER_HOME}/logs/audit/flds/

☐ com.cambridgesemantics.anzo.auditlog.runningQueriesFile

☒ com.cambridgesemantics.anzo.auditlog.splitByType

☒ com.cambridgesemantics.anzo.auditlog.splitFlds

☐ com.cambridgesemantics.anzo.auditlog.standardLog

☒ com.cambridgesemantics.anzo.auditlog.systemEvents

com.cambridgesemantics.anzo.auditlog.trackedGraphs
 None

☒ com.cambridgesemantics.anzo.auditlog.transactionEvents

☒ com.cambridgesemantics.anzo.auditlog.userEvents

- Click the property to make it editable, and then select the checkbox to enable it.

☒ com.cambridgesemantics.anzo.auditlog.standardLog

☒ ☐

- Click the checkmark icon (✓) to save the change.
- Restart Anzo to apply the configuration change.

The Audit Log Packages are now enabled and will start to log the events described above. The log messages are captured in **anzo_full.log** as well as a new file called **anzo_audit_info.log**. All Anzo log files are generated in the `<install_path>/Server/logs` directory on the server. Files in that directory can be viewed and downloaded from the Administration application.

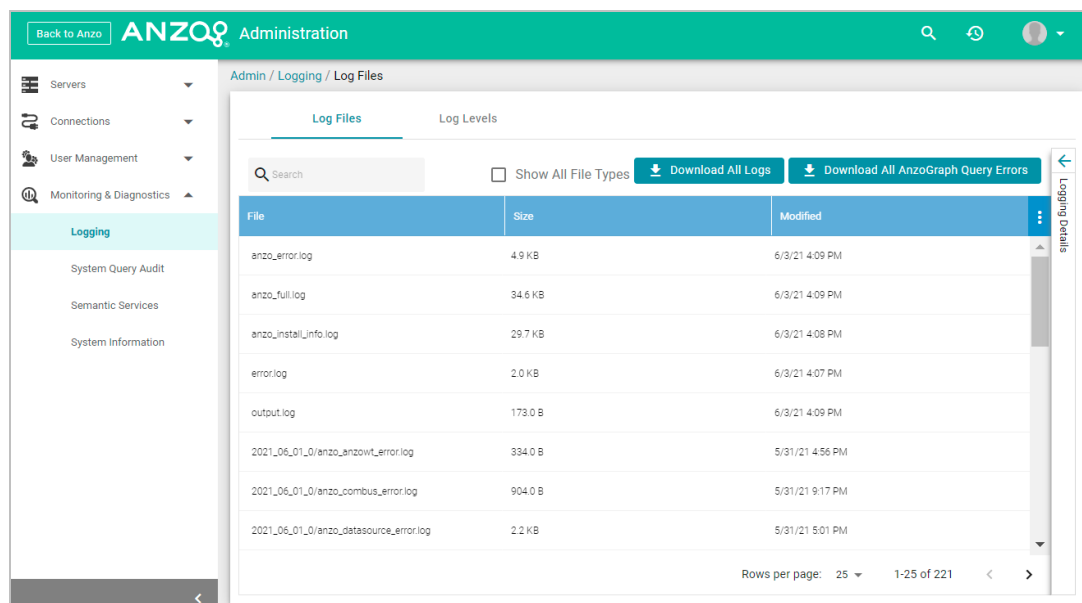
Viewing the Audit Log

Follow the steps below to view the Audit log file in the application. For information about viewing logs on the server, see [Viewing Logs on the Server](#).

Tip

You have the option to split the Audit log into separate files based on the type of event that is being logged, such as a user event or a query event. See [Separating Audit Logs by Type of Event](#) for information. The steps below refer to the default Audit Log where all types of audit events are recorded in a single file.

- In the Administration application, expand the **Monitoring & Diagnostics** menu and select **Logging**. The Log Files tab is displayed on the Logging screen. For example:



Log Packages that have the Log Level set to **Error** log events to files with the suffix **_error**. Operational information that is logged by packages that are set to **Info** is captured in files with the suffix **_info**. The current versions of the log files are shown at the top of the list. Earlier versions of the logs are prefixed with the name of the <date>_<part> subdirectory they are saved in.

2. Select the **anzo_audit_info.log** file. The contents of the file are displayed in the Logging Details section of the screen. For example:

You can expand the details view by clicking the Expand icon (⌵) in the top right corner.

The elements included in each message vary by message type. In general, UserAudit Info messages contain the following information:

- Date and time the event was logged. For example, 2021-04-28 01:06:48.
- The type of message, i.e., the Log Level, such as INFO.
- The type of log. For example, [audit].
- The area of the system or service that processed the event. For example, [UniformSaveService].
- The Log Package that was listening for the event, i.e., UserAudit.
- The message text, such as User Connected or Authentication Failed.
- The unique Operation ID assigned for the operation. For example, [OpId=518ombnsruyv8k6pf0a76y4fc-1414].
- The name of the service that performed the operation. For example, [OpName=executeService].
- The user who performed the operation. For example, [OpUser=r=http://openanzo.org/system/internal/sysadmin] .

Below are examples of the types of messages that are logged (line breaks added for readability):

Successful User Login

```
2021-04-27 16:12:28,754 INFO [audit] [persistent=false#1-1] - UserAudit-
User Connected:sysadmin:<http://openanzo.org/system/internal/sysadmin>,
ConnectionId:ID:anzo-36673-1619539948446-4:1,
RemoteAddress:vm://localhost?broker.persistent=false#0
```

Failed User Login

```
2021-04-28 01:06:48,341 INFO [audit] [erverThreadPool-3323] -
[OpName=ServerRealm.Authenticate]
[OpId=a876f781-5ddf-424d-8d54-c2ea07c87561]
UserAudit-
Authentication Failed:test,
Message:ErrorCode[3844] User test not found.
```

Inactivity Timeout Value Changed

```
2021-04-27 19:50:17,316 INFO [audit] [Service Update Queue] -
[OpName=executeService]
[OpId=518ombnsruyv8k6pf0a76y4fc-1802]
[OpUser=http://openanzo.org/system/internal/sysadmin]
UserAudit- Inactivity Logout Timeout Changed: Old=-1 New=900000
```

New Role Created

```
2021-04-27 18:58:38,276 INFO [audit] [r/UniformSaveService] -
[OpName=executeService]
[OpId=518ombnsruyv8k6pf0a76y4fc-1414]
[OpUser=http://openanzo.org/system/internal/sysadmin]
```



```
UserAudit-
Role Created: <http://cambridgesemantics.com/Role/952810ffb74a42f8b502adc422608e64>
```

Permission Added to a Role

```
2021-04-28 20:41:10,926 INFO [audit] [r/UniformSaveService] -
[OpName=executeService]
[OpId=5q6p7zmp9xn2xujks417pzzl-1808]
[OpUser=http://openanzo.org/system/internal/sysadmin]
UserAudit-
Permission <http://cambridgesemantics.com/permissions/feature/e5c11e5b-afb2-4af0-b1d7-
0e4b620a0378>
added to Role <http://cambridgesemantics.com/Role/952810ffb74a42f8b502adc422608e64>
```

Related Topics

[Introduction to Anzo Logging](#)

[Limiting the Age \(and Size\) of Audit Logs](#)

[Separating Audit Logs by Type of Event](#)

[Configuring a User Inactivity Timeout](#)

[Enabling and Viewing AnzoGraph Query Logs](#)

Retrieving AnzoGraph Diagnostic Files

When Cambridge Semantics Support requests AnzoGraph diagnostic files for troubleshooting an issue, you can quickly retrieve the files from the Diagnostics tab on the AnzoGraph page in the Anzo Administration application. This topic provides information about the AnzoGraph diagnostics and instructions for retrieving the files.

Introduction to AnzoGraph Diagnostic Files

There are two types of AnzoGraph diagnostic files:

- **XRay:** X Rays are generated on-demand. If you encounter an error and the database remains running, you generate an XRay to produce the diagnostic files.
- **Crash Dump:** If you encounter an error that crashes the database, AnzoGraph automatically generates a crash dump that contains diagnostic information about the crash.

Xrays and crash dumps are valuable tools that enable Cambridge Semantics to diagnose and fix issues without access or any other visibility into a customer's data or database system. They can also be used to report on overall and detailed system performance, resulting in improved query performance for future releases of AnzoGraph.

Xrays and crash dumps harvest the diagnostic data that is stored in AnzoGraph's system tables. They include information such as:

- A low level, de-identified log of the requests that were sent to the database.
- Statistics like query operation step execution times, number of rows processed, and amount of memory used.

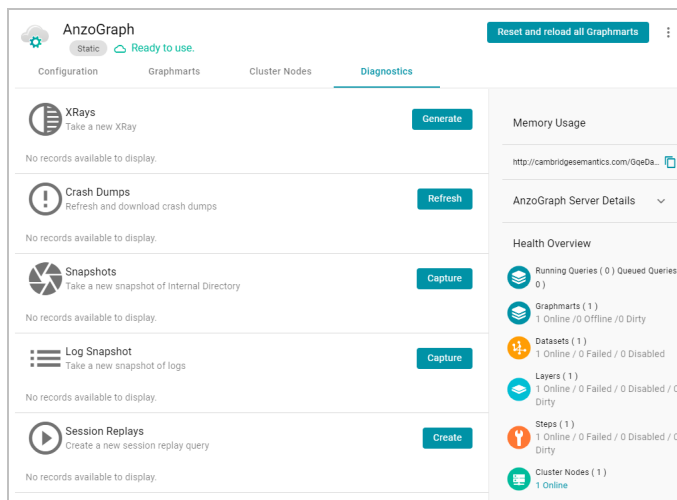
- Detailed but de-identified trace information for errors that were encountered.
- Configuration information such as the number of nodes in the cluster and AnzoGraph system settings values.

Xrays and crash dumps are designed to be anonymous and can be safely shared with Cambridge Semantics Support. They do NOT capture user information or any of the data that is loaded into memory by a user, nor do they expose details that could be used to reveal the nature of the data being queried.

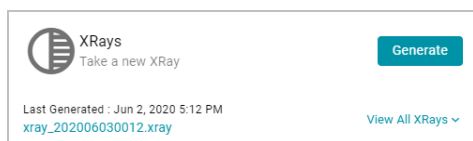
Retrieving the Files

Follow the instructions below to download an xray or crash dump to send to Cambridge Semantics Support.

1. In the Administration application, expand the **Connections** menu and select **AnzoGraph**. Anzo displays the AnzoGraph screen, which lists the connected AnzoGraph instances.
2. Click the name of the AnzoGraph instance for which you want to download an xray or crash dump. Anzo displays the Graphmarts screen for the instance.
3. Click the **Diagnostics** tab. Anzo displays the available options. For example:



4. If you want to retrieve an xray, click the **Generate** button for Xrays. Anzo creates the xray and produces a tarball with a .xray extension. For example:



Click the xray file name to download the tarball to your computer for sending to Cambridge Semantics.

Note

The files in the tarball are compressed. Do not compress the .xray file before sending it to Cambridge Semantics.

- If you want to retrieve a crash dump, click the **Refresh** button next to Crash Dumps to refresh the list of available crash dump files. Click the file name that you want to download. Anzo downloads the file to your computer.

Related Topics

[Monitoring AnzoGraph](#)

[AnzoGraph Server Administration](#)

Monitoring AnzoGraph

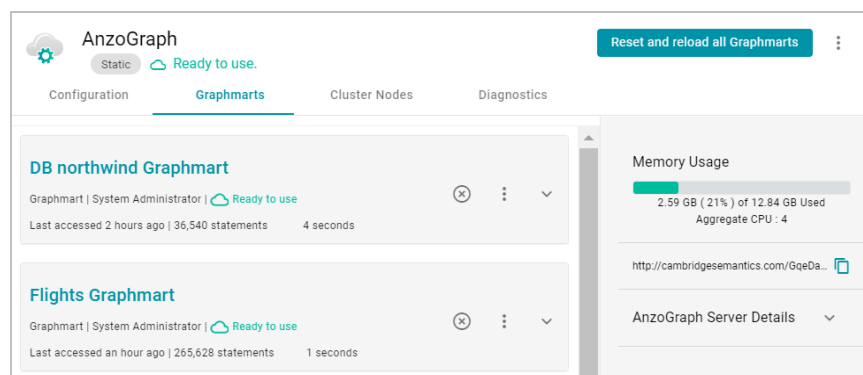
This topic provides information about viewing AnzoGraph's memory usage, query performance statistics, and network bandwidth.

- [Viewing Current Memory Usage](#)
- [Reviewing Query Performance Statistics](#)
- [Evaluating Network Performance on Clusters](#)

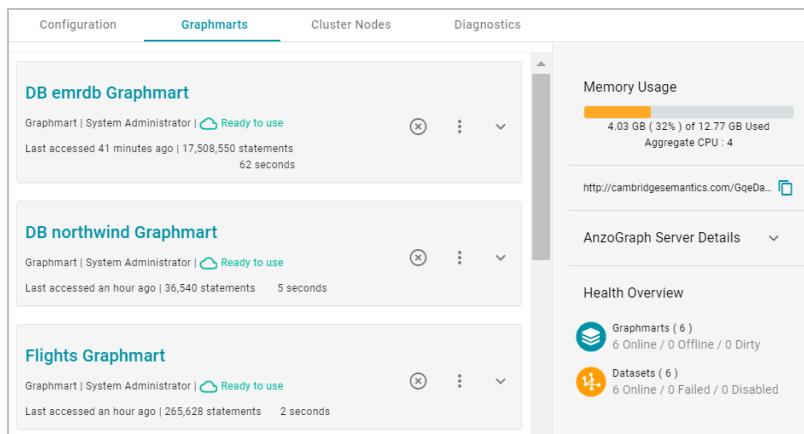
Viewing Current Memory Usage

Follow the steps below to view AnzoGraph's current memory usage.

- In the Administration application, expand the **Connections** menu and select **AnzoGraph**. Anzo displays the AnzoGraph screen, which lists the connected AnzoGraph instances.
- Click the name of the instance that you want to evaluate. Anzo displays the Graphmarts screen for that instance. The memory usage details are displayed in the top right corner on all of the tabs. For example, the test instance below shows that 21% of the available memory is in use:



Ideally, the data at rest should use only 25%-30% of the available memory because query execution and intermediate result storage can temporarily consume a very large amount of RAM, especially when multiple users run queries concurrently. When memory usage increases so that the data uses more than 25% - 30% of the available memory, the status bar changes color to orange as a warning . For example:



If memory usage for the data at rest remains above 50%, Cambridge Semantics recommends that you increase the amount of RAM available. For more information about memory usage, see [Sizing Guidelines for In-Memory Storage](#).

Reviewing Query Performance Statistics

The System Query Audit log provides details about all system events. Users can filter the query audit log to view query execution times for AnzoGraph queries.

Viewing AnzoGraph Query Statistics

1. In the Administration application, expand the **Monitoring & Diagnostics** menu and select **System Query Audit**. Anzo displays the Query Events log. For example:

Query Events

Query Errors

Longest Running Queries

Query Blacklist

Formula Events

Inflight Queries

Sort By: Date Queried

View:

Clear All

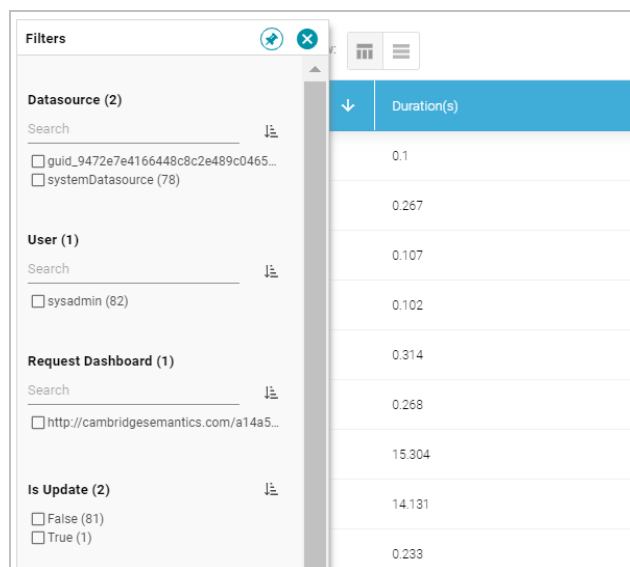
Date Queried	Duration(s)	Query Total Solutions
<div><div></div>a few seconds ago</div>	0.1	38
<div><div></div>a few seconds ago</div>	0.267	1
<div><div></div>a few seconds ago</div>	0.107	1
<div><div></div>a few seconds ago</div>	0.102	1
<div><div></div>a few seconds ago</div>	0.314	23
<div><div></div>a few seconds ago</div>	0.268	23
<div><div></div>a minute ago</div>	15.304	20
<div><div></div>a minute ago</div>	14.131	1
<div><div></div>6 minutes ago</div>	0.233	1
<div><div></div>6 minutes ago</div>	0.103	1

Rows per page: 20

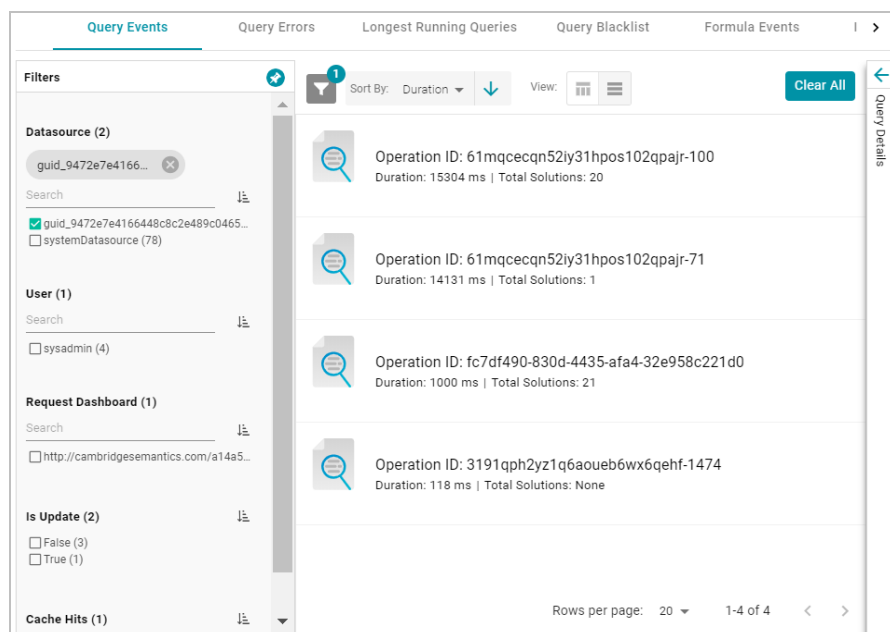
1-20 of 82

By default, the log shows an overview of all query events for all data sources. The table lists the date queried, the duration in milliseconds, and total number of solutions returned for each query event. You can select an event in the table to view details about that event, such as the target data source and query text, on the right side of the screen.

- To filter the events to display only AnzoGraph queries, open the Filters panel by clicking the filter icon (🔍) in the top left corner of the screen. For example:



- In the Filters panel under **Datasource**, select the checkbox for the AnzoGraph data source. Typically the name starts with **guid_**. The table of events is filtered to display AnzoGraph events. At the top of the screen, you can choose between a table view (📊) or list view (☰), and you can sort by date, duration, or total solutions. For example, the image below shows a list view of AnzoGraph query events sorted by duration:



- Select any query in the list to view the event overview on the right side of the screen. For example:

The screenshot shows the 'Query Events' interface. On the left, there is a list of queries with columns for 'Operation ID', 'Duration', and 'Total Solutions'. The first query is '61mqceqcn52iy31hpos102qpajr-100' with a duration of 15304 ms and 20 total solutions. On the right, the 'Query Details' panel is open, showing the 'Overview' tab. It displays the 'Operation ID' as '61mqceqcn52iy31hpos102qpajr-100', the 'User URI' as 'http://openanzo.org/system/internal/sysadmin', and the 'Data Source URI' as 'http://cambridgesemantics.com/GqeDatasource/guid_9472e7e4166448c8c2e489c04656e73a'. The 'Query' tab shows a SQL query: 'WITH (QUERY <tempQuery> //baseResources> as SELECT DISTINCT'.

To view more details about the query event, click the additional tabs to the right of the Overview tab.

Evaluating Network Performance on Clusters

The AnzoGraph Diagnostics screen provides a network benchmark that you can run to evaluate the network bandwidth of a cluster.

Note

Network performance is not applicable for single servers. The benchmark described below is not available for single-server AnzoGraph deployments.

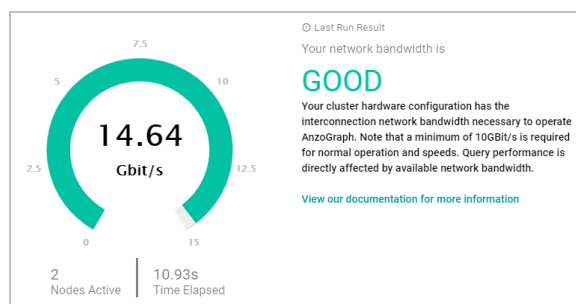
Running the Network Benchmark

- In the Administration application, expand the **Connections** menu and select **AnzoGraph**. Anzo displays the AnzoGraph screen, which lists the connected AnzoGraph instances.
- Click the name of the cluster that you want to evaluate. Anzo displays the Graphmarts screen for the cluster.
- Click the **Diagnostics** tab and find the Network Benchmarking option at the bottom the screen. For example:

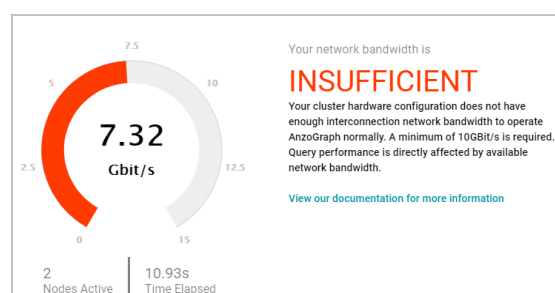
The screenshot shows the 'Network Benchmarking' interface. It has a title 'Network Benchmarking' and a subtitle 'Run Benchmark to check the network bandwidth.' Below this, there is a 'Test data in gigabytes (GB):' field with a value of '20' and a 'GB' unit. To the right of this field is a 'RUN BENCHMARK' button. Below the input field, there is a large circular gauge showing '0 Gbit/s' with a scale from 0 to 12.5. To the right of the gauge, there is a text box that says 'Run benchmark to view results here.' and a link 'View our documentation for more information'. At the bottom left, there is a '2 Nodes Active' indicator and a '0.00s Time Elapsed' indicator.

- By default, the benchmark is set to distribute 20 GB of data per node over the network. Each node in the cluster sends 20 GB to every other node. You can specify a different size if necessary. Note that increasing the value also increases the time to run the benchmark.

5. To run the test, click the **Run Benchmark** button. Anzo runs the benchmark and displays the results. For example:



If the bandwidth is less than 10 Gbit/s, Anzo displays an "Insufficient" result. For example:



When the results are insufficient, Cambridge Semantics recommends that you increase the network bandwidth. You can continue to use the cluster with the expectation of slower performance for network-bound operations.

Related Topics

[Retrieving AnzoGraph Diagnostic Files](#)

[AnzoGraph Server Administration](#)

System Query Audit

The System Query Audit screen enables administrators to quickly view a log of query events, query errors, the duration time for the longest running queries, and a list of any queries that have been blacklisted. The audit log also includes a Queued Queries tab that displays a list of the queries that are queued behind currently running queries. Administrators can cancel queries from the list and remove them from the queue. This topic provides information about using the System Query Audit log.

- [Viewing the System Query Audit Log](#)
- [AnzoGraph Detailed Query Timing Reference](#)

Viewing the System Query Audit Log

In the Administration application, expand the **Monitoring & Diagnostics** menu and select **System Query Audit**. Anzo displays the Query Events log. For example:

Query Events			Query Errors	Longest Running Queries	Query Blacklist	Formula Events	Inflight Queries
Sort By:	Date Queried	View:					
Date Queried	Duration(s)	Query Total Solutions					
a few seconds ago	0.1	38					
a few seconds ago	0.267	1					
a few seconds ago	0.107	1					
a few seconds ago	0.102	1					
a few seconds ago	0.314	23					
a few seconds ago	0.268	23					
a minute ago	15.304	20					
a minute ago	14.131	1					
6 minutes ago	0.233	1					
6 minutes ago	0.103	1					
Rows per page: 20			1-20 of 82				

By default, the log shows an overview of all query events for all data sources. The table lists the date queried, the duration in milliseconds, and total number of solutions returned for each query event. You can select an event in the table to view details about that event, such as the target data source and query text, on the right side of the screen.

Note

The System Query Audit log does not report on queries that complete in less than 100 milliseconds. In addition, queries that reuse the query cache from a previous run are not captured in the log. However, if a query takes less than 100 ms and uses cache, the original entry for the query is updated to increase the Cache Hit count.

AnzoGraph Detailed Query Timing Reference

In the Advanced settings for the AnzoGraph connection configuration, there is an **Enable Detailed Query Timing** setting (shown in the image below) that controls the level of information that is displayed for AnzoGraph queries in the System Query Audit log. This section describes the differences in logging when the setting is enabled and disabled.

Advanced

AnzoGraph Concurrent Queries
10

AnzoGraph connection timeout (seconds)
60

☒ Use AnzoGraph persistence if available
☒ Reload previously deployed Graphmarts on startup

☒ Deploy Graphmart data during AnzoGraph startup

Callback HostName
None

☐ Readonly Replica
☒ Bulk Load from Anzo
☒ Vacuum
☒ Gather Statistics on Load

☒ Use Priority Queue Query Manager
☐ Enable Detailed Query Timing

AnzoGraph Management Port
5600

Enable Detailed Query Timing is disabled by default, meaning that Anzo will not run the additional statistics gathering queries unless you enable the setting. When Enable Detailed Query Timing is disabled, the System Query Audit log

displays fewer query timing details. For example, the images in the table below show a comparison between the **Result Details** tab when Enable Detailed Query Timing is disabled versus enabled. When the setting is disabled, details such as query Compilation Time are not recorded.

Enable Detailed Query Timing Disabled		Enable Detailed Query Timing Enabled	
Query Duration (ms) 10431	Cache Hits -	Date Queried a minute ago	Original Query Date -
Query Total Solutions 14	Query Results Cached true	Query Duration (ms) 13396	Cache Hits -
Is Update false	Cache Hit false	Query Total Solutions 1	Query Results Cached true
Is Error false	Dataset Cache Hit -	Is Update false	Cache Hit false
Query Canceled false	Query Results Valid true	Is Error false	Dataset Cache Hit -
Query Queued Time 0	Query Already Compiled -	Query Canceled false	Query Results Valid true
	Compilation Time (ms) -	Query Queued Time 3	Query Already Compiled false
	Query Execution Time (ms) 10424.964		Compilation Time (ms) 17025.002
			Query Execution Time (ms) 13388.833

In addition, the images in the following table show a comparison between the **Query Statistics** tab when Enable Detailed Query Timing is disabled versus enabled. When the setting is disabled, the Compilation Stats and Query Summary tables are empty.

Enable Detailed Query Timing Disabled					Enable Detailed Query Timing Enabled																																																																																																																																																																																																																											
<div><div><</div><div>Request Dataset</div><div>Resolved Dataset</div><div>Request Details</div><div>Errors/Warnings</div></div>					<div><div>Overview</div><div>Result Details</div><div>Request Dataset</div><div>Resolved Dataset</div><div>Request Details</div><div>Errors/Warnings</div><div>Query Statistics</div></div>																																																																																																																																																																																																																											
Compilation Stats -					Compilation Stats																																																																																																																																																																																																																											
Query Summary -					<table><tr><th>?query</th><th>?segment</th><th>?compile</th><th>?optimized</th><th>?secondpass</th><th>?duration</th><th>?bytes</th><th>?codeid</th><th>?path</th><th>?usertime</th><th>?systemtime</th><th>?rss</th><th>?starttime</th></tr><tr><td>1843</td><td>0</td><td>1</td><td>0</td><td>0</td><td>115168</td><td>31349</td><td>292</td><td>"code/292/0.dylib"</td><td>40</td><td>81</td><td>482564</td><td>2020-04-24T20:39</td></tr><tr><td>1843</td><td>0</td><td>1</td><td>1</td><td>1</td><td>165937</td><td>31349</td><td>292</td><td>"code/292/0.dylib"</td><td>30</td><td>138</td><td>482564</td><td>2020-04-24T20:39</td></tr><tr><td>1843</td><td>1</td><td>1</td><td>0</td><td>0</td><td>135463</td><td>30557</td><td>293</td><td>"code/293/0.dylib"</td><td>42</td><td>101</td><td>482564</td><td>2020-04-24T20:39</td></tr><tr><td>1843</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1852621</td><td>30557</td><td>293</td><td>"code/293/0.dylib"</td><td>82</td><td>1788</td><td>482564</td><td>2020-04-24T20:39</td></tr><tr><td>1843</td><td>2</td><td>1</td><td>0</td><td>0</td><td>131465</td><td>28153</td><td>294</td><td>"code/294/0.dylib"</td><td>36</td><td>102</td><td>482564</td><td>2020-04-24T20:39</td></tr><tr><td>1843</td><td>2</td><td>1</td><td>1</td><td>1</td><td>179733</td><td>28153</td><td>294</td><td>"code/294/0.dylib"</td><td>43</td><td>140</td><td>482564</td><td>2020-04-24T20:39</td></tr><tr><td>1843</td><td>3</td><td>1</td><td>0</td><td>0</td><td>134088</td><td>29009</td><td>295</td><td>"code/295/0.dylib"</td><td>41</td><td>101</td><td>482564</td><td>2020-04-24T20:39</td></tr><tr><td>1843</td><td>3</td><td>1</td><td>1</td><td>1</td><td>1106167</td><td>29009</td><td>295</td><td>"code/295/0.dylib"</td><td>62</td><td>1062</td><td>482564</td><td>2020-04-24T20:39</td></tr><tr><td>1843</td><td>4</td><td>1</td><td>0</td><td>0</td><td>161024</td><td>40411</td><td>296</td><td>"code/296/0.dylib"</td><td>37</td><td>131</td><td>482564</td><td>2020-04-24T20:39</td></tr><tr><td>1843</td><td>4</td><td>1</td><td>1</td><td>1</td><td>261154</td><td>40411</td><td>296</td><td>"code/296/0.dylib"</td><td>35</td><td>232</td><td>482564</td><td>2020-04-24T20:39</td></tr><tr><td>1843</td><td>5</td><td>1</td><td>0</td><td>0</td><td>153252</td><td>34179</td><td>297</td><td>"code/297/0.dylib"</td><td>35</td><td>126</td><td>482564</td><td>2020-04-24T20:39</td></tr><tr><td>1843</td><td>5</td><td>1</td><td>1</td><td>1</td><td>240542</td><td>34179</td><td>297</td><td>"code/297/0.dylib"</td><td>46</td><td>201</td><td>482564</td><td>2020-04-24T20:39</td></tr><tr><td>1843</td><td>6</td><td>1</td><td>0</td><td>0</td><td>133239</td><td>24909</td><td>298</td><td>"code/298/0.dylib"</td><td>35</td><td>105</td><td>482564</td><td>2020-04-24T20:39</td></tr><tr><td>1843</td><td>6</td><td>1</td><td>1</td><td>1</td><td>197318</td><td>24909</td><td>298</td><td>"code/298/0.dylib"</td><td>37</td><td>164</td><td>482564</td><td>2020-04-24T20:39</td></tr><tr><td>1843</td><td>7</td><td>1</td><td>0</td><td>0</td><td>196144</td><td>40500</td><td>299</td><td>"code/299/0.dylib"</td><td>44</td><td>161</td><td>482564</td><td>2020-04-24T20:39</td></tr></table>												?query	?segment	?compile	?optimized	?secondpass	?duration	?bytes	?codeid	?path	?usertime	?systemtime	?rss	?starttime	1843	0	1	0	0	115168	31349	292	"code/292/0.dylib"	40	81	482564	2020-04-24T20:39	1843	0	1	1	1	165937	31349	292	"code/292/0.dylib"	30	138	482564	2020-04-24T20:39	1843	1	1	0	0	135463	30557	293	"code/293/0.dylib"	42	101	482564	2020-04-24T20:39	1843	1	1	1	1	1852621	30557	293	"code/293/0.dylib"	82	1788	482564	2020-04-24T20:39	1843	2	1	0	0	131465	28153	294	"code/294/0.dylib"	36	102	482564	2020-04-24T20:39	1843	2	1	1	1	179733	28153	294	"code/294/0.dylib"	43	140	482564	2020-04-24T20:39	1843	3	1	0	0	134088	29009	295	"code/295/0.dylib"	41	101	482564	2020-04-24T20:39	1843	3	1	1	1	1106167	29009	295	"code/295/0.dylib"	62	1062	482564	2020-04-24T20:39	1843	4	1	0	0	161024	40411	296	"code/296/0.dylib"	37	131	482564	2020-04-24T20:39	1843	4	1	1	1	261154	40411	296	"code/296/0.dylib"	35	232	482564	2020-04-24T20:39	1843	5	1	0	0	153252	34179	297	"code/297/0.dylib"	35	126	482564	2020-04-24T20:39	1843	5	1	1	1	240542	34179	297	"code/297/0.dylib"	46	201	482564	2020-04-24T20:39	1843	6	1	0	0	133239	24909	298	"code/298/0.dylib"	35	105	482564	2020-04-24T20:39	1843	6	1	1	1	197318	24909	298	"code/298/0.dylib"	37	164	482564	2020-04-24T20:39	1843	7	1	0	0	196144	40500	299	"code/299/0.dylib"	44	161	482564	2020-04-24T20:39
?query	?segment	?compile	?optimized	?secondpass	?duration	?bytes	?codeid	?path	?usertime	?systemtime	?rss	?starttime																																																																																																																																																																																																																				
1843	0	1	0	0	115168	31349	292	"code/292/0.dylib"	40	81	482564	2020-04-24T20:39																																																																																																																																																																																																																				
1843	0	1	1	1	165937	31349	292	"code/292/0.dylib"	30	138	482564	2020-04-24T20:39																																																																																																																																																																																																																				
1843	1	1	0	0	135463	30557	293	"code/293/0.dylib"	42	101	482564	2020-04-24T20:39																																																																																																																																																																																																																				
1843	1	1	1	1	1852621	30557	293	"code/293/0.dylib"	82	1788	482564	2020-04-24T20:39																																																																																																																																																																																																																				
1843	2	1	0	0	131465	28153	294	"code/294/0.dylib"	36	102	482564	2020-04-24T20:39																																																																																																																																																																																																																				
1843	2	1	1	1	179733	28153	294	"code/294/0.dylib"	43	140	482564	2020-04-24T20:39																																																																																																																																																																																																																				
1843	3	1	0	0	134088	29009	295	"code/295/0.dylib"	41	101	482564	2020-04-24T20:39																																																																																																																																																																																																																				
1843	3	1	1	1	1106167	29009	295	"code/295/0.dylib"	62	1062	482564	2020-04-24T20:39																																																																																																																																																																																																																				
1843	4	1	0	0	161024	40411	296	"code/296/0.dylib"	37	131	482564	2020-04-24T20:39																																																																																																																																																																																																																				
1843	4	1	1	1	261154	40411	296	"code/296/0.dylib"	35	232	482564	2020-04-24T20:39																																																																																																																																																																																																																				
1843	5	1	0	0	153252	34179	297	"code/297/0.dylib"	35	126	482564	2020-04-24T20:39																																																																																																																																																																																																																				
1843	5	1	1	1	240542	34179	297	"code/297/0.dylib"	46	201	482564	2020-04-24T20:39																																																																																																																																																																																																																				
1843	6	1	0	0	133239	24909	298	"code/298/0.dylib"	35	105	482564	2020-04-24T20:39																																																																																																																																																																																																																				
1843	6	1	1	1	197318	24909	298	"code/298/0.dylib"	37	164	482564	2020-04-24T20:39																																																																																																																																																																																																																				
1843	7	1	0	0	196144	40500	299	"code/299/0.dylib"	44	161	482564	2020-04-24T20:39																																																																																																																																																																																																																				

To enable detailed query timing, edit the AnzoGraph connection and select the **Enable Detailed Query Timing** checkbox. You do not need to restart Anzo or AnzoGraph after changing the setting.

Important

Enabling detailed query timing increases the AnzoGraph workload and may decrease overall query performance.



Related Topics

[Connecting to AnzoGraph](#)

AnzoGraph Server Administration

The topics in this section provide reference information and instructions for performing administrative tasks on an AnzoGraph server. Some tasks, such as modifying server configuration settings, cannot be done via the Anzo Administration application. Other tasks, such as starting and stopping AnzoGraph using the system manager, are documented as alternate methods of managing AnzoGraph if the Administration application is unavailable or you prefer to use the AnzoGraph command line interface.

- [Starting and Stopping AnzoGraph](#)
- [Configuring AnzoGraph for Kerberos Authentication](#)
- [Using the AnzoGraph CLI](#)
- [Changing AnzoGraph Configuration Settings](#)
- [AnzoGraph System Settings Reference](#)
- [Generating Diagnostic Files with the System Manager](#)

Starting and Stopping AnzoGraph

This topic provides instructions for starting and stopping AnzoGraph.

Note

The system management daemon, **azgmgrd**, should remain running at all times. When you restart the database, do not stop and start the daemon. There are two circumstances that require you to restart azgmgrd:

1. When [Upgrading AnzoGraph](#).
2. When making changes to the `<install_path>/config/ip_addrs.conf` file if you add or remove servers from an AnzoGraph cluster.

Follow the appropriate instructions below, depending on the current state of AnzoGraph and your use case:

- [Stop the Database and Leave the System Management Daemon Running](#)
- [Start the Database \(the System Management Daemon is Running\)](#)
- [Stop the Database and the System Management Daemon](#)
- [Start the System Management Daemon and the Database](#)
- [Reinitializing the Database](#)

Stop the Database and Leave the System Management Daemon Running

To stop the database, run one of the following commands from the **leader server**:

- If services are set up, run the following command:

```
sudo systemctl stop anzograph
```

- If services are not set up, stop the database with the following system manager command:

```
<install_path>/bin/azgctl -stop
```

Important

Make sure that you are logged in as the Anzo service account user any time you start and stop AnzoGraph using the system manager commands.

If queries are running, the system manager waits the number of seconds in [stop_timeout](#) (the default value is 30 seconds) for any outstanding queries to complete and then stops the database.

Start the Database (the System Management Daemon is Running)

To start the database, run one of the following commands from the **leader server**:

- If services are set up, run the following command:

```
sudo systemctl start anzograph
```

- If services are not set up, start the database with the following system manager command:

```
<install_path>/bin/azgctl -start
```

Important

Make sure that you are logged in as the Anzo service account user any time you start and stop AnzoGraph using the system manager commands.

Stop the Database and the System Management Daemon

To stop the database and system management daemon, run the appropriate commands from the **leader server**:

- If services are set up, run the following commands on the leader server to stop the database and daemon on all servers in the cluster:

```
sudo systemctl stop anzograph
```

```
sudo systemctl stop azgmgrd
```

- If services are not set up, run the following commands on the leader server to stop the database and daemon on all servers in the cluster:

```
<install_path>/bin/azgctl -stop
```

```
<install_path>/bin/azgctl -stopdaemon
```

Important

Make sure that you are logged in as the Anzo service account user any time you start and stop AnzoGraph using the system manager commands.

Start the System Management Daemon and the Database

To start the system management daemon, run one of the following commands. On clusters, run the command on **each server in the cluster**:

- If services are set up, run the following command on all servers in the cluster:

```
sudo systemctl start azgmgrd
```

- If services are not set up, run the following command on all servers in the cluster:

```
<install_path>/bin/azgmgrd
```

Important

Make sure that you are logged in as the Anzo service account user any time you start and stop AnzoGraph using the system manager commands.

To start the database after the system management daemon is running, run one of the following commands on the **leader node**:

- If services are set up, run the following command:

```
sudo systemctl start anzograph
```

- If services are not set up, start the database with the following system manager command:

```
<install_path>/bin/azgctl -start
```

Reinitializing the Database

If you need to reinitialize the database to remove the generated code and any persisted data, run the following command. The system management daemon (azgmgrd) should be running.

```
<install_path>/bin/azgctl -start -init
```

Configuring AnzoGraph for Kerberos Authentication

If you plan to load data to AnzoGraph from an HDFS file store that uses Kerberos authentication, follow the steps below to configure AnzoGraph for Kerberos authentication.

1. In order to be able to generate an authentication token for requesting encrypted ticket-granting tickets (TGT) from the key distribution center (KDC), each AnzoGraph host server must include the Kerberos workstation package,

krb5-workstation. On each server in the cluster, run the following command to install the package:

```
sudo yum install -y krb5-workstation
```

2. In order to establish a connection to the KDC, AnzoGraph must have a copy of the KDC's **krb5.conf** file. Place a copy of **krb5.conf** in the **/etc** directory on each AnzoGraph host server.
3. In addition to **krb5.conf**, each AnzoGraph server needs a copy of the **.keytab** file from the principal node. The **keytab** file and principal name are used to generate an authentication token.

Note

To find the location of the **.keytab** file and the principal name, you can look up the

`dfs.web.authentication.kerberos.keytab` and

`dfs.web.authentication.kerberos.principal` values in **hdfs-site.xml** on the HDFS master node.

Copy the **.keytab** file to any location on each AnzoGraph host server, and then run the following command to generate the authentication token:

```
kinit -p <principal_name> -k -t <path>/<keytab_file>
```

Where **<principal_name>** is the Kerberos principal name and **<path>/<keytab_file>** is the location and name of the **.keytab** file.

Related Topics

[Connecting to a File Store](#)

Using the AnzoGraph CLI

You can use the **azgi** command line interface (CLI) in the **<install_path>/bin** directory to issue commands directly to the database.

Important

The **azgi** CLI works on the SPARQL HTTPS port and is enabled only when SSL protocol is enabled. SSL access is controlled by the [enable_ssl_protocol](#) setting. If you disabled HTTPS access and want to enable it so that you can use the command line, see [Changing AnzoGraph Configuration Settings](#) for instructions.

This section describes the available **azgi** commands. To view the list of options from the command line, run **azgi -help**.

AZGI Usage

```
azgi [-f filename] [-c "command"] [-set param=value] [-h host_url] [-p port]
    [-u username:password] [-v] [-timer] [-raw] [-csv] [-json] [-xml] [-silent]
```

```
[-nohead] [-noprogess] [-maxwid width] [-wide]
[-o file] [-noss1] [-certs directory] [-context json_file]
```

Option	Description
-f filename	<p>Runs the specified SPARQL query file. For example, the following command runs the query or queries in the query.rq file:</p> <pre>azgi -f /home/user/query.rq</pre>
-c "command"	<p>Runs the command in quotation marks. For example, this command runs a query:</p> <pre>azgi -c "select distinct ?eventname from <ticket> where {?event <eventname> ?eventname} limit 100"</pre> <p>You can include multiple -c options to run multiple commands. For example, this command runs two queries:</p> <pre>azgi -c "select * from <ticket> where {?s ?p ?o} limit 100" -c "select distinct ?likes from <ticket> where {?person <like> ?likes}"</pre> <p>And this command sets the query_label configuration setting to "events" before running the query:</p> <pre>azgi -c "set query_label to 'events'" -c "select distinct ?eventname from <ticket> where {?event <eventname> ?eventname} limit 100"</pre>
-set param =value	<p>Sets or changes parameter values in query files. For example this command runs the query in the query_summary.rq file with the \$query parameter set to 2:</p> <pre>azgi -set query=2 -f query_summary.rq</pre>
-h host_url	<p>Connects to a remote AnzoGraph server. For example, the following statement runs a query against AnzoGraph installed on host 10.104.55.27:</p> <pre>azgi -h 10.104.55.27 -c "select * from <ticket> where {?s ?p ?o} limit 100"</pre>
-p port	<p>Used to connect to AnzoGraph on a non-default port. The default azgi port is 8256.</p>

Option	Description
-u username :password	<p>Connects to the database with credentials (HTTP basic authentication). If you type -u username and exclude the password, the client prompts for the password. For example, the following command uses basic authentication to run a query:</p> <pre>azgi -u admin:Passw0rd1 -c "select ?g where {graph ?g {?s ?p ?o}} limit 100"</pre>
-v	<p>Displays verbose output such as client connection details. For example:</p> <pre>azgi -v -c "select distinct ?p from <ticket> where {<person1> ?p ?o}"</pre> <pre>Connecting to host=localhost port=8256 IPv4: connected POST /sparql HTTP/1.1 Host: Anon Accept: application/sparql-results+xml User-Agent: azgi Connection: keep-alive Content-Length: 38 Content-Type: application/sparql-query select distinct ?p from <ticket> where {<person1> ?p ?o} HTTP/1.1 200 OK Date: Tue, 30 Jun 2020 00:24:42 GMT Server: AnzoGraph Access-Control-Allow-Origin: * X-AnzoGraph-QueryExecution-Time: 20 Connection: close Content-Type: application/sparql-results+xml; charset=utf-8 ...</pre>
-timer	Reports query execution time in milliseconds.
-raw	Displays query results in raw XML, JSON, or CSV format, depending on what format you request.

Option	Description
-csv	<p>Displays results in CSV format. For example:</p> <pre>azgi -csv -c "select * from <ticket> where {<person1> ?p ?o} order by ?p limit 10"</pre> <pre>p,o birthday,1939-11-19 card,3876972207981477 city,Kent dislike,jazz dislike,broadway email,Etiam.laoreet.libero@sodalesMaurisblandit.edu firstname,Rafael friend,person13826 friend,person33618 friend,person15410</pre>
-json	Displays results in JSON format.
-xml	Displays results in XML format
-silent	Suppresses the query output.
-nohead	Suppresses headings in query results.
-noprogress	Suppresses the progress messages that are displayed for queries that are in flight.
-maxwid width	<p>Overrides the default maximum column width of 50 characters for tabular query results. For example, for a data set with long graph names, you can expand column width to view the entire name:</p> <pre>azgi -maxwid 10000 -c "select ?g where {graph ?g {?s ?p ?o}} limit 100"</pre> <p>Using the -wide option described below is equivalent to "maxwid 60000."</p>
-wide	Increases the column width for tabular query results from the default 50 characters to 60,000 characters. Equivalent to <code>-maxwid 60000</code> .

Option	Description
-noss1	<p>Instructs the client to make a non-SSL (HTTP) connection to the database. When using AZGI to send a request to a remote AnzoGraph server, include the <code>-h <i>host_url</i></code> and <code>-p <i>port</i></code> options when using <code>-noss1</code>. The default HTTP port is 7070. For example:</p> <pre>azgi -noss1 -h 10.100.0.20 -p 7070 -c "select (count(*) as ?cnt) where {?s ?p ?o}"</pre>
-o <i>file</i>	Writes the response to the specified file. If the file exists, it is overwritten.
-certs directory	<p>Instructs the client to make a certified secure connection to the database. The AnzoGraph certificates are ca.crt, serv.crt (public key), and serv.key (private key) in the <code>install_path/config</code> directory. When sending requests to a remote AnzoGraph server, you can copy the AnzoGraph certificates to the server where you are using AZGI. For example, the following command runs a query on a remote AnzoGraph server. The command makes a certified connection using the AnzoGraph certificates, which were copied to the <code>/home/user/certs</code> directory:</p> <pre>azgi -h 10.10.10.01 -certs /home/user/certs -c "select ?g where {graph ?g {?s ?p ?o}} limit 100"</pre> <p>This command runs the same query from the AnzoGraph server.</p> <pre>azgi -certs /opt/anzograph/config -c "select ?g where {graph ?g {?s ?p ?o}} limit 100"</pre>
-context json_file	<p>Specifies the query context file to use with the request. Context files are JSON-formatted files with key-value pairs that provide connection details, such as user credentials, keys, and tokens, for authentication against data sources. For example:</p> <pre>{ "url": "jdbc:mysql://10.111.4.9:3306/NORTHWIND", "username": "sysadmin", "password": "admin123" }</pre>

Related Topics

[Changing AnzoGraph Configuration Settings](#)

[AnzoGraph System Settings Reference](#)

Changing AnzoGraph Configuration Settings

The default AnzoGraph system configuration is optimized for most AnzoGraph installations. If Cambridge Semantics Support recommends that you change the configuration, you can edit the configuration file, `install_path/config/settings.conf`, to modify or add settings. Each time you start the database, AnzoGraph reads this file and stores the configuration in memory. **On a cluster, change `settings.conf` on the leader server only.** See the [AnzoGraph System Settings Reference](#) for information about the units of measurement for the settings as well as any special instructions.

- The commented lines in the file show the default configuration values. To customize the value for a setting that is commented out, uncomment the line and edit the value portion of `setting_name=value`.
- To add settings to `settings.conf`, add the setting and new value in the format below. Type each setting and value pair on a new line.

```
setting_name=value
```

Note

AnzoGraph applies settings from the top to the bottom of the file. If the same setting appears more than once, AnzoGraph applies the value for the last instance of the setting. The last instance overrides any previous instances.

- To revert AnzoGraph to a previous configuration from a backup file, rename the existing `settings.conf` file and then change the name of the desired backup file to **`settings.conf`**.

Important

After you change `settings.conf`, you must restart AnzoGraph for the settings to take effect. See [Starting and Stopping AnzoGraph](#) for instructions.

Related Topics

[Relocating AnzoGraph Directories](#)

[Using AnzoGraph Persistence \(Preview\)](#)

[Ignoring Missing Graphs](#)

[Changing the Default FROM Clause Behavior](#)

[Managing the Automatic Restart Feature](#)

[Enabling Paged Data Mode \(Preview\)](#)

[AnzoGraph System Settings Reference](#)

Relocating AnzoGraph Directories

Follow the instructions in this section to designate alternate locations for certain directories included in the AnzoGraph installation. You have the option to relocate the **persistence** directory where the system saves the data in memory to the file system, the **internal** directory where the system saves database-related files such as logs and generated code, and the **spill** directory where the system saves any temporary query files that spill to disk.

You can change the settings described in this section at any time. Once you restart the database, AnzoGraph starts saving any new files in the directory locations that you specify.

Note

The system does not relocate any existing directories or files. You can move the existing files manually if needed.

1. Stop the database. See [Stop the Database and Leave the System Management Daemon Running](#) for instructions.
2. **On the leader node**, open the AnzoGraph settings file, **settings.conf**, in a text editor. The file is in the `<install_path>/config` directory.
3. Uncomment the lines for any of the following settings in settings.conf. Then edit the value portion of *setting=value* to specify the desired directory.
 - **internal_directory**: The directory where you want AnzoGraph to save internal database-related files such as generated code, logs, and query plans.
 - **persistence_directory**: The directory where you want AnzoGraph to save data when writing data to disk.
 - **spill_directory**: The directory where you want the AnzoGraph to save any temporary query files that spill to disk.

Important

AnzoGraph uses O_DIRECT to read the spill files into the database. If you relocate the spill directory, make sure to place it on an ext4 file system that supports O_DIRECT

4. Save and close settings.conf.
5. Restart the database to apply the configuration change. See [Start the Database \(the System Management Daemon is Running\)](#) for instructions.

Related Topics

[Changing AnzoGraph Configuration Settings](#)

[Starting and Stopping AnzoGraph](#)

Using AnzoGraph Persistence (Preview)

By default, Anzo manages the data in AnzoGraph by automatically reloading graphmart data into memory when AnzoGraph is restarted. You also have the option to enable persistence on the AnzoGraph instance. When persistence is enabled, AnzoGraph saves the data in memory to disk after every transaction. Each time AnzoGraph is restarted, the persisted data is automatically loaded back into memory. Once the data is loaded into memory, rather than automatically reloading active graphmarts, Anzo checks to see if the last updated timestamp in AnzoGraph matches the last updated value in Anzo. If the timestamps match, Anzo does not initiate a reload. If there is a mismatch, Anzo reloads the active graphmarts to update the data in memory to the latest version.

Note

The AnzoGraph persistence feature is available as a **Preview** release, which means the implementation has recently been completed but is not yet thoroughly tested with Anzo 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.

This topic lists important information to consider before enabling persistence and provides instructions for enabling persistence in the AnzoGraph configuration file.

Important Considerations

Before enabling persistence, consider the following important notes:

- In general, each AnzoGraph server needs access to about twice as much disk space as RAM on the server. By default, AnzoGraph saves data to the `install_path/persistence` directory on the local file system. You can also configure AnzoGraph to save data to a mounted file system. For more information, see [Relocating AnzoGraph Directories](#).
- Persisted data is unique to each AnzoGraph version and cannot be re-used after an upgrade. If you upgrade AnzoGraph and persistence is enabled, the database will not start until it is reinitialized to remove the persisted data. See [Reinitializing the Database](#) for instructions.
- When persistence is enabled, transactional workloads that perform many concurrent write operations may experience a performance degradation due to the overhead of writing the data from each transaction to disk.

Enabling Persistence

Follow the steps below to enable the AnzoGraph save to disk option.

1. Stop the database. See [Stop the Database and Leave the System Management Daemon Running](#) for instructions.
2. **On the leader node**, open the AnzoGraph settings file, `settings.conf`, in a text editor. The file is in the `<install_path>/config` directory.

3. In `settings.conf`, find the following line in the file:

```
enable_persistence=false
```

4. Change the `enable_persistence` value to **true**:

```
enable_persistence=true
```

5. Save and close `settings.conf`.
6. Restart the database to apply the configuration change. See [Start the Database \(the System Management Daemon is Running\)](#) for instructions.

After each transaction, AnzoGraph saves the data in memory to disk in the location specified in the `persistence_directory` setting. Each time AnzoGraph is restarted, the persisted data is automatically loaded back into memory.

Note

To avoid unnecessary reloads, make sure that the AnzoGraph connection in Anzo is configured to enable the **Use AnzoGraph persistence if available** option. See [Connecting to AnzoGraph](#) for more information.

Related Topics

[Connecting to AnzoGraph](#)

[Relocating AnzoGraph Directories](#)

[Starting and Stopping AnzoGraph](#)

Ignoring Missing Graphs

By default, AnzoGraph returns a "No such graph or view" error and aborts the query if a query references a graph that does not exist. You can configure AnzoGraph to conform to the SPARQL specification and return an empty result instead of an error, however, if a query references a missing graph. Follow the instructions below to configure the system to return empty results instead of an error when a referenced graph does not exist.

1. Stop the database. See [Stop the Database and Leave the System Management Daemon Running](#) for instructions.
2. **On the leader node**, open the AnzoGraph settings file, `settings.conf`, in a text editor. The file is in the `<install_path>/config` directory.
3. In `settings.conf`, uncomment the `enable_unbound_variables=false` line and change the value to true:

```
enable_unbound_variables=true
```

4. Save and close `settings.conf`.
5. Restart the database to apply the configuration change. See [Start the Database \(the System Management Daemon is Running\)](#) for instructions.

Note

In addition to allowing queries that reference non-existent graphs to succeed, setting `enable_unbound_variables` to `true` also configures AnzoGraph to ignore unbound variables elsewhere in queries. For example, by default (when `enable_unbound_variables=false`), if a query includes a variable in the SELECT list that is not referenced in a WHERE clause pattern, AnzoGraph aborts the query and returns a "Named variable not in contained WHERE clause" error. When `enable_unbound_variables=true`, AnzoGraph does not warn the user about unbound variables. Instead, the results are empty for the unbound variable. For example:

```
SELECT ?unbound ?person ?name
FROM <http://cambridgesemantics.com/people>
WHERE {?person <http://cambridgesemantics.com/people#firstname> ?name}
LIMIT 5
```

```
unbound | person      | name
-----+-----+-----
      | person35632 | Ross
      | person20216 | Quin
      | person35859 | Kellie
      | person2551  | Maris
      | person24963 | Madonna
5 rows
```

Related Topics

[Changing AnzoGraph Configuration Settings](#)

[AnzoGraph System Settings Reference](#)

Changing the Default FROM Clause Behavior

By default, if a query omits FROM clauses, the scope of the query is limited to the default graph (DEFAULTSET). Triples in named graphs will not be included in the scope of the query. The default behavior is controlled by the `sparql_spec_default_graph` configuration setting. To configure AnzoGraph to conform to the SPARQL specification and include the default graph and all named graphs in the scope of a query that omits the FROM clause, follow the instructions below.

1. Stop the database. See [Stop the Database and Leave the System Management Daemon Running](#) for instructions.
2. **On the leader node**, open the AnzoGraph settings file, `settings.conf`, in a text editor. The file is in the `<install_path>/config` directory.
3. In `settings.conf`, uncomment the `sparql_spec_default_graph=false` line and change the value to `true`:

```
sparql_spec_default_graph=true
```

4. Save and close settings.conf.
5. Restart the database to apply the configuration change. See [Start the Database \(the System Management Daemon is Running\)](#) for instructions.

Related Topics

[Changing AnzoGraph Configuration Settings](#)

[AnzoGraph System Settings Reference](#)

Managing the Automatic Restart Feature

AnzoGraph can be configured so that the system manager automatically restarts the database and evaluates the queries that were running if AnzoGraph shuts down unexpectedly. This topic describes the process that occurs when AnzoGraph automatically restarts and provides information about the configuration settings that control the functionality as well as administrative information for managing the evaluated queries.

- [Automated Restart Procedure](#)
- [Automated Restart System Settings](#)
- [Removing a Query from the Block List](#)

Automated Restart Procedure

The steps below describe what occurs during the automatic restart process after AnzoGraph has crashed:

1. The system manager restarts the database in **safe mode**. In safe mode, AnzoGraph is locked to users and returns the following message if a user runs a query: "AnzoGraph is running in safe-mode. Cannot execute query." In addition, running `azgctl -status` to check the status of the database returns the message "AnzoGraph is running in safe-mode." If persistence is enabled, the data that was in memory at the time of the crash is reloaded into memory.
2. While in safe mode, AnzoGraph runs any queries that were inflight at the time of the crash. By executing the queries that were running, AnzoGraph tries to determine if the crash was directly caused by one of the inflight queries.
3. Depending on the outcome of running the inflight queries, AnzoGraph does the following:
 - If all inflight queries run to completion in safe mode, they are all added to the **warned_list**. In addition, each query is copied to a file named `<query_ID>.txt` in the `<install_path>/internal/auto_restart/<timestamp>/warned_list` directory.

Note

When all inflight queries complete successfully, that means it is unlikely that any one of the queries on its own is the culprit for the crash. However, all of the queries are added to the warned list because it is possible that the combination of queries run concurrently could have caused the crash.

- If any of the inflight queries fail or crash the database in safe mode, those queries are added to the **denied_list**. In addition, each query is copied to a file named `<query_ID>.txt` in the `<install_path>/internal/auto_restart/<timestamp>/denied_list` directory.

Note

If an inflight query fails, none of the inflight queries are added to the warned list. Instead, the failed queries are added to the denied list.

- If AnzoGraph runs a query in safe mode and cannot determine if it should be added to the denied or warned list, those queries are copied to a file named `<query_ID>.txt` in the `<install_path>/internal/auto_restart/<timestamp>/unanalyzed_list` directory.
- Metadata about the warned_list, denied_list, and unanalyzed_list queries is captured in the **stc_blocklist** system table.

Note

The **auto_restart_directory** setting in the system configuration file, `<install_path>/config/settings.conf`, controls the location of the auto_restart directories listed above. For more information about the setting, see the [Automated Restart System Settings](#) section below.

4. After the inflight queries have been run, AnzoGraph restarts the database, loads the persisted data back into memory, and returns the system to normal operation.

To help prevent the circumstance that caused the database to crash, any queries that were added to the **denied** list are blocked from being executed when the system returns to normal operation. When a user runs a query, AnzoGraph compares that query with the denied list. If the query is on the list, the query is terminated and AnzoGraph returns an "Attempting to execute a denied-listed query" error message. Queries on the warned list are not blocked. A denied list query cannot be run unless it is removed from the denied list. This behavior is controlled by the **ignore_deniedlist_queries** setting. For more information about the setting, see the [Automated Restart System Settings](#) section below. For information about removing queries from the denied list, see [Removing a Query from the Block List](#) below.

Automated Restart System Settings

The automatic restart feature is controlled by the following four settings in `<install_path>/config/settings.conf`:

- **auto_restart_max_attempts**: This setting specifies the number of times the system manager should attempt to start the database after a crash. The default value is **5**, which means the system manager will attempt to restart the database a maximum of 5 times. Changing `auto_restart_max_attempts` to **0** disables the auto-restart feature.
- **auto_restart_time**: This setting specifies the number of seconds to spend attempting to restart the database. If all attempts fail and this time limit is reached, the system manager stops trying to restart the database. The

default value is **600**, which means that the system manager will attempt to restart the database for a maximum of 600 seconds (10 minutes).

- **auto_restart_directory**: This setting specifies the base location of the **auto_restart** directory, which contains the **denied_list**, **warned_list**, and **unanalyzed_list** directories. The default value is `<install_path>/internal`.
- **ignore_deniedlist_queries**: This setting controls whether denied list queries are blocked from running or are allowed to be run when the database is returned to normal operation. The default value is **false**, which means denied list queries are not ignored and are therefore blocked from running. If **ignore_deniedlist_queries** is **true**, incoming queries are not compared with the denied list and are run.

Important

Changing the **auto_restart_max_attempts**, **auto_restart_time**, or **auto_restart_directory** values requires a restart of the system management daemon, **azgmgrd**, as well as the database. See [Starting and Stopping AnzoGraph](#) for instructions.

Removing a Query from the Block List

AnzoGraph stores metadata about the denied and warned list queries in the **stc_blocklist** system table. To remove a query from either list, you remove the entry from the **stc_blocklist** table by running the **REMOVE_FROM_BLOCKLIST** command.

```
REMOVE_FROM_BLOCKLIST '<list_name>' <query_ID>
```

Where **<list_name>** is the name of the list that the query is on and **<query_ID>** is the ID number for the query. To retrieve the list name and query ID values, run the following query to return the **stc_blocklist** contents:

```
SELECT * WHERE { TABLE 'stc_blocklist' } ORDER BY ?blocklist
```

For example:

```
/opt/anzograph/bin/azgi -c "select * where {table 'stc_blocklist'} order by ?blocklist"
```

query	blocklist	updated	query_text	part
3587	denied_list	2020-08-25 14:29:27	select * from <http://an..	0
3592	denied_list	2020-08-25 14:29:32	select * where {?s ?p ?o}	0
3612	warned_list	2020-08-25 14:32:15	select * from <http://an..	0

In the results, the **<list_name>** is the value in the **blocklist** column, and **<query_id>** is the value in the **query** column. Running the following command removes the first entry from the **stc_blocklist** table, which removes that query from the denied list.

```
REMOVE_FROM_BLOCKLIST 'denied_list' 3587
```

Related Topics

[Changing AnzoGraph Configuration Settings](#)

[AnzoGraph System Settings Reference](#)

[Starting and Stopping AnzoGraph](#)

Enabling Paged Data Mode (Preview)

By default, AnzoGraph is configured as an in-memory database. In memory mode, all graphs are stored in memory and all queries are run against the data in memory. Data is persisted to disk only for backup purposes as well as automatic loading of graphs back into memory when the database is restarted. You have the option, however, to configure AnzoGraph as a disk-based database, where all of the data is stored on disk and then paged into memory on-demand for running analytics.

Note

The Paged Data feature is available as a **Preview** release in **2.3.x** versions of AnzoGraph, 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.

How Does Paged Data Mode Work?

The procedure below gives an overview of how AnzoGraph operates in paged data mode:

1. First, just like in-memory mode, you load data into AnzoGraph before running queries.
2. As data is loaded, it passes through memory to be converted to AnzoGraph's internal storage format, and then it is saved to disk in the persistence directory. The persistence directory location is configurable, and the speed of the disk that hosts the directory has an impact on query performance. For the best performance, store the persistence directory on a fast disk, such as SSD.
3. AnzoGraph keeps the most recently accessed data cached in memory for queries. By default, the size of the cache is 20% of the total available memory. The percentage of memory to use for paged data caching is configurable. For more information, see [paged_cache_memory_percent](#).
4. As queries are run, AnzoGraph keeps track of the data that is accessed most often and keeps that data cached in memory. If a query requests data that is not currently cached, AnzoGraph releases the least accessed data from memory and loads the relevant data into memory.

Enabling and Configuring Paged Data Mode

Follow the steps below to configure AnzoGraph for paged data storage. Before changing the configuration, make sure that your environment meets the requirements in [Sizing Guidelines for Disk-Based Storage \(Preview\)](#).

Important

Though enabling paged data does not change the way users interact with the database, i.e., data loading and query operations remain the same, the performance of user operations will likely be slower compared to the default in-memory operation. In addition, enabling paged data requires you to re-initialize the database to remove the existing persistence.

1. Stop the database. See [Stop the Database and Leave the System Management Daemon Running](#) for instructions.
2. **On the leader node**, open the AnzoGraph settings file, **settings.conf**, in a text editor. The file is in the `<install_path>/config` directory.
3. In **settings.conf**, locate the `# paged_data=false` line. This setting enables and disables paged data storage. Uncomment the line and change the value to **true** to enable paged data.

```
paged_data=true
```

4. The following settings are also related to paged data operations. If necessary, uncomment the lines for any of these settings and modify the values as needed:
 - **paged_cache_memory_percent**: This setting controls the amount of memory (as a percentage of total memory) to use for caching the most often accessed data. The default value is **20**, which means AnzoGraph is configured to use 20% of the total available memory for caching data for analytics. If a query requests data that is not currently cached, AnzoGraph releases the least used data from memory and loads the relevant data into memory.

Important

Cambridge Semantics recommends that you do not set this value higher than 30.

- **enable_persistence**: Persistence must be enabled when using paged data mode. This setting is **false** by default. See [Using AnzoGraph Persistence \(Preview\)](#) for information about enabling AnzoGraph persistence.
 - **persistence_directory**: The directory where AnzoGraph saves the data that is persisted to disk. By default, the data is saved in the `<install_path>/persistence` directory. To persist data to an alternate disk, such as a separate SSD, specify the path and directory name.
5. Save and close **settings.conf**.
 6. Restart and re-initialize the database to apply the configuration change and remove any existing persisted data. See [Reinitializing the Database](#) for instructions. When AnzoGraph starts, reload the database from your original files or insert queries.

Related Topics

[Changing AnzoGraph Configuration Settings](#)

[AnzoGraph System Settings Reference](#)

AnzoGraph System Settings Reference

This section provides reference information for the AnzoGraph system configuration settings. For instructions on changing settings, see [Changing AnzoGraph Configuration Settings](#).

The table below describes the basic-level settings. Additional advanced-level settings are available for use by system administrators or users with an advanced level of knowledge about AnzoGraph or databases in general. See the configuration file, `<install_path>/config/settings.conf`, for descriptions of the advanced settings.

Setting	Description	Default Value (Type)
enable_persistence	Controls AnzoGraph's save data to disk option. For more information, see Using AnzoGraph Persistence (Preview) .	false (boolean)
enable_sparql_protocol	Whether to enable the HTTP SPARQL protocol service. The sparql_protocol_port setting controls the port to use. Note Enabling the SPARQL HTTP protocol opens the standard SPARQL-compliant HTTP endpoint. Unlike the Anzo protocol endpoint, the SPARQL HTTP endpoint is not secured.	false (boolean)
enable_ssl_protocol	Whether to enable the HTTPS SPARQL protocol service. The ssl_protocol_port setting controls the port to use. Note Enabling the SPARQL HTTPS protocol opens the standard SPARQL-compliant HTTPS endpoint. Unlike the Anzo protocol endpoint, the SPARQL HTTPS endpoint is encrypted but not authenticated.	false (boolean)

Setting	Description	Default Value (Type)
internal_directory	The directory where AnzoGraph should save internal database-related files such as generated code, logs, and query plans. For more information, see Relocating AnzoGraph Directories .	Not set (char) The default directory is <install_path>/internal .
max_memory	Specifies the amount of memory (in MB) that is available for AnzoGraph. The default is system-based; at startup, AnzoGraph determines the amount of RAM that is available and sets max_memory. In test environments where AnzoGraph may be co-located with other programs, you can set the max_memory value to put a limit on the amount of memory AnzoGraph can use. However, Cambridge Semantics recommends that you do not set max_memory unless instructed by Support.	System-based (int)
output_format	Specifies the default output format for AnzoGraph responses. Valid values are xml , json , or csv .	xml (char)
persistence_directory	The directory where AnzoGraph should save data when it is persisted to disk. For more information, see Relocating AnzoGraph Directories .	Not set (char) The default directory for persisted data is <install_path>/persistence .
sparql_protocol_port	SPARQL service HTTP port to use if enable_sparql_protocol is true .	7070 (int)
sparql_spec_default_graph	Controls the default scope of SPARQL queries when FROM clauses are excluded from a query. When false , queries without FROM clauses target the default graph (DEFAULTSET) only. Triples in named graphs will not be included in the scope of the query. When true , AnzoGraph conforms to the SPARQL specification and includes the default graph and all named graphs in the scope of a query that omits the FROM clause. For more information, see Changing the Default FROM Clause Behavior .	false (boolean)

Setting	Description	Default Value (Type)
spill_directory	<p>The directory where AnzoGraph should save temporary query files that spill to disk. For more information, see Relocating AnzoGraph Directories.</p> <div> <p>Important</p> <p>AnzoGraph uses O_DIRECT to read the spill files into the database. If you relocate the spill directory, make sure to place it on an ext4 file system that supports O_DIRECT.</p> </div>	<p>Not set (char)</p> <p>The default directory for spill files is <install_path>/spill.</p>
ssl_protocol_port	SPARQL service HTTPS port to use if enable_ssl_protocol is true.	8256 (int)
startup_info	Specifies how verbose the startup message is: - 0 -quiet, 1 -ready, 2 -ports, 3 -more.	1 (int)
stop_timeout	The number of seconds to wait for queries to finish before stopping the database.	30 (int)
truncate_clob	Specifies whether to truncate large strings to the maximum string size (1 MB).	false (boolean)
use_custom_ssl_files	Specifies whether to use custom SSL files containing fully qualified domain names.	false (boolean)
user_queues	Sets the limit on the number of queries that can run concurrently.	40 (int)
anzo_protocol_port	The Anzo protocol (gRPC) port for secure communication between AnzoGraph and Anzo.	5700 (int)
enable_root_user	Whether to allow a user running with root privileges to start AnzoGraph.	false (boolean)

Setting	Description	Default Value (Type)
auto_restart_directory	Specifies the base location of the auto_restart directory, which contains the denied_list , warned_list , and unanalyzed_list directories. For more information about the auto-restart feature, see Managing the Automatic Restart Feature .	Not set (char) The default location for the auto_restart directory is <install_path>/internal .
auto_restart_max_attempts	Specifies the number of times the system manager should attempt to start the database after a crash. The default value is 5 , which means the system manager will attempt to restart the database a maximum of 5 times. Changing auto_restart_max_attempts to 0 disables the auto-restart feature. For more information about the auto-restart feature, see Managing the Automatic Restart Feature .	5 (int)
auto_restart_time	Specifies the number of seconds to spend attempting to restart the database. If all attempts fail and this time limit is reached, the system manager stops trying to restart the database. The default value is 600 , which means that the system manager will attempt to restart the database for a maximum of 600 seconds (10 minutes). For more information about the auto-restart feature, see Managing the Automatic Restart Feature .	600 (int)
ignore_deniedlist_queries	Controls whether denied list queries are blocked from running or are allowed to be run when the database is returned to normal operation. The default value is true , which means denied list queries are ignored. Incoming queries are not compared with the denied list and are permitted to run. If ignore_deniedlist_queries is false , denied list queries are not ignored and are therefore blocked from running until they are removed from the denied list. For more information about the auto-restart feature, see Managing the Automatic Restart Feature .	true (boolean)

Setting	Description	Default Value (Type)
enable_unbound_variables	Controls whether AnzoGraph returns an empty result or an error if a query references a missing graph or includes unbound variables. This value is set to false by default, which means AnzoGraph returns an error. For more information, see Ignoring Missing Graphs .	false (boolean)
jvm_max_memory	Specifies the maximum size of the heap that can be used by the embedded Java virtual machine (JVM). Use k , m , or g (case insensitive) for KiB, MiB, or GiB. You can also specify % to indicate a percentage of the total memory that is available to AnzoGraph. By default, this value is not set, which means jvm_max_memory defaults to either 5% of the total memory or 4g , whichever value is smaller.	Not set (char) When not set, the default is 5% or 4g, depending on which value is smaller.
jvm_options	Lists any optional parameters to use for configuring the embedded JVM. Use a semicolon-delimited (;) list to specify multiple parameters. For information about JVM options, see Options in the Java Documentation.	Not set (char)
aws_log_level	AnzoGraph uses an AWS C++ SDK for loading data from S3. This setting controls the logging level for the AWS SDK. The default value is 2 , which is Error level logging.	2 (int) Valid values: <ul style="list-style-type: none"> • 0 (Off) • 1 (Fatal) • 2 (Error) • 3 (Warn) • 4 (info) • 5 (Debug) • 6 (Trace)
aws_search_regions	Lists the regions to search for AWS S3 buckets.	Not set

Setting	Description	Default Value (Type)
log_directory	<p>Specifies where to write system management daemon (azgmgrd) log files. These types of logs (azgmgrd.log, azgctl-<user>.log, azgpdmgr.log, and azgpids.log) are created before the system is initialized and may be written before the <install_path>/internal/log directory exists. Therefore, they are located outside of the AnzoGraph file system, /tmp by default. If you change the log_directory value, Cambridge Semantics recommends that you choose another location that is outside the internalAnzoGraph directories.</p>	<p>Not set</p> <p>When not set, the default location is /tmp.</p>
paged_data	<p>Enables or disables AnzoGraph's paged data feature, which controls whether data is stored in memory or on disk. When this option is false (the default value), data is stored in memory. Setting this option to true changes data storage from in-memory to on-disk (in the persistence_directory).</p> <div> <p>Important</p> <p>Enabling this option changes underlying database operations. Before enabling paged data, make sure that the performance and storage impacts are well-understood and that your environment meets the requirements. See Sizing Guidelines for Disk-Based Storage (Preview) for details.</p> </div>	false (boolean)

Setting	Description	Default Value (Type)
paged_cache_memory_percent	<p>When paged_data is enabled, this setting controls the amount of memory (as a percentage of total memory) to use for caching the most recently requested data. The default value is 20, which means AnzoGraph is configured to use 20% of the total available memory for caching data for analytics. For example, if you have 1 TB of data on disk and 300 GB of available RAM, AnzoGraph caches in memory 60 GB of the most recently accessed data. If a query requests data that is not currently cached, AnzoGraph releases the least accessed data from memory and loads the relevant data into memory. Note that a portion of the paged cache memory percent is used for the overhead of tracking the pages that are accessed. For more information, see Enabling Paged Data Mode (Preview).</p> <div> <p>Important</p> <p>Cambridge Semantics recommends that you do not set this value higher than 30.</p> </div>	20 (int)

Related Topics

[Changing AnzoGraph Configuration Settings](#)

Generating Diagnostic Files with the System Manager

When Cambridge Semantics Support requests AnzoGraph diagnostic files for troubleshooting an issue, you can use the AnzoGraph system manager to generate the required system information. If you encounter an error and the database remains running, you run an XRAY command to produce the diagnostic files. If you encounter an error that crashes the database, you run a CRASHFETCH command to produce a "crashdump" that includes the diagnostic files. This section provides instructions for generating the diagnostic files using the AnzoGraph system manager. For instructions on retrieving diagnostic files from the Anzo Administration application, see [Retrieving AnzoGraph Diagnostic Files](#).

- [Generating an X-ray on a Running Database](#)
- [Generating a Crashdump after a Crash](#)

Generating an X-ray on a Running Database

If you encounter an error and the database remains running, run the following command to take an x-ray from the command line on the AnzoGraph leader server. This command creates a tarball that includes the necessary diagnostic files:

```
<install_path>/bin/azgctl -xray /<path>/<name>.xray
```

- **path:** The location on the server where you want to save the tarball.
- **name:** The name for the tarball. The name must be unique; AnzoGraph will not overwrite existing files.
- **.xray:** All x-ray files must be named with the .xray extension.

For example, this command runs an x-ray on the leader server:

```
/opt/anzograph/bin/azgctl -xray /tmp/query_error.xray
```

Generating a Crashdump after a Crash

If you encounter an issue that stops the database, AnzoGraph automatically generates diagnostic files for Support. Follow the instructions below to retrieve the files after a crash.

Note

The database does not need to be running to collect the crashdump.

1. Run the following command on the leader server to view a list of the available crash diagnostics.

```
<install_path>/bin/azgctl -crashlist
```

The results show a list of available crash dumps by timestamp. For example:

Crash ID	Time
520460982	2017-06-28 20:30:35
520457655	2017-06-28 20:28:25

2. Run the following command to retrieve the appropriate crash files. This command creates a tarball that includes the necessary files:

```
<install_path>/bin/azgctl -crashfetch crash_id /path/name.xray
```

- **crash_id:** The ID for the crash that you want to retrieve, as shown in the crashlist from the previous step. To automatically retrieve the latest crash files, omit the crash_id.
- **path:** The location on the server where you want to save the tarball.
- **name:** The name for the tarball. The name must be unique; AnzoGraph will not overwrite existing files.
- **.xray:** All crashdumps files must be named with the .xray extension.

For example, this command runs a crashfetch to capture the diagnostics with the ID 520457655:

```
/opt/anzograph/bin/azgctl -crashfetch 520457655 /tmp/query_crash.xray
```

This command captures the most recent crash diagnostic files:

```
/opt/anzograph/bin/azgctl -crashfetch /tmp/query_crash.xray
```

Tip

You can run the following command to remove all crash dumps from the server.

```
<install_path>/bin/azgctl -crashtoss
```

Related Topics

[Retrieving AnzoGraph Diagnostic Files](#)

Anzo Admin CLI

The Anzo command line interface (CLI) utility, called **anzo**, is an advanced administration tool for managing Anzo. It is primarily used for migrations and deployments. The topics in this section provide information about the CLI.

Note

To script user interface operations or control Anzo with the CLI, please contact Cambridge Semantics.

- [Setting up the Admin CLI](#)
- [Querying Graphmart Data](#)
- [Accessing a Graph's Metadata](#)
- [Specifying an Output Format](#)

Setting up the Admin CLI

Important

The anzo CLI is an advanced administration tool for managing Anzo. It is primarily used for migrations and deployments. To script user interface operations or control Anzo with the CLI, please contact Cambridge Semantics.

This topic provides instructions for configuring the admin command line interface, **anzo**, and viewing the help menu. The anzo client is in the `<install_path>/Client` directory.

- [Adding the CLI to the Anzo Service User PATH](#)
- [Configuring the CLI](#)
- [Viewing the CLI Help Menu](#)

Adding the CLI to the Anzo Service User PATH

Follow the instructions below to configure the PATH environment variable to include the Client directory so that you call the anzo CLI from anywhere.

1. If necessary, run the following command to become the Anzo service user:

```
sudo su - <anzo_user_name>
```

For example:

```
sudo su - anzo
```

2. Open `~/.bash_profile` in a text editor.

3. Change the PATH to the following value:

```
PATH=$PATH:$HOME/.local/bin:$HOME/bin:Anzo_install_path/Client
```

For example:

```
PATH=$PATH:$HOME/.local/bin:$HOME/bin:/opt/Anzo/Client
```

4. Save and close the file, and then run the following command:

```
source ~/.bash_profile
```

5. Type **anzo** to verify that you can access the CLI. For example:

```
[anzo@anzo-server ~]$ anzo
Anzo Command Line Client.
Copyright (c) 2017 - 2019 Cambridge Semantics Inc and others.
All rights reserved.
Version: 4.4.0.r201910171220
Type anzo help for usage
```

Configuring the CLI

Follow the instructions below to configure a settings file that specifies the default Anzo CLI configuration values for parameters such as host, port, user, and password. Specifying these details in the settings file eliminates the need to include those options in subsequent commands.

To create and populate the settings file, **settings.trig**, in your home directory, run the following command:

```
anzo setup <options>
```

Where *options* include the following choices:

-beep , --beep	beep when command is completed
-ds , --datasource <datasource>	URI of the datasource to query, if other than primary datasource.
	Option not available for dataset queries.
-h , --host <hostname>	anzo server hostname
-http , --http	Use http connection to server.
-p , --port <int>	anzo server port
-pause , --pause-exit	Wait for a user key entry before an abnormal exit.
-ssl , --use-ssl	Use SSL for connection.
-t , --timeout <timeout>	override the default 30 second timeout for operations
-timer , --timer	Print out the total operation time
-trace , --show-trace	Show stack trace for errors.
-trust , --trust-all	Trust all certificates including invalid ones
-u , --user <string>	username to connect with
-w , --password <string>	user's password

<code>-x , --exclude-prefixes</code>	Do not use prefixes defined in user settings to expand options,
	arguments, or to write RDF.
<code>-z , --settings <file></code>	override the default settings file location

For example:

```
anzo setup -h localhost -p 61616 -u sysadmin -w @nz0
```

Anzo creates the `settings.trig` file in the `~/user/.anzo` directory. You can edit the file as needed. The installation also includes a sample settings file, **settings_example.trig**, in the `Client` directory. You can view the sample file for reference. For example:

```
### standard prefixes
@prefix foaf      : <http://xmlns.com/foaf/0.1/> .
@prefix rdfs      : <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dc        : <http://purl.org/dc/elements/1.1/> .
@prefix xsd       : <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf       : <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
#### anzo prefixes:
@prefix cli : <http://openanzo.org/cli/> .
@prefix system : <http://openanzo.org/ontologies/2008/07/System#> .
@prefix anzo : <http://openanzo.org/ontologies/2008/07/Anzo#> .
@prefix ld : <http://cambridgesemantics.com/ontologies/2009/05/LinkedData#> .
@prefix anzwt : <http://cambridgesemantics.com/ontologies/2009/05/AnzoWebToolkit#> .
@prefix reg : <http://cambridgesemantics.com/registries/> .
@prefix ontserv : <http://cambridgesemantics.com/semanticServices/OntologyService#> .
@prefix ldserv : <http://cambridgesemantics.com/semanticServices/LinkedData#> .
cli:config {
  cli:config
#      system:user "" ;
#      system:password "" ;
  system:timeout "0";
  system:useSsl "false";
  system:port "61616";
  system:keystoreFile "${ANZO_CLI_HOME}/../Common/ssl/client.ks";
  system:keystoreType "JCEKS";
  system:keystorePassword "p@ssw0rd";
  system:truststoreFile "${ANZO_CLI_HOME}/../Common/ssl/client.ts";
  system:truststoreType "JCEKS";
  system:truststorePassword "p@ssw0rd";
  .
}
```

Viewing the CLI Help Menu

The CLI help menu lists all of the available subcommands. To view the subcommands, run **anzo help**.

usage: anzo <subcommand> [options] [args]

Anzo Command Line Client.

Type 'anzo help <subcommand>' for help with a specific subcommand.

Available subcommands:

acls	Ensure the graphs in a dataset inherit their ACLs from the dataset
analyze	Provides several flavors of analysis for Anzo request/response logs
call	Calls an anzo semantic service and prints the service response to the console
collapse	Collapse all URI arguments to prefixed URIs (CURIEs) using user defined prefixes
collapseGraph	In specified graph(s), collapse object properties with only one literal value into a datatype property
convert	Converts between the various RDF file formats
count	Counts the statements in an RDF file
create	Creates named graphs in the repository from the provided RDF
csv	Export instances of an ontology class with all of their property values
deploy	Import, export, or delete a linked data set and related components
deregister	Deregister given resource from appropriate registries based on rdf:type of resource
expand	Expands all prefixed URI (CURIE) arguments to expanded URIs using user defined prefix map
find	Retrieves statements from the server via simple pattern find
gen	Generates code for the ontologies as supplied by the input RDF or arguments
get	Retrieves named graphs from the server
graph2lds	Creates a Linked Data Set from the statements in a graph(s)
import	Imports statements into the repository, creating graphs in the repository as needed
inspectOntology	Inspects a dataset for an ontology
link	Link an excel workbook using a layout
load	Loads file based linked datasets
loadXML	Imports xml as statements into a graph in the repository as needed
ls	List resources from appropriate registries based on type of resource
play	Play back a sequence of recorded requests
query	Executes a SPARQL query against the repository or a local RDF file
rdfformats	Show available rdf formats
register	Register given resource to appropriate registries based on rdf:type of resource.

Supported types:[

- <http://cambridgesemantics.com/ontologies/2009/05/LinkedData#LinkedDataSet>
- <http://cambridgesemantics.com/ontologies/2009/05/LinkedData#LinkedDataCollection>
- <http://cambridgesemantics.com/ontologies/2009/05/LinkedData#LinkedDataCollectionInstance>
- <http://www.w3.org/2002/07/owl#Ontology>
- <http://openanzo.org/ontologies/2008/07/SemanticService#SemanticService>

```

    http://cambridgesemantics.com/ontologies/2009/05/Spreadsheets#LinkedWorkbook
    http://cambridgesemantics.com/ontologies/2009/05/AnzoWebToolkit#Component
    http://cambridgesemantics.com/ontologies/Graphmarts#Graphmart
    http://cambridgesemantics.com/ontologies/Graphmarts#Step
    http://cambridgesemantics.com/ontologies/Graphmarts#Layer
    http://cambridgesemantics.com/ontologies/Graphmarts#View
  ]
remove          Removes named graphs from the repository
replace         Replaces named graphs in the repository with the provided RDF
reset           Resets the repository, replacing all contents of repository with rdf
provided
retrieve        Retrieves content from the binary store and saves it in a local file
setup           Set up settings.trig file
sortedConvert    Converts between the various RDF file formats
store           Stores a local file in the Anzo server's binary store
union           Unions RDF from the arguments and optionally from STDIN as well
update          Updates existing graphs in the repository
uploadBundle     Upload bundle to server
uploadCertificate Upload trusted certificate to server
watch           Listens for changes to a graph and prints them out
xray            Export system tables into trig file

```

URI arguments to commands may either be fully qualified URIs ("http://...") or prefixed URIs ("dc:title").

The prefix mapping is defined in the users settings file.

User settings are loaded from a user's "~/.anzo/settings.trig" file.

See documentation for details.

To view the help for a specific subcommand, run **anzo help *command_name***. For example, the following command displays help for the find command:

```

[user@anzo Client]# ./anzo help find
usage: anzo find [options] [NAMED-GRAPH-URI...]
Retrieves statements from the server via simple pattern find.
-beep , --beep                beep when command is completed
-ds , --datasource <datasource>  URI of the datasource to query, if other than primary
datasource.

                                Option not available for dataset queries.
-f , --output-file <file>        write the find results to a file
-h , --host <hostname>          anzo server hostname
-http , --http                 Use http connection to server.
-lang , --literal-language <string> The literal language
-lit , --literal-object <string>  The literal object of find pattern
-n , --count                   Outputs only the total number of matching statements
-o , --output-format <rdf-Format> Override the default RDF format associated with the

```

```

RDF output(s)
-p , --port <int>                anzo server port
-pause , --pause-exit            Wait for a user key entry before an abnormal exit.
-pred , --predicate <URI>        The predicate of find pattern
-pretty , --pretty-print          PrettyPrint output (currently only json)
-ssl , --use-ssl                  Use SSL for connection.
-sub , --subject <subject>        The subject of find pattern
-t , --timeout <timeout>          override the default 30 second timeout for operations
-timer , --timer                  Print out the total operation time
-trace , --show-trace             Show stack trace for errors.
-trust , --trust-all             Trust all certificates including invalid ones
-type , --literal-datatype <URI> The literal datatype
-u , --user <string>              username to connect with
-uri , --uri-object <URI>         The uri object of find pattern
-w , --password <string>          user's password
-x , --exclude-prefixes           Do not use prefixes defined in user settings to expand
options,                          arguments, or to write RDF.
-z , --settings <file>           override the default settings file location

'help rdfformats' for list of available RDF formats.
Filename arguments default to the file format matching their filename extension.
STDIN and STDOUT default to 'trig'.

```

Querying Graphmart Data

Important

The anzo CLI is an advanced administration tool for managing Anzo. It is primarily used for migrations and deployments. To script user interface operations or control Anzo with the CLI, please contact Cambridge Semantics.

This topic provides information about using the anzo CLI to query graphmart data in AnzoGraph.

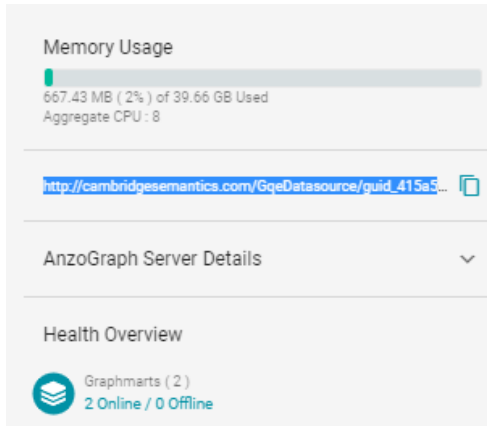
Use the **query** subcommand to access the data in graphmarts that are loaded in AnzoGraph:

```
anzo query "<query_text>" -ds <AZG_URI> -dataset <graphmart_URI>
```

If you saved the query in a file, run the following command to run the query in the file:

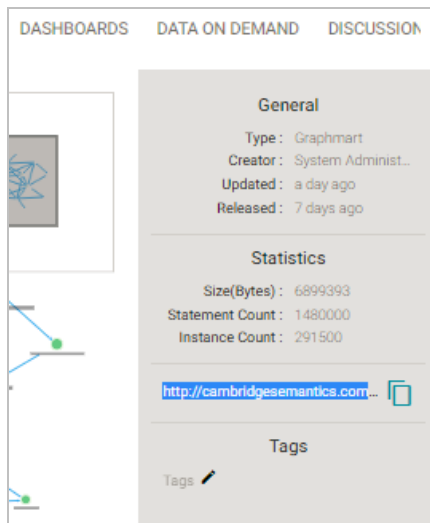
```
anzo query -f <filename>.rq -ds <AZG_URI> -dataset <graphmart_URI>
```

Where <filename>.rq is the path to and name of the query file and <AZG_URI> is the Datasource URI shown on the **Connections > AnzoGraph** screen in the Administration application. For example:



And <graphmart_URI> is the URI for graphmart. To view the URI for a graphmart:

1. Click **Graphmarts** in the **Blend** menu in the Anzo application.
2. On the Graphmarts screen, click the graphmart that you want to query.
3. On the details screen for the graphmart, you can view the graphmart URI in the statistics section. For example:



4. Click the clipboard icon (📋) to copy the graphmart URI to your clipboard.

Examples

The example below queries a data set to list its classes:

```
anzo query "SELECT DISTINCT ?p WHERE { ?s ?p ?o.} LIMIT 100"
-ds http://cambridgesemantics.com/GqeDatasource/guid_b833b32453694342c7bbc22422035e07
-dataset http://cambridgesemantics.com/Graphmart/f4bc354ebe9540329eef561f66e42454
```

This example runs a query in a file:

```
anzo query -f /home/user/queries/classes.rq
-ds http://cambridgesemantics.com/GqeDatasource/guid_b833b32453694342c7bbc22422035e07
-dataset http://cambridgesemantics.com/Graphmart/f4bc354ebe9540329eef561f66e42454
```

Accessing a Graph's Metadata

Important

The anzo CLI is an advanced administration tool for managing Anzo. It is primarily used for migrations and deployments. To script user interface operations or control Anzo with the CLI, please contact Cambridge Semantics.

Each graph has a metadata graph associated with it. The metadata graph includes details such as ACL information, the last modified date, and which user created and modified the graph. To include the metadata graph when you retrieve graph details, use the **get** subcommand with the **-m** option:

```
anzo get -m <URI>
```

The **-m** option indicates that you want to see the metadata graph for the specified URI. For example, the following command retrieves the metadata graph for a graphmart:

```
anzo get -m http://cambridgesemantics.com/Graphmart/89baf53cc5644600961778c88bd3d7fd
```

In addition to showing the graphmart details for the `<http://cambridgesemantics.com/Graphmart/89baf53cc5644600961778c88bd3d7fd>` graph, the results include the additional metadata for the graph:

```
...
<http://openanzo.org/metadataGraphs
(http%3A%2F%2Fcambridgesemantics.com%2FGraphmart%2F89baf53cc5644600961778c88bd3d7fd)>
{
  <http://cambridgesemantics.com/Graphmart/89baf53cc5644600961778c88bd3d7fd> a
  anzo:NamedGraph ;
    anzo:createdBy <http://openanzo.org/system/internal/sysadmin> ;
    anzo:lastModifiedByUser <http://openanzo.org/system/internal/sysadmin> ;
    anzo:created "2020-03-24T17:25:48.004Z"^^xsd:dateTime ;
    anzo:datasource datasource:systemDatasource ;
  ...
}
```

Specifying an Output Format

Important

The anzo CLI is an advanced administration tool for managing Anzo. It is primarily used for migrations and deployments. To script user interface operations or control Anzo with the CLI, please contact Cambridge Semantics.

The Anzo CLI enables you to request results in the following formats: TriG (default), RDF, RDFS, XML, NT, N3, TTL, TriX, and JSON. To change the format for results, you use the `-o` option with Anzo subcommands such as `find`, `get`, `query`, `call`, and `analyze`.

For example, the following `get` subcommand returns data set details in XML format:

```
anzo get -o xml http://csi.com/FileBasedLinkedDataSet/059060234accd1d2d44b6bbb4207ee54
```

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ld="http://cambridgesemantics.com/ontologies/2009/05/LinkedData#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://csi.com/DataLocation/059060234accd1d2d44b6bbb4207ee54">
    <fileConnection xmlns="http://cambridgesemantics.com/ontologies/DataSources#"
      rdf:resource="http://cambridgesemantics.com/File_Connection/local"/>
    <filePath xmlns="http://cambridgesemantics.com/ontologies/DataSources#"
      /nfs/data/store/LoadMovies_223d3/</filePath>
    <isPrimary xmlns="http://cambridgesemantics.com/ontologies/DataSources#"
      rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">true</isPrimary>
  </rdf:type
  rdf:resource="http://cambridgesemantics.com/ontologies/DataSources#DataLocation"/>
  <rdf:type
  rdf:resource="http://cambridgesemantics.com/ontologies/DataSources#PathConnection"/>
</rdf:Description>
```

Developer Guide

The Developer Guide provides information about using the Anzo Java software development kit (SDK) to develop custom extensions for Anzo.

The Anzo system, including the SDK, is built using the Open Service Gateway Initiative (OSGi) as a packaging mechanism. OSGi is a Java packaging and runtime environment that enables Anzo to load and unload extensions easily. Certain components, such as Anzo Semantic Services, are packaged into an OSGi bundle and then loaded into the server. For an introductory description of OSGi, see [What is OSGi?](#) on the OSGi Alliance website. Note that a deep understanding of OSGi is not necessary for creating Anzo extensions with the Anzo SDK.

The topics in this section list the SDK requirements and provide instructions for deploying, testing, and using the Anzo SDK.

- [Deploying the Anzo Java SDK](#)

Deploying the Anzo Java SDK

This topic provides instructions for setting up an Anzo development environment using the Anzo software development kit (SDK) and Eclipse integrated development environment (IDE). The sample instructions below deploy the Anzo SDK in a Windows environment with Eclipse IDE for Java Developers Version 4.12.0. Anzo SDK and Eclipse can also be deployed on Linux and Mac operating systems.

Requirements

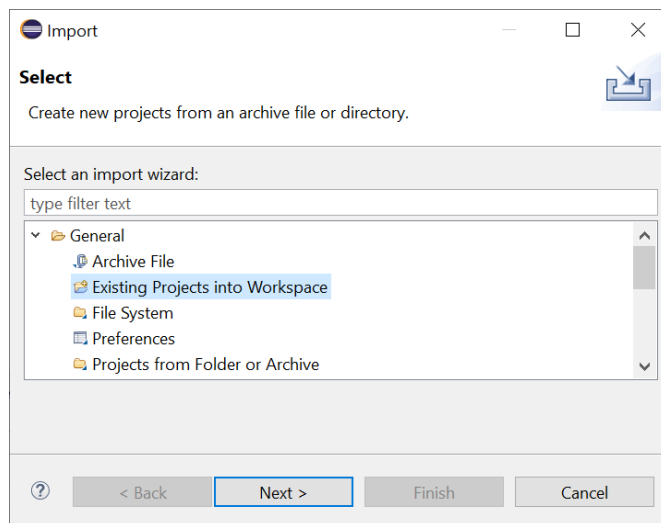
Make sure that the Anzo development server meets the requirements in [Anzo Requirements](#). In addition, install the following programs for working with the Anzo Java SDK:

- Eclipse for Java Developers Version 4.7.3+: Install the **Eclipse IDE for Java Developers** or **Eclipse IDE for Enterprise Java Developers**.
- Java Runtime Environment Version 8: Eclipse and the Anzo SDK require JDK version 8. Cambridge Semantics tests with jdk1.8.0_181.

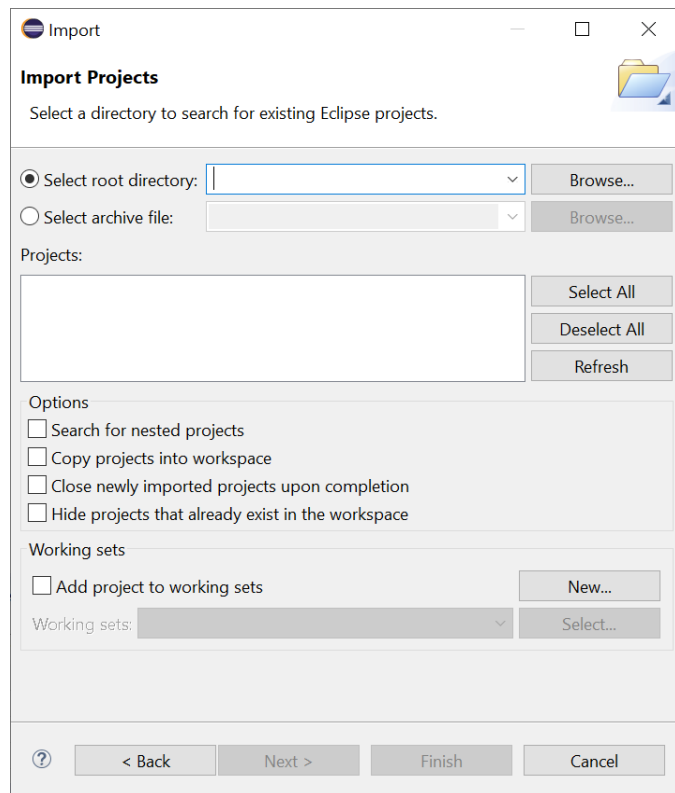
Deploying the Anzo SDK with Eclipse

Follow the instructions below to import the Anzo Java SDK to Eclipse and configure and test the environment.

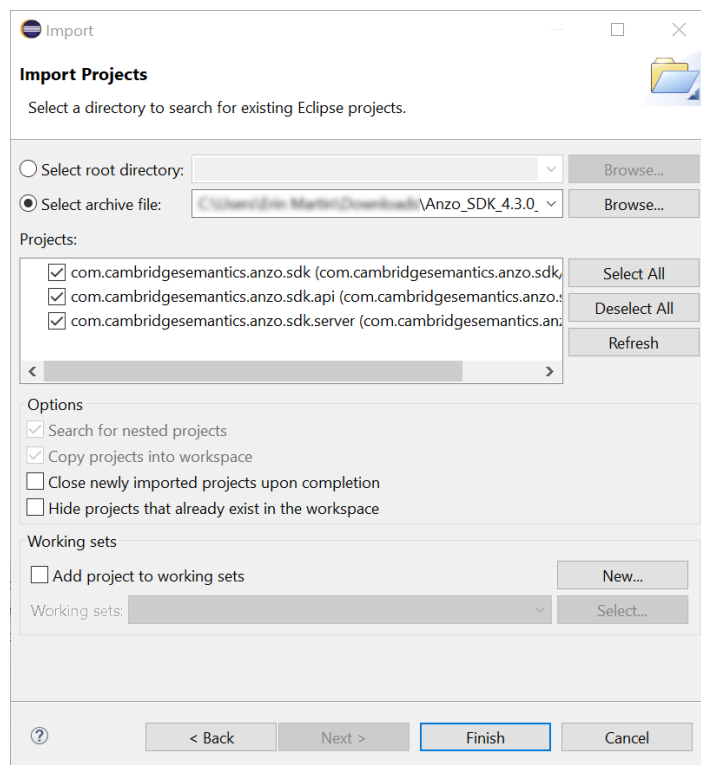
1. Download the Anzo SDK .zip file to the host server. Do not unpack the file.
2. In Eclipse, click the **File** menu and select **Import**. Eclipse opens the Import dialog box. For example:



3. In the Import dialog box, expand the **General** folder and select **Existing Projects into Workspace** and click **Next**. Eclipse opens the Import Projects dialog box. For example:



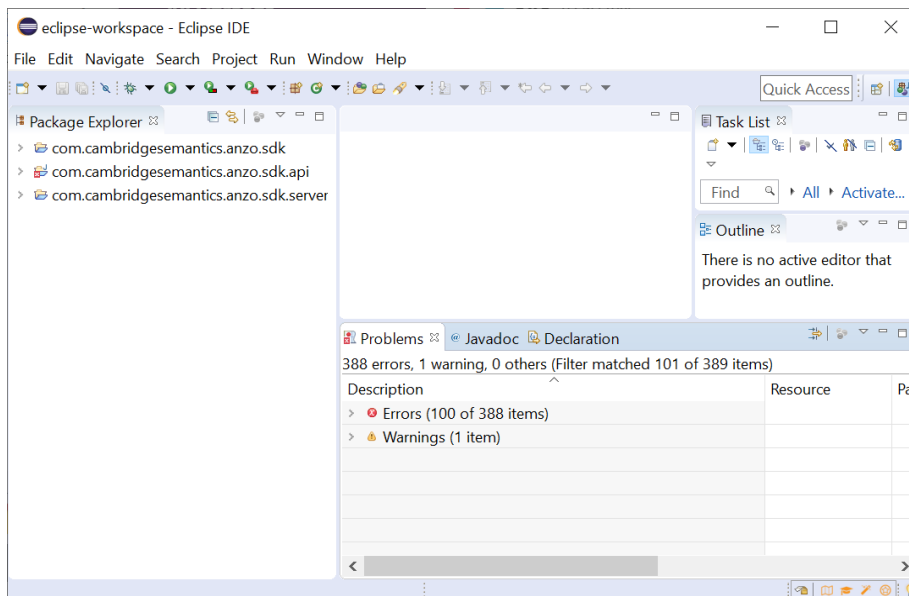
4. Select the **Select archive file** radio button and then browse to and select the Anzo SDK .zip file. Eclipse loads the .zip file and lists the contents in the Projects field. For example:



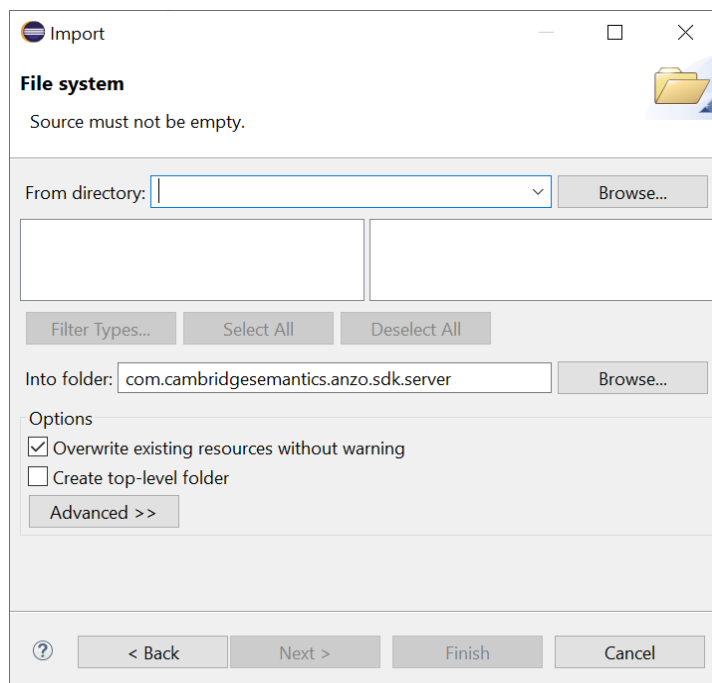
The Anzo SDK contains three projects:

- **com.cambridgesemantics.anzo.sdk**: This core project is required for creating solutions. It contains the Anzo libraries that provide the Anzo APIs and extension points as well as the libraries that enable Anzo to run in the development environment.
- **com.cambridgesemantics.anzo.sdk.server** This core project is required for creating solutions. It contains configuration files for running Anzo as well as a launcher for starting the Anzo server.
- **com.cambridgesemantics.anzo.sdk.api**: This is an example project that contains sample Java programs that illustrate several aspects of the Anzo client APIs. Each program is a simple example that demonstrates how to communicate with the Anzo server to read, write, and query data. See the comments in each example for an explanation of what each one demonstrates.

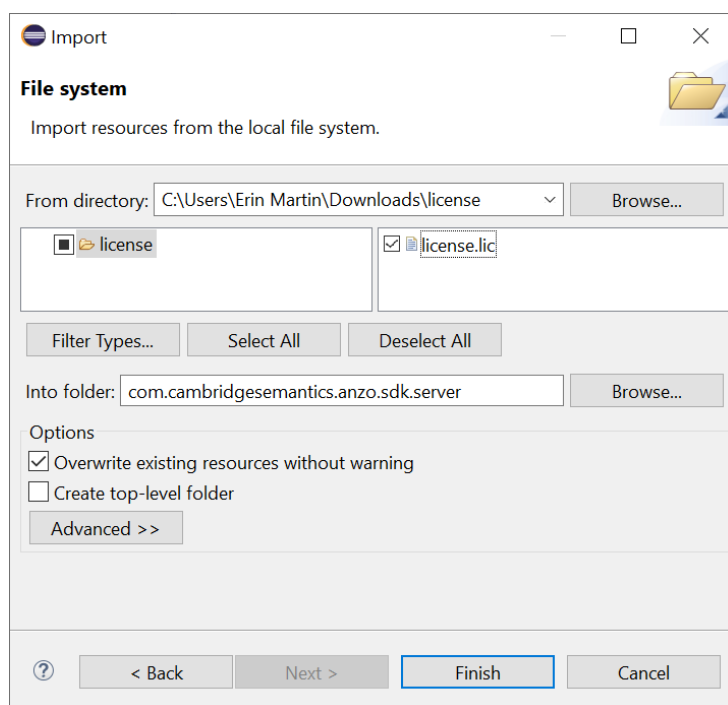
5. Click **Finish** to import the Anzo SDK .jar files. The process may take a few minutes. When the import is complete, Eclipse opens the workspace. At this point in the process, expect to see several errors in the workspace. For example:



6. Import your Anzo license:
 - a. Make sure that you have a copy of the Anzo license on the server. If necessary, you can view and download a copy from the [Cambridge Semantics Support Center](#).
 - b. Rename the license file so its file extension is .lic. For example, **license.lic**.
 - c. In the Eclipse Package Explorer, right-click **com.cambridgesemantics.anzo.sdk.server** and select **Import**.
 - d. In the Import dialog box, expand the **General** folder and select **File System**. Then click **Next**. Eclipse opens the File System Import dialog box. For example:



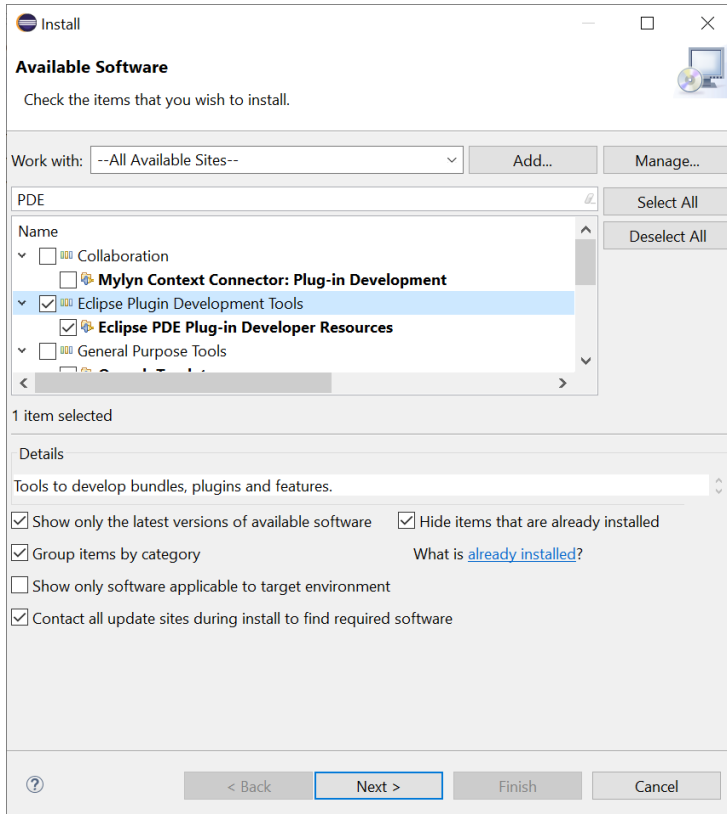
- e. Click the **Browse** button next to the From directory field and select the directory that contains the license file. Eclipse displays the directory and its contents.



- f. Select the license file in the right pane, and then click **Finish**.

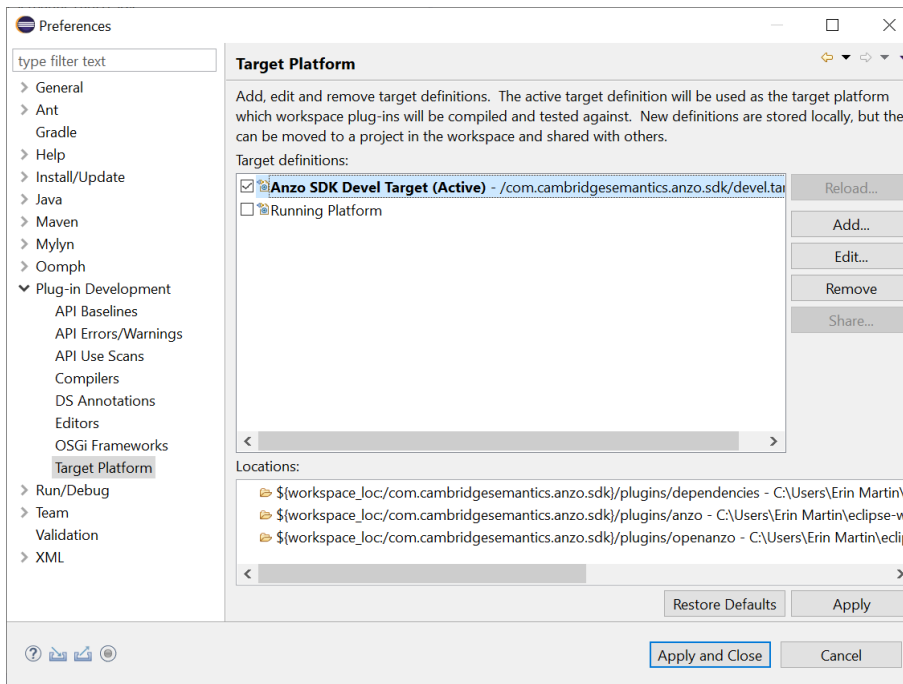
7. Install the Eclipse Plugin Development Tools:

- a. Click the **Help** menu and select **Install New Software**. Eclipse opens the Install dialog box.
- b. In the Install dialog box, click the **Work with** drop-down list and select **All Available Sites**. In the search field below the Work with field, type "PDE" and wait for Eclipse to find the plugin tools. Select the checkbox next to **Eclipse Plugin Development Tools**, including **Eclipse PDE Plug-in Developer Resources**. For example:



- c. Click **Next** and accept the license agreement, then click **Finish**. Eclipse installs the software and then prompts you to restart the application.
8. After restarting Eclipse, load the Anzo SDK Target Platform:
- a. Click the **Window** menu and select **Preferences**.
 - b. In the Preferences dialog box, expand **Plug-in Development** and select **Target Platform**.

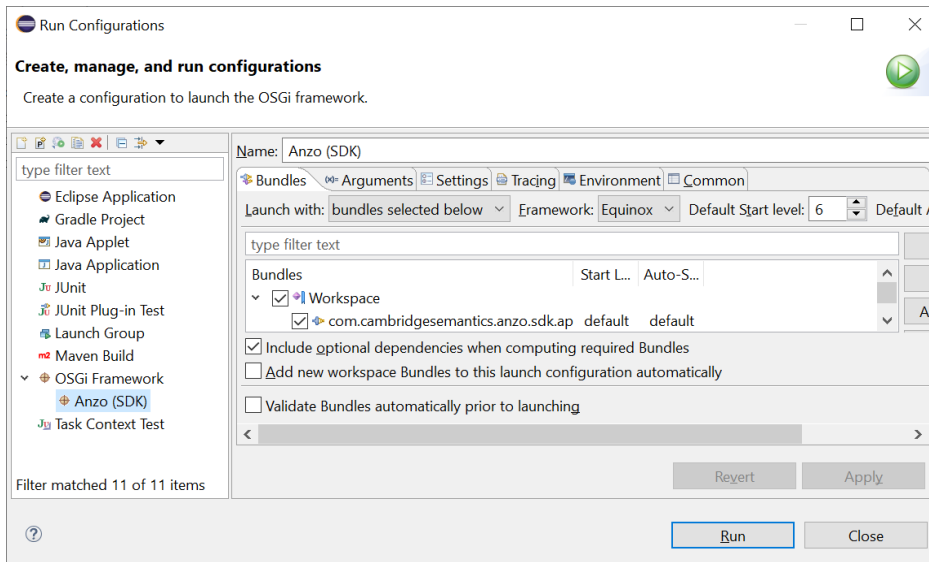
- c. In the Target Platform definitions, select the **Anzo SDK Devel Target** checkbox. For example:



- d. Click **Apply and Close**. Eclipse loads the Anzo SDK Target Platform.

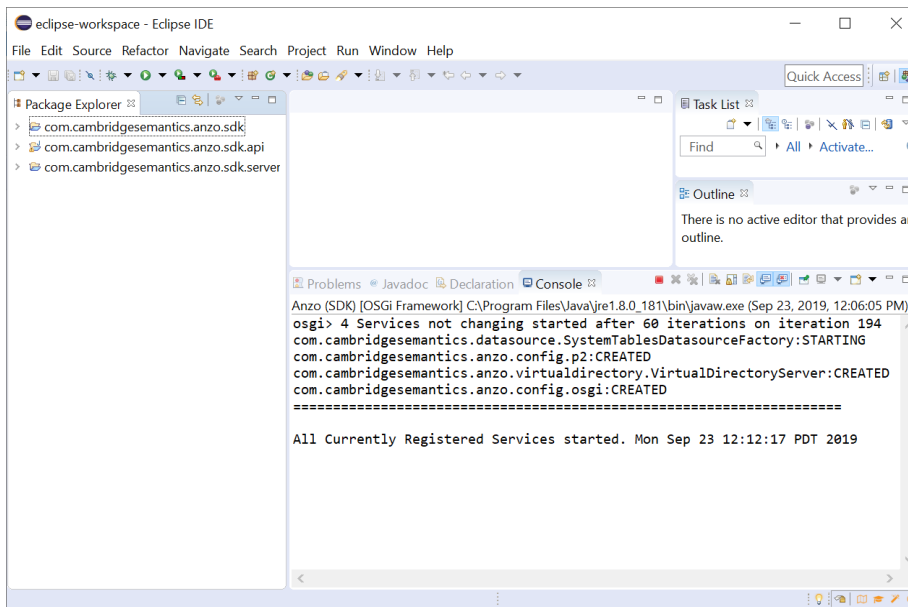
9. Test the environment:

- a. In the Eclipse workspace, click the **Run** menu and select **Run Configurations**. Eclipse opens the Run Configurations dialog box.
- b. On the left side of the dialog box, expand the **OSGi Framework** folder and select **Anzo (SDK)**. For example:



- c. Click **Run** to run the Anzo SDK target platform. A Console tab opens in Eclipse and shows the status messages. When Anzo starts, the console displays the message "All Currently Registered Services started." For

example:

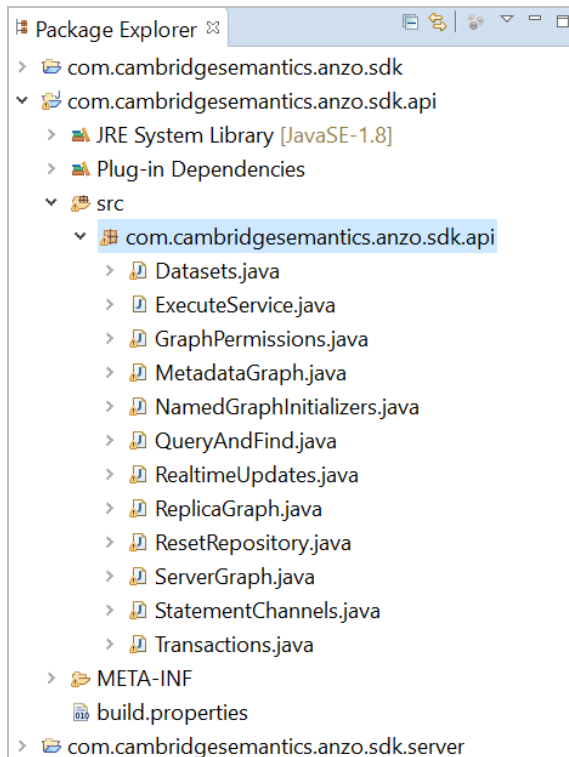


If Anzo fails to start, one of the common reasons for the failure is that one or more of the Anzo ports are in use by other software. See [Firewall Requirements](#) for information about the ports that Anzo uses.

Note

If you deployed the Anzo SDK on Windows, Eclipse displays Spark-related error messages such as "java.io.FileNotFoundException: Source '...\com.cambridgesemantics.anzo.sdk.server\spark' does not exist." The errors occur because Spark is not supported on Windows operating systems. You cannot run ETL jobs locally, but the errors do not affect the ability to develop Anzo extensions.

To explore the sample Java programs that are included in the Anzo SDK, expand the **com.cambridgesemantics.anzo.sdk.api** package in the Package Explorer. In the package, expand the **src** directory and then the **com.cambridgesemantics.anzo.sdk.api** directory to see the list of sample programs. For example:



To run a program, right-click the .java file and select **Run As > Java Application**. For more information about using the Anzo SDK, see the **Anzo Java SDK Guide.pdf** that is distributed in the SDK .zip file.

Troubleshooting

The topics in this section provide troubleshooting information for Anzo components.

- [Getting Information from the Anzo Log Files](#)
- [Viewing the Current Stack in a Browser](#)
- [Error Message Reference](#)

Getting Information from the Anzo Log Files

You can review the Anzo log files to get more detailed information about errors or to obtain more granular information about server operations. The server writes logs to the `<install_path>/Server/logs` directory and adds timestamps to all logged statements. Major issues are logged in files with the suffix "error," and other server information is logged in files with the suffix "info." For information about viewing and managing Anzo logs, see [Managing Anzo Logging](#).

Related Topics

[Viewing the Current Stack in a Browser](#)

[Error Message Reference](#)

Viewing the Current Stack in a Browser

When the System Monitor service is configured to save heap and/or stack dumps, those dumps are saved to disk and cannot be viewed from the Administration application. However, the sysadmin user can quickly review the stack for the current state of the JVM in a browser. Follow the instructions below to view the stack.

Note

Only a user with sysadmin access can view the stack in a browser. The sysadmin credentials are required to log in to the stack page.

To review the stack for the current state, go to the following URL in a browser:

```
https://<Anzo_server>:<HTTPS_admin_port>/status?stack
```

Where <Anzo_server> is the IP address or host name for the Anzo server and <HTTPS_admin_port> is the HTTPS port for the Administration application. For example:

```
https://10.11.0.12:8946/status?stack
```

The browser prompts you to log in as the **sysadmin** user. Supply the credentials and click **Sign in**.

The current state is displayed. For example:

```
2:Reference Handler
  Cpu: 0.00%
  Priority: 10 WAITING
  BlockedCount:487 BlockedTime:-1
  WaitedCount:432 WaitedTime:-1
  LockName:java.lang.ref.Reference$Lock@b3a29cf
  LockOwnerId:-1
  LockOwnerName:null
  LockClassName:java.lang.ref.Reference$Lock
  LockMonitors:
  LockSynchronizers:
  Stack:
    java.lang.Object.wait(Native Method)
    java.lang.Object.wait(Object.java:502)
    java.lang.ref.Reference.tryHandlePending(Reference.java:191)
    java.lang.ref.Reference$ReferenceHandler.run(Reference.java:153)

3:Finalizer
  Cpu: 0.00%
  Priority: 8 WAITING
  BlockedCount:1599 BlockedTime:-1
  WaitedCount:416 WaitedTime:-1
  LockName:java.lang.ref.ReferenceQueue$Lock@7941cc81
  LockOwnerId:-1
  LockOwnerName:null
  LockClassName:java.lang.ref.ReferenceQueue$Lock
  LockMonitors:
  LockSynchronizers:
  Stack:
    java.lang.Object.wait(Native Method)
    java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:144)
    java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:165)
    java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:216)
```

You can also check specifically for blocked or deadlocked threads by replacing **stack** in the URL with **block** or **deadlock**. To check for blocked threads, go to the following URL:

```
https://<Anzo_server>:<HTTPS_admin_port>/status?block
```

For example:

```
https://10.11.0.12:8946/status?block
```

To check for deadlocks, go to the URL below:

```
https://<Anzo_server>:<HTTPS_admin_port>/status?deadlock
```

For example:

```
https://10.11.0.12:8946/status?deadlock
```

Related Topics

[Managing Anzo Logging](#)

[Enabling and Configuring the System Monitor Service](#)

Error Message Reference

This topic provides information about Anzo and AnzoGraph and error messages.

- [Anzo Error Messages](#)
- [AnzoGraph Error Messages](#)

Anzo Error Messages

This section includes the possible causes and solutions for Anzo error messages. Click a message in the list below to view details about that error:

- [Application Service Failure](#)
- [Elasticsearch exception \[type=circuit_breaking_exception, reason=\[parent\] Data too large, data for \[<http_request>\]...](#)
- [Sparkler Exception: java.io.IOException: Unable to connect to provided ports 10000~10010](#)

Application Service Failure

This message indicates that the Anzo server cannot bind to the Application Port defined on the Server Settings page in the Administration application. The problem has two likely causes:

- Another program is bound to the defined Anzo Server Application Port.
- You are not running as the root user and lack the required permission.

To resolve this issue, make sure that no other application is running on the defined Application port and log in as the root user if Anzo is installed on a UNIX operating system.

Elasticsearch exception [type=circuit_breaking_exception, reason=[parent] Data too large, data for [<http_request>]...

This message indicates that the Elasticsearch heap size is not large enough to process the request. By default, Elasticsearch is configured to use a maximum heap size of 1 GB. Cambridge Semantics recommends that you increase the amount to 50% of the memory that is available on the server. To change the configuration, open the `<elasticsearch_install_dir>/config/jvm.options` file in an editor. At the top of the file, modify the **Xms** and **Xmx** values to replace the **1** with the new value. For example:

```
# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space

-Xms15g
-Xmx15g
```

Sparkler Exception: java.io.IOException: Unable to connect to provided ports 10000~10010

This message indicates that the Sparkler Livy RSC client ran out of the ports that it uses internally for running jobs. Increase the range of ports by adjusting the `livy.rsc.launcher.port.range` value in the `livy-client.conf` file. If you use the embedded Anzo Sparkler compiler, the file is in the `<install_path>/Server/spark/csi-livy-spark/conf` directory.

Cambridge Semantics recommends that you set `livy.rsc.launcher.port.range = 10000~10110`. Restart the Livy server after changing the configuration file.

AnzoGraph Error Messages

This section includes the possible causes and solutions for AnzoGraph error messages. Click a message in the list below to view details about that error:

- [Exiting: Error - Cannot execute as user 'root'. To override this security protection, set 'enable_root_user=true': Invalid user id](#)
- [Invalid Certificate](#)
- ["Compilation Failed" at Startup](#)

Exiting: Error - Cannot execute as user 'root'. To override this security protection, set 'enable_root_user=true': Invalid user id

This message indicates that you tried to start AnzoGraph as the root user and root access is disabled. Log in as the correct user, and then run the command again.

Invalid Certificate

This message indicates that you replaced the default AnzoGraph certificates with your own trusted certificates and the certificates are invalid. Certificates can be invalid because they expired or they were generated or signed incorrectly.

"Compilation Failed" at Startup

If AnzoGraph fails to start and you receive a "Compilation failed" message, it may indicate that some of the required GNU Compiler Collection (GCC) libraries are missing. Specifically, AnzoGraph requires the `glibc`, `glibc-devel`, and `gcc-c++` libraries. Typically when you install GCC by running `yum install gcc` those libraries are included as part of the package. In some cases, depending on the host server configuration, installing GCC excludes certain libraries. To install the missing libraries, run the following command:

```
sudo yum install glibc glibc-devel gcc-c++
```

Then start AnzoGraph again.

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 graph 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 AnzoGraph load files that are generated during the ETL process. All installations require at least one graph data source. You can create one graph data source and configure all pipelines to write to that graph source (each ETL run automatically creates a new sub-directory under the graph data source) or you can create multiple graph data sources to use for different data sets. For information about creating graph sources, see [Creating an Anzo Data Store](#).

What is the difference between a Dataset Pipeline and an ETL Pipeline?

Dataset pipelines are used to ingest data into Anzo. They produce new data sets in the Catalog and generate RDF files for loading data to AnzoGraph. All auto-ingested projects 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 data set 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 data set 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.

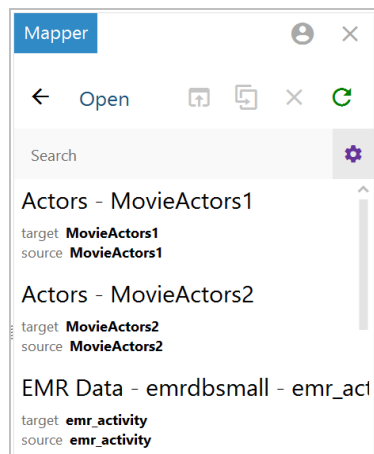
Update Type	File Location	Update Process
Added, deleted, or changed rows – Columns did not change	Uploaded from your computer and copied to the base upload path as described in Setting a Base File Store Path for File Uploads .	<ol style="list-style-type: none">1. Replace the file on the file store with the updated version of the file.2. Re-publish the job for this file or the entire pipeline to update the existing data set.3. Reload any graphmarts that include the updated data set and then refresh the affected Hi-Res Analytics dashboards to view the updated data.
	Selected from the File Store.	<ol style="list-style-type: none">1. Replace the file on the file store with the updated version of the file.2. Re-publish the job for this file or the entire pipeline to update the existing data set.3. Reload any graphmarts that include the updated data set and then refresh the affected Hi-Res Analytics dashboards to view the updated data.

Update Type	File Location	Update Process
Added, deleted, or changed columns and rows	Uploaded from your computer and copied to the base upload path as described in Setting a Base File Store Path for File Uploads .	<ol style="list-style-type: none"> 1. Replace the file on the file store with the updated version of the file. 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 Import Selected button to import the updated file. 3. Click Ingest and re-ingest the data source. 4. Publish the pipeline to update the existing data set. 5. Reload any graphmarts that include the updated data set and then refresh the affected Hi-Res Analytics dashboards to view the updated data.
	Selected from the File Store.	<ol style="list-style-type: none"> 1. Replace the file on the file store with the updated version of the file. 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 Import Selected button to import the updated file. 3. Click Ingest and re-ingest the data source. 4. Publish the pipeline to update the existing data set. 5. Reload any graphmarts that include the updated data set and then refresh the affected Hi-Res Analytics dashboards to view the updated data.

How do I duplicate a mapping?

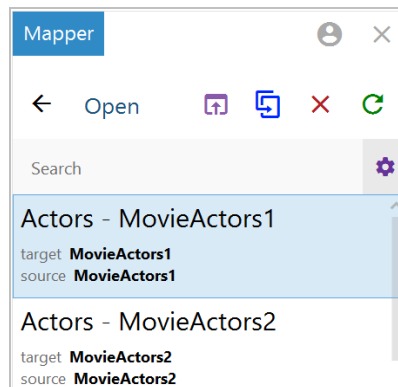
Using the Anzo for Office Excel plugin, users can duplicate mappings to use as a template for a new mapping. Follow the instructions below to duplicate a mapping.


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:

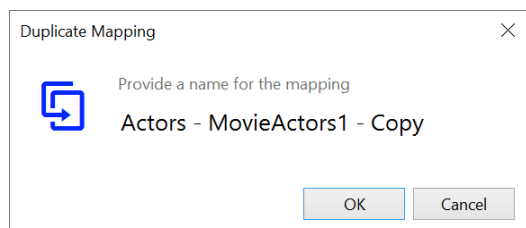


- Click the mapping that you want to duplicate. Selecting a mapping activates the buttons at the top of the screen.

For example:



- Click the **Duplicate** icon () to copy the selected mapping. Anzo displays the Duplicate Mapping dialog box. For example:



- 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.

- In the Anzo application, expand the **Blend** menu and click **Datasets**. Anzo displays the Dataset catalog, which lists the existing data sets.

- Click the data set that you want to add a model to. Anzo displays the Explore screen for the data set.
- 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.

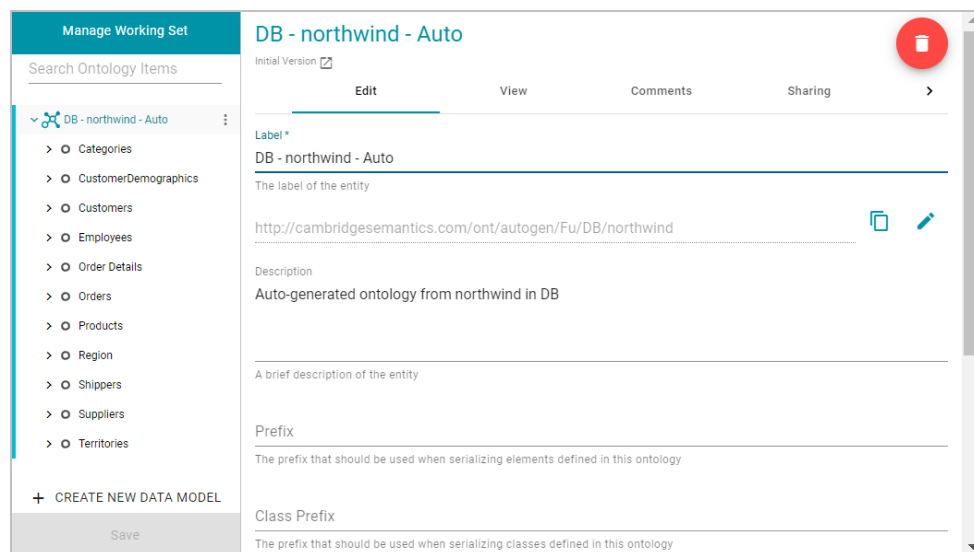
- In the Anzo application, click **Model**. Anzo displays the Manage Data Model Working Set screen. For example:

Manage Data Model Working Set			
Search		Sort By: Title	View: [Grid] [List] [Chart]
<input type="checkbox"/>	Title	Class #	Description
<input type="checkbox"/>	DB - emrdb - Auto	11	Auto-generated ontology from emrdb
<input type="checkbox"/>	DB - northwind - Auto	11	Auto-generated ontology from north
<input type="checkbox"/>	Flights - Auto	1	Auto-generated ontology from Flight
<input type="checkbox"/>	SKOS Vocabulary	4	
<input type="checkbox"/>	Ticket - Auto	7	Auto-generated ontology from Ticket

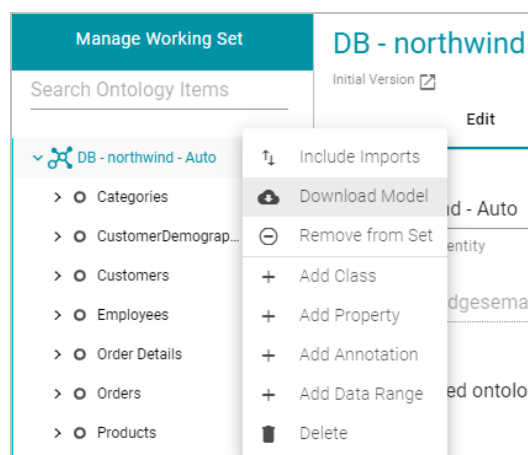
Rows per page: 20 1-5 of 5

UPLOAD MODELS CANCEL OK

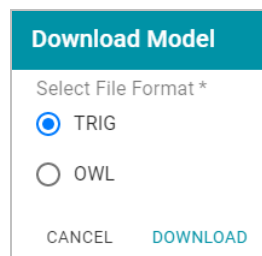
- On the Manage Working Set screen, select the checkbox next to the model that you want to export, and then click **OK**. Anzo opens the selected model in the editor. For example:



- 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:



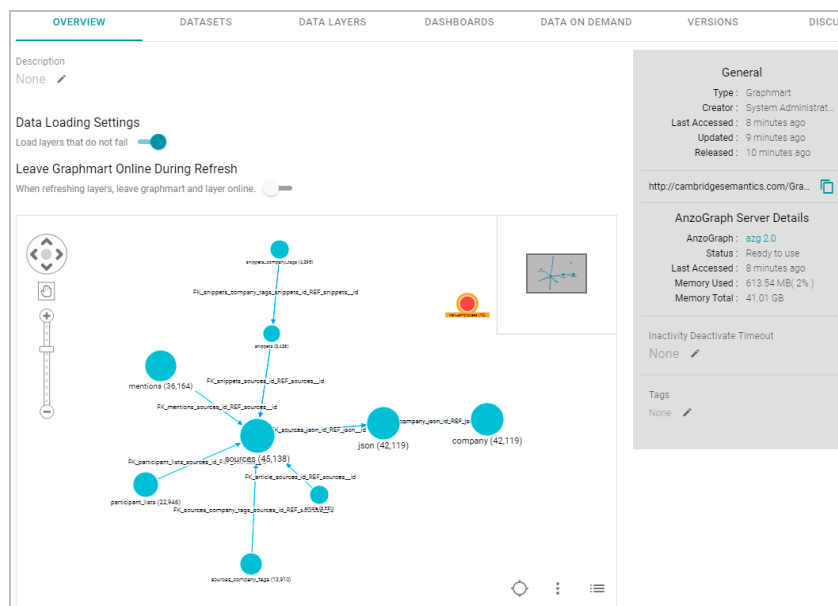
- In the Download Model dialog box, select the format to save the model in. By default Anzo saves models in **TRIG** format. If you want to save the file in OWL format, select the **OWL** radio button. Then click **Download**.

Anzo downloads the model to your computer in the selected format.

How do I see which Models are included in a Graphmart?

Anzo displays Graphmart details, such as a list of the Models in the Graphmart, on the Overview screen for the Graphmart. Follow the steps below to view the list of Models in a Graphmart.

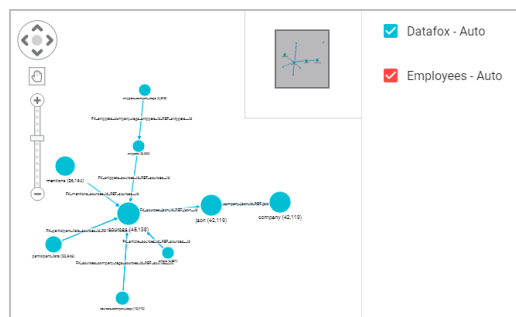
1. In the Anzo application, expand the **Blend** menu and click **Graphmarts**. Anzo displays the Graphmarts screen.
2. In the list of Graphmarts, click the name of the Graphmart for which you want to view the included Models. Anzo displays the Graphmart Overview. For example:



In the bottom right corner of the graph view in the center of the screen, there are three icons:



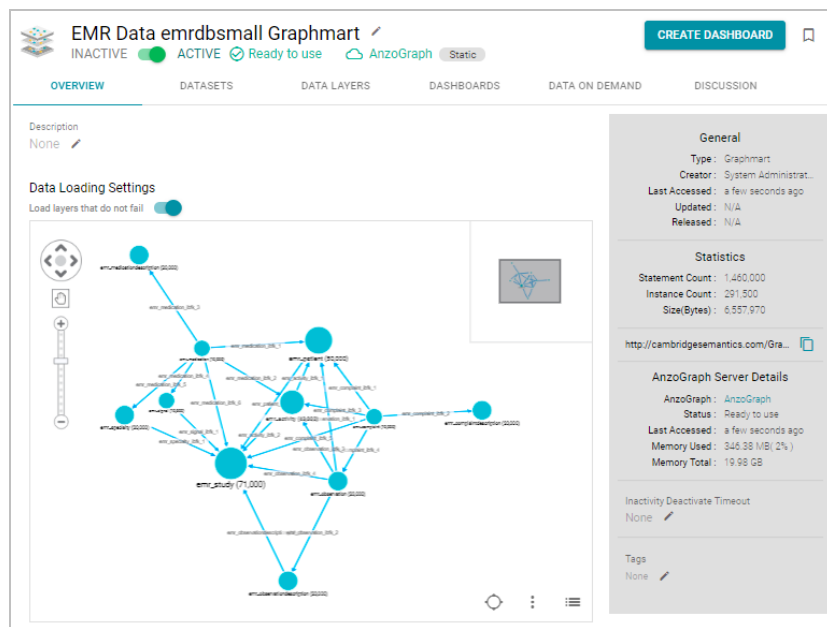
3. To view the associated Models, click the contents icon (☰) on the right. For example, the image below shows a Graphmart with two Models:



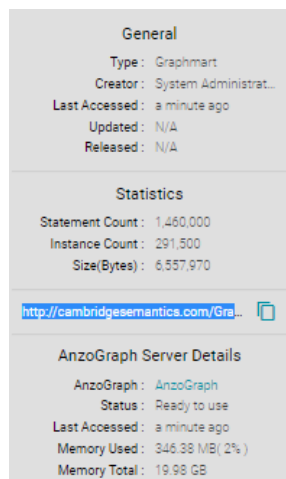
How do I find the URI for a Graphmart?

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:

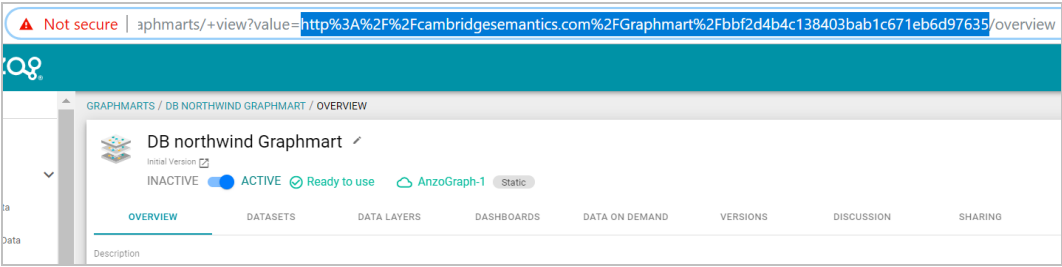


3. View the Graphmart URI in the statistics section on the right side of the screen. For example:



You can click the clipboard icon (📋) to copy the URI to your clipboard.

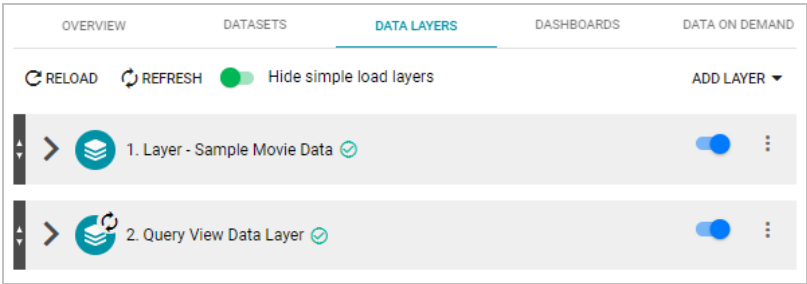
You can also copy a URL-encoded version of the Graphmart URI from the address bar in the browser when viewing the Graphmart Overview. For example:



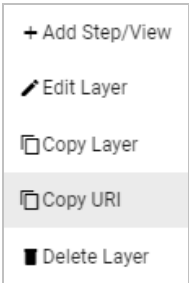
How do I find the graph URI for a Data Layer in a Graphmart?

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**.

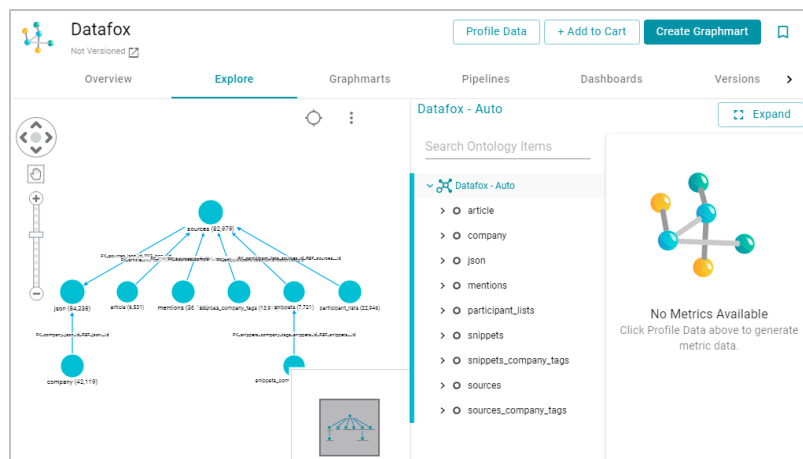


How do I find the URI for a Dataset?

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:

Title	Description	Most Recent Published Date	Actions
Default Edition	Contains the latest successfully...	12/07/2020 09:18AM	

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 Query Builder. At the top, there are two tabs: 'Query' and 'Find', with 'Find' being the active tab. Below the tabs, there is a 'Source' dropdown menu currently set to 'System Datasource'. Underneath, there is a table with four columns: 'Subject', 'Predicate', 'Object', and 'Graph'. At the bottom right of the table, there are three buttons: 'CLEAR', 'ADD STATEMENT', 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. At the top, it says 'Result(14)'. To the right, there is a 'Quick Filter' section with checkboxes for 'Subject', 'Predicate', 'Object', and 'Named Graph'. The 'Subject', 'Predicate', and 'Object' checkboxes are checked. Below this, there is a table with three columns: 'Subject', 'Predicate', and 'Object'. The table contains 14 rows of results, each representing a quad. The first row shows a Subject URI starting with 'http://openanzo.org/datasets#NamedGraphs', a Predicate URI starting with 'http://openanzo.org/ontologies/2008/07/Anzo#namedGraph', and an Object URI starting with 'http://csi.com/FileBasedLinkedDataSet/001e517db4f0eaea9f279427e4e2a828'. The subsequent rows show similar results with different URIs. The last row shows a Subject URI starting with 'http://cambridgesemantics.com/registries/LinkedDataSets', a Predicate URI starting with 'http://openanzo.org/ontologies/2008/07/Anzo#defaultNamedGraph', and an Object URI starting with 'http://csi.com/FileBasedLinkedDataSet/001e517db4f0eaea9f279427e4e2a828'.

Subject	Predicate	Object
<http://openanzo.org/datasets#NamedGraphs>	<http://openanzo.org/ontologies/2008/07/Anzo#namedGraph>	<http://csi.com/FileBasedLinkedDataSet/001e517db4f0eaea9f279427e4e2a828>
<http://openanzo.org/catEntry(%5Bhttp%3A%2F%2Fcsi.com%2FFileBasedLinkedDataSet%2F001e517db4f0eaea9f279427e4e2a828%5D%40%5Bhttp%3A%2F%2Fopenanzo.org%2Fdatasource%2FsystemDatasource%5D)>	<http://cambridgesemantics.com/ontologies/2009/05/LinkedData#linkedDataset>	<http://csi.com/FileBasedLinkedDataSet/001e517db4f0eaea9f279427e4e2a828>
<http://openanzo.org/catEntry(%5Bhttp%3A%2F%2Fcsi.com%2FFileBasedLinkedDataSet%2F001e517db4f0eaea9f279427e4e2a828%5D%40%5Bhttp%3A%2F%2Fopenanzo.org%2Fdatasource%2FsystemDatasource%5D)>	<http://cambridgesemantics.com/ontologies/2009/05/LinkedData#dataset>	<http://csi.com/FileBasedLinkedDataSet/001e517db4f0eaea9f279427e4e2a828>
<http://csi.com/FileBasedLinkedDataSet/001e517db4f0eaea9f279427e4e2a828>	<http://cambridgesemantics.com/ontologies/2009/05/LinkedData#dataset>	<http://csi.com/FileBasedLinkedDataSet/001e517db4f0eaea9f279427e4e2a828>
<http://cambridgesemantics.com/registries/LinkedDataSets>	<http://openanzo.org/ontologies/2008/07/Anzo#defaultNamedGraph>	<http://csi.com/FileBasedLinkedDataSet/001e517db4f0eaea9f279427e4e2a828>

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:

Semantic Service Details

Overview Operations **Service Builder**

clearWorkingEdition

http://cambridgesemantics.com/semanticServices/LinkedData#clearWorkingEdition

Request Statements

```

1 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
2 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
3 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
4 @prefix dc: <http://purl.org/dc/elements/1.1/> .
5 @prefix dcterms: <http://purl.org/dc/terms/> .
6 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
7 @prefix owl: <http://www.w3.org/2002/07/owl#> .
8 @prefix system: <http://openanzo.org/ontologies/2008/07/System#> .
9 @prefix anzo: <http://openanzo.org/ontologies/2008/07/Anzo#> .
10 @prefix ld: <http://cambridgesemantics.com/ontologies/2009/05/LinkedData#> .
11 @prefix graphmart: <http://cambridgesemantics.com/ontologies/Graphmarts#> .
12
13 <http://serviceRequesta9e72cca-a79b-42f6-94a7-1ef9dfb1e65c> {
14   <http://serviceRequesta9e72cca-a79b-42f6-94a7-1ef9dfb1e65c> a ld:ClearWorkingEditionRequest ;
15   ld:fldsToClear <temp://value_to_fill_in_1> ;
16   ld:typesToClear <temp://value_to_fill_in_0> .
17
18   <temp://value_to_fill_in_1> a ld:FileBasedLinkedDataSet .
19 }
20

```

Run Service

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/ee8d3d5792fd218a03b70fdf850b6a4c> ;
    ld:typesToClear <temp://value_to_fill_in_0> .

  <http://csi.com/FileBasedLinkedDataSet/ee8d3d5792fd218a03b70fdf850b6a4c> 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/ee8d3d5792fd218a03b70fdf850b6a4c> ;
    # ld:typesToClear <temp://value_to_fill_in_0> .

  <http://csi.com/FileBasedLinkedDataSet/ee8d3d5792fd218a03b70fdf850b6a4c> 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#> .
```

```

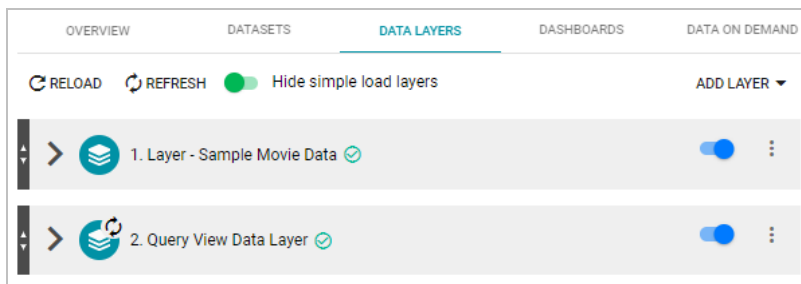
n-1060687345:effbda5b-ce9c-4190-abb1-6f6efd07f5d8 {
  n-1060687345:effbda5b-ce9c-4190-abb1-6f6efd07f5d8 ld:wasWorkingEditionCleared
    "true"^^<http://www.w3.org/2001/XMLSchema#boolean> .
  n-1060687345:effbda5b-ce9c-4190-abb1-6f6efd07f5d8 rdf:type
ld:ClearWorkingEditionResponse .
  n-1060687345:effbda5b-ce9c-4190-abb1-6f6efd07f5d8 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](#).