

Altair Panopticon™ v2025.1

**STREAMS SERVER INSTALLATION AND
REFERENCE GUIDE**

TABLE OF CONTENTS

[1] INTRODUCTION	1
Acronyms	1
Terminology	1
Overview	1
Panopticon Streams Applications	2
Panopticon Streams Operators	3
Panopticon Streams Inputs	3
Panopticon Streams Outputs	4
System Requirements	4
System Hardware Requirements	5
Development / Test	5
Small Scale Deployment	5
Medium Scale Deployment	5
Large Scale Deployment	5
[2] GETTING STARTED	6
Setting Up Confluent Kafka Enterprise on a Local Machine Using Docker	6
Setting Up ZooKeeper, Kafka, and Schema Registry	7
Additional Notes on Setting Up the Schema Registry	8
Setting Up Panopticon Streams	9
Importing the Bundle of Example Applications	13
Configuration of the Client Properties	13
Configuration of The Client URL Query Parameters	16
Licensing	16
Using Altair Units License in Altair's License Server	17
Using Managed Altair Units License via Altair One	18
Migration to Streams Server 2025.1 from an Older version	21
1. Copy All Content	21
2. Delete Old Content	22
3. One-time Conversion	22
4. Applications, Data Sources, and Data files	22
5. Do Not Make Changes on Both Servers	23
6. Post-migration Cleanup	23
Upgrade	23
[3] AUTHENTICATION	24
Introduction	24
Mapping Users to Roles	24
Token	26
Tomcat Realm	26

Tomcat User Base	27
Tomcat Memory Configuration for Windows.....	28
Tomcat Memory Configuration for Linux.....	28
Encrypting Passwords in tomcat-users.xml.....	29
LDAP	30
Active Directory.....	31
SAML	32
OAuth 2.0.....	33
Example.....	33
Filter.....	34
Creating a Custom Filter	34
Header	36
[4] PCLI: COMMAND UTILITIES FOR PANOPTICON	37
Export Data Sources	37
Parameters	37
Example 1: Export data sources from a workbook	37
Example 2: Export data sources from all workbooks example	37
[5] USING ALTAIR PANOPTICON STREAMS	38
Connecting to or Disconnecting from the CEP Engine.....	38
Connecting to the CEP Engine	40
Disconnecting from the CEP Engine.....	40
[6] MANAGING THE STREAMS SYSTEM	41
Viewing and Managing Kafka Properties	43
Reloading Configurations	44
Logging/Monitoring.....	44
View Logs	44
Setting the Logging Level	47
Setting the Server Metrics Publisher.....	47
Scheduling Task to Clear Topic Data.....	48
Modify a Scheduled Task.....	50
[7] AUTHORIZATION	51
Secure Access	51
Creating Folders	51
Creating Folders on the Applications Tab.....	52
Creating Folders on the Data Sources Tab	53
Adding Groups and Users with Allowed Authorization.....	55
Adding Groups and Users with Denied Access	58
Creating Subfolders	60

Updating Folder or Subfolder Properties.....	63
[8] MANAGING APPLICATIONS	65
Uploading Applications.....	66
Folders and Applications Display View	70
Importing an Application Bundle.....	72
Exporting an Application or Folder Bundle	76
Sorting the List of Applications	77
Searching for Applications	78
Renaming Applications or Folders	79
Viewing Application History and Republishing	80
Viewing and Managing Application Topic Usages.....	82
Clearing the Data In an Application Topic.....	83
Clearing the Schema in an Application Topic	84
Sorting Application Topics	84
Viewing the Application Data Sources Usage	85
Moving Applications	86
Copying Applications.....	87
Downloading an Application.....	88
Deleting an Application	89
Deleting Applications Using the Toolbar	89
Deleting Applications Using the Context Menu	89
Creating a New Application.....	90
Adding an Operator.....	92
Adding an Input Operator.....	93
Example	96
Adding An Aggregation Operator.....	96
Example	101
Building the Expression	101
Using the Expression Builder	102
Supported Aggregation Functions.....	104
Converting Timestamp to/from Integer	105
Adding a Branch Operator	106
Example	107
Example 2	108
Adding a Calculation Operator	108
Supported Operators and Calculation Functions	112
Supported Operators.....	113
Supported Calculation Functions.....	114
Example	121
Adding a Conflate Operator	121
Example	123
Adding an External Input.....	123
Adding a Filter Operator.....	127
Example	129
Adding a Join Operator	129
Example	132

Fixing Broken Joins	132
Adding a Metronome Input Operator	137
Example	139
Adding a Python Transform Operator	139
Example	143
Additional Best Practice Recommendations in Using Python with Panopticon.....	143
Adding a Rank Operator	144
Example	148
Adding a Rekey Operator	148
Example	150
Adding a REST Transform Operator.....	150
Adding an R Transform Operator.....	155
Additional Best Practice Recommendations in Using R with Panopticon	158
Adding a Scatter Operator	158
Example	160
Adding a Table to Stream Operator	160
Example	162
Adding a Union Operator	163
Adding an Output Operator.....	164
Adding Application-specific Properties	166
Example	167
Saving an Application.....	168
Editing an Application.....	169
Validating and Fixing Application Issues	170
Starting an Application	171
Starting an Application on the Applications Tab.....	172
Starting an Application on the Application Page	174
Stopping an Application	178
Stopping an Application on the Applications Tab.....	178
Stopping an Application on the Application Page	179
[9] MANAGING DATA SOURCES	181
Uploading Data Sources	182
Folders and Data Sources Display View.....	185
Sorting the List of Data Sources.....	187
Searching for Data Sources	188
Renaming Data Sources or Folders	189
Viewing Application Usages	190
Moving Data Sources.....	192
Copying Data Sources	193
Downloading a Data Source	195
Deleting Data SourceS.....	195
Creating a Data Source.....	196
Common Data Source Settings	200
Selecting and Defining the Data Connector File Source.....	200
Defining the Message Type in Data Sources	206
Defining the Format in Data Sources.....	208

Saving or Loading Column Definitions in the Data Sources	210
Defining Real-time Settings	212
Using the Data Source Toolbar	213
Date/Time Key Elements	214
Creating Email Output Connector	216
Creating InfluxDB 1.x Output Connector.....	217
Creating JDBC Legacy Output Connector	218
Creating Apache Kafka Output Connector.....	219
Creating Kx kdb+ Output Connector	220
Creating a MQTT Output Connector.....	221
Creating a REST Output Connector	222
Creating Text Output Connector	225
Creating ActiveMQ Input Data Source.....	226
Creating Altair AI Hub Input Data Source	229
Creating Azure Input Data Source	230
Creating AMPS Input Data Source	232
Creating Cassandra Input Data Source	235
Creating DolphinDB Input Data Source	237
Creating DolphinDB – Streaming Input Data Source	237
Creating Google Analytics Input Data Source.....	239
Creating Google Cloud Input Data Source.....	241
Creating Google Cloud Pub/Sub Input Data Source	243
Creating an InfluxDB 1.x Input Data Source	247
Creating JDBC Legacy Input Data Source.....	248
Creating JDBC Input Data Source	252
Creating a JSON Input Data Source	257
Creating Apache Kafka Input Data Source	259
Creating Kx kdb+ Input Data Source	264
Kx kdb+ - Deferred Sync Query.....	268
Creating Kx kdb+Tick Input Data Source	268
Creating ksqlDB Input Data Source	272
Creating ksqlDB - Streaming Input Data Source.....	274
Creating MongoDB Input Data Source.....	275
Row-Wise Array Expansion.....	279
Column-Wise Array Expansion.....	280
Bson-Wise Array Expansion.....	280
Creating MQTT Input Data Source	280
Creating MQTT Publisher Input Data Source	283
Creating MS Excel Input Data Source	284
Creating OneTick Input Data Source	286
Creating OneTick CEP Input Data Source.....	288
Creating Python Input Data Source	290
Creating RabbitMQ Input Data Source	291
Creating Amazon S3 Input Data Source.....	294
Creating Rserve Input Data Source	295
Creating Solace Input Data Source	297
Creating Stream Simulator Input Data Source.....	299
Creating StreamBase Input Data Source.....	303

Creating StreamBase LiveView Input Data Source	304
Creating Text Input Data Source	306
Creating Text Entry Input Data Source	309
Creating WebSocket Input Data Source	310
Creating Web Data Input Data Source.....	312
Creating XML Input Data Source	318
Modifying Data Sources	319
 [10] MANAGING DATA PRODUCERS	321
Refresh Data Producers	322
Starting or Stopping Data Producers	322
 [11] MONITORING ENGINE METRICS AND APPLICATION TOPICS	323
Managing Topics.....	325
Filter Topics	326
Sorting the List of Topics	326
Moving to Other Topics List Pages	326
 [12] MANAGING PARAMETERS.....	327
Adding Parameters	327
Modifying Parameters.....	328
Deleting Parameters	329
Refresh Parameters.....	329
Sorting the List of Parameters	329
 [13] EXAMPLE APPLICATIONS.....	330
 [14] PANOPTICON RESOURCES	332
 [APPENDIX]	333
Properties: Streams	333

[1] INTRODUCTION

Fundamental to understanding Panopticon Streams are these acronyms and terminologies:

ACRONYMS

Component	Description
CEP	Complex Event Processing
PCLI	Panopticon Command-line Interface

TERMINOLOGY

Component	Description
Apache Kafka or Kafka	Used for building the real-time data pipelines and streaming applications. It is horizontally scalable, fault-tolerant, fast and runs in production in thousands of companies.
Apache ZooKeeper or ZooKeeper	A centralized service for maintaining configuration information, naming, providing both distributed synchronization and group services.
Confluent	The free, open-source streaming platform based on Apache Kafka. The Confluent Platform is the complete streaming platform for large-scale distributed environments. Unlike a traditional messaging system or streaming processing API, Confluent Enterprise enables your interfaces to be connected to anywhere in the world and help make decisions with all your internal systems in real-time.
Schema registry	Part of the Confluent distribution package. Stores a versioned history of all schemas and allows the evolution of schemas according to the configured compatibility settings. Also provides a plug-in to clients that handles schema storage and retrieval for messages that are sent in Avro format.
Panopticon Streams	The name of the Panopticon CEP platform.

OVERVIEW

Event processing is a method of tracking and analyzing streams of information of an event, and eventually deriving a conclusion from what transpired. CEP is an event processing method which combines data from multiple sources to infer events or patterns that may demonstrate unusual activities or anomalies, consequently requiring immediate action.

The CEP engine provided by Panopticon is named **Panopticon Streams** and it is built to work with different CEP engines. However, for this version, it will only support Kafka.

Kafka is a distributed streaming platform that lets you publish and subscribe to streams of records. Each record consists of a **key**, a **value**, and a **timestamp** and stores streams of records in categories called **topics**. Kafka is mainly used for two reasons:

- Building real-time streaming data pipelines that reliably get data between systems or applications
- Building real-time streaming applications that transform or react to the streams of the data

Refer to <https://kafka.apache.org/intro.html> for more information.

Panopticon Streams enables you to create streaming data pipelines which both transforms and reacts to streaming data. Aside from Kafka, it is also using ZooKeeper and Schema Registry that are provided by Confluent. ZooKeeper is a key component when using Kafka since it allows the configuration and management of clusters in the Kafka servers. The Schema Registry stores a versioned history of all schemas used by Kafka and provides a RESTful interface for storing and retrieving Avro schemas.

Panopticon Streams Applications

The main task of Panopticon Streams is to execute and manage streams **applications**. An application describes how data should be piped, transformed, and processed. Applications consist of a set of **inputs**, **operators**, and **outputs** and are described or constructed in an XML file.

It can be viewed as a directed graph with a set of nodes (or operators) and a set of edges (or streams) that are interconnected with each other.

Component	Description
ID	The ID of the application config. It should be the same with the filename when loading an application config from the system.
operators	A list of operators (actions and functions).
Streams	A list of streams that describe the connection and the flow between operators.
properties	Application-specific defined properties.

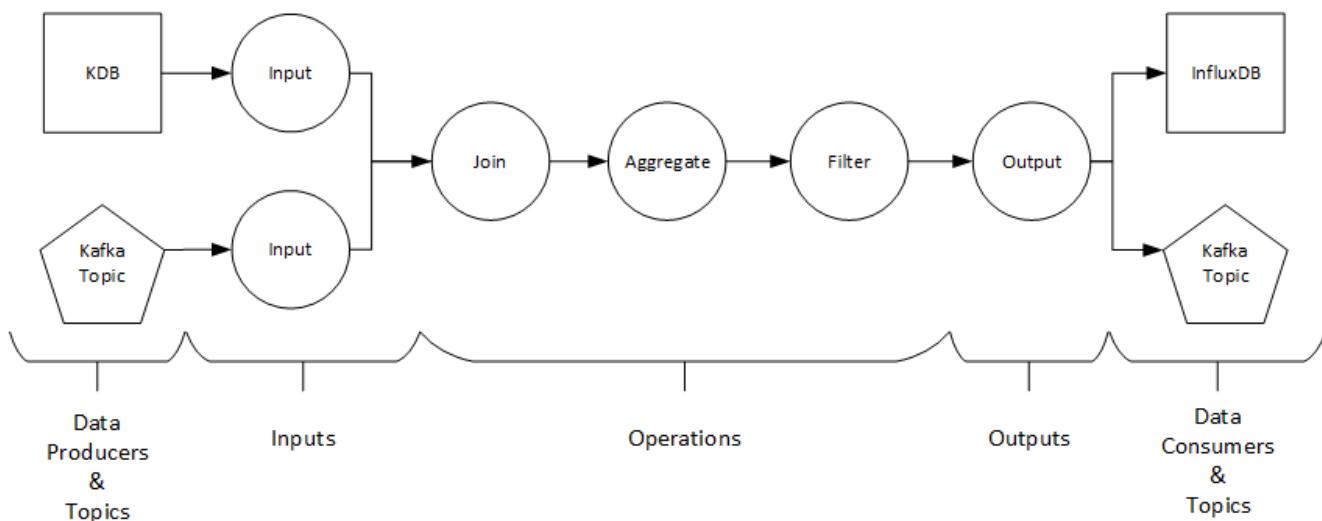


Figure 1-1. Panopticon Streams Framework

An application can either use **Kafka topics** or **data producers**, which generate data from a data source. The data producer also demonstrates to be the connection between the Panopticon Streams framework and the Panopticon core.

The Panopticon core has data connectors such as Kx kdb+, OneTick, and MS Excel that serve as data sources. Just like the application, the data source is also constructed or described in an XML file.

An application refers to a data source through its ID (or filename). There are several ways to create a data source of an application:

- Export data source with the [PCLI tool](#)

The PCLI tool extracts the already defined data sources in workbooks and saves them as CEP data sources.

- [Using Panopticon Streams](#)

Panopticon Streams Operators

An **operator** is a single task responsible for processing the data and publishing it as an output. Currently, Panopticon Streams supports the following operators:

- [Aggregation](#)
- [Branch](#)
- [Calculation](#)
- [Conflate](#)
- [External Input](#)
- [Filter](#)
- [Input](#)
- [Join](#)
- [Metronome](#)
- [Rank](#)
- [Rekey](#)
- [Scatter](#)
- [To_stream](#)
- [Output](#)
- [Union](#)
- [Python Transform](#)
- [REST Transform](#)
- [R Transform](#)

Each operator produces one or more output streams that can be connected and defined as input streams for other operators.

Panopticon Streams Inputs

Panopticon Streams engine allows the combination of multiple data sources and their definition as input channels. The data sources are referred to within Panopticon Streams as **inputs**. The data produced by each input can be processed by one or more operators.

Panopticon Streams Outputs

An **output** produces and publishes streams towards a Kafka topic or a **data consumer**. A data consumer is the opposite of a data producer. It consumes the data produced from an output in Panopticon Streams and publishes the data to a data source.

The most common approach is to publish the data to a Kafka topic which eventually can be consumed or used by Panopticon Real Time or other platforms that support Kafka.

Currently, Panopticon Streams supports publishing of the output data to the following data sources:

- [Email](#)
- [InfluxDB 1.x](#)
- [JDBC Legacy](#)
- [Apache Kafka](#)
- [Kx kdb+](#)
- [Rest](#)
- [Text](#)

SYSTEM REQUIREMENTS

Panopticon Real Time is supported by these operating systems:

Linux which includes the following distributions and versions:

- Red Hat Linux (RHEL) >= 9.4
- Debian >= 11
- Ubuntu >= 22.04 LTS
- Fedora >= 41

Windows operating systems – For Evaluation, Development, and Testing Environments Only

- Windows 10 or higher (64-bit)
- Windows Server 2012 or higher (64-bit)

Panopticon Streams Server also requires:

- Oracle Java SE 17 and Open JDK 17
- Apache Tomcat 10

Panopticon Streams Server is supported for deployment on the following cloud providers:

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform
- Oracle Cloud

Supported browsers include the latest version of:

- Google Chrome

- Safari

System Hardware Requirements

Development / Test

- 1 x Dual Core CPU (Hyper Threaded to 4 Cores/Threads)
- 8GB RAM
- 4GB Disk (Available)
- In Memory Caching limited to available Server RAM

Small Scale Deployment

- 1 x Quad Core CPU Or Equivalent (Hyper Threaded to 8 Cores/Threads)
- 16GB RAM
- 4GB Disk (Available)
- In Memory Caching limited to available Server RAM

Medium Scale Deployment

- 4 x Quad Core CPU Or Equivalent (Hyper Threaded to 32 Cores/Threads)
- 32GB RAM
- 4GB Disk (Available)
- In Memory Caching limited to available Server RAM

Large Scale Deployment

- 8 x Quad Core CPU Or Equivalent (Hyper Threaded to 64 Cores/Threads)
- 64GB RAM
- 4GB Disk (Available)
- In Memory Caching limited to available Server RAM

[2] GETTING STARTED

Running Panopticon Streams can either be done with:

- a [Dockerized Kafka](#) (Confluent Kafka Enterprise platform)
- a local cluster that includes [Zookeeper, Kafka, and Schema Registry](#)

Follow the steps below corresponding to the platform you are using.

SETTING UP CONFLUENT KAFKA ENTERPRISE ON A LOCAL MACHINE USING DOCKER

Steps:

1. Install Docker.

If you install on a Windows machine, you can use Docker Desktop for Windows, see:

<https://docs.docker.com/desktop/windows/install/>

NOTE

If you run Docker on Windows, we recommend that you use the WSL 2 backend. If you choose to use Hyper-V instead, make sure that you select Linux containers, and increase the available memory to at least 8 GB.

2. Download the Docker Compose script from Confluent:

<https://github.com/confluentinc/cp-all-in-one/raw/v6.0.1/cp-all-in-one/docker-compose.yml>

Save this in a convenient location, such as next to the Panopticon Streams directory.

3. Start the Confluent platform from a command prompt in the same directory as the YML file:

`docker compose up -d`

This script defines a number of services that together make up the full Confluent platform. Panopticon Streams really only needs zookeeper, broker, and schema-registry.

4. To verify that everything is working, run `docker compose ps`, and make sure that all services are listed with status **running**.

Once the Confluent services are up, start Tomcat and Panopticon Streams to build, deploy, and execute your applications.

SETTING UP ZOOKEEPER, KAFKA, AND SCHEMA REGISTRY

NOTE

Windows is currently not a supported platform for running Confluent Kafka, ensure that your OS is on the list of supported operating systems:
https://docs.confluent.io/4.0.0/installation/installing_cp.html#system-requirements

Before proceeding, you must install and setup the following prerequisites:

- Java JDK 64-bit, version 1.7 or later
- System Environment variable JAVA_HOME set to the Java JDK 64-bit

Steps:

1. Download one of the Confluent Kafka archives from <http://confluent.io/download>.
2. Extract the contents of the archive to a new location.
3. Below are the top-level folders of the archive:

```
confluent-3.1.1/bin/          # Driver scripts for starting/stopping services
confluent-3.1.1/etc/           # Configuration files
confluent-3.1.1/share/java/   # Jars
```
4. Start the ZooKeeper, Kafka and Schema Registry processes in the correct order. Make sure the previous process has been started before continuing to the next one.
 - Start ZooKeeper

```
$ ./bin/zookeeper-server-start ./etc/kafka/zookeeper.properties
```
 - Start the Kafka broker

```
$ ./bin/kafka-server-start ./etc/kafka/server.properties
```
 - Start Schema Registry

```
$ ./bin/schema-registry-start ./etc/schema-registry/schema-registry.properties
```

When these three processes have been started, you can now connect Panopticon Streams to your local Kafka cluster to execute and deploy your applications.

For more details, refer to the Confluent Kafka Installation-and Quick Start guides:

- <https://docs.confluent.io/3.1.1/installation.html>
- <https://docs.confluent.io/3.1.1/quickstart.html>

NOTE

When connecting to a Kafka broker on a separate machine, exposing different IP addresses internally and externally, you need to configure `KAFKA_ADVERTISED_LISTENERS`. This is typically the case when running Kafka in a Docker container.

The symptoms of the missing configuration are:

- Panopticon Streams can connect to ZooKeeper and the Kafka Broker
- No data is written to topics

In `[Kafka]/etc/kafka/server.properties`, uncomment `advertised.listeners` and replace “`your.host.name`” with the externally exposed host name or IP address.

```
# Hostname and port the broker will advertise to producers and
# consumers. If not set,
# it uses the value for "listeners" if configured. Otherwise,
# it will use the value
# returned from java.net.InetAddress.getCanonicalHostName().
advertised.listeners=PLAINTEXT://your.host.name:9092
```

When using the Confluent Docker image, you can pass the `KAFKA_ADVERTISED_LISTENERS` as a parameter:

```
docker run -d --restart=always \
--net=confluent \
--name=kafka \
-p 9092:9092 \
-e KAFKA_ZOOKEEPER_CONNECT=zookeeper:2181 \
-e KAFKA_ADVERTISED_LISTENERS=PLAINTEXT://your.host.name:9092 \
-e KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR=1 \
confluentinc/cp-kafka:5.1.0
```

Additional Notes on Setting Up the Schema Registry

It is recommended to turn off the compatibility checking in schema registry when used with Panopticon Streams.

To do this, set the Avro compatibility level to **NONE** (as mentioned below) in the `schema-registry.properties` file.

Then there are three cases depending on how Kafka is deployed:

- ❑ On Windows from the ZIP file from Panopticon. Already turned off by default.
- ❑ On Linux manually deployed (“bare metal”). Add the following line to `...etc/schema-registry/schema-registry.properties`

```
Avro.compatibility.level=NONE
```

- ❑ With Docker Compose using the Confluent images

Add the following line to the environment section of the schema-registry service in `docker-compose.yml`

```
SCHEMA_REGISTRY_AVRO_COMPATIBILITY_LEVEL: 'NONE'
```

SETTING UP PANOPTICON STREAMS

Follow the steps and guidelines below to install Panopticon Streams on Windows.

NOTE

If you need to upgrade your previously installed Panopticon Streams, proceed to the [Upgrade](#) section.

Steps:

1. Extract the contents of PanopticonStreamsWAR_<version>.zip file to a new location.

This zip file will contain the following files and folder:

- streams.war
- streams.xml
- [Examples.apz](#)
- CustomMessageParserExample.zip
- CustomTransformExample.zip
- [OpenJDK11Dependencies.zip](#)
- [OpenJDK11Dependencies_README.txt](#)
- User_License.rtf
- Panopticon Streams Reference Guide
- examples folder with sample data files and CEP applications and data sources

2. Create the AppData folder (i.e., **streamsserverdata**) and ensure that the user account **Local Service** running Tomcat has read/write and execute permissions to this folder.

Example: c:\streamsserverdata

3. Copy the extracted streams.xml file into the Tomcat config folder (\Apache Software Foundation\Tomcat 10.0\conf\Catalina\localhost). This file contains the following information:

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
    <Environment name="PanopticonAppData" override="false"
type="java.lang.String" value="c:\streamsserverdata" />
</Context>
```

NOTE

- Instead of setting the path of the environment variable PanopticonAppData on the streams.xml file, you can do so on the System Environment Variables. For example:

Variable	New Value
PanopticonAppData	c:\panopticonstreamsdata

- If the directory path is set in both an environment variable and in the streams.xml file, the value set in the XML file will take precedence.

- Copy the streams.war file into the Tomcat webapps folder (\Apache Software Foundation\Tomcat 10.0\webapps).
- Edit the existing tomcat-users.xml file which is available in the Tomcat config folder (\Apache Software Foundation\Tomcat 10.0\conf) and add the entry:

```
<role rolename="user"/>
<role rolename="designer"/>
<role rolename="admin"/>
<user username="viewer" password="viewer" roles="user" />
<user username="designer" password="designer" roles="designer" />
<user username="admin" password="admin" roles="admin" />
```

For more complex authentication and user directory options, see section [\[3\] Authentication](#).

IMPORTANT

- Before proceeding to step 7, ensure the Tomcat temp folder (e.g., (\Apache Software Foundation\Tomcat 10.0\temp) is available.
- You can opt to choose a different temp folder with the CATALINA_TMPDIR environment variable. For example:

Variable	Value
CATALINA_TMPDIR	C:\tomcat\dev\temp

- Start Tomcat to deploy the .war file.

The streams folder is extracted in the Tomcat webapps folder:

This PC > OS (C) > Program Files > Apache Software Foundation > Tomcat 10.0 > webapps			
Name	Date modified	Type	Size
docs	14/02/2024 1:28 pm	File folder	
examples	14/02/2024 1:28 pm	File folder	
host-manager	14/02/2024 1:28 pm	File folder	
manager	14/02/2024 1:28 pm	File folder	
ROOT	14/02/2024 1:28 pm	File folder	
streams.war	14/02/2024 9:27 am	WAR File	199,286 KB

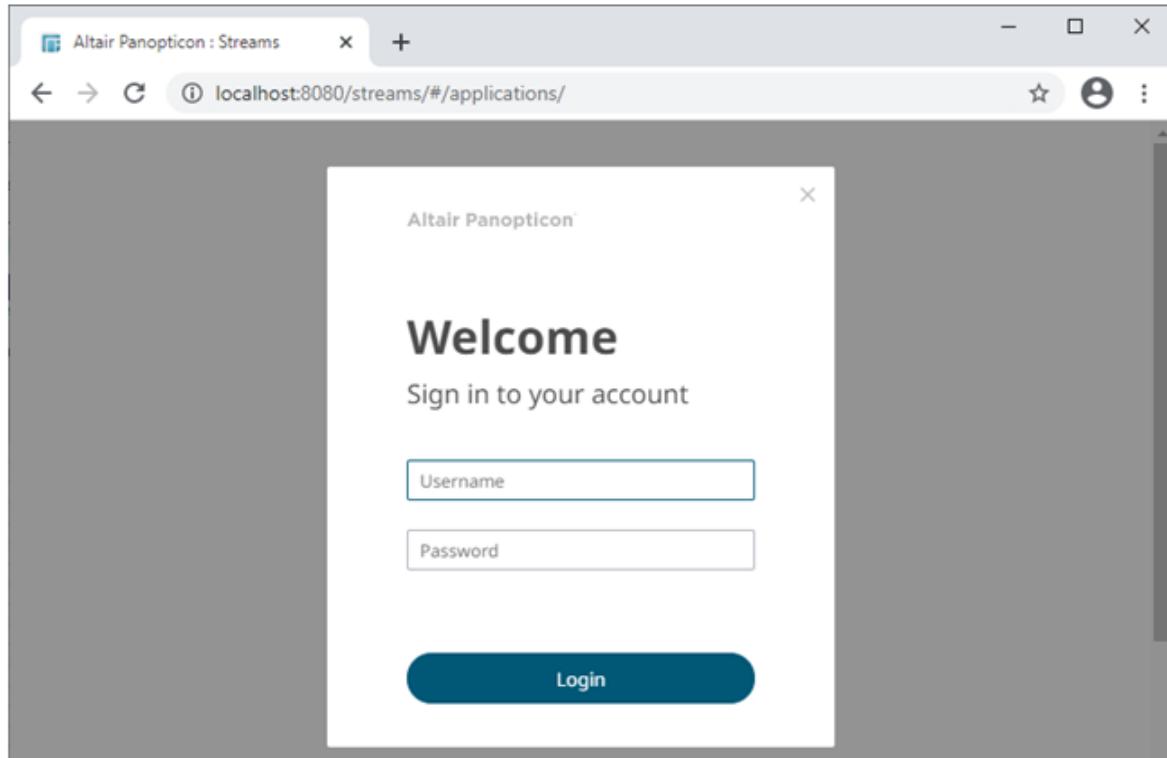
The server initializes the AppData directory with an empty content repository and empty subdirectories for other types of data. The [Streams.properties](#) file is created with the default server properties.

- Specify the [license type](#) that will be used. Use any of the following license types:
 - Volume License file ([PanopticonLicense.xml](#)) that must be copied to the designated AppData folder.
 - Altair Units license. Refer to [Using Altair Units License in Altair's License Server](#) for more information.
 - Managed Altair Units license. Refer to [Using Managed Altair Units via Altair One](#) for more information.
- You should now be able to log on to Panopticon Streams using the following URL:

[Host Name] : [Port] / [Name of your application]

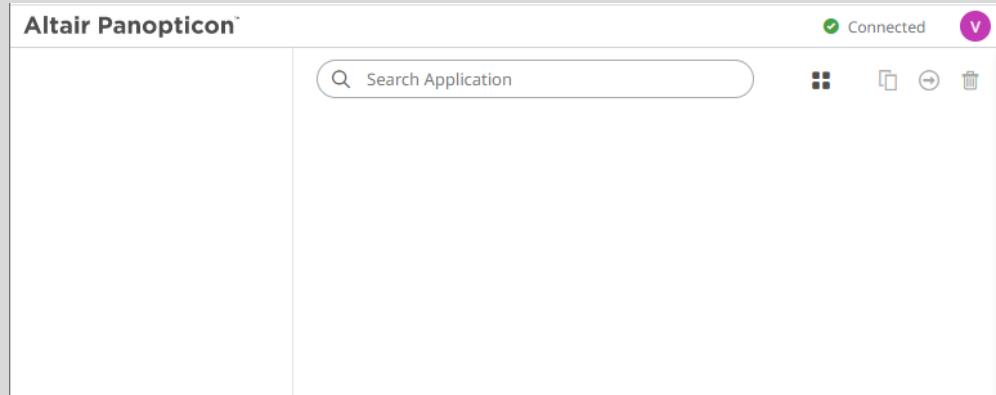
For example:

<http://localhost:8080/streams>

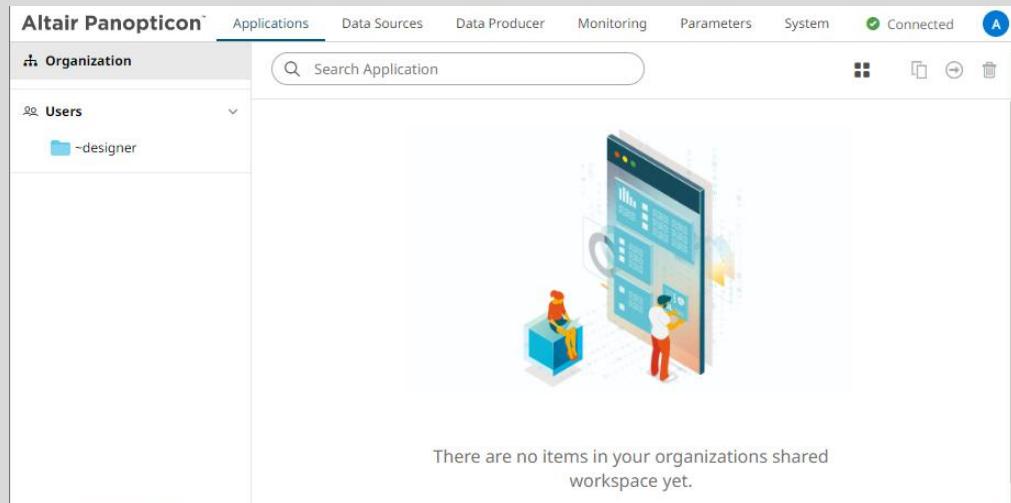


NOTE

Panopticon Streams Server [supports different user roles](#). By default, all users are assigned the VIEWER role. For example, logging on using the viewer user added in step 6, Panopticon Streams Server will only display:



To have full access to all the services, the user is required to have an ADMINISTRATOR role.



A user with a DESIGNER role can create or upload applications and data sources:

For more information on how to set up the user groups and map them to the user roles, refer to [Mapping Users to Roles](#) for more information.

Importing the Bundle of Example Applications

The AltairPanopticonStreamsWAR_<version number>.zip file includes the bundle file (Examples.apz) of the example applications and their associated data sources and data files.

Follow the instructions in [Importing an Application Bundle](#) to import this bundle to Panopticon Streams Server.

Configuration of the Client Properties

Starting with version 2020.1, Panopticon Streams Server generates a streams.json configuration file in the JavaScriptConfiguration directory of the AppData folder (i.e., **c:\streamsserverdata**).

This PC > Windows (C:) > streamsserverdata > JavaScriptConfiguration			
<input type="checkbox"/> Name	Date modified	Type	Size
streams.json	10/11/2020 9:36 PM		1 KB

The default content of the streams.json file has the following objects/names:

```
{
  "baseUrl" : ".",
  "hideAuthenticationButton" : false,
}
```

NOTE

In the JSON files, a dot in the name (e.g., name1.name2) is used to denote a nested object structure:

```
{  
  "name1": {  
    "name2": ...  
  }  
}
```

In the `streams.json` file, you can control the configuration of the following objects/names:

Object/Name	
Object/Name	baseUrl
Description	Location of Panopticon Streams Server.
Default Value	<code>."</code>
Required	Yes
Object/Name	
Description	If set to true , then the real time connection (WebSocket or long polling) to the Panopticon server will be automatically reconnected if it is disconnected.
Default Value	false
Required	No
Object/Name	
Description	Controls the which transport should be used when viewing log from the server. Valid values are "websocket" and "long-polling" . If configured to "websocket" , but the WebSocket connection fails, then the web client will automatically fall back to "long-polling" .
Default Value	"websocket"
Required	No
Object/Name	
Description	Maximum length of text that will be attempted to be put into the system clipboard (copy). If too much text is attempted, then the browser might become unresponsive.
Default Value	500000
Required	No
Object/Name	
Description	Boolean. Hides the login and logout buttons.
Default Value	false
Required	No
Object/Name	

Description	Boolean. If set to true , then the <code>browser navigator.language</code> , <code>navigator.userLanguage</code> on IE11, controls the localization of the UI. Not all languages are supported.
Default Value	true
Required	No
Object/Name	<code>localization.defaultLocale</code>
Description	Locale used if the browser locale is not supported, or if <code>useBrowserLocale</code> is set to false .
Default Value	"en-US"
Required	No
Object/Name	<code>localization.fallbackLocale</code>
Description	Locale used if a resource string is missing from the locale in use. Should be specified if <code>localization.defaultLocale</code> is specified.
Default Value	value of <code>localization.defaultLocale</code>
Required	No
Object/Name	<code>localizationOverride</code>
Description	Nested object with resource strings per language. Used to customize resource strings.
Default Value	
Required	No
Object/Name	<code>logLevel</code>
Description	Controls which types of logs Panopticon will write to the browser dev console. Valid values are: "trace" , "debug" , "info" , "warn" , "error" and "silent" .
Default Value	"info"
Required	No
Object/Name	<code>showFileUploadUI</code>
Description	Controls the visibility of UI that is used to upload a file to the server.
Default Value	true
Required	No
Object/Name	<code>showLinkToFileUI</code>
Description	Controls the visibility of UI that is used to specify a data file on the server local file system as a data source. Per default, this flag is automatically inserted into the client configuration at runtime.
Default Value	true
Required	No

NOTE

If there are no config files available on the server, default ones will be created and saved. After that, you can alter them in any way you would like and keep the configuration even if the server is restarted.

CONFIGURATION OF THE CLIENT URL QUERY PARAMETERS

This section discusses the list of URL query parameters supported by the streams application.

NOTE

- These parameters are not required.
- See [What is URL?](#) and [Query String](#) for additional information on what the query component of an URL is.

Parameter	Description
globalConfig	JSON object that provides overrides to client configuration provided in <code>streams.json</code> . Only the first level properties may be overridden.
debug	Override for assigning the client configuration <code>logLevel</code> in <code>streams.json</code> .

LICENSING

NOTE

In the Panopticon documentation, HyperWorks Units (HWU) and Hosted HyperWorks Units (HHWU) are now named Altair Units.

In the Panopticon product, these license types are still named HyperWorks Units and Hosted HyperWorks Units.

For more information on Altair Units, visit <https://www.altair.com/altair-units/>.

Licensing within Panopticon Streams supports three license types:

- ❑ a volume-based XML file (named **PanopticonLicense.xml**), that is used to store all license information for a specific customer, must be copied to the designated AppData folder (i.e., **c:\streamsserverdata**)
- ❑ [Altair Units license](#) which is available in the Altair's License Server you are connected to (local or over the network)
- ❑ [Managed Altair Units license via Altair One](#)

The license file type you will use is delivered separately from the installation packages.

Using Altair Units License in Altair's License Server

If your license source is Altair's License server, it is required to configure certain properties in the Streams.properties file located in the AppData folder or c:\streamsserverdata:

Property	Service authentication level
Attribute	authentication.required
Description	The property that will make the authentication required. It will force the user to login in order to use any of the services provided by the server. Must be set to true .
Default Value	true
Property	Licensing
Attribute	license.hwu.uri
Description	The path where the License Server is running e.g., 6200@191.255.255.0 where the syntax is PORTNUMBER@HOST . If multiple servers are specified, use the ';' semicolon separator sign for Windows and the ':' colon separator sign for Linux. NOTE: If value is not set in the Streams.properties, the environment variable ALTAIR_LICENSE_PATH serves as the backup path and will be used.
Example	For Windows: license.hwu.uri=6200@192.168.5.51;6200@192.168.5.52 For Linux: license.hwu.uri=6200@192.168.5.51:6200@192.168.5.52
Default Value	
Property	Licensing
Attribute	license.mode
Description	The license mode. Possible values are: FILE or HWU. Must be set to HWU .
Default Value	FILE

For example:

```
authentication.required=true
license.hwu.uri=6200@192.168.5.51;6200@192.168.5.52
license.mode=HWU
```

NOTE

- Panopticon Streams Server supports different user roles which check out different numbers of Altair Units.

Role	Altair Units License Draw
Designer	10
Administrator	2

- Logging in to both Panopticon Real Time and Panopticon Streams Server with the same username levels the unit draw. A total of 10 units are drawn even if the user logs in to both servers.
- Running applications are leveled towards the user who started the application.

Using Managed Altair Units License via Altair One

Using the managed Altair Units licensing will support simplifying the license management by removing all manual aspects of emailing license files, extending evaluation periods, among others.

In addition, managed Altair Units licensing will help small to medium deployment customers who do not want to host on-premise license server.

Before using Altair Units, it is required to configure certain properties in the [Streams.properties](#) file located in the AppData folder or c:\streamsserverdata:

Property	Licensing
Attribute	license.hwu.hosted
Description	Boolean stating if you wish to use Managed or Local Altair Units licensing. Set to true if you wish to use managed licensing.
Default Value	false
Property	Licensing
Attribute	license.hwu.hosted.authorization.username
Description	Username to the Altair One account.
Default Value	
Property	Licensing
Attribute	license.hwu.hosted.authorization.password
Description	Password to the Altair One account.
Default Value	
Property	Licensing
Attribute	license.hwu.hosted.authorization.token

Description	An authorization token generated through the Altair One admin portal. Used to authorize a machine to the managed Altair Units system.
Default Value	

NOTE

- To use the managed Altair Units licensing, set the following properties:


```
license.hwu.hosted=true
license.mode=HWU
authentication.required=true
```
- Add the Panopticon application to your Altair One account.

To authorize the machine against the managed Altair Units system, you have two options.

Option 1

Use an authorization code generated through Altair One:

1. Log on to Altair One (<https://admin.altairone.com>) then navigate to **User Profile** and select **Authorized Machines**.

<input type="checkbox"/> Username	Hostname	Auth T...	Added At	MAC Addresses
<input type="checkbox"/> * (ANY)	AltairOne	a2c1.....	2021-04-27 1...	ANYHOST
<input type="checkbox"/> * (ANY)	AltairOne	0895.....	2021-04-27 0...	ANYHOST
<input type="checkbox"/> * (ANY)	AltairOne	3db2.....	2021-07-27 0...	ANYHOST

2. Click **Generate Auth Code**.

The page displays the auth code and a timer indicating the code's expiration.

Auth Codes can be used with Almutil by administrators to quickly authorize license access by multiple users on a single machine and/or multiple machines.

Your Auth Code is: XW2ZC81G 

Expires in 14 minutes, 41 seconds

Authorized Machines (7)

<input type="checkbox"/>	Username	Hostname	Auth T...	Added At	MAC Addresses
<input type="checkbox"/>	* (ANY)	AltairOne	a2c1.....	2021-04-27 1...	ANYHOST
<input type="checkbox"/>	* (ANY)	AltairOne	0895.....	2021-04-27 0...	ANYHOST
<input type="checkbox"/>	* (ANY)	AltairOne	3db2.....	2021-07-27 0...	ANYHOST

Click **Copy to Clipboard**  to copy the generated auth code.

- Paste the generated code into the `license.hwu.hosted.authorization.token` property in the `Streams.properties` file.
- Start the server.

Option 2

Directly use your Altair One credentials in `Streams.properties`:

- Enter your Altair One credentials into the `license.hwu.hosted.authorization.username` and `license.hwu.hosted.authorization.password` properties in the `Streams.properties` file.
- Start the server.

NOTE

- If a token is entered, this will be tried first. If the token was invalid or not present, and credentials are present, the credentials will be used to authorize the machine towards the managed Altair Units system.
- In Option 1, the generated auth code is only valid for 15 minutes and you should restart your server within that timeframe to properly get access to your licenses.
- In Option 2, Altair One credentials are only required at first restart of the server to generate the auth token and should be removed from the `Streams.properties` file going forward to avoid exposing these credentials.
- A working Internet connection is required to use Altair Units licensing.
- If your company uses proxy, you might need to add exception in your proxy to allow access to

the Managed Altair Licenses.

Please refer to this link for more information:

https://community.altair.com/kb_view.do?sys_kb_id=bb9bf3fc97205590e3b0361e6253af03

- If you don't have an Altair One account, you can sign up for a free trial that will allow you to test the product for 14 days.

MIGRATION TO STREAMS SERVER 2025.1 FROM AN OLDER VERSION

These instructions assume that you:

- Have an existing 2025 or older server installed and want to migrate the content to a new installation of the 2025.1 server.
- Want to keep running the old server while you make sure that the migration was successful, and that the new server is running as it should.

All of the server content is stored in its application data <appdata> folder, the path of which is set in the PanopticonAppData context environment property. For example, in Tomcat this would be in <tomcat_home>/conf/catalina/localhost/streams.xml or similar.

NOTE

Two Panopticon web applications should never share the same <appdata> folder, ensure that the new server is pointed at its own initially empty folder.

Some of the content can simply be copied from the old server to the new one, while some is now stored in a new format and needs to be converted. The applications and data sources themselves can be migrated any number of times, essentially resetting the applications on the new server.

Summary of steps:

1. [Copy all content.](#)
2. [Delete old content.](#)
3. [One-time conversion](#)
4. [Applications, data sources, and data files](#)
5. [Do not make changes on both servers.](#)
6. [Post-migration cleanup](#)

1. Copy All Content

Start by copying all files from <old_appdata> to <new_appdata>. You can selectively copy some files again later to keep the old and new server in sync (e.g., copy over scheduled tasks after they are modified on the old server). This completes the **migration** of the following:

- ❑ License file - The server will not start without a valid <appdata>/PanopticonLicense.xml. Starting in 2020, you also have the option to use [Altair units licensing](#) instead of the XML file.
- ❑ Properties file - The set of properties in <appdata>/Streams.properties that the server understands changes between releases. The first time you start it, it will add a new one and remove old properties.
- ❑ Scheduled tasks - All scheduled tasks are in SCH files in <appdata>/Schedule/.

2. Delete Old Content

On the new server, delete the <new_appdata>/Tokens/ folder. This holds authentication tokens for logged in users, and they are server specific.

3. One-time Conversion

NOTE

Converting applications and data sources is covered in the [next](#) section.

On servers older than version 2020.0, parameters were stored in <old_appdata>/DefaultParameters.xml. They were global and applied to all content (applications and data sources). Starting in 2021.0 you can now organize content in folders, and you can also define parameters that only apply to content in a particular folder. The new server stores them all in <new_appdata>/Parameters.json.

If <appdata>/Parameters.json doesn't exist when the new server starts, it will create it, and if it finds <appdata>/DefaultParameters.xml it will import these into the new file. To repeat the conversion, e.g., if you want to re-import changed parameters from the old server, delete Parameters.json and restart the server.

4. Applications, Data Sources, and Data files

Applications and their change history, and data sources, are stored in a very different format in a repository inside the <appdata>/.streams-repository/ folder. This is preparation for better versioning, content synchronization in a cluster and other things.

On servers older than version 2020.2, all applications were stored as individual APP files in <appdata>/CEP/Applications. Every time an application was updated, a backup was placed in <appdata>/CEP/Archive. Data sources were stored as DSM files in <appdata>/CEP/Datasources.

If the new server starts and the <appdata>/.streams-repository/ folder doesn't exist, it will create one, and then look in the <appdata>/CEP/ folder. Any applications and data source files it finds in there, it will import into the newly created repository. After the import, the <appdata>/CEP/ folder is no longer used.

Optionally, you can also import all application backups from <appdata>/CEP/Archive/. If you do, they will be recorded as application edits in the new repository. While the web UI currently doesn't expose the change history, it may very well do so in the future.

NOTE

To opt out, set `repository.import.archived.applications` to `false` in [Streams.properties](#).

You can repeat this migration as many times as you like: stop the new server, delete the entire <new_appdata>/streams-repository/ folder, then start the new server. This provides a convenient way

to keep the new server in sync with changes on the old server, assuming the old server is still in use. Please note that this process will lose all changes made on the new server only, as they are stored in the repository.

Data sources that use data files (e.g., CSV, JSON, XML) with relative paths expect the path to be relative to the `<appdata>/Data/` folder. You can simply copy the entire `<old_appdata>/Data/` folder to `<new_appdata>`.

5. Do Not Make Changes on Both Servers

After the initial migration you can keep the new server up to date when content changes on the old server by repeating any of the steps above. It is much harder to move content the other way, from the new server to the old one. Therefore, avoid making changes (that you want to keep) on the new server until you've completely migrated and retired the old server.

6. Post-migration Cleanup

When you are satisfied that the new server is running as it should, that all content has been migrated, switched users over to the new server, and are no longer using the old server, you can remove files from `<new_appdata>` that are no longer needed.

- `<appdata>/DefaultParameters.xml` - These are now in the JSON file.
- `<appdata>/CEP/Applications/` - Applications are now stored in the repository.
- `<appdata>/CEP/Archive/` - If you migrated the change history, this is also in the repository now. Otherwise, you can keep it if you want to go back to an earlier application version.
- `<appdata>/CEP/Datasources/` - Data sources are now also in the repository.

UPGRADE

A previously installed Panopticon Streams can be upgraded through the following process:

1. Stop Tomcat.
2. Delete the existing `webapps\streams.war` file.
3. Delete the deployed application: `webapps\streams`.
4. Delete the cache from the working folder (for example):
`work\Catalina\localhost\streams`
5. Deploy the new `streams.war` file by copying it to the Tomcat `webapps` folder.
6. Restart Tomcat.

[3] AUTHENTICATION

INTRODUCTION

Panopticon Streams provides multiple approaches to authentication. It can easily be configured to use different authentication mechanisms depending on the environment and the setup. The server only supports authentication and authorization and does not have any support for user management or administration of users.

There are mainly two properties that manage the authentication on the server. These properties are listed and described in the table below. Please note that more properties might need to be configured depending on the authentication mechanism you are using.

Property	Description	Default Value
authentication.role	The required role or group that the user needs to be identified as a Panopticon user. The property can be left blank if no role or group is required.	
authentication.required	This property will make the authentication required. It will force the user to login in order to use any of the services provided by the server.	true
authentication.type	The type of authentication that should be used when authenticating the user. The property allows the following values: BASIC , FILTER , HEADER , OAUTH2 , SAML	BASIC

Mapping Users to Roles

Depending on the authentication or user management mechanism used, the role that a user should have is specified and then mapped to a group set in [Streams.properties](#).

Property	Description	Default Value
access.administrator.groups	The role that is mapped to the administrator group. Allowed to perform the following: <ul style="list-style-type: none">• connect to or disconnect from the CEP Engine.• create, rename, remove folders and subfolders, upload applications or data sources, and manage users or groups that should be granted or denied access.• import and export application bundles.• rename, view topic or data source usage, move, copy, download, remove, and publish/republish applications to folders to which the user has permission.• rename, view application usage, move, copy, download, and remove data sources.• Administer the server which includes:<ul style="list-style-type: none">◦ refresh, start, and stop data producers.	admin

Property	Description	Default Value
	<ul style="list-style-type: none"> ○ view engine metrics and retrieved messages. ○ add, modify, refresh, and delete parameters. ○ define file logging level or view, pause, resume logging, copy, and clear all logs ○ view Kafka properties. ○ reload configuration. ○ create, modify, and delete clear topic data tasks. 	
access.default.roles	<p>The default roles applied to all users of the server.</p> <p>For example, If <code>access.default.roles=DESIGNER,ADMINISTRATOR</code> and a user with a VIEWER role logs on to the server, then the user will simultaneously have a VIEWER, DESIGNER, and ADMINISTRATOR roles.</p> <p>However, if no default roles are wanted, then leave the property blank.</p> <p>NOTE: The roles that can be assigned in this property can only be ADMINISTRATOR, VIEWER, ANONYMOUS, and/or DESIGNER. This property is case sensitive.</p>	VIEWER
access.designer.groups	<p>The role that is mapped to the designer group.</p> <p>Allowed to perform the following:</p> <ul style="list-style-type: none"> • create, rename, remove folders and subfolders, upload applications or data sources, and manage users or groups that should be granted or denied access. • create, rename, view topic or data source usage, move, copy, download, remove, and publish/republish applications to folders to which the user has permission • create, rename, view application usage, move, copy, download, and remove data sources. • import and export application bundles. 	designer
access.viewer.groups	<p>The role that is assigned to the viewer group.</p> <p>Allowed to view the engine status.</p>	viewer

NOTE

Group sets can be added for a role, by default separated by a comma.

Normally, you should use role mapping to control user access. This way you can manage access in the same place that you manage your users without having to reconfigure the server.

In some scenarios, it may be impossible to set up appropriate roles for Panopticon in your external system, or you may want to make one-off exceptions for specific users. As a workaround for these cases, you can also explicitly list individual users and their access in the server configuration with the [access.administrator.users](#), [access.designer.users](#), and [access.viewer.users](#) properties.

Token

A web token is used when the user has successfully logged into Panopticon Streams when using one of the following authentication types: BASIC or SAML. The token is used to identify the user and represent the user's ongoing session. This is done to prevent user credentials being sent between the user and server more than necessary.

The token is returned from Panopticon Streams in the form of a cookie when the user has been authenticated. The cookie will be stored in the browser as a HttpOnly cookie.

The token can be configured differently to suit your needs and requirement. The token can be configured to be valid at a certain amount of time, if it can refresh itself and/or if it should be persistent or if it should only last for a user session (While the browser is still open). All this can be configured in the [Streams.properties](#). The table below lists all available token properties.

Property	Description	Default Value
authentication.token.persistence	<p>This property is used to determine if the token should persist if the browser is closed or if it should only last while the browser is open. There are two possible values: PERSISTENT and SESSION. PERSISTENT will persist the token in the browser even if the browser has been closed and reopened. SESSION will remove the token from the browser if it is shutdown.</p> <p>IMPORTANT: After modifying the property value to SESSION, ensure to clear the AppData/Token folder before starting the server.</p>	PERSISTENT
authentication.token.refreshable	<p>This property determines if the token can refresh itself. The web client can identify if the token is about to expire and then request a new token with the existing token. A token is refreshable if the property is set to true. The token will expire and invalidate the user session if the property is set to false.</p>	true
authentication.token.secret	<p>The secret is used to sign the token. The secret will be auto-generated when the server starts for the first time.</p> <p>NOTE: This value should be kept a secret.</p>	Auto-generated
authentication.token.validity.seconds	The number of seconds that the token should be valid.	604800

TOMCAT REALM

Panopticon Streams can be configured to use the Tomcat Realm when performing authentication. The Tomcat Realm is configured in the `server.xml` file in the Tomcat `conf` folder. The Tomcat Realm itself can be configured to authenticate towards a variety of different types of authentication source, such as Tomcat user base and LDAP. The sub chapters in this chapter will give examples on how to configure the Tomcat Realm.

Panopticon Streams needs to be configured to use the BASIC type in order to do the authentication towards the Tomcat Realm. To enable Tomcat Realm authentication, set this property in the [Streams.properties](#) file:

`authentication.type=BASIC`

NOTE

It is a common approach to wrap your Tomcat Realm with the LockOutRealm. This is used to prevent brute-force attacks.

```
<Realm className="org.apache.catalina.realm.LockOutRealm">
    <!--Insert your own Tomcat Realm here -->
</Realm>
```

Tomcat User Base

The Tomcat User Base Realm is using a JNDI resource to store user information. By default, the JNDI resource is configured in an XML file. The default file is `tomcat-users.xml` in the Apache Tomcat `conf` folder.

We strongly recommend using this authentication approach for your test or local environment. It is easy to setup and configure. However, it is not designed to be used for large-scale production or when you have a large number of users.

The following Realm has to be added in the `server.xml` file in the Apache Tomcat `conf` folder:

```
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"
resourceName="UserDatabase"/>
```

NOTE

The Tomcat User Database Realm is used as the default. No configurations are required in the `server.xml` file to be able to use the Tomcat Database Realm.

The users and roles are managed in the `tomcat-users.xml` file in the Apache Tomcat `conf` folder. In this file, you can add users and roles as well as assign roles to users.

Add the following role and user to your `tomcat-users.xml` file:

```
<role rolename="admin"/>
<user username="John" password="john" roles="admin"/>
```

By adding these two lines you have achieved the following:

- Created a new role named **admin**
- Created a new user with username **John** and password **john**
- Assigned the newly created user the role **admin**

NOTE

Authentication towards a Tomcat Realm (i.e., Tomcat users, LDAP, AD) in Tomcat 8.5.28 is not supported. This has been supported in all the previous and the succeeding versions.

Tomcat Memory Configuration for Windows

NOTE

It is recommended to increase the Java heap size of Tomcat to avoid the initiation of garbage collection when memory usage hits the set threshold.

The steps may vary depending on how Tomcat was deployed.

Steps:

1. Stop Tomcat.
2. Create a file named `setenv.bat`.
3. Place the file in the Tomcat `bin` folder.
4. Set the minimum and maximum heap size with the JVM `-Xms` and `-Xmx` parameters. A minimum of 1 GB is recommended. For example:

```
set JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=UTF-8 -server -Xms512m -Xmx2g
```

NOTE

Setting the maximum value should be dependent on your system. Ensure that the heap size is not larger than the available free RAM on your system. It is recommended to use 80% of the available RAM not taken by the operating system or other processes of your JVM.

5. Save the file.
6. Restart Tomcat to apply the increase in the heap.

Tomcat Memory Configuration for Linux

NOTE

It is recommended to increase the Java heap size of Tomcat to avoid the initiation of garbage collection when memory usage hits the set threshold.

The steps may vary depending on how Tomcat was deployed.

Steps:

1. Stop Tomcat.
2. Create a file named `setenv.sh`.
3. Place the file in the Tomcat `bin` folder.
4. Set the minimum and maximum heap size with the JVM `-Xms` and `-Xmx` parameters. A minimum of 1 GB is recommended. For example:

```
JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=UTF-8 -server -Xms512m -Xmx2g"
```

NOTE

Setting the maximum value should be dependent on your system. Ensure that the heap size is not larger than the available free RAM on your system. It is recommended to use 80% of the available RAM not taken by the operating system or other processes of your JVM.

5. Save the file.
6. Restart Tomcat to apply the increase in the heap.

Encrypting Passwords in tomcat-users.xml

Tomcat supports encrypted user credentials via the Digested Passwords feature:

https://tomcat.apache.org/tomcat-9.0-doc/realm-howto.html#Digested_Passwords

To secure passwords saved in tomcat-users.xml, do the following:

1. Stop Tomcat.
2. Open [tomcat_home]/conf/server.xml.
3. In server.xml, find the Engine XML element.

Nested inside the Engine element, there is a Realm element named LockOutRealm. Nested inside the LockOutRealm is another Realm element named UserDatabaseRealm that looks like this:

```
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"  
      resourceName="UserDatabase"/>
```

4. Edit the UserDatabaseRealm element into the following:

```
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"  
      resourceName="UserDatabase">  
    <CredentialHandler  
      className="org.apache.catalina.realm.MessageDigestCredentialHandler"  
      algorithm="SHA-256"/>  
</Realm>
```

NOTE

You must add the closing element "</Realm>" for the UserDatabaseRealm, and edit out the closing forward slash "/" at the end of the original Realm element.

5. Generate hash from plain text passwords using the command below:

Linux example:

```
[tomcat_home]/bin/digest.sh -a SHA-256 -h  
org.apache.catalina.realm.MessageDigestCredentialHandler [password]
```

Windows example:

```
[tomcat_home]/bin/digest.bat -a SHA-256 -h  
org.apache.catalina.realm.MessageDigestCredentialHandler [password]
```

NOTE

If your Apache Tomcat installation has the `JAVA_HOME` environment variable set only in the file `catalina.sh` (Linux) or `catalina.bat` (Windows) and not generally on the system, you will also need to set the `JAVA_HOME` variable before running the `digest` command.

Linux example:

```
export JAVA_HOME=/path/to/JavaInstallation
```

Windows example:

```
set JAVA_HOME=/path/to/JavaInstallation
```

The `digest` command will return the password supplied, followed by a colon, and then a hash of the password. Example, for a password **asd123**:

```
asd123:74807befd6bdc1c937dc931a3dfadf015da1df1b99b74cd8d91210788e0141a5$1f21cb2dd667209d639f6be48cf83826a657730032bdacb04465262d221bfc509
```

6. Replace the plain text password in `tomcat-users.xml` with the generated password hash, and save the `tomcat-users.xml` file. **NOTE:** When you have defined a `MessageDigestCredentialHandler` in the `UserDatabaseRealm`, then ALL passwords stored in `tomcat-users.xml` are treated as hash values. You will no longer be able to log in using passwords that are saved as clear text.
7. Start Tomcat.

LDAP

Panopticon Streams can be configured to authenticate towards a Lightweight Directory Access Protocol (LDAP) or source. By configuring the Apache Tomcat Realm, the server can authenticate users and extract their roles by querying the LDAP source.

The realm's connection to the directory is defined by the `connectionURL` attribute. Each user that can be authenticated must be represented in the directory with an individual entry that corresponds to an element in the initial `DirContext` from the `connectionURL`. This user entry must have an attribute containing the `username` that is presented for authentication.

You can add a dedicated user with `connectionName` and `connectionPassword` in a Realm to define a user with a Read access to the user database and roles. If for example the `admin` `cn` name is set as **admin** and the `admin` `password` is set as **admin**, then you need to add these properties as shown in the example below.

The `userPattern` attribute may be used to specify the DN, with “`{ 0 }`” marking where the `username` should be substituted.

The role is usually an LDAP group entry with one attribute containing the name of the role and another one whose values are distinguished names or usernames of the users in that role. The following attributes configure a directory search to find the names of roles associated with the authenticated user:

- **roleBase:** The base entry for the role search. If not specified, the search base is the top-level directory context
- **roleSearch:** The LDAP search filter for selecting role entries
- **roleName:** The attribute in a role entry containing the name of that role
- **roleNested:** Includes nested roles if set to **true**. This means every newly found `roleName` and distinguished Name will be recursively tried for a new role search. The default behavior is **false**.

The following is an example on how the Realm can be configured when using LDAP. Please note that the values should be replaced with details from your own LDAP source.

```

<Realm className="org.apache.catalina.realm.JNDIRealm"
connectionURL="ldap://localhost:389"
connectionName="cn=admin,dc=test,dc=com"
connectionPassword="admin"
userPattern="uid={0},ou=users,dc=test,dc=com"
roleBase="ou=groups,dc=test,dc=com"
roleName="cn"
roleSearch="(uniqueMember={0})"
roleNested="true"

/>

```

Using this configuration, the realm determines the user's distinguished name by substituting the username into the `userPattern`, authenticates by binding to the directory with this DN and the password received from the user, and searches the directory to find the user's roles.

NOTE

If you opt not to have a dedicated user, remove `connectionName` and `connectionPassword`, and then have each user extract information about itself. You do this by adding `userSearchAsUser` and `roleSearchAsUser` in a Realm, and setting both values to true. The recommended usage, however, is to have a dedicated user. This allows you to always have the rights to query a LDAP, unlike using `userSearchAsUser` and `roleSearchAsUser` where there is no guarantee that each user is authorized to extract these details.

Active Directory

Panopticon Streams can be configured to authenticate towards an Active Directory server. Panopticon Streams is using LDAP to interact and communicate with the Active Directory server. Therefore, the configuration is very similar to the LDAP configuration in the previous section.

The following is an example on how the Realm can be configured when using Active Directory. Please note that the values should be replaced with details from your own LDAP source.

```

<Realm className="org.apache.catalina.realm.JNDIRealm"
connectionURL="ldap://ad.dwch.com:3268"
alternateURL="ldap://ad.dwch.com:389"
authentication="simple"
referrals="follow"
connectionName=admin@DWCH.com
connectionPassword="admin"
userBase="cn=Users,dc=DWCH,dc=com"
userSearch="(sAMAccountName={0})"
userSubtree="true"
roleBase="cn=Users,dc=DWCH,dc=com"
roleName="cn"
roleSearch="(member={0})"
roleSubtree="true"
roleNested="true"

/>

```

NOTE

Similar with LDAP, you can opt not to have a dedicated user by removing `connectionName` and `connectionPassword` and instead let each user extract information about itself by adding `userSearchAsUser` and `roleSearchAsUser` in a Realm. Set both values to true. As mentioned in the LDAP section, the recommended usage is to have a dedicated user since there is no guarantee that each user is authorized to extract these details.

SAML

Panopticon Streams supports Security Assertion Markup Language, SAML2. Upon a login request, Panopticon Streams will redirect the user to an Identity provider (IdP). The IdP will authenticate the user and redirect the user back to Panopticon Streams. The response message will be controlled and validated. Username and roles will be extracted from the response message and used within Panopticon Streams.

Panopticon Streams will redirect the user back to the IdP upon a logout request. The IdP logout service should then invalidate the SAML token.

Property	Description
<code>authentication.saml.assertion.roles</code>	User attribute for roles configured in the IdP.
<code>authentication.saml.assertion.username</code>	User attribute for username configured in the IdP.
<code>authentication.saml.assertionconsumerservice.url</code>	The URL to the Panopticon assertion consumer service. URL: [Protocol]://[Host]:[Port]/[Context]/server/rest/auth/login
<code>authentication.saml.certificate.name</code>	The name of the certificate used to validate signature and/or sign outgoing SAML messages
<code>authentication.saml.certificate.password</code>	The password of the certificate used to validate signature and/or sign outgoing SAML messages..
<code>authentication.saml.identityprovider.logout.url</code>	The URL to the IdP logout service.
<code>authentication.saml.identityprovider.url</code>	The URL to the IdP login service.
<code>authentication.saml.keystore.file</code>	The location of the Keystore file that contains the certificate.
<code>authentication.saml.keystore.password</code>	The password to the Keystore file.
<code>authentication.saml.serviceprovider.id</code>	The ID of the service provider configured in the IdP.
<code>authentication.saml.identityprovider.certificate.file</code>	Takes a file path to a certificate file that contains the IdP's public key.
<code>authentication.saml.identityprovider.signature.validation.required</code>	Specifies whether to require a valid IdP signature to be present on the SAML response. Default value is false .
<code>authentication.saml.provider</code>	The IdP provider. Possible values are OPENSAML , OPENAM . Default value is OPENSAML .

Property	Description
authentication.saml.keystore.type	The key store type. Possible values are JKS , JCEKS , PKCS12 . Default value is JKS .
authentication.saml.openam.meta.alias	The meta alias for the IdP if you are using OpenAM.

OAUTH 2.0

This section discusses how to configure Panopticon Streams to use the OAuth 2.0 for authorization. Upon a logon request, Panopticon Streams will redirect the user to the Login page provided by the OAuth 2.0.

Note that OAuth 2.0 does not normally provide support on how to authenticate the user, Panopticon Streams will only know if the user is authorized or not. To authenticate the user, Panopticon Streams can be configured to use a REST service to extract the user identity with an access token retrieved from the OAuth 2.0 provider. In addition to the standard OAuth 2.0 configurations, the server includes properties (i.e., `authentication.oauth2.*`) that are specifically used to extract the user details.

`authentication.type=OAUTH2`

Property	Description
<code>authentication.oauth2.client.id</code>	The ID of the OAuth 2.0 client.
<code>authentication.oauth2.client.secret</code>	The secret used by the OAuth 2.0 client.
<code>authentication.oauth2.identity.attribute.username</code>	The attribute that will be extracted from the identity response and used as the username.
<code>authentication.oauth2.identity.url</code>	The URL to the REST service that provides details about the authenticated user.
<code>authentication.oauth2.login.callback.url</code>	The callback URL. The URL should be the same as one of the specified callback URLs used by the client. The URL should refer to Panopticon Streams
<code>authentication.oauth2.login.response.type</code>	The response type. The only response type that is currently supported is CODE . The value can also be left blank.
<code>authentication.oauth2.login.scope</code>	The requested scope. The field can be left blank.
<code>authentication.oauth2.login.url</code>	The URL to the OAuth 2.0 login resource.
<code>authentication.oauth2.logout.url</code>	The URL to the OAuth 2.0 logout resource. This field can be left blank.
<code>authentication.oauth2.token.method</code>	The method on how the token should be retrieved. Supported values are QUERY , BODY , and HEADER .
<code>authentication.oauth2.token.url</code>	The URL to the OAuth 2.0 token resource.

Example

```
authentication.oauth2.client.id=ClientId
authentication.oauth2.client.secret=ClientSecret
authentication.oauth2.identity.attribute.username=name
```

```

authentication.oauth2.identity.url=https://oauth2/me
authentication.oauth2.login.callback.url=http://localhost:8080/panopticon/server/rest/auth/login
authentication.oauth2.login.response.type=CODE
authentication.oauth2.login.scope=
authentication.oauth2.login.url=https://oauth2/authorize
authentication.oauth2.logout.url=
authentication.oauth2.token.method=QUERY
authentication.oauth2.token.url=https://oauth2/access_token
authentication.type=OAUTH2

```

FILTER

Custom authentication filters can be applied to the server and the application when the default authentication settings are not sufficient. This type of authentication is referred to as **Filter authentication**. When Panopticon Streams is configured to use filter authentication, it means that the incoming requests have already been authenticated and authorized before reaching the server. Follow the steps below to configure filter authentication:

1. Open the `Streams.properties` file in the `AppData` folder (`c:\streamsserverdata`).
2. Enable `authentication.type=FILTER` in `Streams.properties`.
3. Apply the following URL pattern to your own filter: `/*`
4. Save the changes and restart the Tomcat.

Creating a Custom Filter

The custom filter will be a basic authentication filter which will authenticate the user with hardcoded values. The Principal forwarded by the filter will be used to authenticate the user.

The filter will require the following dependencies:

- Javax Servlet
- Tomcat embed core

Steps:

1. Create a HTTP request wrapper.

The class will contain the following:

- the original incoming HTTP request
- the Principal which contains both the credentials and the roles for the authenticated user.

The HTTP wrapper will be forwarded to Panopticon Streams instead of the original incoming HTTP request.

```

import org.apache.catalina.realm.GenericPrincipal;
import org.apache.catalina.users.MemoryUser;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;
import java.security.Principal;

public class FilterRequestWrapper extends HttpServletRequestWrapper {

    private final GenericPrincipal principal;

```

```

        public FilterRequestWrapper(final HttpServletRequest request, final
GenericPrincipal principal) {
    super(request);
    this.principal = principal;
}

@Override
public Principal getUserPrincipal() {
    return principal;
}

@Override
public boolean isUserInRole(final String role) {
    if (principal != null) {
        return principal.hasRole(role);
    }
    return super.isUserInRole(role);
}
}
}

```

2. Create a custom filter. The filter will create a new Principal which includes both the credentials and the groups/roles for the user.

In this example, the class `GenericPrincipal` contains `username`, `password`, and `groups`. Panopticon Streams is only able to extract the groups from `GenericPrincipal` class or the `MemoryUser` class. Both the Principal and the original HTTP request will be wrapped in an instance of `FilterRequestWrapper`. The wrapper will then be forwarded towards Panopticon Streams.

```

import org.apache.catalina.realm.GenericPrincipal;
import org.apache.catalina.users.MemoryUser;
import javax.servlet.*;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.security.Principal;
import java.util.Arrays;
import java.util.List;

public class ExampleFilter implements Filter{

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {}

    @Override
    public void doFilter(final ServletRequest servletRequest, final
ServletResponse servletResponse, FilterChain filterChain) throws
IOException, ServletException {
        if (!(servletRequest instanceof HttpServletRequest ||

! (servletRequest instanceof HttpServletResponse))) {
            return;
        }

        final HttpServletRequest request = (HttpServletRequest)
servletRequest;
        final HttpServletResponse response = (HttpServletResponse)
servletResponse;
        final String username = "username";
    }
}

```

```

        final String password = "password";
        final List<String> groups = Arrays.asList("Group1", "Group2");
        final GenericPrincipal principal = new GenericPrincipal(username,
password, groups);
        filterChain.doFilter(new FilterRequestWrapper(request, principal),
response);
    }

    @Override
    public void destroy() {}
}

```

3. When these classes have been created, you can compile them and package them in a jar file.
4. Copy the jar file to the WEB-INF/lib folder in the panopticon war file (or the extracted folder).
5. Enable the filter by adding the following code to the web.xml file in panopticon WEB-INF folder:

```

<filter>
    <filter-name>ExampleFilter</filter-name>
    <filter-class> com.panopticon.server.filter.ExampleFilter </filter-
class>
</filter>
<filter-mapping>
    <filter-name>ExampleFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

```

HEADER

It is possible to use a web-facing Panopticon Streams behind a proxy server that will handle the authentication of users. The proxy server forwards the name of the user and roles to Panopticon Streams as HTTP headers for every request.

For requests where headers are blank or missing, they are treated like anonymous requests while requests where the user HTTP header is valid are treated like authenticated requests with that specific username.

Requests from the proxy server are fully trusted and checks are no longer performed at Panopticon Streams regarding the validity of the username. The authorization on workbooks and administration will work as usual.

To activate the Header authentication, add or update the following properties in the [Streams.properties](#) file:

```

authentication.type=HEADER
authentication.header.role.delimiter=,
authentication.header.roles={roles header, ie. X-Roles}
authentication.header.username=={userid header, ie. X-User}

```

[4] PCLI: COMMAND UTILITIES FOR PANOPTICON

Panopticon Streams is supplied with a command line utility PCLI.jar.

EXPORT DATA SOURCES

THE PCLI provides functionality to export data sources from one or all workbooks in a directory. The exported data sources can be [uploaded](#) and used directly by Panopticon Streams.

Parameters

Parameter	Description	Required
-w, --workbook	The name of the workbook.	Yes (or -wd)
-od, --output-directory	The output directory where the data source will be exported to.	No
-wd, --workbook-directory	The directory of the workbooks folder.	Yes (or -w)
-dd, --data-directory	The directory of the data folder.	Yes
-l, --license-file	The path of the license file.	Yes

Example 1: Export data sources from a workbook

```
java -jar pcli.jar exportdatasource
-w "C:/vizserverdata/Workbooks/VizGuide.exw"
-l "C:/vizserverdata/PanopticonLicense.xml"
-dd "C:/vizserverdata/Data"
-od "C:/streamsserverdata/CEP/Datasources"
```

Example 2: Export data sources from all workbooks example

```
java -jar pcli.jar exportdatasource
-wd "C:/vizserverdata/Workbooks"
-l "C:/vizserverdata/PanopticonLicense.xml"
-dd "C:/vizserverdata/Data"
-od "C:/streamsserverdata/CEP/Datasources"
```

Where:

- C:\vizserverdata is the AppData folder of Panopticon Real Time
- C:\streamsserverdata is the AppData folder of the Streams server

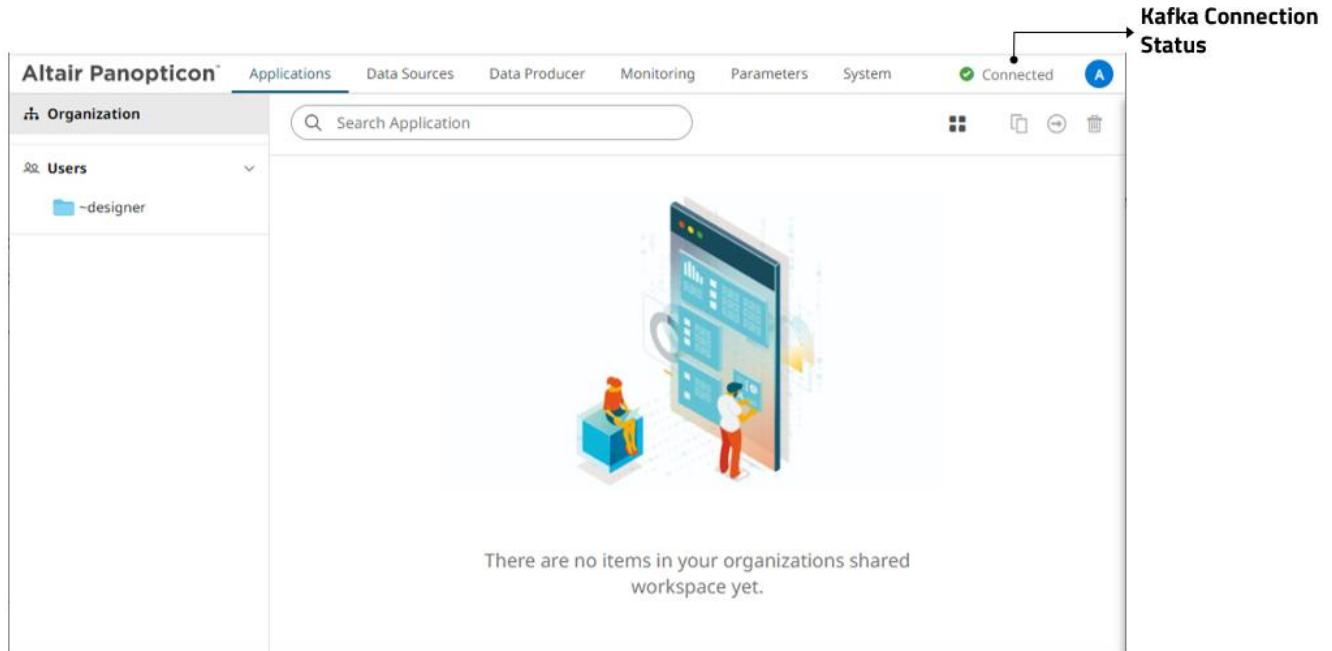
[5] USING ALTAIR PANOPTICON STREAMS

CONNECTING TO OR DISCONNECTING FROM THE CEP ENGINE

NOTE

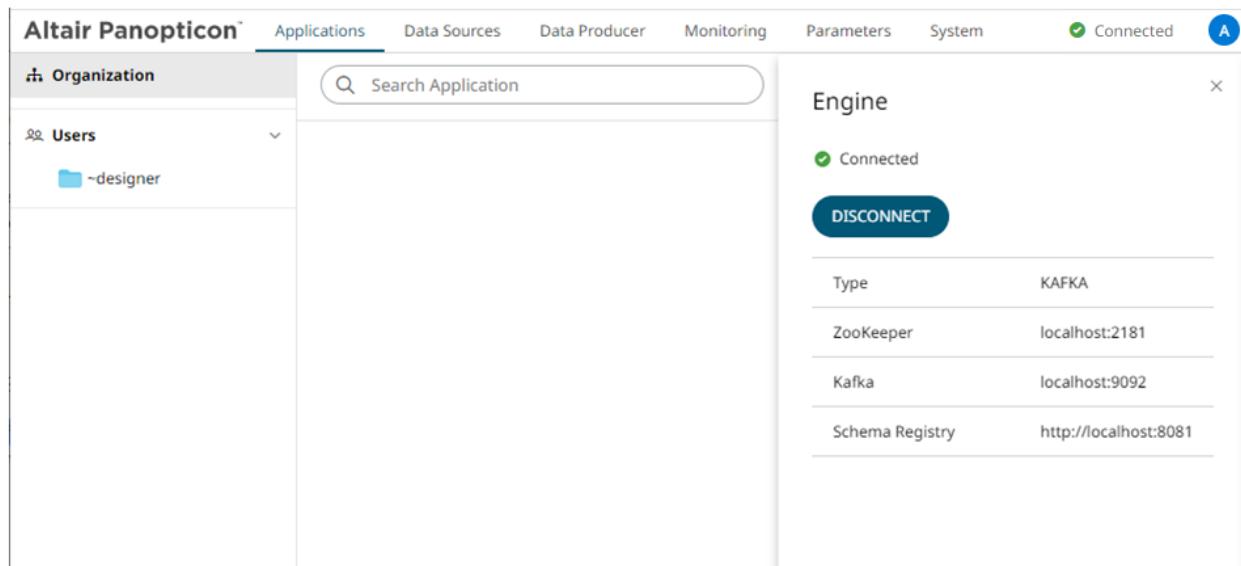
Panopticon Streams Server will be connected to the CEP engine after start up if any of the following settings is true:

- The default setting of the localhost for the Kafka broker is available.
- The Kafka settings in the [Streams.properties](#) file are correct.



Click **Kafka Connection Status** to expand and display the *Engine* pane and view the settings.

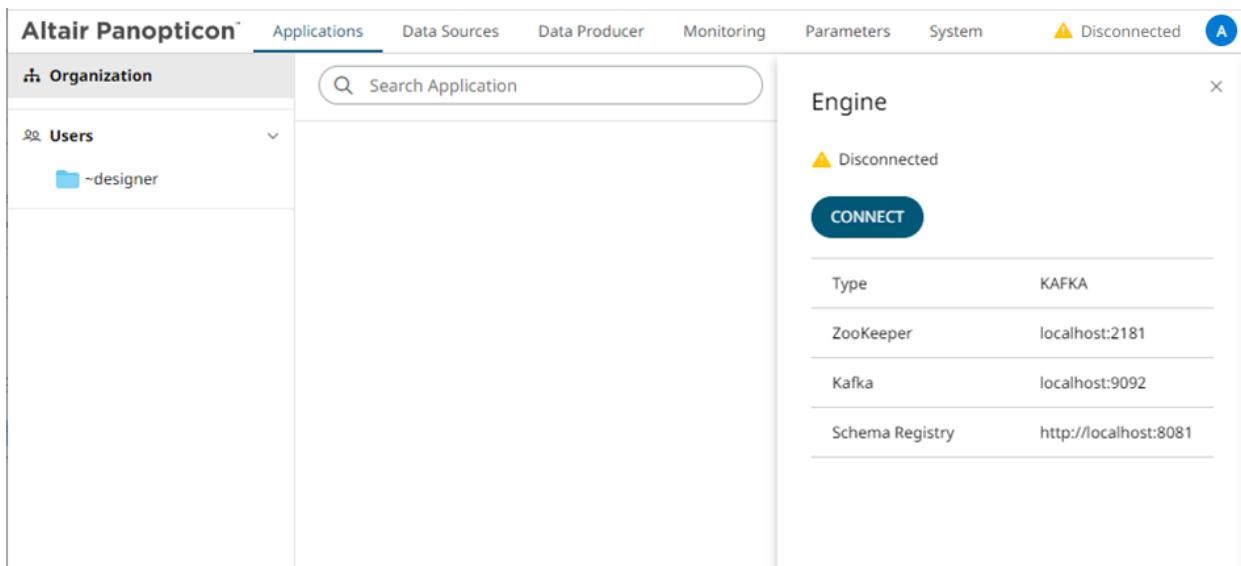
For  Connected



The screenshot shows the Altair Panopticon Streams interface with a "Connected" status. The "Engine" section on the right displays the following configuration:

Type	KAFKA
ZooKeeper	localhost:2181
Kafka	localhost:9092
Schema Registry	http://localhost:8081

For  Disconnected



The screenshot shows the Altair Panopticon Streams interface with a "Disconnected" status. The "Engine" section on the right displays the following configuration:

Type	KAFKA
ZooKeeper	localhost:2181
Kafka	localhost:9092
Schema Registry	http://localhost:8081

Property	Description
Status	Displays whether Panopticon Streams is connected to or disconnected from the CEP Engine (Kafka).
Type	The CEP Engine that Panopticon Streams engine will work with (KAFKA).
ZooKeeper	The URL to the ZooKeeper servers. Default is localhost:2181 .
Kafka	The URL of all the Kafka servers. Default is localhost:9092 .
Schema Registry	The URL to the Schema Registry. Default is http://localhost:8081 .

Connecting to the CEP Engine

Starting with version 2021.0, the “local” or “internal” Kafka connectivity is deprecated. To connect to the CEP engine, use the external setup.

NOTE

Before connecting to the CEP engine, ensure the following are running:

- [Confluent Kafka Enterprises services](#) if you are using a Dockerized Kafka.
- ZooKeeper, Kafka, and Schema Registry batch files if you are using a local cluster.

CONNECT

Click **CONNECT** to connect to the external Kafka.

Disconnecting from the CEP Engine

DISCONNECT

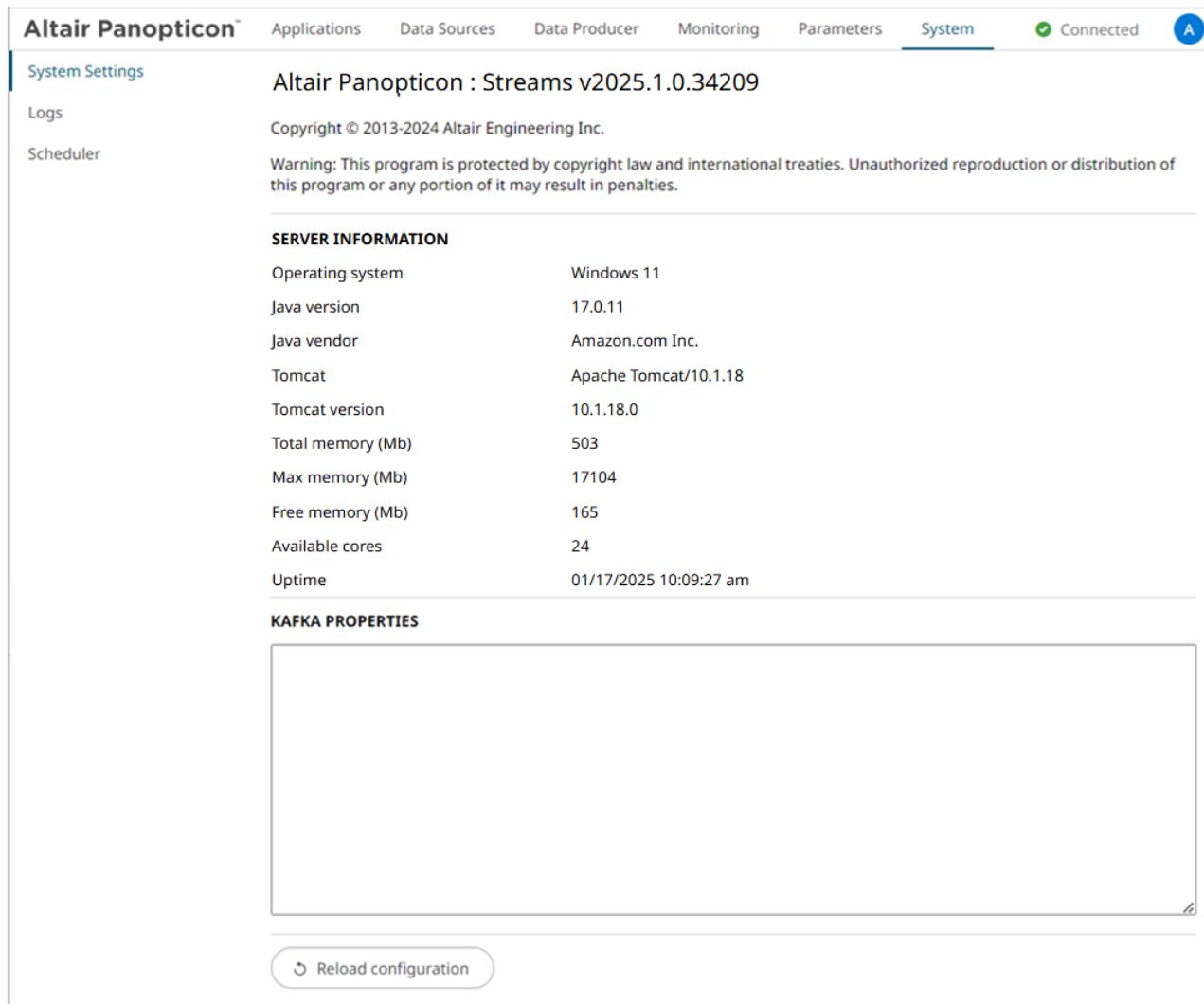
Click **DISCONNECT**. Consequently, the applications cannot be started and the input and output topics will not be generated.

[6] MANAGING THE STREAMS SYSTEM

The **System** tab displays the following sections where an administrator can:

- View the active license
- View the server properties
- View [Kafka properties](#)
- [Reload configurations](#)
- [Schedule tasks](#)

If the licensing used is the [Altair Units license](#), the page will be displayed as:



The screenshot shows the 'System' tab of the Altair Panopticon interface. The top navigation bar includes tabs for Applications, Data Sources, Data Producer, Monitoring, Parameters, and System, with 'System' being the active tab. A status bar on the right indicates 'Connected' with a green checkmark and a letter 'A'.

System Settings

Altair Panopticon : Streams v2025.1.0.34209

Copyright © 2013-2024 Altair Engineering Inc.

Warning: This program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program or any portion of it may result in penalties.

SERVER INFORMATION

Operating system	Windows 11
Java version	17.0.11
Java vendor	Amazon.com Inc.
Tomcat	Apache Tomcat/10.1.18
Tomcat version	10.1.18.0
Total memory (Mb)	503
Max memory (Mb)	17104
Free memory (Mb)	165
Available cores	24
Uptime	01/17/2025 10:09:27 am

KAFKA PROPERTIES

A large empty rectangular area for Kafka properties.

[Reload configuration](#)

If the [licensing](#) used is the volume-based XML file (named **PanopticonLicense.xml**), the page will be displayed as:

Altair Panopticon : Streams v2025.1.0.34209

Copyright © 2013-2024 Altair Engineering Inc.

Warning: This program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program or any portion of it may result in penalties.

LICENSE

```
<PanopticonLicense CustomerName="Test License nr 1" SerialNumber="1" Created="2024-01-24" SchemaVersion="1.1" xmlns="http://panopticon.com/PanopticonLicense/2007/11">
  <Product Name="Panopticon Developer Java">
    <Fallback ExpiryDate="2025-01-31" Evaluation="False" OEM="False" />
    <Visualizations />
    <Datasources>
      <TypeLicense Type="com.panopticon.excelplugin.Plugin" ExpiryDate="2025-01-31" Evaluation="False" OEM="False" />
      <TypeLicense Type="com.panopticon.activemqplugin.Plugin" ExpiryDate="2025-01-31" Evaluation="False" OEM="False" />
      <TypeLicense Type="com.panopticon.aleripugin.Plugin" ExpiryDate="2025-01-31" Evaluation="False" OEM="False" />
      <TypeLicense Type="com.panopticon.cassandraplugin.Plugin" ExpiryDate="2025-01-31" Evaluation="False" OEM="False" />
      <TypeLicense Type="com.panopticon.cloudantplugin.CloudantPlugin" ExpiryDate="2025-01-31" Evaluation="False" OEM="False" />
    </Datasources>
  </Product>
</PanopticonLicense>
```

Location on the server: c:\streamsserverdata

SERVER INFORMATION

Operating system	Windows 11
Java version	17.0.11
Java vendor	Amazon.com Inc.
Tomcat	Apache Tomcat/10.1.18
Tomcat version	10.1.18.0
Total memory (Mb)	662
Max memory (Mb)	17104
Free memory (Mb)	470
Available cores	24
Uptime	10/21/2024 02:12:37 pm

KAFKA PROPERTIES

Reload configuration

VIEWING AND MANAGING KAFKA PROPERTIES

The user-defined `Kafka.properties` file contains properties for controlling the Kafka configurations in Panopticon Streams.

Below is a sample properties file:

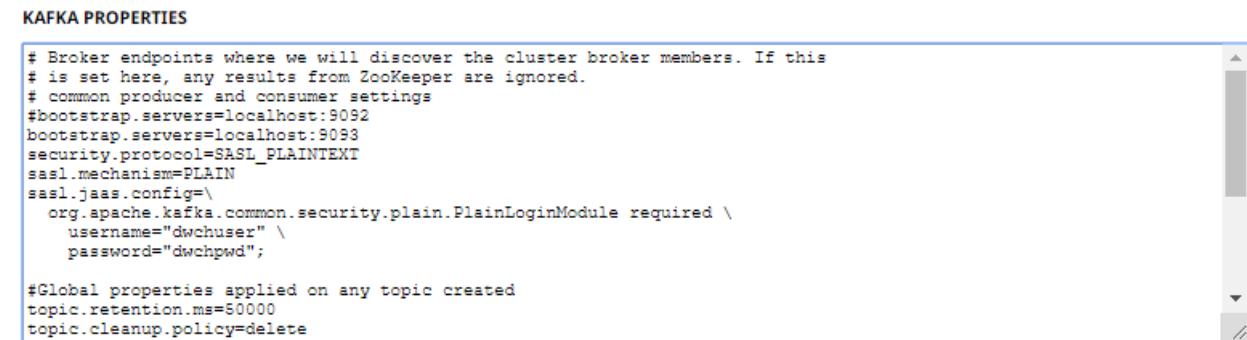
```
# Broker endpoints where we will discover the cluster broker members.
# If this
# is set here, any results from ZooKeeper are ignored.
# common producer and consumer settings
#bootstrap.servers=localhost:9092
bootstrap.servers=localhost:9093
security.protocol=SASL_PLAINTEXT
sasl.mechanism=PLAIN
sasl.jaas.config=\
    org.apache.kafka.common.security.plain.PlainLoginModule required \
        username="dwchuser" \
        password="dwchpwd";

#Global properties applied on any topic created
topic.retention.ms=50000
topic.cleanup.policy=delete

aggregate.cachingEnabled=true

#Specific operator/node applicationId.operatorId.propertyname
AggregationExample.Input.retention.ms=60000
AggregationExample.Output.retention.ms=30000
AggregationExample.Aggregation.cachingEnabled=false
```

When Panopticon Streams is started, it checks the `AppData` folder for the `kafka.properties` file and loads the properties in the *Kafka Properties* box.



However, if the `kafka.properties` file is not available, the *Kafka Properties* box will display a blank *Kafka Properties* box:

KAFKA PROPERTIES

If you opt to copy the `kafka.properties` file to a different location, open the `Streams.properties` file and set the attribute `cep.kafka.properties` to the value of the file path along with the Kafka properties file name. For example:

```
cep.kafka.properties=c:\kafkafile\kafka.properties
```

NOTE

- The values in the *Kafka Properties* box is not editable on the **System** tab. Changes can be made in the actual `kafka.properties` file. To reload the properties on the **System** tab, click **Reload Configuration**.
- The `kafka.properties` file supports any Kafka configurations available on their documentation
- The configurations made in the `kafka.properties` will supersede any of the Kafka-related properties in the `streams.properties` file
- Some of the configurations in the `kafka.properties` file can be overridden by the settings made in Panopticon Streams applications

RELOADING CONFIGURATIONS

 [Reload configuration](#)

On the *System Settings* page under the **System** tab, click [Reload configuration](#). This will stop and restart applications, reload data sources, and Kafka properties along with the administrators and parameters from the file system.

LOGGING/MONITORING

View Logs

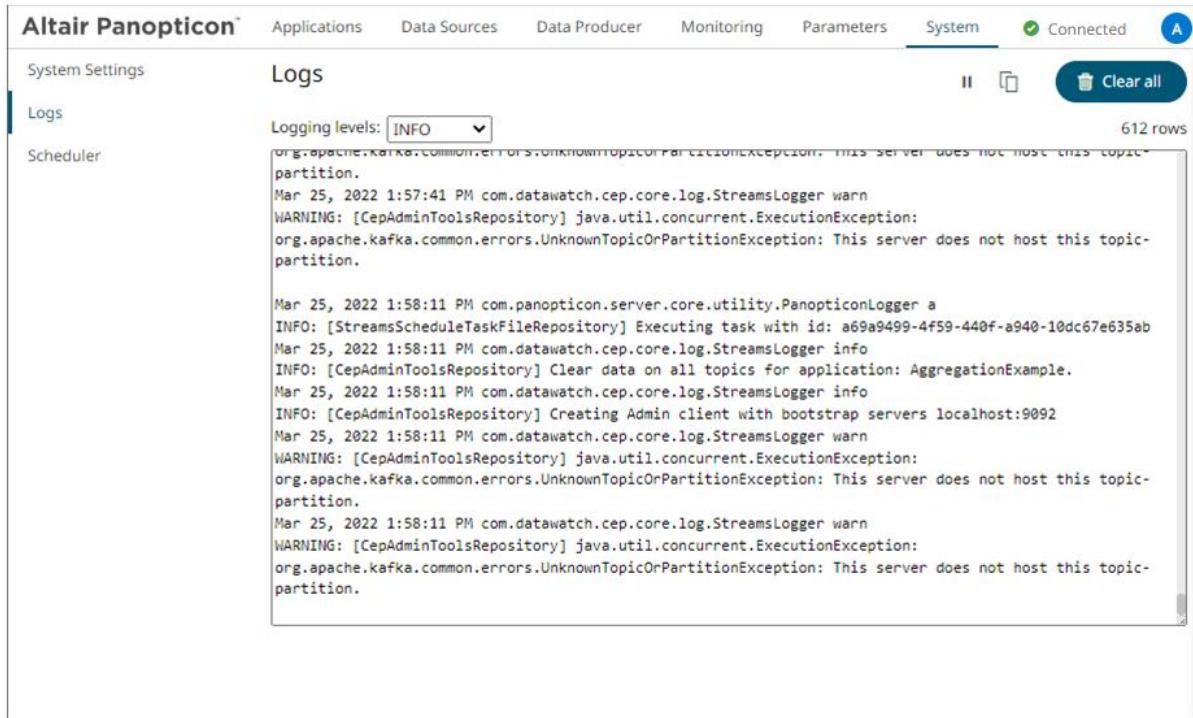
View the latest 300 rows of a *Logging Level* on the *Logs* page:

- FINEST (lowest level)
- FINER

- FINE
- CONFIG
- INFO (default level)
- WARNING
- SEVERE (highest level)

Steps:

- Under the **System** tab, click **Logs**. Initially, the default level (**INFO**) logs are displayed.



The screenshot shows the Altair Panopticon interface with the 'Logs' tab selected. The left sidebar has 'System Settings' and 'Logs' sections. The 'Logs' section is active, showing a dropdown menu for 'Logging levels' set to 'INFO'. The main area displays log entries from March 25, 2022, at 1:58:11 PM. The logs include several 'WARNING' and 'INFO' messages related to Kafka and Datawatch tasks, with one specific warning about a topic partition not being hosted by the server.

```

org.apache.kafka.common.errors.UnknownTopicOrPartitionException: This server does not host this topic-partition.
Mar 25, 2022 1:58:41 PM com.datawatch.cep.core.log.StreamsLogger warn
WARNING: [CepAdminToolsRepository] java.util.concurrent.ExecutionException:
org.apache.kafka.common.errors.UnknownTopicOrPartitionException: This server does not host this topic-partition.

Mar 25, 2022 1:58:11 PM com.panopticon.server.core.utility.PanopticonLogger a
INFO: [StreamsScheduleTaskFileRepository] Executing task with id: a69a9499-4f59-440f-a940-10dc67e635ab
Mar 25, 2022 1:58:11 PM com.datawatch.cep.core.log.StreamsLogger info
INFO: [CepAdminToolsRepository] Clear data on all topics for application: AggregationExample.
Mar 25, 2022 1:58:11 PM com.datawatch.cep.core.log.StreamsLogger info
INFO: [CepAdminToolsRepository] Creating Admin client with bootstrap servers localhost:9092
Mar 25, 2022 1:58:11 PM com.datawatch.cep.core.log.StreamsLogger warn
WARNING: [CepAdminToolsRepository] java.util.concurrent.ExecutionException:
org.apache.kafka.common.errors.UnknownTopicOrPartitionException: This server does not host this topic-partition.
Mar 25, 2022 1:58:11 PM com.datawatch.cep.core.log.StreamsLogger warn
WARNING: [CepAdminToolsRepository] java.util.concurrent.ExecutionException:
org.apache.kafka.common.errors.UnknownTopicOrPartitionException: This server does not host this topic-partition.

```

- Select another *Logging Level* in the drop-down.

For example: **FINEST**

The latest 300 rows of the selected log level or higher are fetched.

3. You can also click any of the following buttons:

- to pause the logging, it changes to

-  to resume the logging
-  to copy log to clipboard
-  **Clear all** to clear the logs

Setting the Logging Level

Changes to the logging level can be made by altering the value of `logger.level.file` in the [Streams.properties](#) file. The server will not log messages with a lower/finer level than this value. The separate logging configuration still applies to route whatever messages that the server does log. This means that if you have set the file handler's level to **INFO** in the configuration, setting the property to **FINE** has no effect.

The default value of the property is set to **INFO**. At this level, most information needed for troubleshooting is logged, including many data queries, timing, and parameters. With a lower/finer level performance will be affected due to the amount of information logged.

Setting the Server Metrics Publisher

The server performance metrics can be used to report, monitor, and configure the server's health and limits. The collected metrics may include the following information:

- Long polling, WebSocket, and total number of connections
- CPU loading percentage
- Maximum, size, and used Heap Bytes
- Subscription alerts, users, and total
- Number of parallel data loading and live threads
- Average data load time or refresh rate

You can configure the following properties in the [Streams.properties](#) in the AppData folder or `C:\streamsserverdata`:

Property	Server Metrics
Attribute	<code>metrics.authorization.level</code>
Description	Specifies the required authorization level to get server metrics. Available values are ANONYMOUS, VIEWER, DESIGNER, ADMINISTRATOR . NOTE: This property is case sensitive.
Default Value	ADMINISTRATOR
Property	Server Metrics
Attribute	<code>metrics.collection.rate</code>
Description	Specifies the rate at which metrics are collected in milliseconds.
Default Value	1000

Property	Server Metrics
Attribute	metrics.file.flush.rate
Description	Specifies how often metrics should be saved to disk in milliseconds. Only used if the metrics.publisher.type is set to FILE .
Default Value	10000
Property	Server Metrics
Attribute	metrics.memory.queue.size
Description	Specifies how many metric entries are stored in memory. When the number of metrics goes above the specified value, the oldest value is removed to make room for the newest one (FIFO). Only used if the metrics.publisher.type is set to MEMORY .
Default Value	100
Property	Server Metrics
Attribute	metrics.publisher.configuration
Description	Specifies the id for which metric publisher configuration to use.
Default Value	
Property	Server Metrics
Attribute	metrics.publisher.type
Description	Specifies the current metric publisher that is used. Available values are NONE, MEMORY, FILE, EMAIL, INFLUX_DB, JDBC, KAFKA, KDB, MQTT, REST, TEXT .
Default Value	MEMORY

SCHEDULING TASK TO CLEAR TOPIC DATA

Panopticon Streams supports scheduling of tasks such as daily deletion of application topics.

Steps:

1. Under the **System** page, click **Scheduler**.

2. Click **New Task**. The *New Task* pane displays.

Altair Panopticon

System Settings Applications Data Sources Data Producer Monitoring Parameters System Connected A

Logs Scheduler Activated

Trigger Period CRON

Interval (sec) 3600

Type Clear Topic Data

Description

Application

Run Save

- Enter the *Name* of the task. Ensure the name is unique.
- Tap the *Activated* slider to turn it on.
- Select the *Trigger*. You can either select:
 - Period** then enter the *Interval* (in seconds), or

Trigger Period CRON

Interval (sec) 3600

- CRON** then enter a *CRON Expression* on the format:
sec mins hours dayofmonth month dayofweek (e.g., 09 02 18 ? * MON-FRI)

Trigger Period CRON

CRON Expression

- Select the task *Type*: **Clear Topic Data**.
- Enter the *Description* of the task.
- Select the *Application* in the drop-down list. These are the applications available on the **Applications** tab.

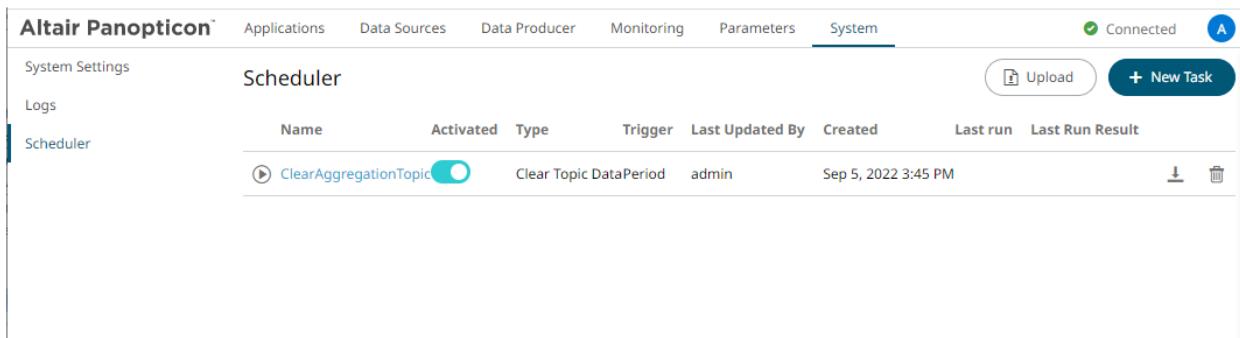
Save

9. Click

Run

- Once saved, you can opt to click **Run** to manually run the task.

- Click  to go back to the Tasks pane. The new task is added in the list.



A task displays the following columns: *Name*, *Activated*, *Type*, *Trigger*, *Last Updated By*, *Created*, *Last Run*, and *Last Run Result*.

Modify the sorting of the list by clicking the  or  button of any of these columns. The icon beside the column that was used for the sorting will indicate if it was in an ascending or descending order.

Tasks can also be:

- manually started

Instead of waiting for the set Period interval or CRON Expression, you can manually execute the task by clicking .

The *Last Run* and *Last Run Result* (**Success** or **Failed**) are displayed. For failed results, you can hover on the tooltip to view the error.

- [modified](#)
- deleted

Click  of a task. A confirmation message displays. Click .

Modify a Scheduled Task

Steps:

- On the *Scheduler* page under the **System** tab, click the link of a task to modify.
The properties of the task are displayed.
- Apply the desired changes.

- Click .

[7] AUTHORIZATION

NOTE

Starting with version 2020.0, mapping of administrators through `Administrators.txt` and `AdministratorGroups.txt` is no longer supported. The property `access.administrator.groups` should be used instead.

If the customer's authentication method relied to the use of the `Administrators.txt` or `AdministratorGroups.txt` file, they can still do so by additionally using the [`tomcat-users.xml`](#) to replicate the usage of these administrator text files.

For example, in the `tomcat-users.xml`, they can assign groups from the administrator text files to specific users like this:

```
<user username="admin" password="admin" roles="role1,otherRole"/>
<user username="admin2" password="admin2" roles="role2"/>
```

Then in the [`Streams.properties`](#) file, use the `access.administrator.groups` property to map the admins (i.e., admin and admin2) to the administrator groups by adding their roles:

```
access.administrator.groups=role1,role2
```

SECURE ACCESS

Panopticon [applications](#) and [data sources](#) published to the folders or subfolders in Panopticon Streams Server can be secured by granting allowed or denied permissions.

Creating Folders

A user with an Administrator or Designer role can create folders.

NOTE

Users that log on with a Designer role will have their own personal folder created and displayed on the Applications and Data Sources tabs (e.g., `~designer`).

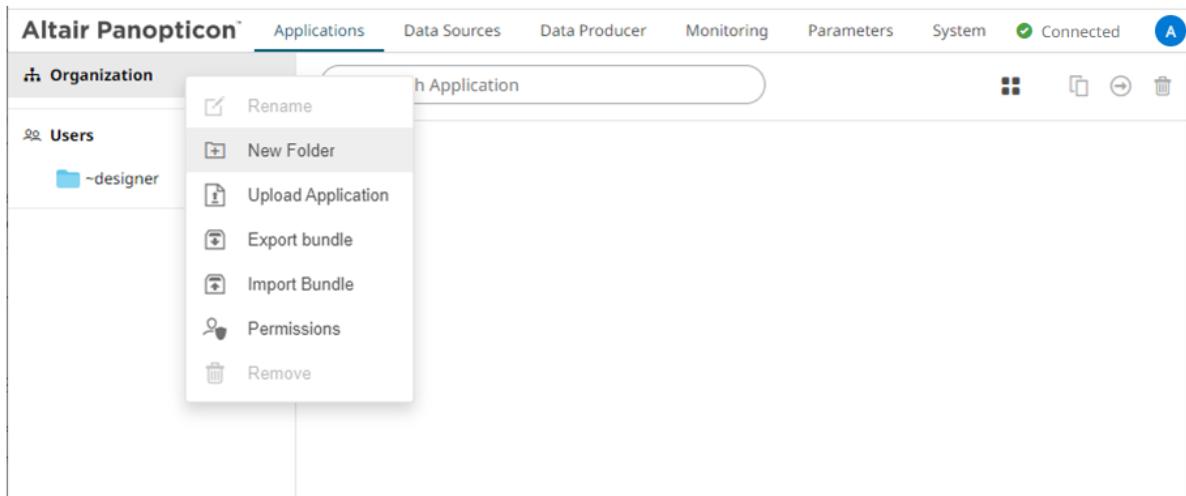
The personal folders:

- are displayed and can be accessed for users with an Administrator or Designer role.
- are where Designers can create applications and data sources. For more information, see [Creating a New Application](#) or [Creating a Data Source](#) sections.

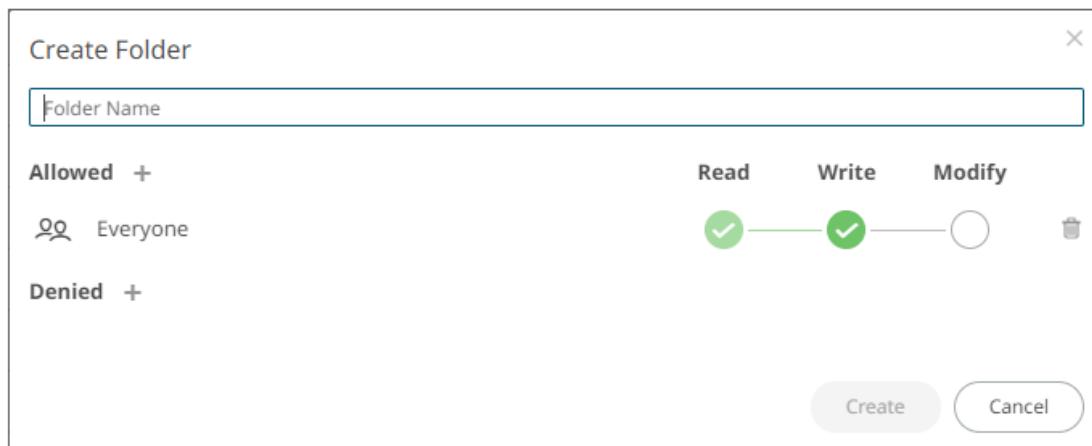
Creating Folders on the Applications Tab

Steps:

1. On the **Applications** tab, right-click on the topmost folder (**Organization**) or the *Applications* pane and select **New Folder**.



The *Create Folder* dialog displays.



NOTE

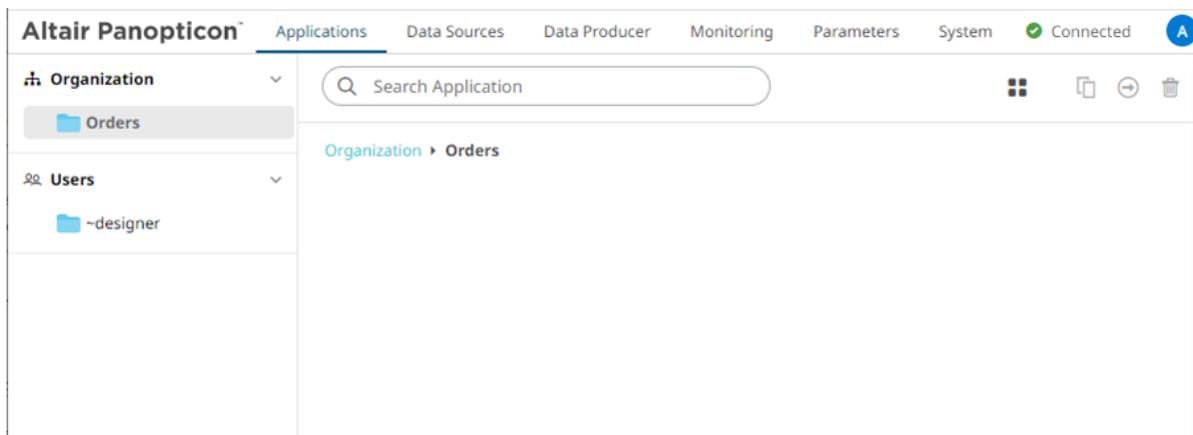
- Everyone is available in the *Allowed* section by default.
- Removing the Everyone group will mean that the folder and its subfolders will not be available for public access.

2. Enter a *Folder Name*.
3. Proceed to defining the Authorization to *Allowed* or *Denied* groups and users.

Create

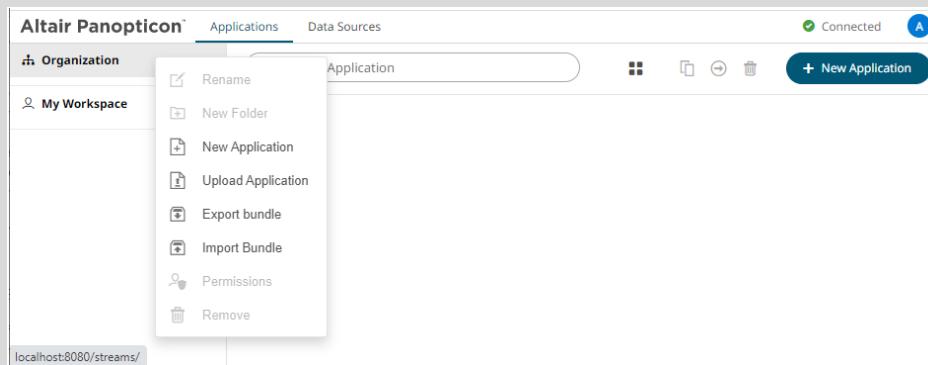
4. Click **Create**.

The new folder is displayed on the expanded *Folder* hierarchy list and on the *Root Folder* list.



NOTE

- A user with a Designer role is not allowed to create a folder on the root folder.

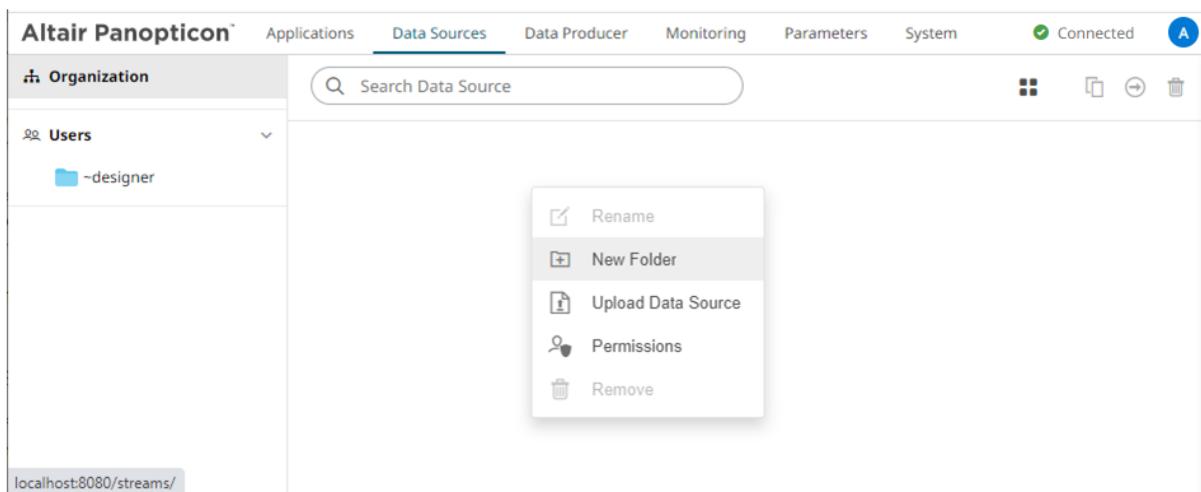


- Folders and subfolders can be deleted as long as they do not contain applications or data sources.
- The folders and subfolders on the **Applications** tab will also be available on the **Data Sources** tab.

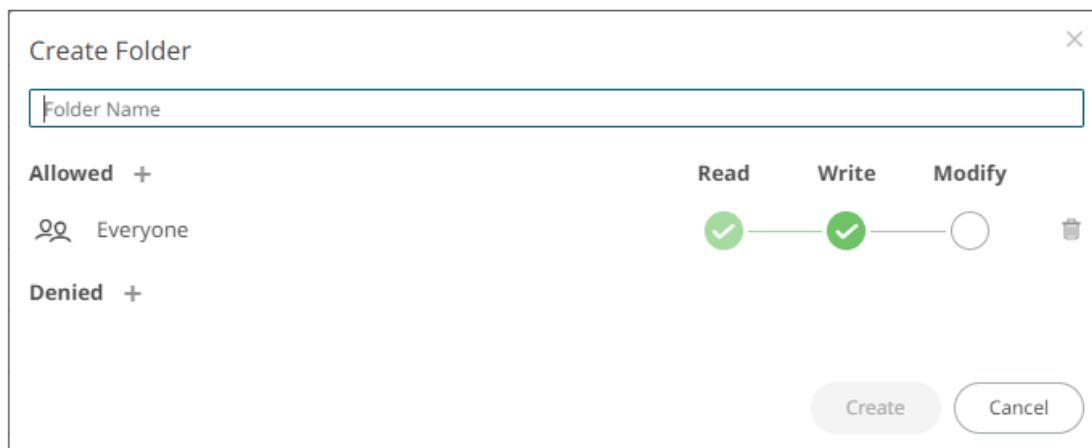
Creating Folders on the Data Sources Tab

Steps:

1. On the **Data Sources** tab, right-click on the topmost folder or the *Data Sources* pane and select **New Folder**.



The *Create Folder* dialog displays.



NOTE

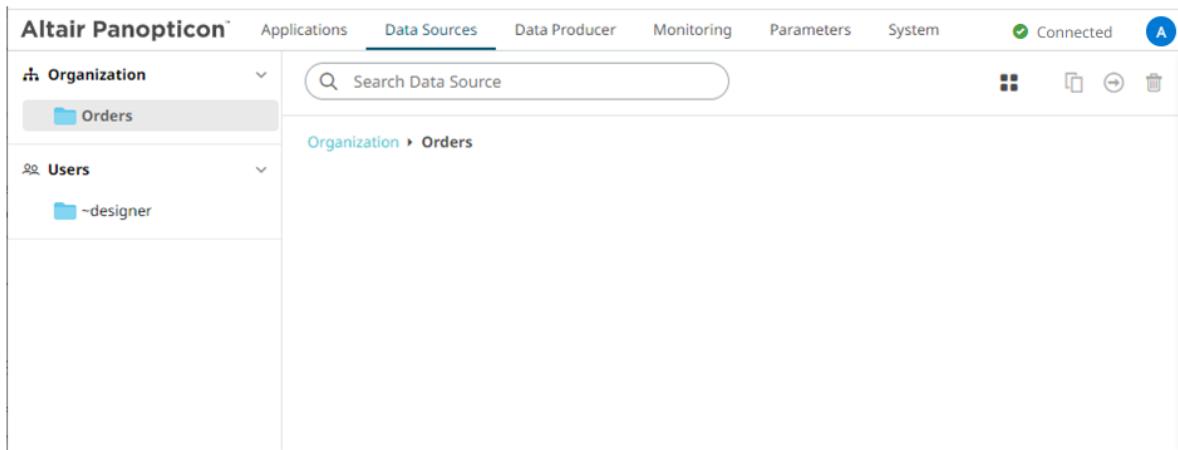
- Everyone is available in the *Allowed* section by default.
- Removing the Everyone group will mean that the folder and its subfolders will not be available for public access.

- Enter a *Folder Name*.
- Proceed to defining the Authorization to *Allowed* or *Denied* groups and users.

Create

- Click **Create**.

The new folder is displayed on the expanded *Folder* hierarchy list and on the *Root Folder* list.

**NOTE**

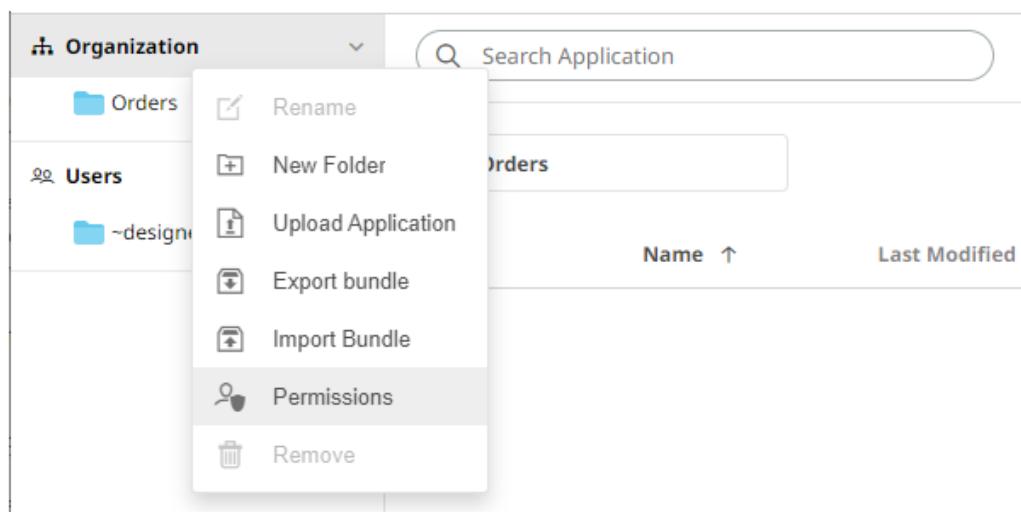
Folders and subfolders can be deleted as long as they do not contain applications or data sources.

Adding Groups and Users with Allowed Authorization

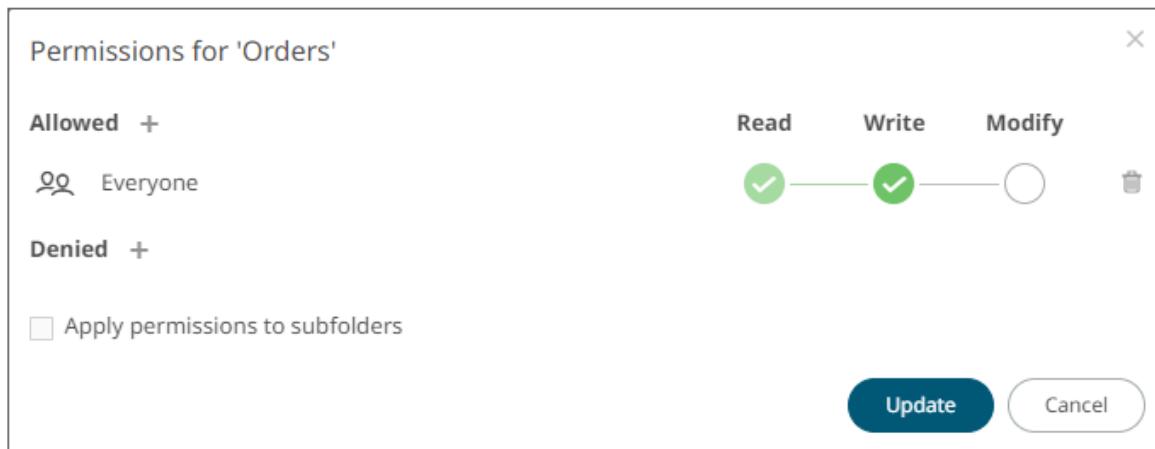
A user with an Administrator or Designer role can grant permission for users or groups access to application or data source folder or subfolder.

Steps:

1. Right-click on a folder (except the root folder) and select **Permissions** in the context menu.

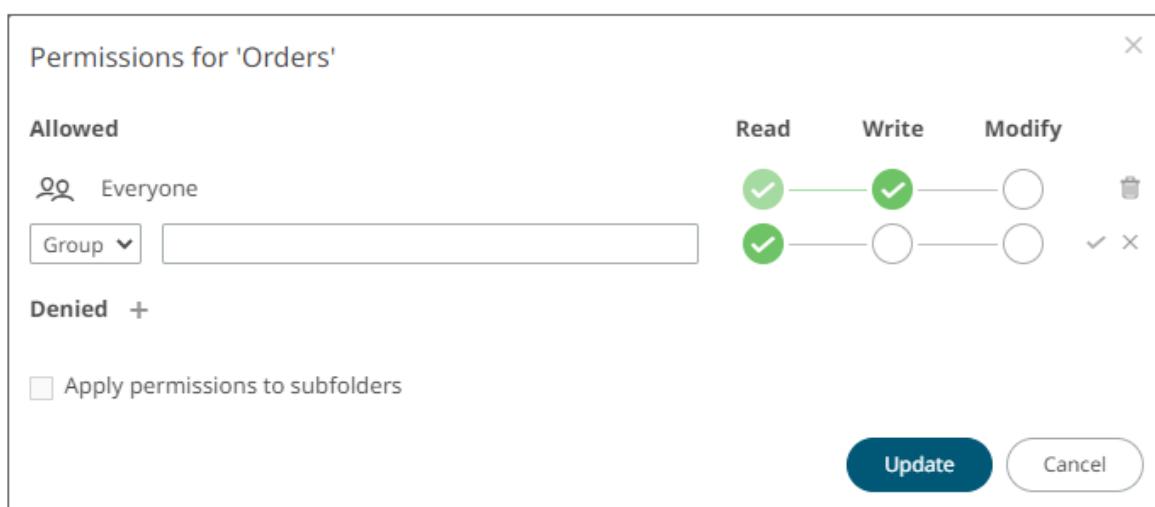


The *Permissions* dialog displays.



2. Under the *Allowed* section, click the **Add**  icon.

A new *User/Group Allowed* section is displayed.



3. Select **User** or **Group** to be given permission in the drop-down list.



4. Enter the user or group *Name*.

5. Select the permission level that will be granted to the user or group:

- **READ**

Permission to read the folder.

- **READ + WRITE**

Permission to write to the folder and read.

- **MODIFY + WRITE + READ**

Permission to read, modify, and write to the folder as well as create subfolders.

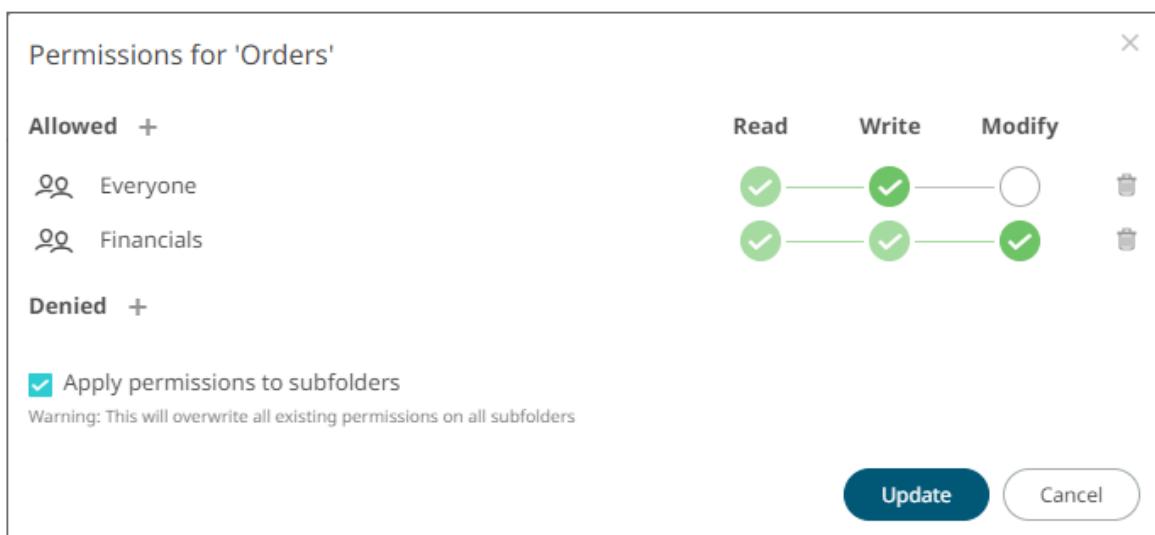


6. Click . The user or group is added under the *Allowed* list.



7. You can either:

- select the **Apply Permissions to Subfolders** check box



This means the permissions that will be used on all of the subfolders will be fetched from the root folder.

NOTE

The **Apply Permissions to Subfolders** check box is only enabled when there is an [existing subfolder](#).

- leave the **Apply Permissions to Subfolders** box unchecked and [modify the permission properties](#) of the subfolders

8. Click **Update** to save the changes.

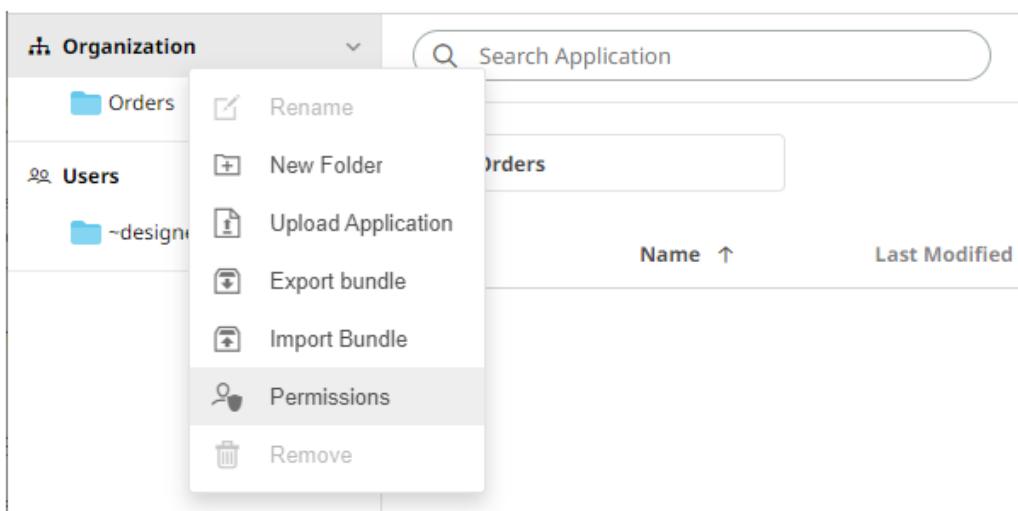
NOTE

You can copy the user names in the *Permissions* dialog by highlighting the text then right-clicking, and selecting **Copy** in the context menu.

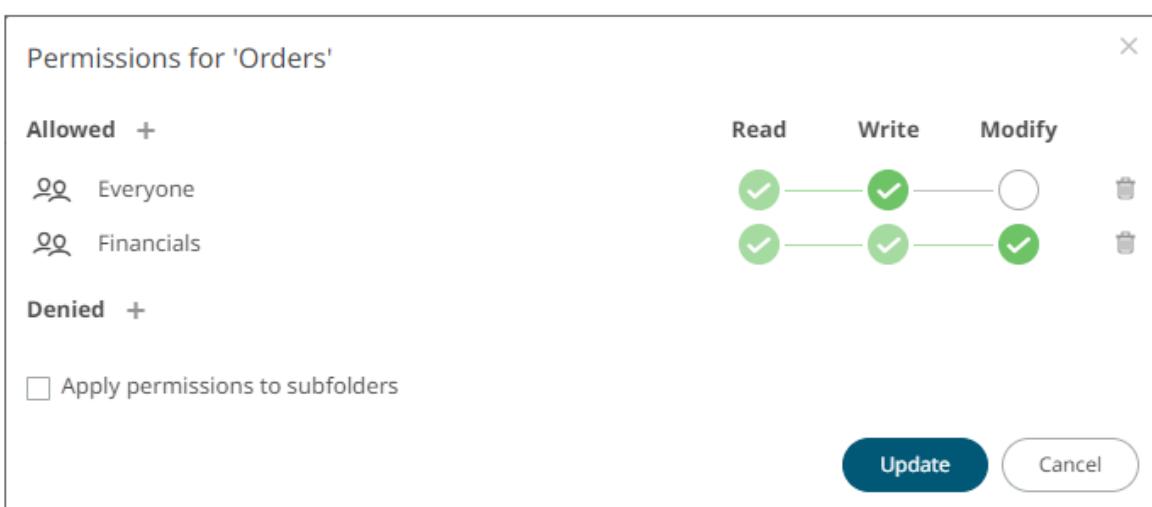
Adding Groups and Users with Denied Access

Steps:

1. Right-click on a folder and select **Permissions** in the context menu.



The *Permissions* dialog displays.



2. Under the *Denied* section, click the **Add**  icon.

A new *User/Group Denied* section is displayed.

Permissions for 'Orders'

Allowed +	Read	Write	Modify
Everyone	✓	✓	○
Financials	✓	✓	✓

Denied

Group

Read	Write	Modify
○	✗	✗

Apply permissions to subfolders

Update **Cancel**

3. Select **User** or **Group** that will be given denied permission in the drop-down list.
4. Enter the user or group *Name*.
5. Select the denied permission level that will be granted to the user or group:
 - **MODIFY**
Prevent user or group to modify and create subfolders.
 - **WRITE + MODIFY**
Prevent user or group to modify and write to the folder.
 - **READ + WRITE + MODIFY**
Prevent user or group to modify and create subfolders, modify and write to the folder, as well as read the folder.

Denied

User

John

✗

✓ ✗

6. Click . The user or group is added under the *Denied* list.

Denied +

John

✗

✗

Repeat until all of the users with denied access are added.

7. You can either:
 - select the **Apply Permissions to Subfolders** check box, or



This means the permissions that will be used on all of the subfolders will be fetched from the root folder.

NOTE

The **Apply Permissions to Subfolders** check box is only enabled when there is an [existing subfolder](#).

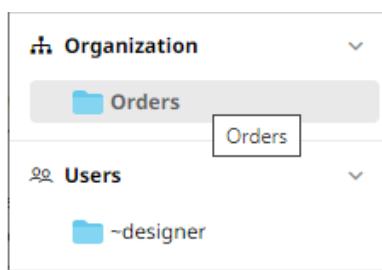
- leave the **Apply Permissions to Subfolders** box unchecked and [modify the permission properties](#) of the subfolders.

8. Click **Update** to save the changes.

Creating Subfolders

Steps:

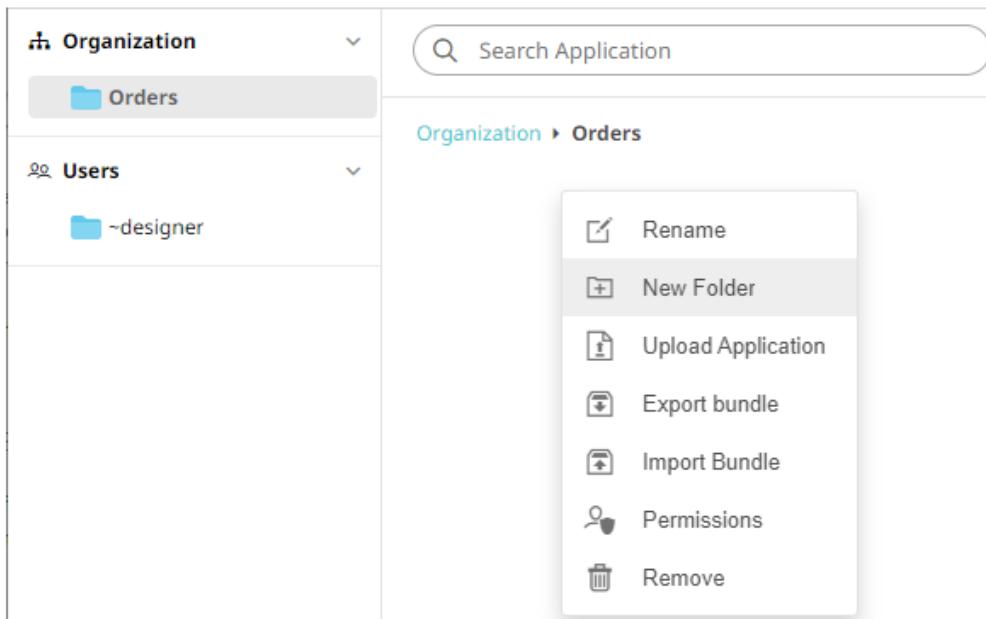
- To create subfolders, you can either click a folder:
 - on the expanded *Folder* hierarchy list, or



- on the Root folders list.

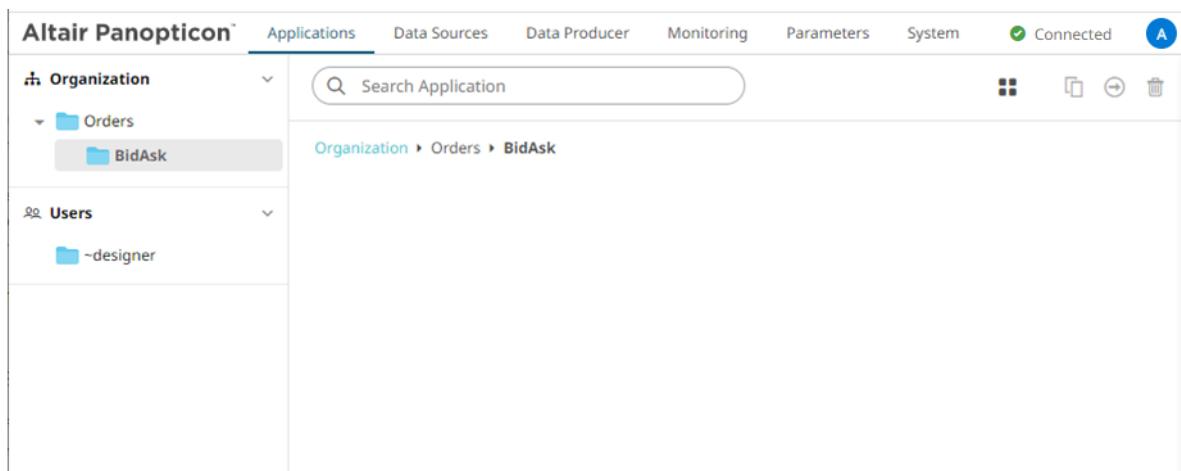
The *Folders* page is displayed.

- Right-click on the folder or on the *Applications* pane or *Data Sources* pane and select **New Folder**.

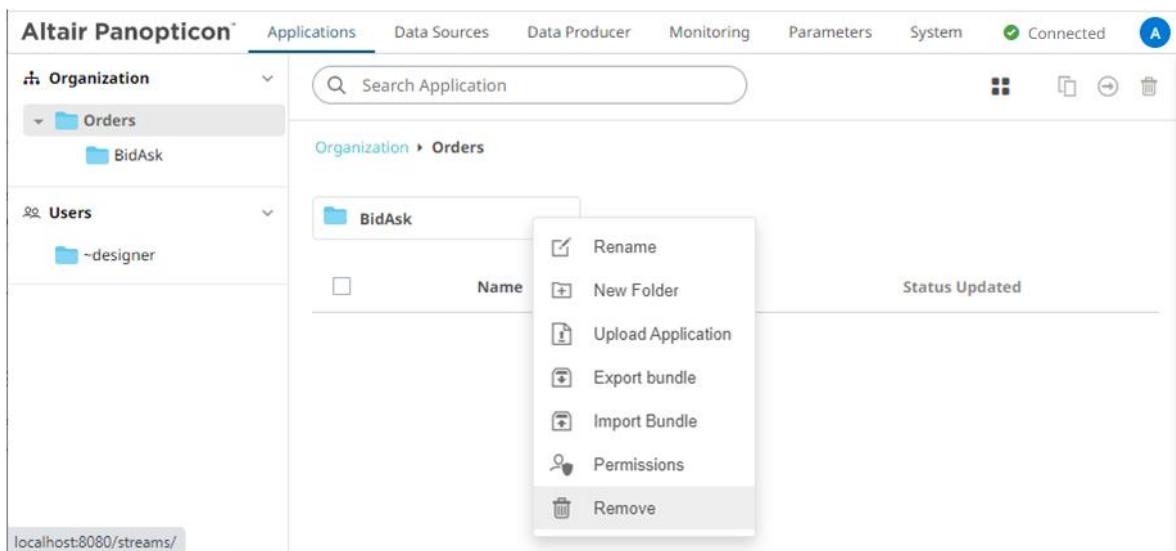


Refer to [Creating Folders](#) for the steps in creating the subfolders. Also, [Adding Groups and Users with Allowed Authorization](#) and [Adding Groups and Users with Denied Access](#) for more information on adding users and groups with allowed or denied authorization.

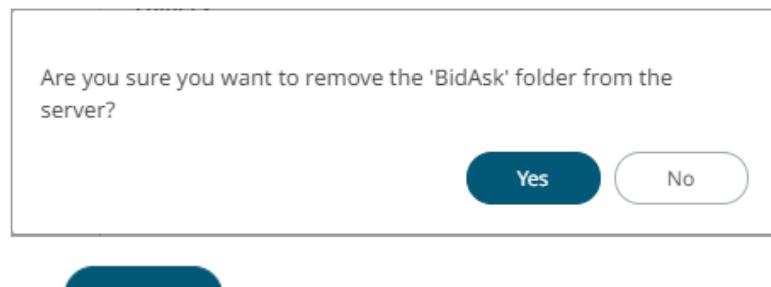
The subfolder is added.



3. You can also opt to delete a subfolder by right-clicking on the folder and selecting **Remove** in the context menu as long as it does not contain applications or data sources.



A confirmation message displays.

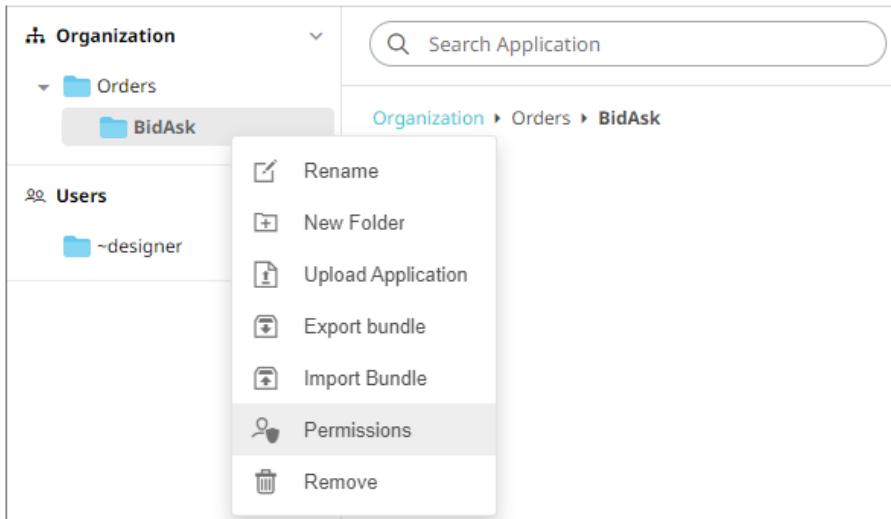


Click **Yes**

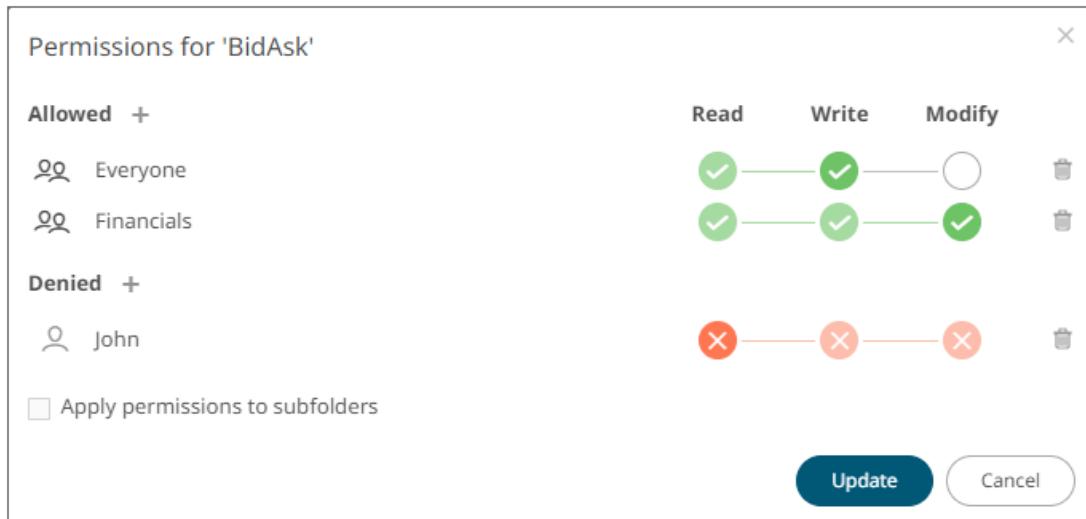
Updating Folder or Subfolder Properties

Steps:

1. To update folder properties, click a folder or a subfolder.
2. Right-click on the folder or subfolder and select **Permissions**.



The corresponding *Permissions* dialog displays.



4. Make the necessary changes such as new folder name, add or delete users and groups.
5. You can either:
 - select the **Apply Permissions to Subfolders** check box
This means the permissions that will be used on all of the subfolders will be fetched from the root folder.
 - leave the **Apply Permissions to Subfolders** box unchecked and modify the permission properties of the subfolders

NOTE

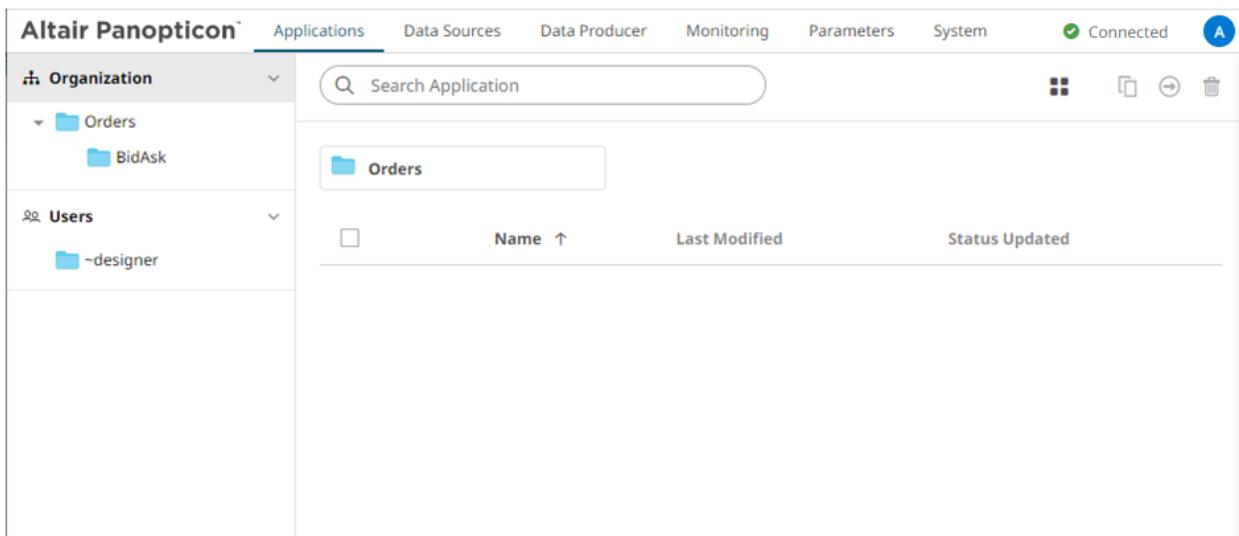
The **Apply Permissions to Subfolders** check box is not enabled when defining the permissions for a subfolder.

6. Click **Update** to save the changes.

[8] MANAGING APPLICATIONS

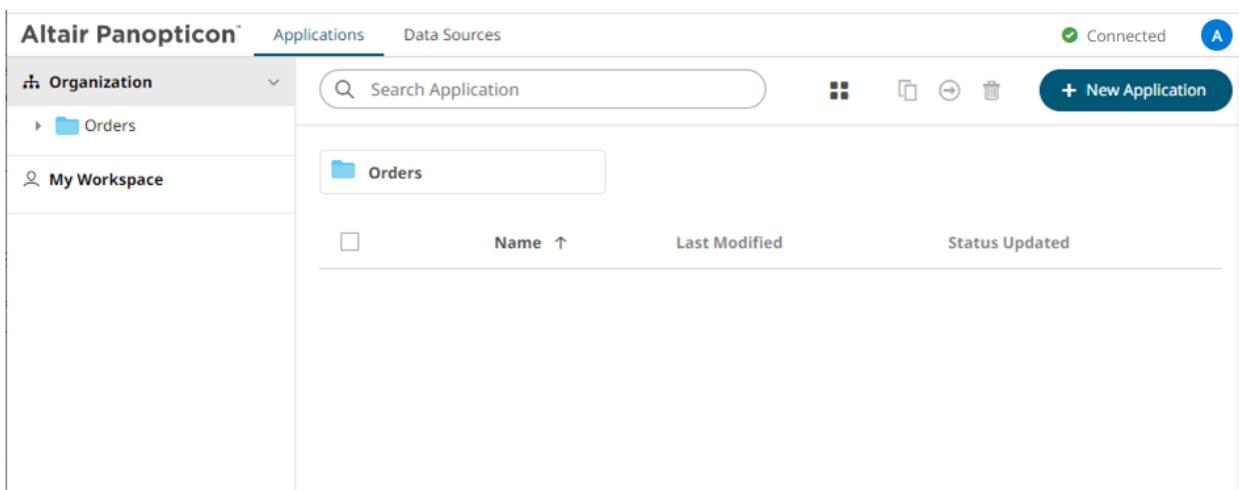
On the **Applications** tab, users with Administrator or Designer role can:

- ❑ [import](#) and [export](#) application bundles
- ❑ [upload](#) applications
- ❑ [rename](#) applications
- ❑ view [topic](#) or [data source](#) usage
- ❑ [move](#) or [copy](#) applications to folders or subfolders to which the user has permission
- ❑ [download](#) applications
- ❑ [remove](#) applications
- ❑ publish/[republish](#) applications to folders to which the user has permission



The screenshot shows the Altair Panopticon interface with the 'Applications' tab selected. The left sidebar shows a tree view of 'Organization' and 'Users'. Under 'Organization', there is a 'Orders' folder containing a 'BidAsk' subfolder. The main panel displays a list of applications. A single application, 'Orders', is listed in the 'Orders' folder. The list includes columns for a checkbox, 'Name' (sorted by 'Last Modified'), and 'Status Updated'. The 'Status Updated' column is empty for this entry. The top right of the interface shows a 'Connected' status with a green checkmark and a blue 'A' icon.

To [create a new application](#), a user must have a Designer role.



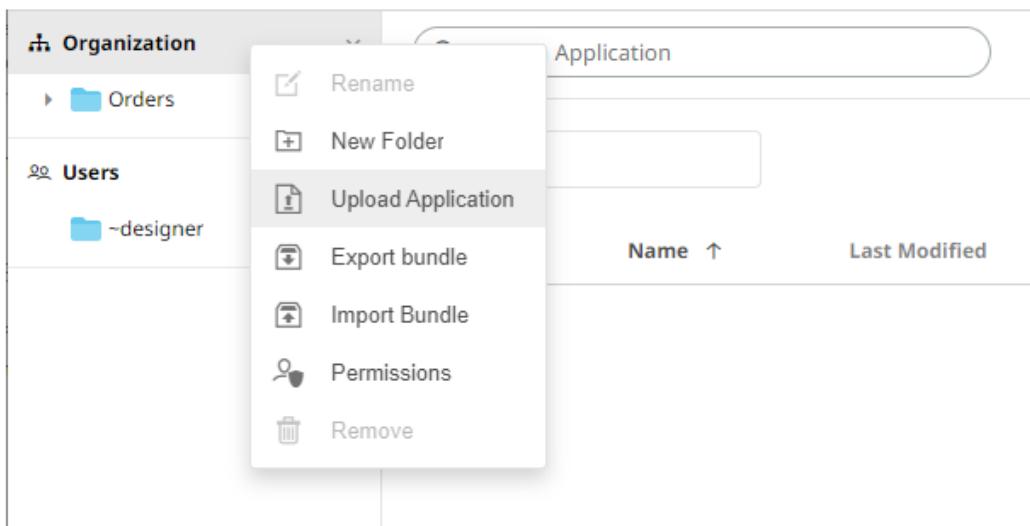
The screenshot shows the Altair Panopticon interface with the 'Applications' tab selected. The left sidebar shows a tree view of 'Organization' and 'My Workspace'. Under 'Organization', there is a 'Orders' folder. Under 'My Workspace', there is also a 'Orders' folder. The main panel displays a list of applications. A single application, 'Orders', is listed in the 'Orders' folder under 'My Workspace'. The list includes columns for a checkbox, 'Name' (sorted by 'Last Modified'), and 'Status Updated'. The 'Status Updated' column is empty for this entry. The top right of the interface shows a 'Connected' status with a green checkmark and a blue 'A' icon. A prominent blue button with a white plus sign and the text '+ New Application' is located in the top right corner of the main panel.

UPLOADING APPLICATIONS

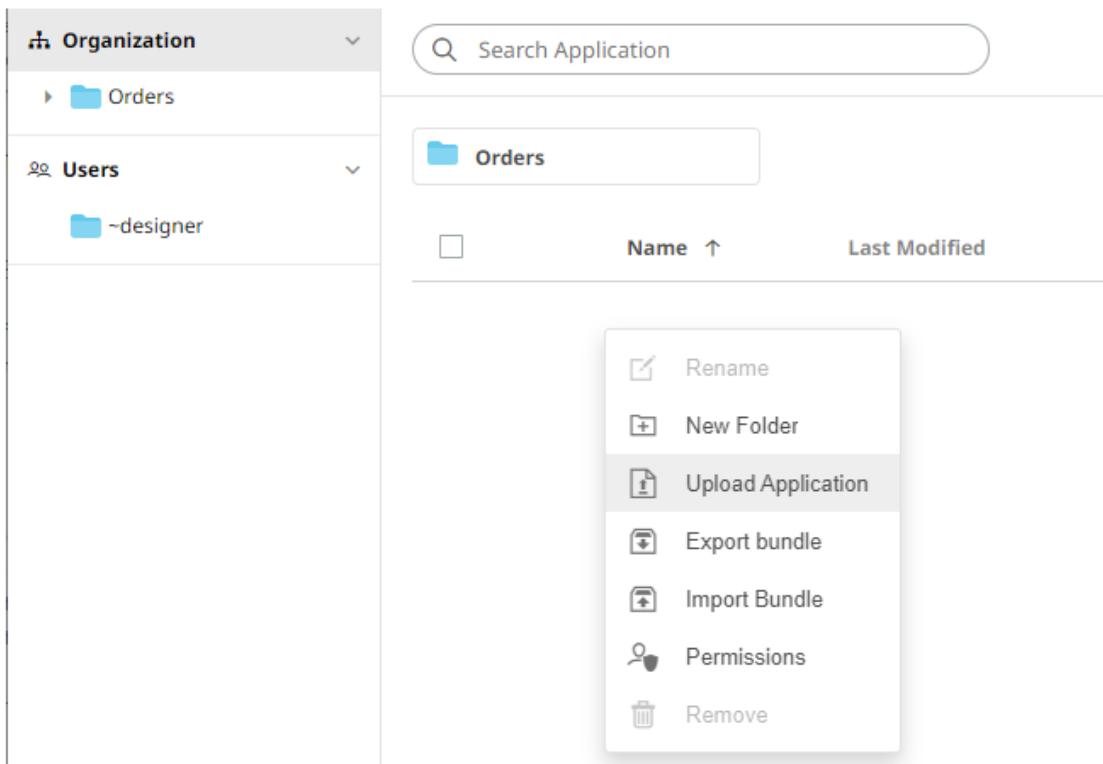
Users with Administrator or Designer role can upload applications to folder or subfolders that they have permission.

Steps:

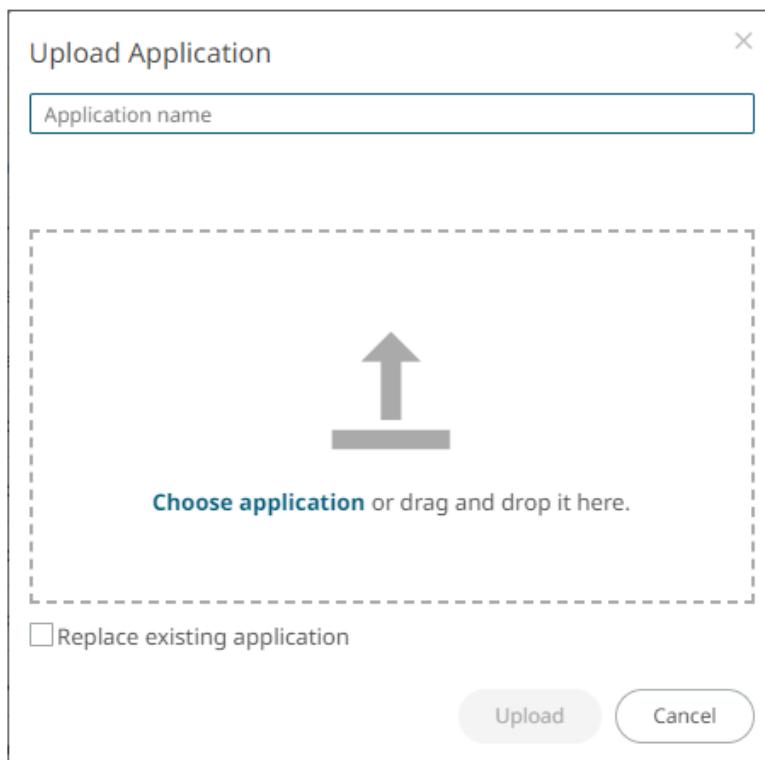
1. To upload applications, you can either right-click a folder or subfolder then select **Upload Application**:
 - on the expanded *Folder* hierarchy list



- on the Root folders list

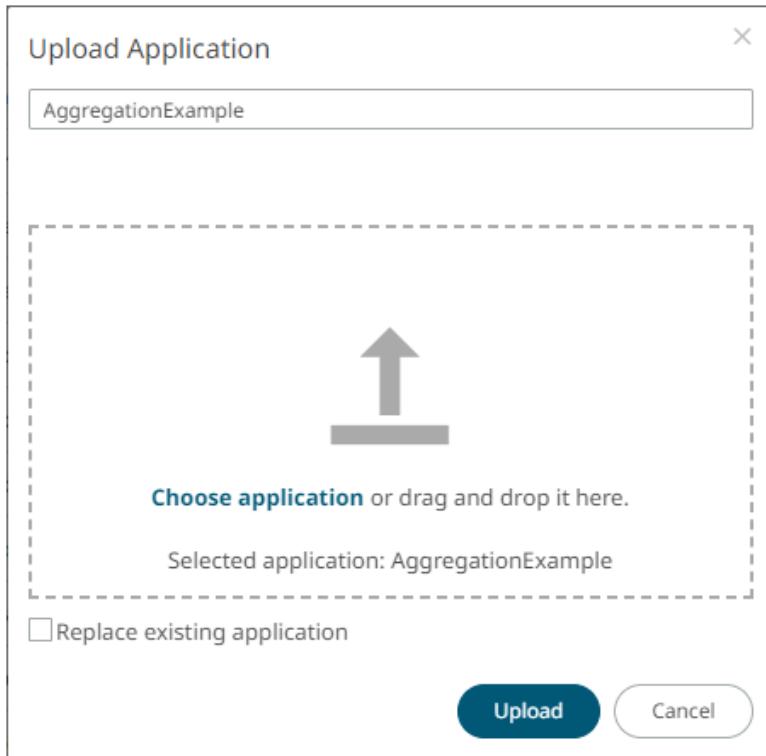


The *Upload Application* dialog displays.



2. To upload an application, you can either:
 - drag it from your desktop and drop in the dialog, or
 - click **Choose Application** and select one in the *Open* dialog that displays.

The name of the application is displayed on the uploaded application area and in the *Name* box.



3. You can opt to rename the application.

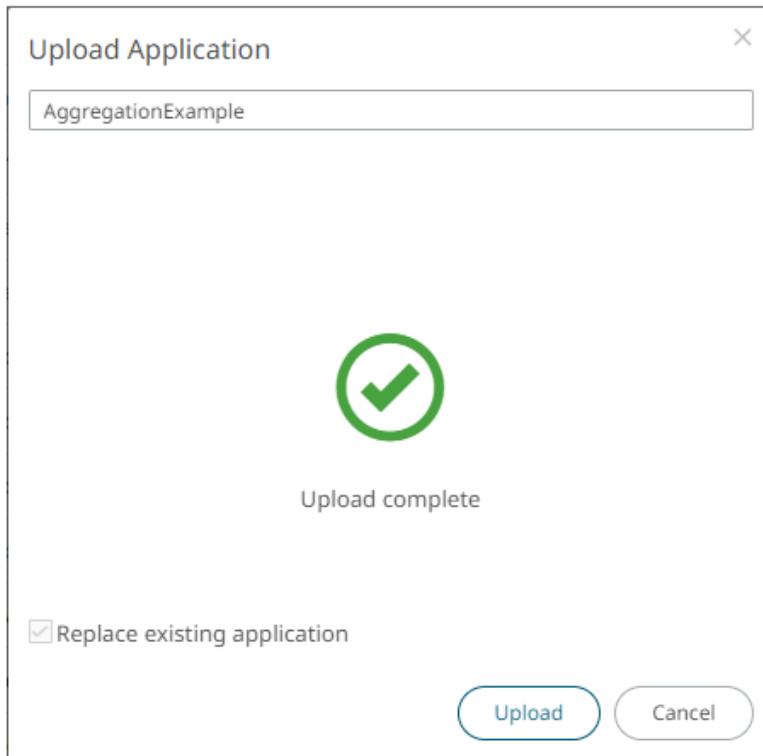
NOTE

The application name must start with a letter (a to Z) or underscore. Also, it can only contain letters (a to Z), numbers (0 to 9), and underscores.

4. To replace an existing application, select the **Replace existing application** check box.

5. Click **Upload**.

You will be notified when the application has been uploaded.



The application is added and displayed on the **Applications** tab.

	Name	Last Modified	Status Updated
	AggregationExample	May 24, 2022 6:11 PM	

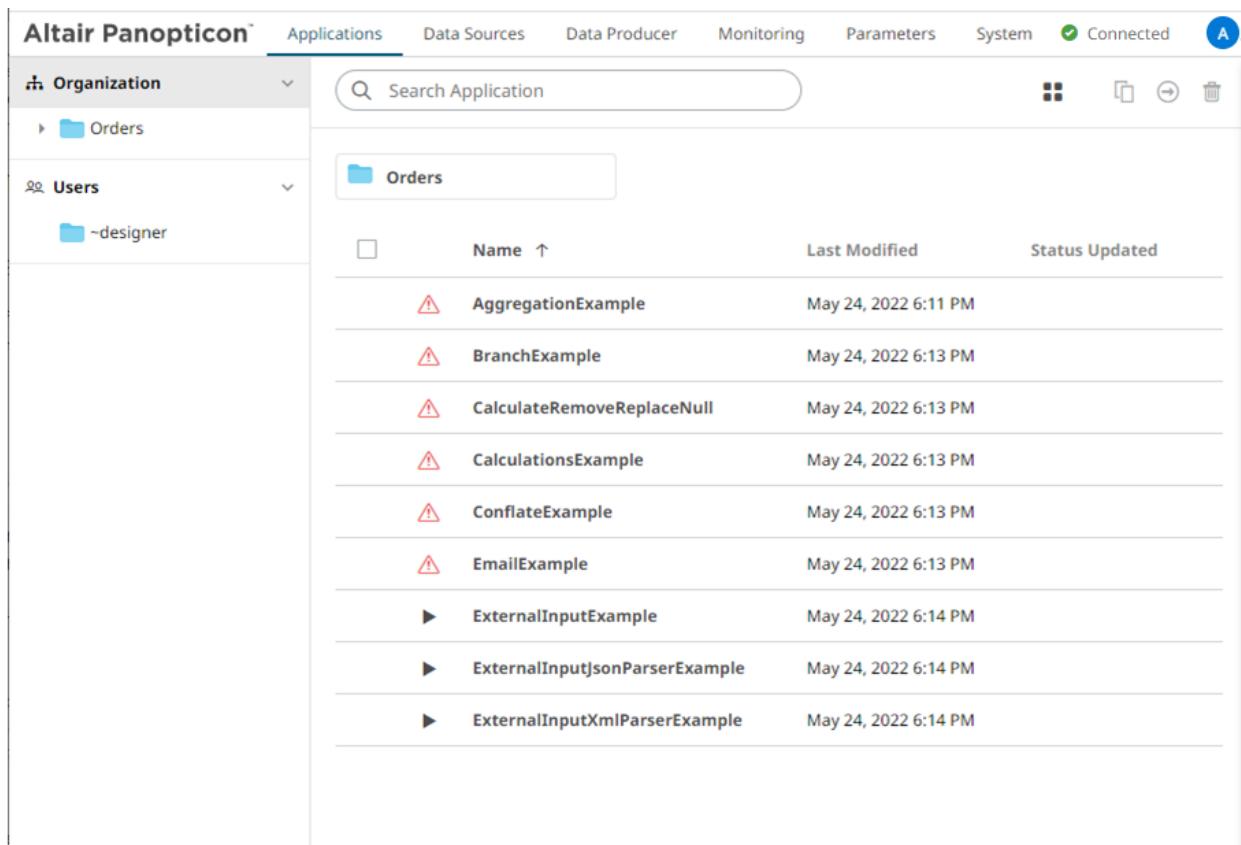
NOTE

A icon displays before the application name. This means the required data source is not available. Refer to [Uploading Data Sources](#) for more information.

When the data source is available, the icon changes to .

Folders and Applications Display View

Folders and applications can be displayed either on a *List* or *Grid View*. By default, the applications are displayed in a *List View*.



The screenshot shows the Altair Panopticon interface with the 'Applications' tab selected. On the left, there is a sidebar with 'Organization' and 'Users' sections, each containing a folder icon and a 'Orders' folder. A search bar labeled 'Search Application' is at the top. The main area displays a table of applications under the 'Orders' folder. The table has columns for 'Name', 'Last Modified', and 'Status Updated'. Each application row has a warning icon and a link to its details. The applications listed are: AggregationExample, BranchExample, CalculateRemoveReplaceNull, CalculationsExample, ConflateExample, EmailExample, ExternalInputExample, ExternalInputJsonParserExample, and ExternalInputXmlParserExample. All applications were last modified on May 24, 2022, at 6:13 PM, except for ExternalInputExample which was modified at 6:14 PM.

	Name	Last Modified	Status Updated
⚠	AggregationExample	May 24, 2022 6:11 PM	
⚠	BranchExample	May 24, 2022 6:13 PM	
⚠	CalculateRemoveReplaceNull	May 24, 2022 6:13 PM	
⚠	CalculationsExample	May 24, 2022 6:13 PM	
⚠	ConflateExample	May 24, 2022 6:13 PM	
⚠	EmailExample	May 24, 2022 6:13 PM	
▶	ExternalInputExample	May 24, 2022 6:14 PM	
▶	ExternalInputJsonParserExample	May 24, 2022 6:14 PM	
▶	ExternalInputXmlParserExample	May 24, 2022 6:14 PM	

Click **Grid View** . The folders and applications are displayed as thumbnails.

Altair Panopticon

Applications Data Sources Data Producer Monitoring Parameters System Connected A

Organization

Orders

Users

~designer

Search Application

Name ↑

Folders

Orders 0 applications

Applications

AggregationExample Modified 5 minutes ago

BranchExample Modified 3 minutes ago

CalculateRemoveReplaceNull Modified 3 minutes ago

CalculationsExample Modified 3 minutes ago

ConflateExample Modified 3 minutes ago

EmailExample Modified 2 minutes ago

ExternalInputExample Modified 2 minutes ago

ExternalInputJsonParserExample Modified 2 minutes ago

ExternalInputXmlParserExample Modified 2 minutes ago

Click **List View**  to return to the standard listing.

Altair Panopticon

Applications Data Sources Data Producer Monitoring Parameters System Connected A

Organization

Orders

Users

~designer

Search Application

Name ↑

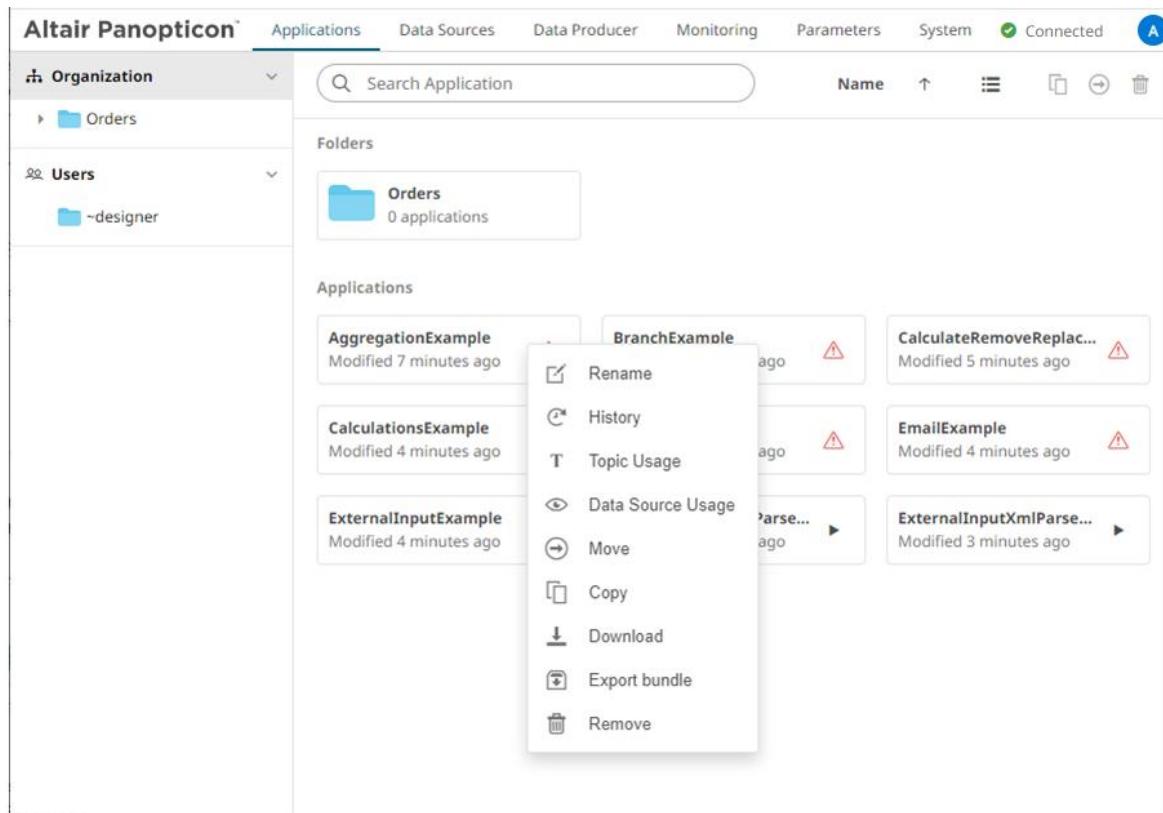
Last Modified

Status Updated

	Name ↑	Last Modified	Status Updated
⚠	AggregationExample	May 24, 2022 6:11 PM	
⚠	BranchExample	May 24, 2022 6:13 PM	
⚠	CalculateRemoveReplaceNull	May 24, 2022 6:13 PM	
⚠	CalculationsExample	May 24, 2022 6:13 PM	
⚠	ConflateExample	May 24, 2022 6:13 PM	
⚠	EmailExample	May 24, 2022 6:13 PM	
▶	ExternalInputExample	May 24, 2022 6:14 PM	
▶	ExternalInputJsonParserExample	May 24, 2022 6:14 PM	
▶	ExternalInputXmlParserExample	May 24, 2022 6:14 PM	

On either display view style:

- clicking on an application title or thumbnail displays the application
- right-clicking on an application displays the context menu

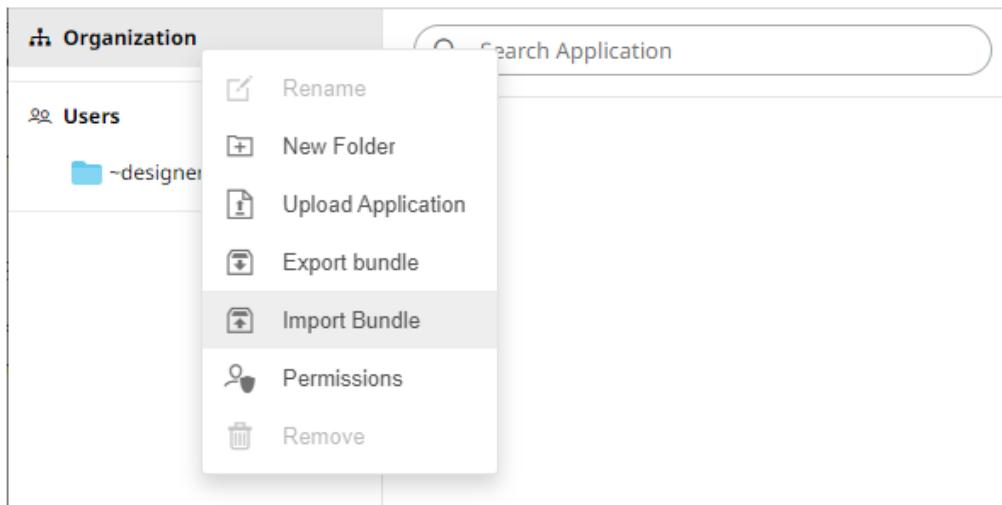


IMPORTING AN APPLICATION BUNDLE

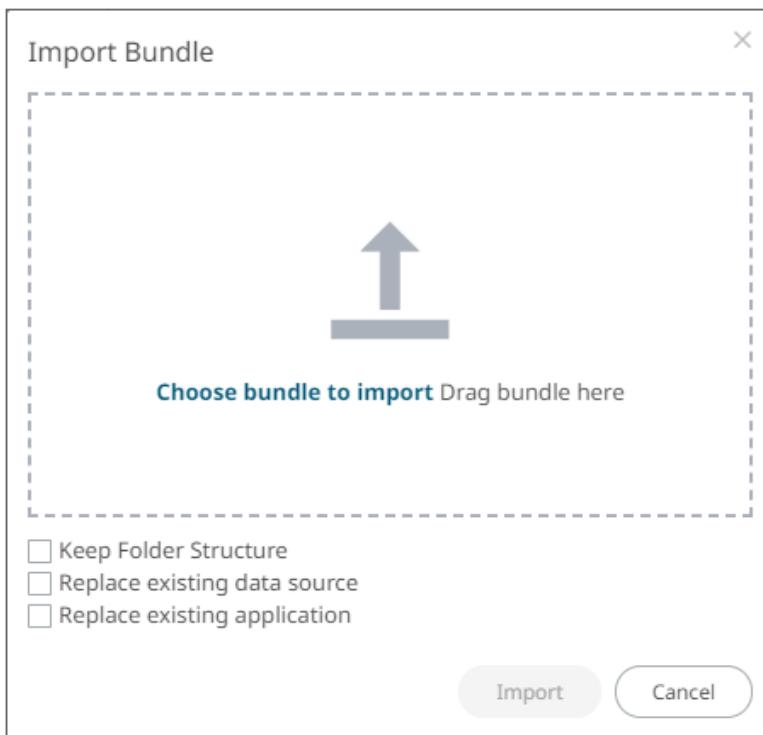
Follow the instructions below to import an application bundle to Panopticon Streams.

Steps:

1. On the **Applications** tab, right-click on a folder and select **Import Bundle** in the context menu.

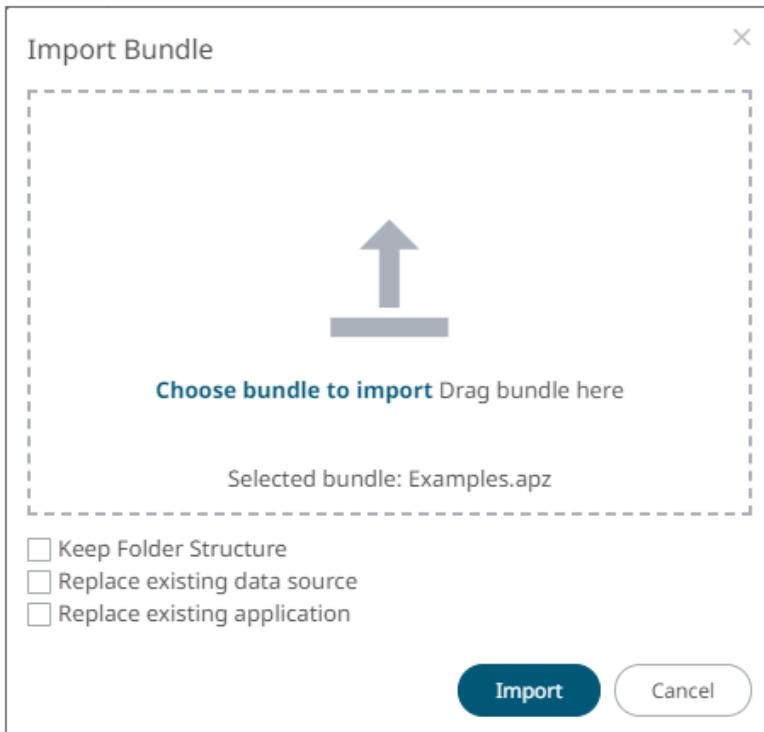


The *Import Bundle* dialog displays.



2. To import the bundle, you can either:
 - drag it from your desktop and drop in the dialog, or
 - click **Choose Bundle to Import** and select one on the *Open* dialog that displays.

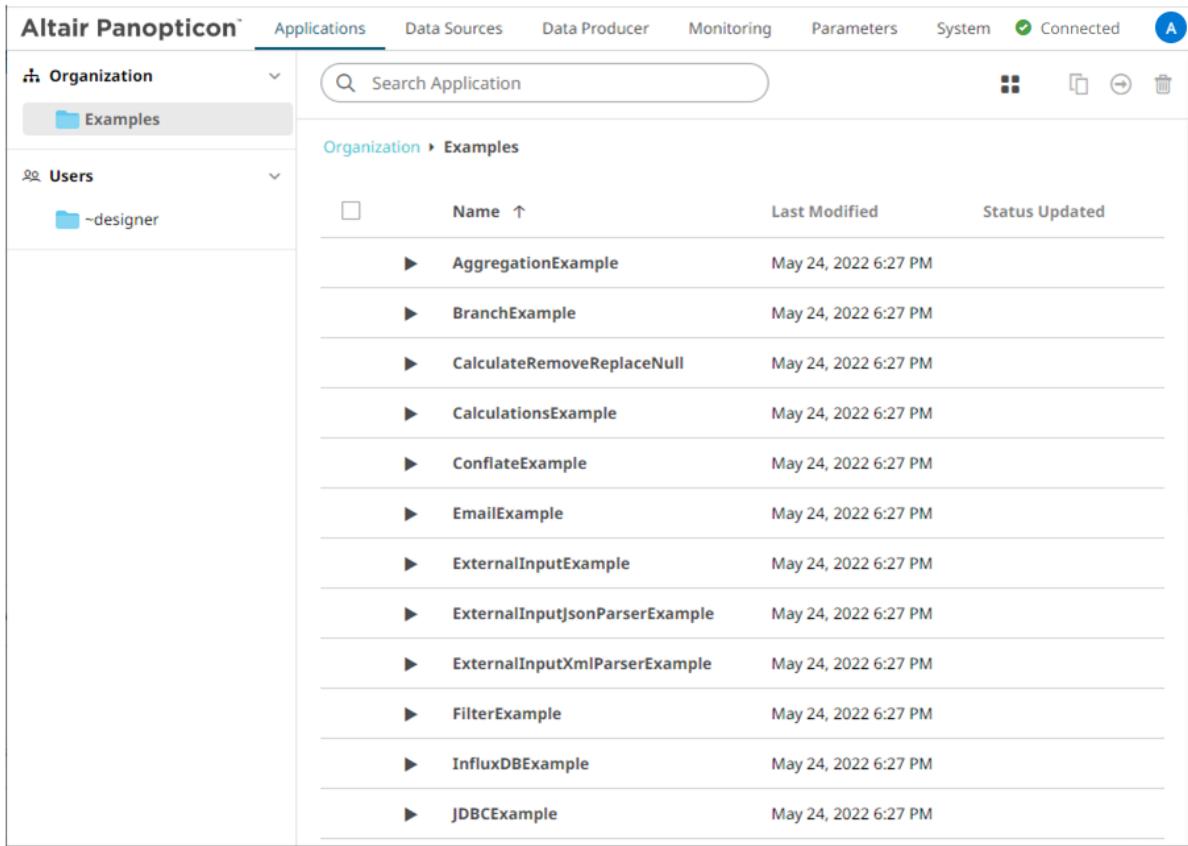
The name of the selected bundle is displayed in the dialog box.



3. Select the **Keep Folder Structure** check box if you opt to maintain the exported folder structure when uploading the bundle. If the folders do not exist on the server, they will be created.
4. To replace an existing data source, select the **Replace existing data source** check box.
5. To replace an existing application, select the **Replace existing application** check box.

6. Click **Import**.

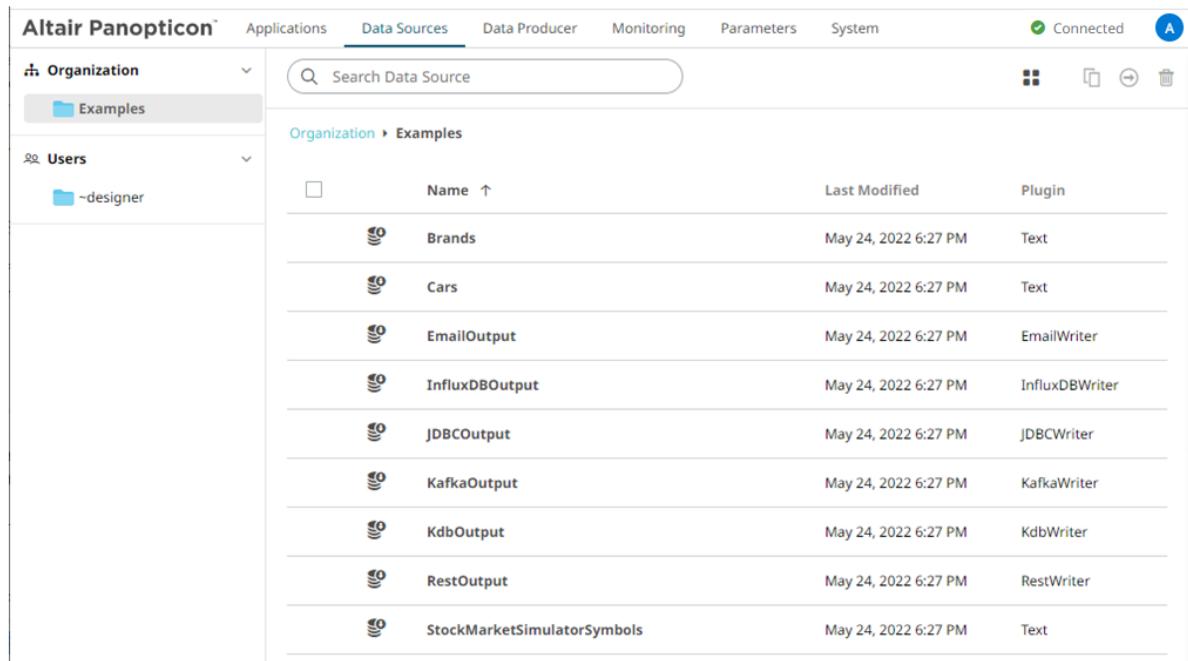
The applications and data sources that you can view and explore are imported.



The screenshot shows the Altair Panopticon interface with the Applications tab selected. The left sidebar shows the Organization structure with 'Examples' selected. The main content area displays a list of sample applications under 'Organization > Examples'. The table has columns for Name, Last Modified, and Status Updated. The applications listed are: AggregationExample, BranchExample, CalculateRemoveReplaceNull, CalculationsExample, ConflateExample, EmailExample, ExternalInputExample, ExternalInputJsonParserExample, ExternalInputXmlParserExample, FilterExample, InfluxDBExample, and JDBCExample. All entries were modified on May 24, 2022, at 6:27 PM.

	Name ↑	Last Modified	Status Updated
▶	AggregationExample	May 24, 2022 6:27 PM	
▶	BranchExample	May 24, 2022 6:27 PM	
▶	CalculateRemoveReplaceNull	May 24, 2022 6:27 PM	
▶	CalculationsExample	May 24, 2022 6:27 PM	
▶	ConflateExample	May 24, 2022 6:27 PM	
▶	EmailExample	May 24, 2022 6:27 PM	
▶	ExternalInputExample	May 24, 2022 6:27 PM	
▶	ExternalInputJsonParserExample	May 24, 2022 6:27 PM	
▶	ExternalInputXmlParserExample	May 24, 2022 6:27 PM	
▶	FilterExample	May 24, 2022 6:27 PM	
▶	InfluxDBExample	May 24, 2022 6:27 PM	
▶	JDBCExample	May 24, 2022 6:27 PM	

Clicking the **Data Sources** tab displays the associated [data sources](#) of the sample applications.



The screenshot shows the Altair Panopticon interface with the Data Sources tab selected. The left sidebar shows the Organization structure with 'Examples' selected. The main content area displays a list of data sources under 'Organization > Examples'. The table has columns for Name, Last Modified, and Plugin. The data sources listed are: Brands, Cars, EmailOutput, InfluxDBOutput, JDBCOutput, KafkaOutput, KdbOutput, RestOutput, and StockMarketSimulatorSymbols. All entries were modified on May 24, 2022, at 6:27 PM.

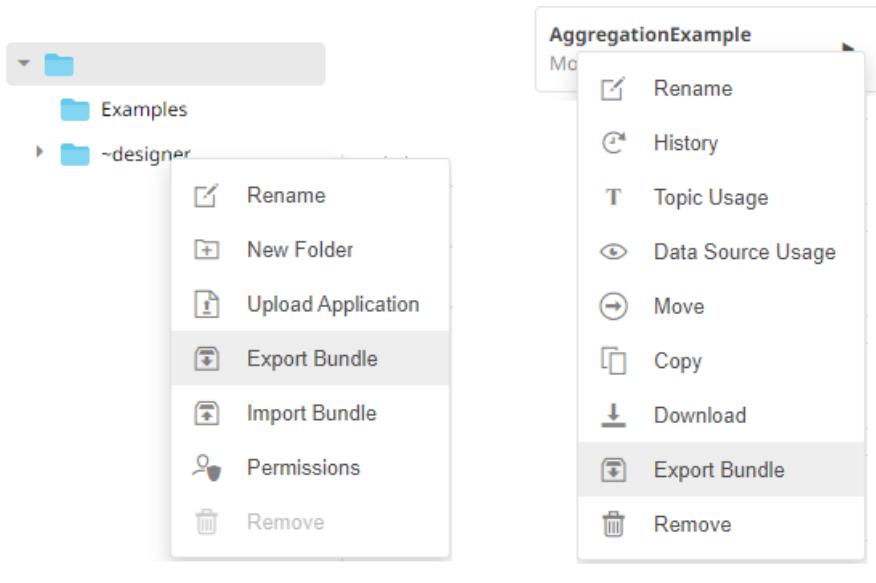
	Name ↑	Last Modified	Plugin
⌚	Brands	May 24, 2022 6:27 PM	Text
⌚	Cars	May 24, 2022 6:27 PM	Text
⌚	EmailOutput	May 24, 2022 6:27 PM	EmailWriter
⌚	InfluxDBOutput	May 24, 2022 6:27 PM	InfluxDBWriter
⌚	JDBCOutput	May 24, 2022 6:27 PM	JDBCWriter
⌚	KafkaOutput	May 24, 2022 6:27 PM	KafkaWriter
⌚	KdbOutput	May 24, 2022 6:27 PM	KdbWriter
⌚	RestOutput	May 24, 2022 6:27 PM	RestWriter
⌚	StockMarketSimulatorSymbols	May 24, 2022 6:27 PM	Text

EXPORTING AN APPLICATION OR FOLDER BUNDLE

Users with an Administrator or Designer role have the ability to export application or folder bundle and the associated data files.

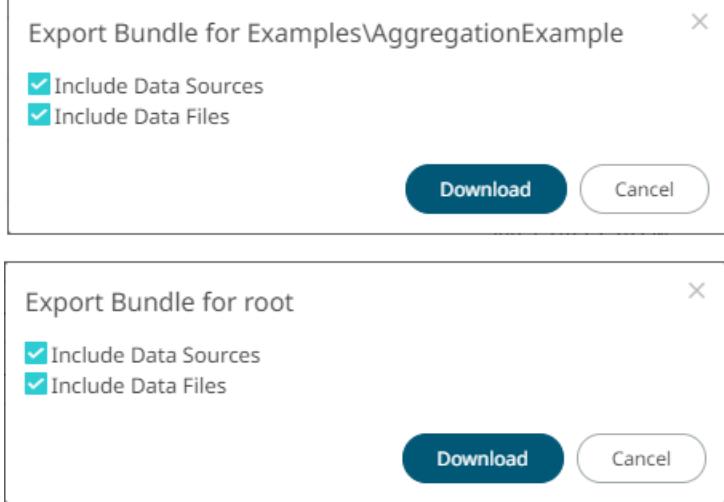
Steps:

1. Right-click on a application or folder and select **Export Bundle** in the context menu.



Application Folder or Subfolder Context Menu Application Context Menu

The corresponding notification message displays.



The **Include datasources** and **Include data files** check boxes are selected by default. This means the associated application data sources and data files will be included in the download.

Download

2. Click **Download**. A copy of the application or folder bundle is downloaded.

SORTING THE LIST OF APPLICATIONS

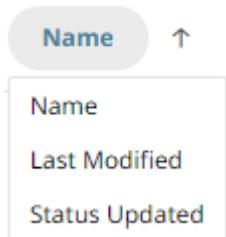
Sorting applications can be done by *Name*, *Last Modified*, or *Status Updated*.

Sort By	Default Sort Order
Name	Ascending
Last Modified	Descending
Status Updated	Descending

Steps:

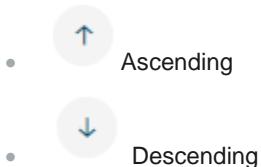
On the *Folders and Applications Summary* layout, either:

- click the **Sort By** option on the *Toolbar* of the *Grid View*
By default, the sorting is by **Name**.



- Name
- Last Modified
- Status Updated

Then click the *Sort Order*:



- click on the **Name**, **Last Modified**, or **Status Updated** column header of the *List View*

The screenshot shows the Altair Panopticon application interface. The left sidebar has sections for 'Organization' (with 'Examples' selected) and 'Users' (with '-designer' listed). The main area is titled 'Organization > Examples'. A search bar at the top right contains the placeholder 'Search Application'. Below the search bar are three column headers: 'Name ↑', 'Last Modified', and 'Status Updated'. A list of ten application examples is displayed, each with a small triangle icon and a timestamp of 'May 24, 2022 6:27 PM'.

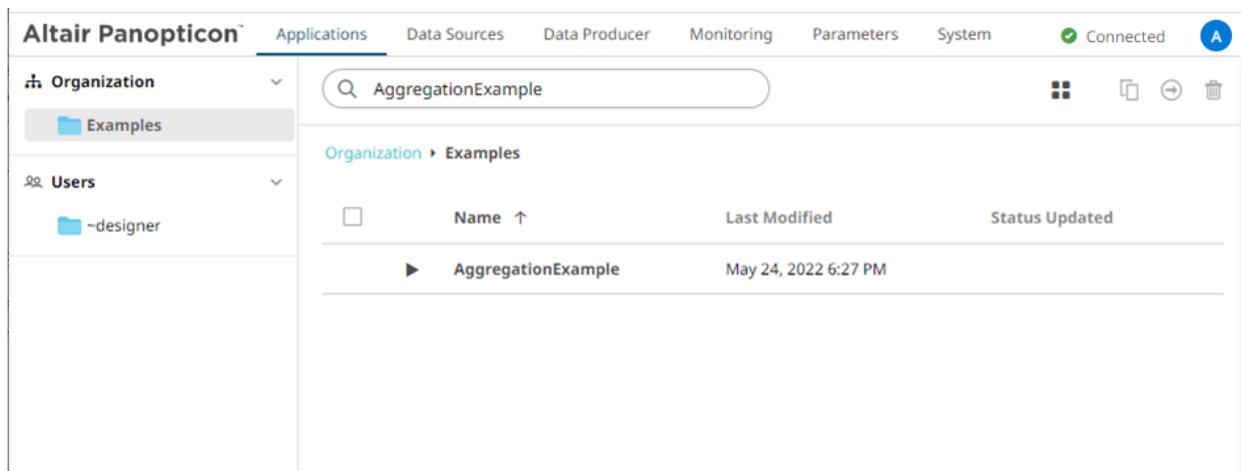
	Name ↑	Last Modified	Status Updated
▶	AggregationExample	May 24, 2022 6:27 PM	
▶	BranchExample	May 24, 2022 6:27 PM	
▶	CalculateRemoveReplaceNull	May 24, 2022 6:27 PM	
▶	CalculationsExample	May 24, 2022 6:27 PM	
▶	ConflateExample	May 24, 2022 6:27 PM	
▶	EmailExample	May 24, 2022 6:27 PM	
▶	ExternalInputExample	May 24, 2022 6:27 PM	
▶	ExternalInputJsonParserExample	May 24, 2022 6:27 PM	
▶	ExternalInputXmlParserExample	May 24, 2022 6:27 PM	
▶	FilterExample	May 24, 2022 6:27 PM	
▶	InfluxDBExample	May 24, 2022 6:27 PM	
▶	JDBCExample	May 24, 2022 6:27 PM	

Then click the *Sort Order*:

- Ascending
- Descending

SEARCHING FOR APPLICATIONS

To search for a particular application, enter it in the *Search Application* box.



Altair Panopticon

Applications Data Sources Data Producer Monitoring Parameters System Connected A

Organization Examples

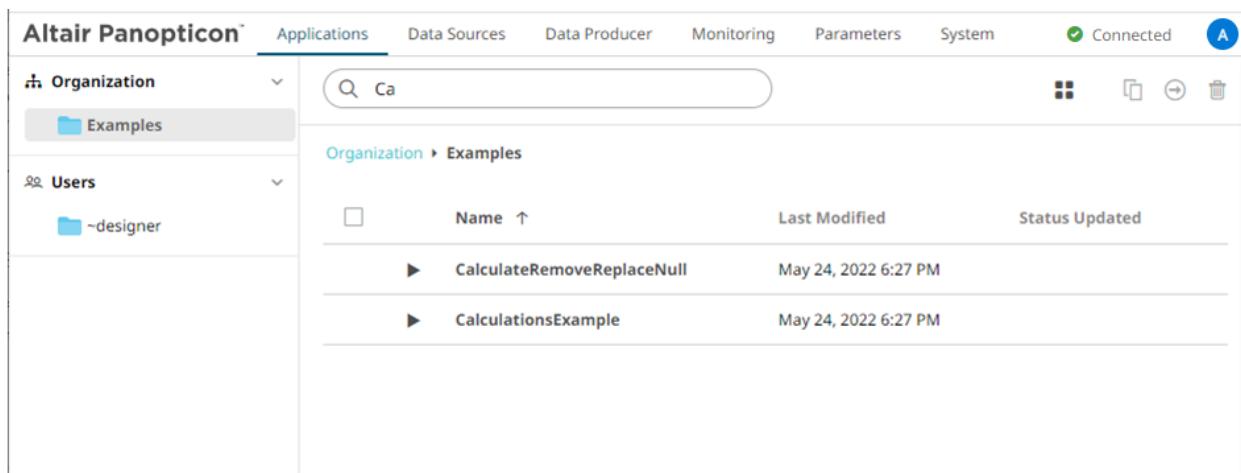
Users ~designer

Search Application: AggregationExample

Organization > Examples

	Name ↑	Last Modified	Status Updated
▶	AggregationExample	May 24, 2022 6:27 PM	

You can also enter one or more characters into the *Search Application* box then click **Enter**. The list of applications that matched the entries will be displayed.



Altair Panopticon

Applications Data Sources Data Producer Monitoring Parameters System Connected A

Organization Examples

Users ~designer

Search Application: Ca

Organization > Examples

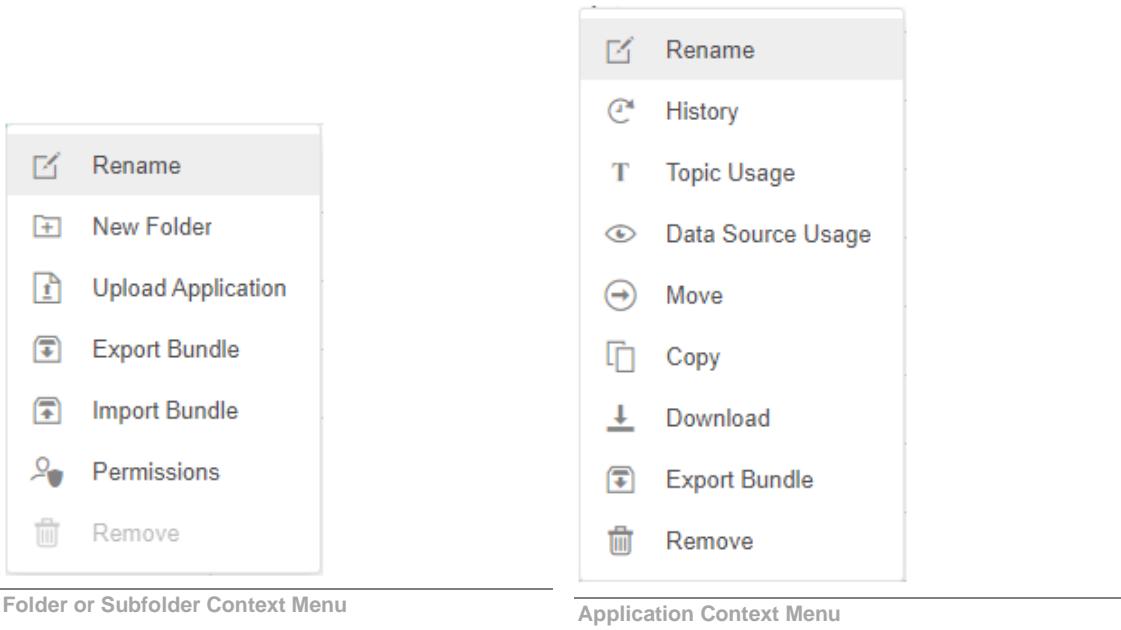
	Name ↑	Last Modified	Status Updated
▶	CalculateRemoveReplaceNull	May 24, 2022 6:27 PM	
▶	CalculationsExample	May 24, 2022 6:27 PM	

RENAMING APPLICATIONS OR FOLDERS

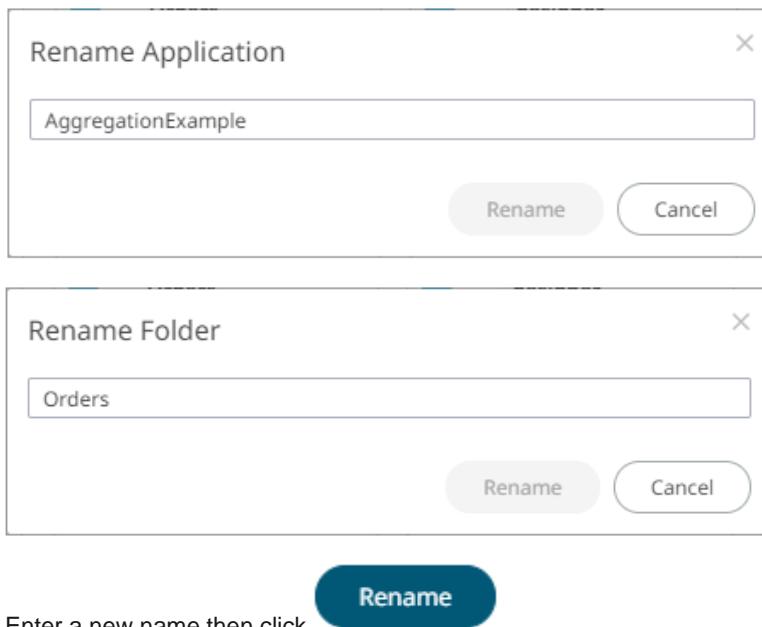
A user with an Administrator or Designer role can rename applications and folders.

Steps:

1. Right-click on an application or folder then select **Rename** in the context menu.



The *Rename Application* or *Rename Folder* dialog displays.



2. Enter a new name then click

VIEWING APPLICATION HISTORY AND REPUBLISHING

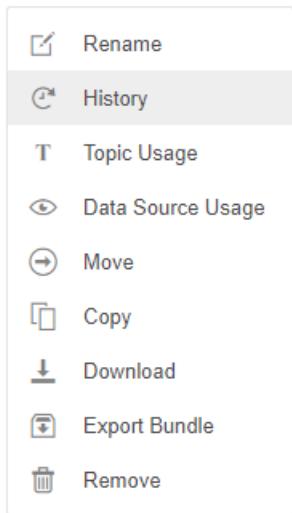
A user with either an Administrator or Designer role can also perform the following:

- View the change history of applications
- Republish an archived application to the recent version of Panopticon Streams Server

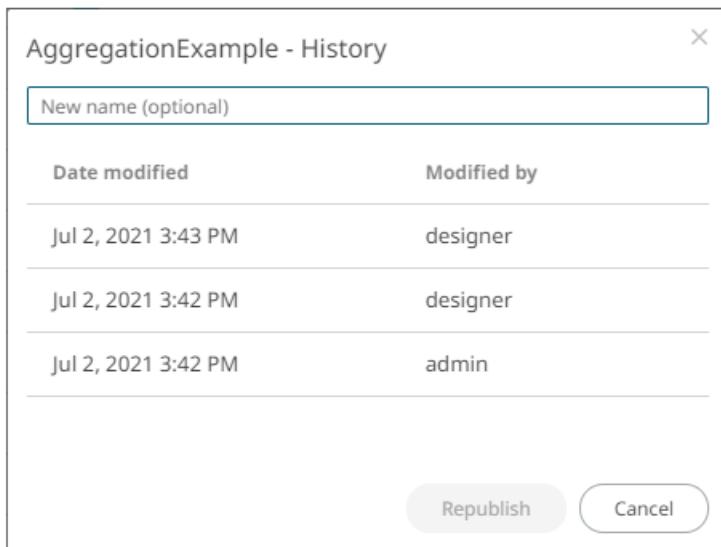
- Rename an archived application

Steps:

1. On the **Applications** tab, right-click on an application and select **History** in the context menu.



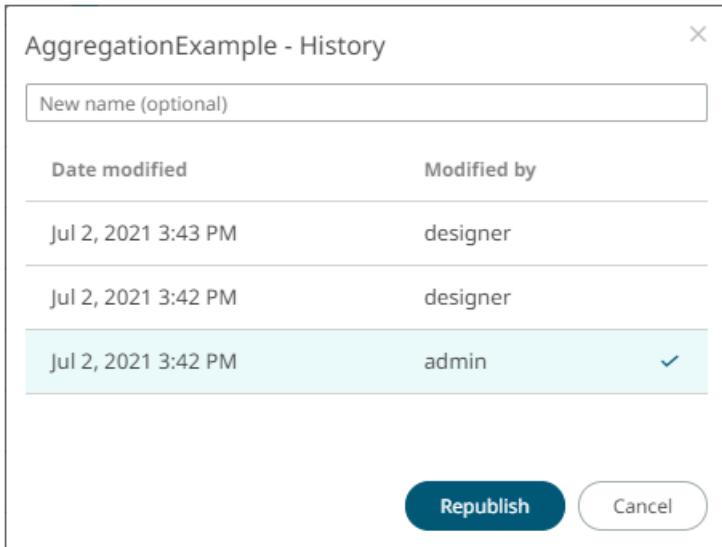
The *<Application Name> - History* dialog displays:



Sort the archival list either through the *Date Archived* or *Archived By* by clicking on the or button.

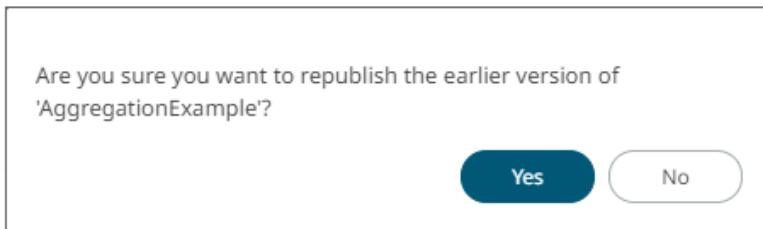
Also, move to the other pages of the list by clicking on a page or clicking the or button.

2. You may opt to rename an archived application by entering a new one in the *New Name* box.
3. Click on an archived application in the list.



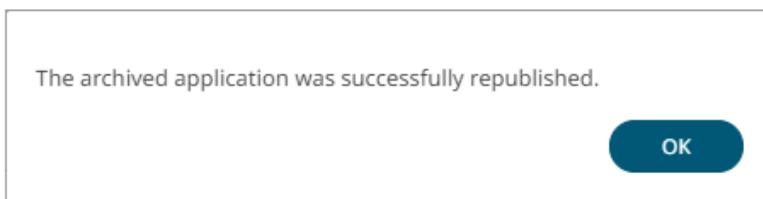
Republish

Then click **Republish**. A notification message displays.



4. Click **Yes**.

A confirmation message displays.

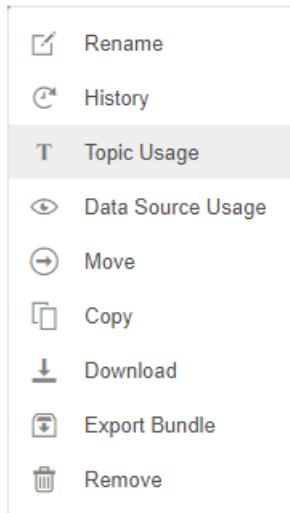


VIEWING AND MANAGING APPLICATION TOPIC USAGES

On the **Applications** tab, you can view the input and output topic usages of an application.

Steps:

1. On the **Applications** tab, right-click on an application and select **Topic Usage** in the context menu.



The *<Application Name> Topics* dialog displays.

AggregationExample Topics

TOPICS			
Name	Type	# Messages	Messages/sec
.designer.AggregationExample.Input	INPUT	0	0
.designer.AggregationExample.Output	OUTPUT	0	0

Clear Schemas **Clear All Data**

Close

If the application has been executed, the *#Messages* column will display the number of retrieved messages, while the *Messages/sec* column will display the number of retrieved messages per second.

If the application is not yet executed, both the *#Messages* and *Messages/sec* columns will display 0 values.

2. You can then opt to:

- [clear data](#)
- [clear schemas](#)
- [sort topics](#)

3. Click **Close**.

Clearing the Data In an Application Topic

You can recycle an application by:

- [stopping](#) the application
- deleting data in the topics
- [restarting](#) the application

Follow the steps below to clear the data in an application topic.

Steps:

1. You can either:

- Click  to delete the data in a topic, or
- Click  **Clear All Data** to delete the data in all of the topics in an application

 **Close**

2. Click  .

Clearing the Schema in an Application Topic

Schema registry can be cleared in any application topic.

Steps:

1. You can either:

- click  to delete the schema in a topic, or
- click  **Clear Schemas** to delete the schema in all of the topics in an application

 **Close**

2. Click  .

Sorting Application Topics

By default, the list of topics is sorted by *Name* in an ascending order.

AggregationExample Topics					X
TOPICS					
Name	Type	# Messages	Messages/sec	Clear Schemas	Clear All Data
.designer.AggregationExample.Input	INPUT	0	0	 	
.designer.AggregationExample.Output	OUTPUT	0	0	 	

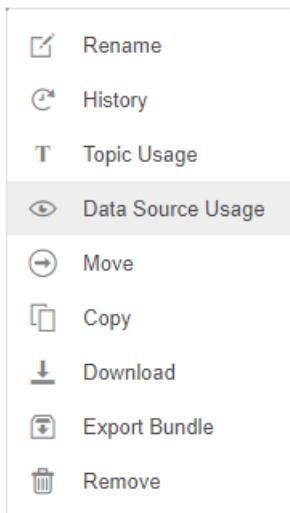
Modify the sorting of the list by clicking the \downarrow or \uparrow button of the *Name*, *Type*, *#Messages* or *Messages/sec* columns. The icon beside the column that was used for the sorting will indicate if it was in an ascending or descending order.

VIEWING THE APPLICATION DATA SOURCES USAGE

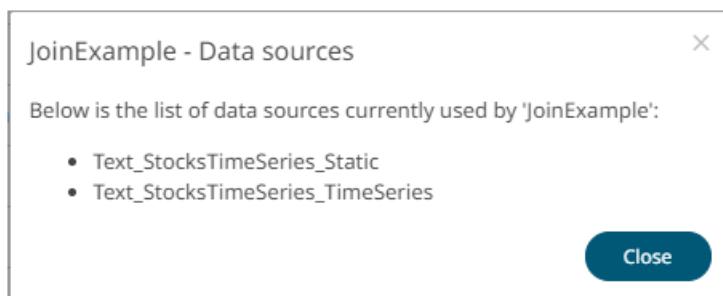
On the **Applications** tab, you can view the data sources that are currently used by an application.

Steps:

1. On the **Applications** tab, right-click on an application and select **Data Source Usage** in the context menu.



The list of data sources that is currently used by the application displays.



2. Click **Close**.

MOVING APPLICATIONS

Users with Administrator or Designer role are allowed to move an application to another folder or subfolder to which they have permission.

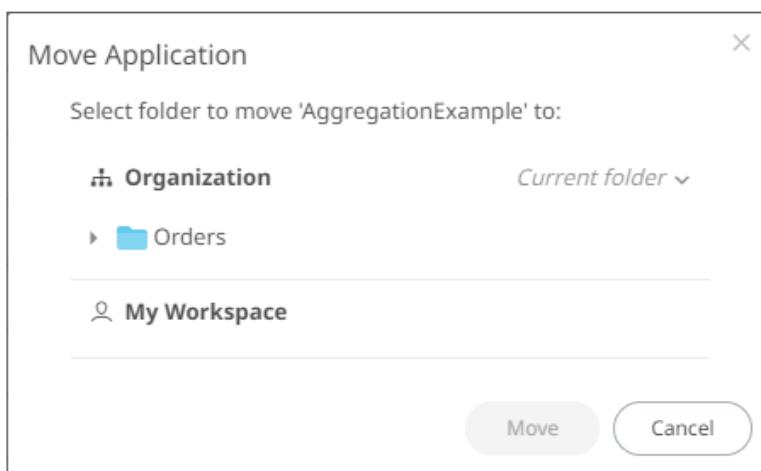
Moving applications can be done either through the toolbar or context menu.

Steps:

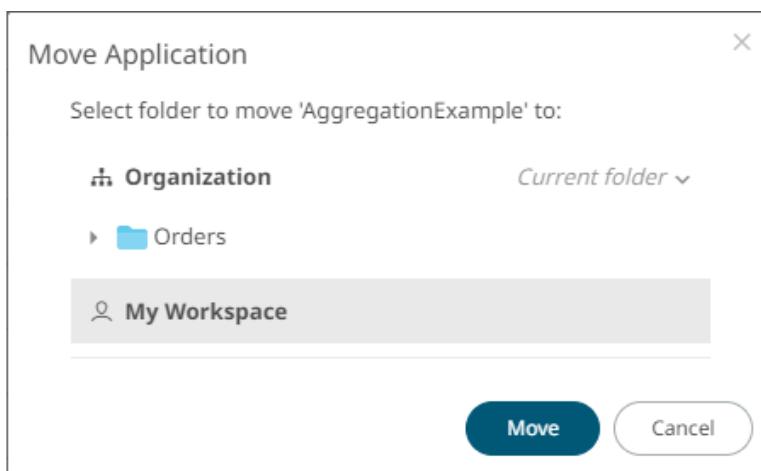
1. Select the check the box of one or multiple applications either on the *Grid View* or *List View*.
2. Then select either:

- the **Move**  icon on the toolbar
- **Move** in the content menu

The *Move Application* dialog displays with the folder or subfolders to which the user is allowed to move the application.



3. Select the folder or subfolder.



4. Click **Move**.

NOTE

If applications with the same name are already in the selected folder, a notification message displays if they will be replaced.

A application with the same name already exists in the selected folder. Do you want to replace it?

Yes

No

Click **Yes** to replace or **No** to move a copy of the same applications.

The application is moved and displayed on the selected folder.

COPYING APPLICATIONS

Users with Administrator or Designer role are allowed to copy an application to another folder or subfolder to which they have permission.

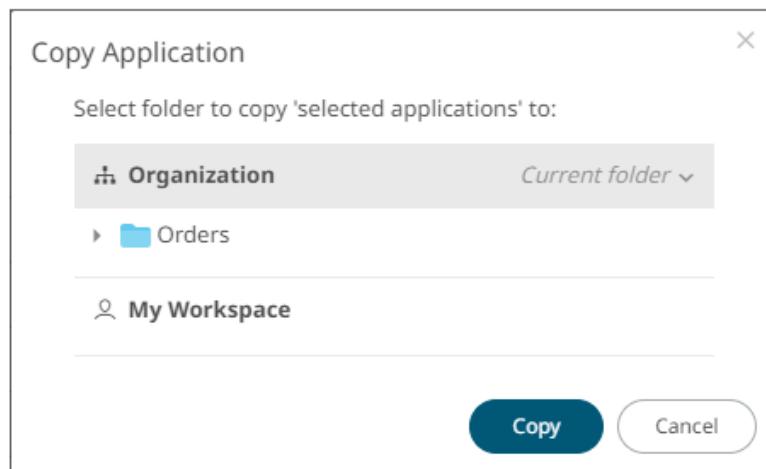
Copying applications can be done either through the toolbar or context menu.

Steps:

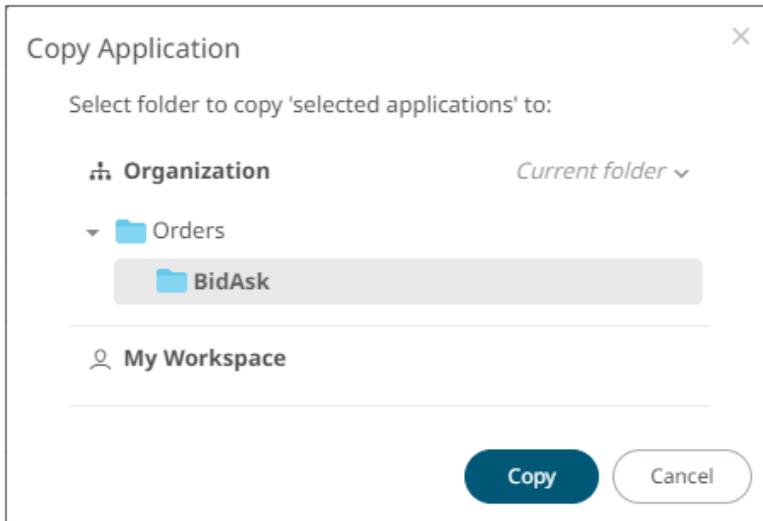
1. Select the check box of one or multiple applications either on the *Grid View* or *List View*.
2. Then select either:

- the **Copy** icon on the toolbar
- **Copy** in the content menu

The *Copy Application* dialog displays with the folder or subfolders the user is allowed to copy the applications to.



3. Select the folder or subfolder.



4. Click **Copy**.

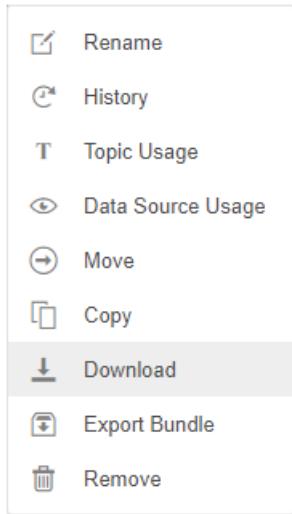
NOTE

If applications with the same name are already in the selected folder, a copy of the applications are added.

The applications are copied and displayed on the selected folder.

DOWNLOADING AN APPLICATION

Users with an Administrator or Designer role are allowed to download a copy of an application by right-clicking on an application and selecting **Download** in the context menu.



The application is downloaded.

DELETING AN APPLICATION

Users with Administrator or Designer role are allowed to delete an application which can be done either through the [toolbar](#) or [context menu](#).

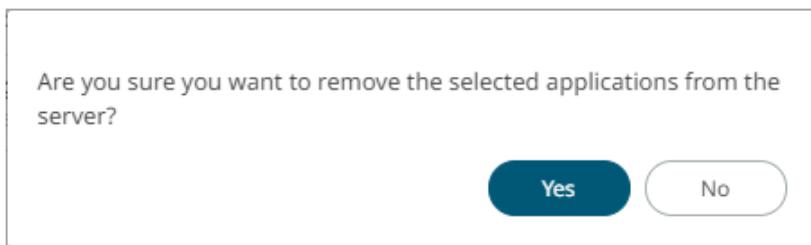
Deleting Applications Using the Toolbar

Steps:

1. Select the check box of applications either on the *Grid View* or *List View*.

2. Click  on the toolbar.

A notification message displays.

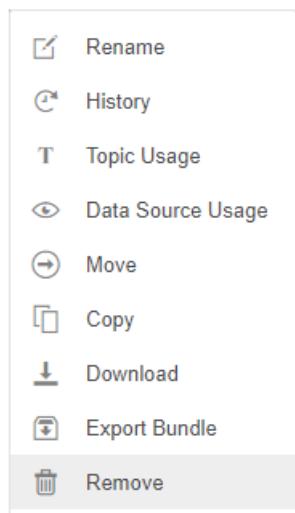


3. Click  to remove.

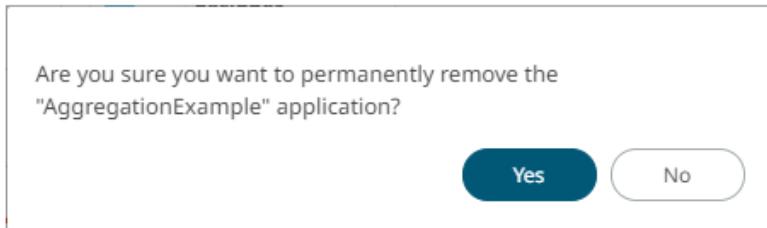
Deleting Applications Using the Context Menu

Steps:

1. Right-click on an application and select **Remove** in the context menu.



A confirmation message displays.



2. Click **Yes** to delete.

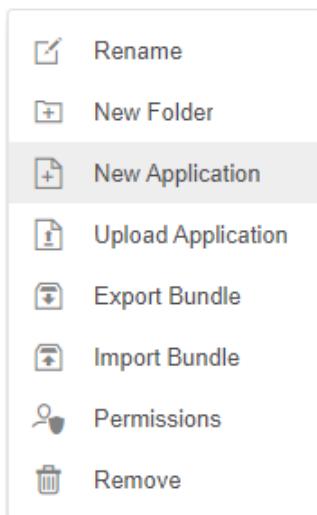
CREATING A NEW APPLICATION

A user with a Designer role can create new applications to folders or subfolders to which they have permission.

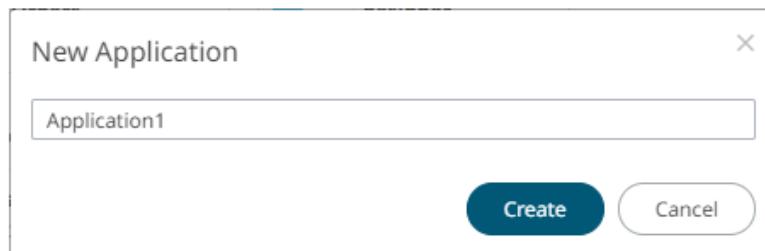
Steps:

1. On the **Applications** tab:

- click **+ New Application** on the toolbar
- right-click on a folder or subfolder and select **New Application**.



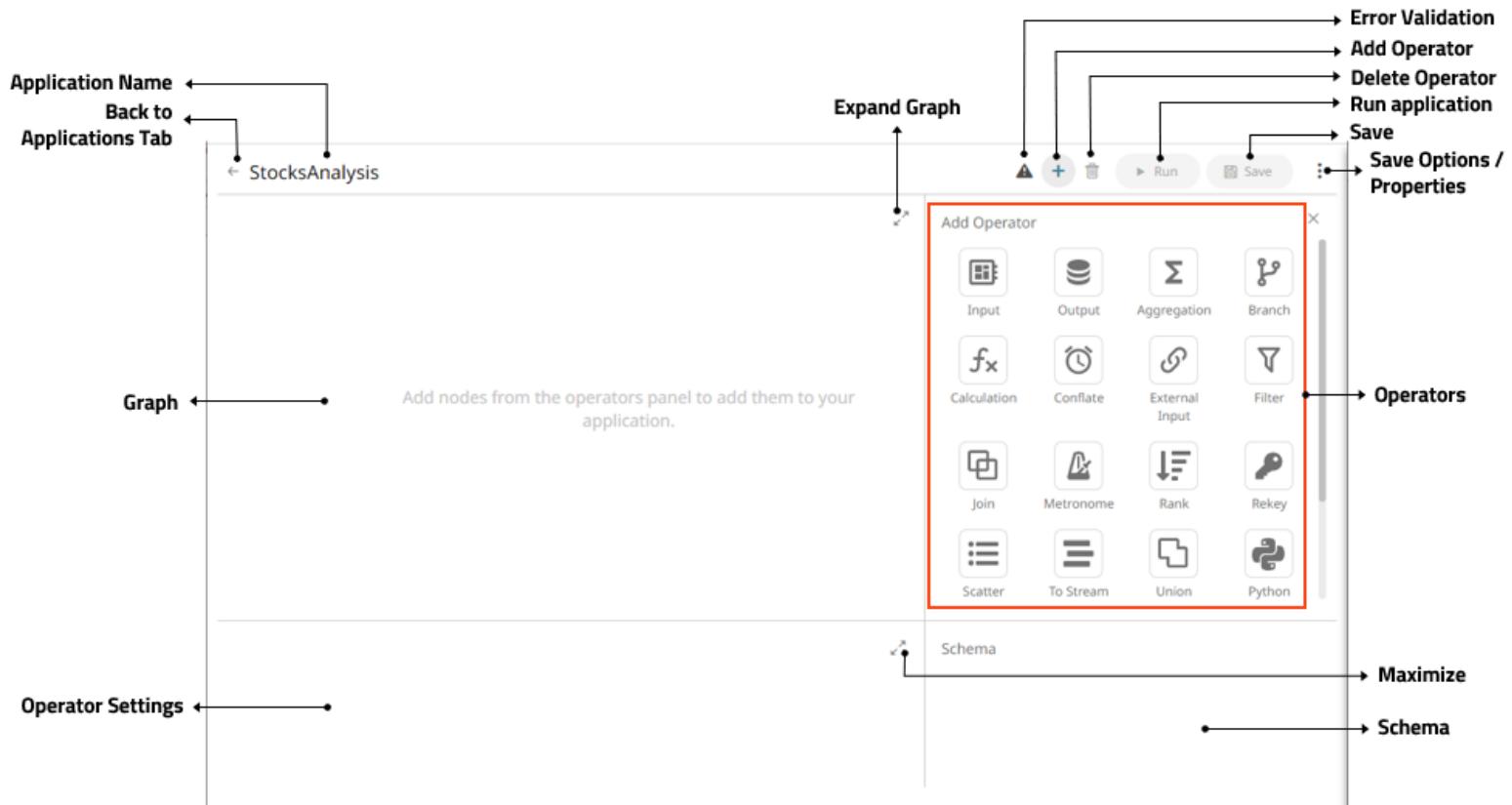
The *New Application* dialog displays.



Create

2. Enter the *Name* of the application and click

The *Application* page displays with the following sections. Initially, the *Operator Settings* pane is displayed.



Section/Pane	Description
Application Name	Name of the application. Click the button to go back to the <i>Applications</i> listing page.
Expand Graph	Expands the <i>Graph</i> pane.
Error Validation	After <u>saving</u> the changes in the application, this allows <u>error validation</u> . If there are definition issues (red node) or if there is no traffic on the topic (yellow node), you can click to help fix the errors. If there are no issues, is no longer displayed in the <i>Application</i> page.
Add Operator	When enabled , displays the <i>Add Operator</i> pane and allows operators to be added in the application. Click to close the <i>Add Operator</i> pane.
Delete Operator	Deletes the selected operator.

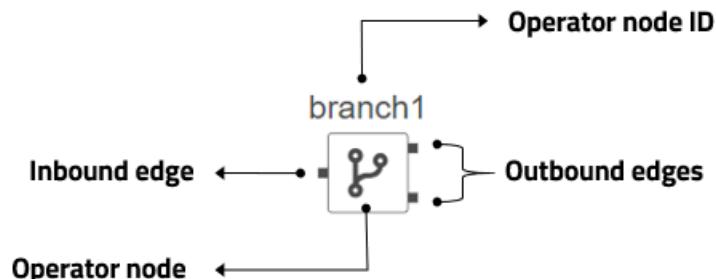
Run Application	Runs or executes an application.
Save	Saves the changes made in the <i>Application</i> page.
Save Options and Properties	Allows saving of changes made in the application or saving another copy. The application properties are also displayed and allows adding new ones.
Graph	Display the nodes and edges of the application model. It allows single node selection.
Operators	Available operators that can be added in the application.
Operator Settings	When an operator has been added or a node is selected in the application graph, the corresponding operator settings are displayed in the <i>Operator Settings</i> pane to allow editing. When the application is running, the operator settings are displayed but are not editable.
Maximize	Expands the <i>Operator Settings</i> pane.
Schema	Displays the preview of the data.

Refer to the sections below to proceed in creating an application.

ADDING AN OPERATOR

Operators can be added in any order. The sequence or direction of the graph will be based on the inputs or outputs that will connect the nodes.

When adding an operator, its node will be displayed in the *Graph* pane.



NOTE

- The edges (inbound and/or outbound) will depend on the operator.
- For best practice, start by adding Input operators (i.e., Input, External Input, or Metronome) and end with the Output operator.

Also, the corresponding *Operator Properties* and *Schema* are displayed on the *Application* page.

The screenshot shows the Panopticon Streams Application page. At the top, there is a header with a back arrow, the title 'Application1', and various buttons for 'Run', 'Save', and settings. Below the header, the main area is divided into three sections: a 'Graph' pane on the left containing a node labeled 'input1' with a table icon, an 'Add Operator' pane on the right listing various operators (Input, Output, Aggregation, Branch, Calculation, Conflate, External Input, Filter, Join), and a 'Settings' pane on the left. The 'Settings' pane contains tabs for 'Metrics' and 'Settings'. Under 'Settings', there are fields for 'Node ID' (input1), 'Topic' (input1), 'Partition Count' (empty), 'Input Type' (STREAM), 'Priority' (HIGH), 'Data source' (empty), and 'Key columns' (empty). To the right of the 'Settings' pane is a 'Schema' pane with a table header: 'Field Name' (colored blue), 'Type' (colored red), and 'OUTPUT' (colored green). The 'OUTPUT' column is currently empty.

Adding an Input Operator

Used to define the input data for the application model.

Steps:

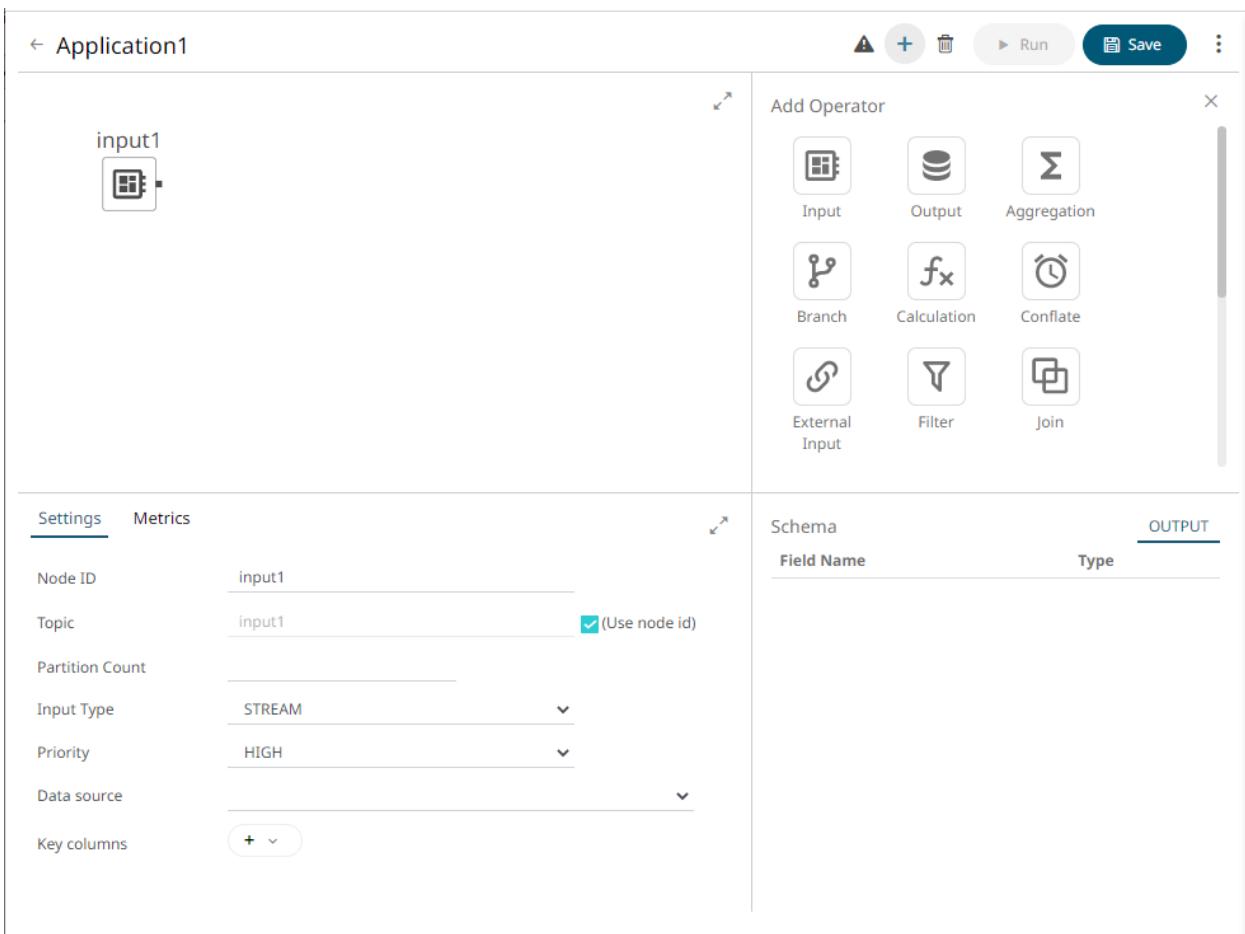


1. In the *Application* page, click  and select **Input**  in the *Add Operator* pane.

input1



The **Input** node  icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.



This operator serves as the initial source of the data in the application. The right (outbound) edge allows you to connect to other operators.

2. In the *Operator Settings* pane, define or select the following properties:

Property	Description
Node ID	The ID of the input operator.
Topic	<p>The stream of records or input you will be subscribed to. Select the <i>Use Node ID</i> check box to use the value entered in the <i>Input ID</i>. Otherwise, uncheck the box and enter a new <i>Topic ID</i>.</p> <p>When adding Topic IDs, ensure they:</p> <ul style="list-style-type: none"> • must be unique across an application • must be specified • must start with a letter (a to Z) or an underscore. Also, it can only contain letters (a to Z), numbers (0 to 9), and underscores
Partition Count	Enter the number of partitions for the Kafka topics that will be created for the Input operator.
Input Type	Select the input type: STREAM , TABLE , or GLOBAL_TABLE .

	STREAM will treat incoming data as a stream of records while TABLE creates a “materialized view” or snapshot table, representing the latest state of received key/value pairs. GLOBAL_TABLE can be seen as a materialized view that is distributed across all partitions. This is useful for keeping small, relatively static, data sets that needs to be joined with streaming data.
Priority	Select the priority of the node's startup: <ul style="list-style-type: none"> • APPLICATION – running and successful completion of the node is critical in the application startup. • HIGHEST – highest priority but not critical. • HIGH (Default) – high priority but not critical. • STANDARD – standard priority. • LOW – low priority.
Data Source	Select the data source. <p>NOTES:</p> <ul style="list-style-type: none"> • It is recommended to upload the data source first so they will be available for selection. • Selecting a non-streaming data source displays the <i>Refresh Period</i> (ms) property. Enter the refresh period for the data. This value determines when to periodically reload the data (from the beginning). • Setting the <i>Refresh Period</i> to any value less than or equal to zero will disable automatic data reload. <p>The preview of the data (OUTPUT) is displayed in the <i>Schema</i> pane.</p>
Key Columns	In Kafka, all messages are processed in a key/value fashion where the value represents the actual data payload or record. The key is used to determine how the key/value pairs are distributed across available partitions. If the key is null a round-robin approach is used to determine partition. For the TABLE and GLOBAL_TABLE input type, key is also essential for defining how records are segregated (keyed) in the table. Not providing a key will result in a single-row table. Post input, keying of records can be changed by using either the Rekey or Aggregation operators.

NOTE

Node ID, Topic, Input Type, and Data Source properties are required.

3. In the *Key Columns* section, click  to add a key column from the data source schema. Repeat to add more.

Settings

Node ID: input1

Topic: input1 (Use node id)

Partition Count: 1

Input Type: STREAM

Priority: HIGH

Data source: Text_StocksStatic

Key columns: Industry X, Region X, Country X, +

Refresh period (ms): 0

Schema

Field Name	Type
Region	string (key, null)
Country	string (key, null)
Exchange	string (null)
Name	string (null)
Forex	string (null)
Symbol	string (null)
ISIN	string (null)
SEDOL	string (null)
Close_local	double (null)
Mcap_local	double (null)
Mcap_USD	double (null)
Industry	string (key, null)
Supersector	string (null)

You can also delete a key column in the list by clicking .

4. [Save](#) the changes.

Example

```

<input>
  <id>Input</id>
  <topic>Input</topic>
  <dataProducer>
    <id>StocksStatic</id>
    <refreshPeriod>1000</refreshPeriod>
    <keyColumns>
      <field>Id</field>
    </keyColumns>
  </dataProducer>
  <inputType>TABLE</inputType>
</input>

```

Adding An Aggregation Operator

The aggregation operator aggregates the data based on a grouping key and a set of aggregated fields.

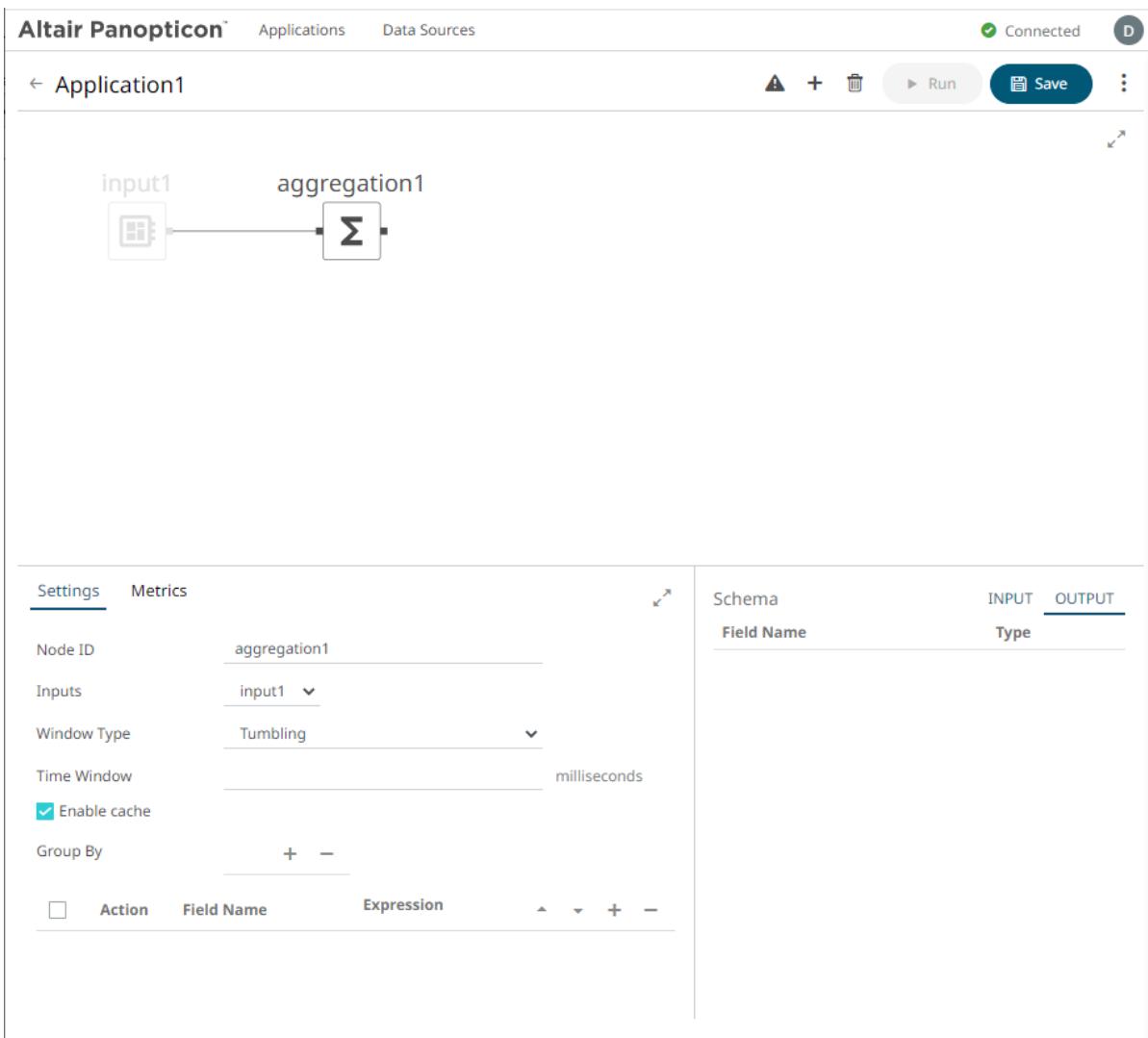
Steps:



1. On the *Application* page, click  and select **Aggregation**  in the *Add Operator* pane.



The **Aggregation** node  icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.



This operator has left (inbound) and right (outbound) edges that allow connection to other operators in the application.

2. In the *Operator Settings* pane, define or select the following required properties:

Property	Description
Node ID	The ID of the aggregation operator.
Inputs	Automatically connects to the currently selected operator. You can select another ID of the operator that will be the source of the data in the <i>Inputs</i> drop-down list. The preview of the data (INPUT) is displayed in the <i>Schema</i> pane.
Window Type	Select either: <ul style="list-style-type: none"> • Tumbling (default) A series of fixed-sized, non-overlapping, and adjoining time intervals.

	<p>This window type is a moving window whose window size (<i>Time Window</i>) is equal to its advance interval. Since tumbling windows never overlap, a data record will belong to only one window.</p> <ul style="list-style-type: none"> Hopping <p>This window type models fixed-sized, scheduled overlapping windows.</p> <p>Defined by the window's size (<i>Time Window</i>) and advance interval (<i>Advance Period</i>).</p>
Time Window	The window's size (in milliseconds).
Advance Period	The advance interval for the <i>Hopping Window</i> (in milliseconds). Specifies by how much a window moves forward relative to the previous one.
Enable Cache	Specifies whether to start or stop caching tables. When caching is stopped, it gets every event input into the table to produce an output event. This is necessary for delta/prev aggregates.
Group By	The name/IDs of the columns that the data will be grouped by. (Proceed to step 3.)
Fields List	A set of aggregated fields (with actions and expressions). (Proceed to step 5.)

3. In the *Group By* section, click . A column is added in the list. Click the drop-down list to select another column.
4. Select a column that will be used to group the data.

The INPUT and OUTPUT schema are displayed.

Schema	INPUT	OUTPUT
	Field Name	Type
Region		string (key, null)
Country		string (null)
Exchange		string (null)
Name		string (null)
Forex		string (null)
Symbol		string (null)
ISIN		string (null)
SEDOL		string (null)
Close_local		double (null)
Mcap_local		double (null)
Mcap_USD		double (key, null)
Industry		string (key, null)

Schema	INPUT	OUTPUT
Field Name	Type	
Region	string (key, null)	

Repeat steps 3 and 4 to add more.

Settings Metrics

Node ID: aggregation1

Inputs: input1

Window Type: Tumbling

Time Window: 1000 milliseconds

Enable cache

Group By: Region Mcap

Action	Field Name	Expression	▲	▼	+	-
<input type="checkbox"/>						

Schema

Field Name	INPUT	OUTPUT
Region	string (key, null)	
Mcap_USD		double (key, null)

You can also delete a column in the *Group By* list by selecting its check box and clicking .

- Under the *Field List* section, click . A new field entry displays.

<input type="checkbox"/>	Action	Field Name	Expression	▲	▼	+	-
<input type="checkbox"/>	Add						

Field Name is required

6. Enter the *Field Name* and the *Expression* that will be evaluated for each incoming record.

Example:

Field Name: Count

Expression: count()

7. Repeat steps 5 and 6 to add more aggregated fields.

Action	Field Name	Expression	
Add	Count	count()	fx
Add	Samples	samples(Mcap_USD)	fx
Add	Sum_Mcap_USD	sum(Mcap_USD)	fx
Add	First_Close_local	first(Close_local)	fx
Add	Last_Close_local	last(Close_local)	fx
Add	Min_One_Day_Change	min(One_Day_Change)	fx
Add	Max_One_Day_Change	max(One_Day_Change)	fx
Add	Avg_One_Day_Change	avg(One_Day_Change)	fx
Add	Varp_One_Day_Change	varp(One_Day_Change)	fx
Add	Vars_One_Day_Change	vars(One_Day_Change)	fx

You can also:

- check the topmost box to select all of the fields
- change the order of the fields by checking a field's box and clicking either the or button
- delete a field entry in the Field List by checking its box and clicking

The OUTPUT schema is updated based on the added aggregations.

Schema	INPUT	OUTPUT
Field Name	Type	
Industry		string (key, null)
Count		long (not null)
Sum_Mcap_USD		double (not null)
First_Close_local		double (null)
Last_Close_local		double (null)
Min_One_Day_Change		double (null)
Max_One_Day_Change		double (null)
Samples		long (not null)
Avg_One_Day_Change		double (null)
Varp_One_Day_Change		double (null)
Vars_One_Day_Change		double (null)
Sdevp_One_Day_Change		double (null)
Sdevs_One_Day_Change		double (null)

8. [Save](#) the changes.

Example

```

<aggregation>
  <id>Aggregation</id>
  <fields>
    <field>
      <id>ColumnId</id>
      <action>ADD</action>
      <expression>Sum(Mcap_USD) </expression>
    </field>
  </fields>
  <groupBy>
    <field>Industry</field>
  </groupBy>
</aggregation>

```

Building the Expression

To build the expression, you can either:

- manually enter into *Expression* text box

Take note that the column name is case sensitive.

A validation error displays with a suggestion to help build the expression.

Examples:

<input type="checkbox"/>	Add	Samples	samples(MCAP_USD)
--------------------------	-----	---------	-------------------

Col 8: Unable to find column MCAP_USD , did you mean [Mcap_USD](#)

Click the link (e.g., [Mcap(USD)]) to apply the correct entry.

<input type="checkbox"/>	Add	Samples	samples()
--------------------------	-----	---------	-----------

col.8: Something missing? Got <nothing> expected or ')

Complete the expression as necessary.

- use the [Expression Builder](#)

Using the Expression Builder

Create new fields using data from existing columns in your input operator and supported [aggregation](#) or [calculation](#) functions and operators.

Steps:

1. On the *Fields List* section, click  to add a new field instance.

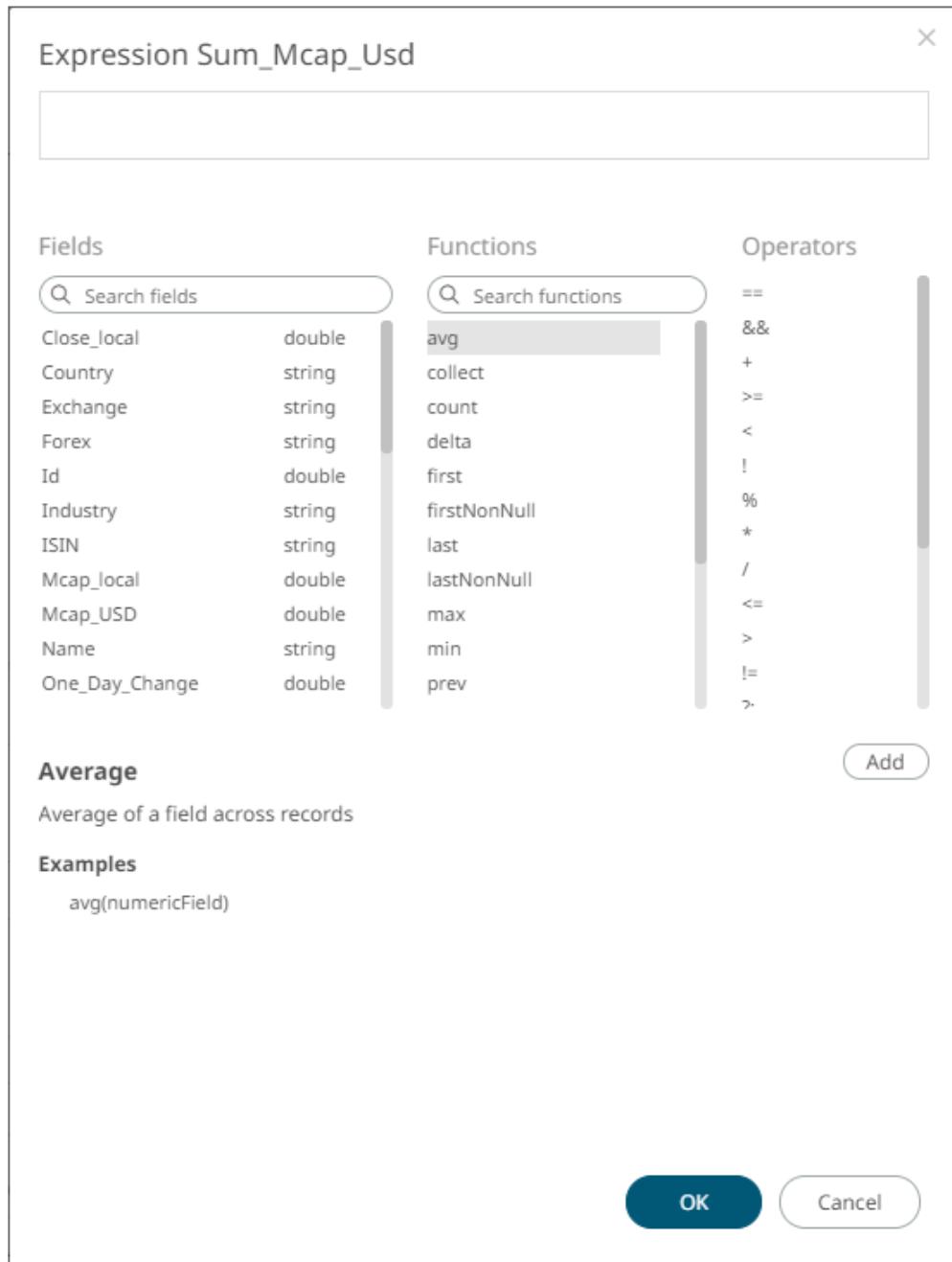
<input type="checkbox"/>	Action	Field Name	Expression	▲	▼	+	-
<input type="checkbox"/>	Add						

Field Name is required

2. Enter the *Field Name*.

3. Click the **New Expression**  icon.

The *Expression <Field Name>* dialog displays.



4. Build an expression by double-clicking in the list of *Functions*, *Fields*, and *Columns*.

You can also click on a function and operator then **Add**.

To search for a particular column or function, enter it in the *Search Fields/Search Functions* box.

Functions

Search fields

samples

Or enter one or more characters/alphabets into the *Search Fields* box and the suggested list of columns that matched the entries will be displayed.

For example, after entering **One**, the list will be displayed such as below:

Fields

One_Day_Change double
One_Day_Change_USD double
One_Day_Close double
One_Month_Change double
One_Month_Change_USD double
One_Month_Close double
One_Week_Change double
One_Week_Change_USD double
One_Week_Close double

OK

5. Click **OK**. The new expression is added to the field instance.

Add **Sum_Mcap_USD** `sum(Mcap_USD)` *fx*

Supported Aggregation Functions

This section lists the aggregation functions that are only supported in aggregation operator expressions.

NOTE

Panopticon Streams also supports nullability where:

- a field may or may not allow null/empty/missing/NA values.
- functions or operators may or may not allow null arguments (e.g., you can't divide seven by null).

Aggregation Function	Description	Example	Nullability
<code>avg(X)</code>	Average of a field across records.	<code>avg(numericField)</code>	✓
<code>collect(X)</code>	Returns last n values of a field across records. collect(fieldName,valuesCount) Parameters: <ul style="list-style-type: none">• fieldName – Field name whose last n values should be retained.• valuesCount – Number of values to be retained.	<code>collect(symbol,3)</code>	✓
<code>count(X)</code>	Counts the number of records.	<code>count()</code>	

Aggregation Function	Description	Example	Nullability
delta(X)	Returns the difference of the last and second last value of the integer field.	delta(numericField)	✓
first(X)	First value of a field.	first(fieldName)	✓
firstNonNull(X)	First non-null value of a field.	firstNonNull(fieldName)	✓
last(X)	Last value of a field.	last(fieldName)	✓
lastNonNull(X)	Last non-null value of a field.	lastNonNull(fieldName)	✓
max(X)	Maximum of an integer field across records.	max(fieldName)	✓
min(X)	Minimum of an integer field across records.	min(fieldName)	✓
prev(X)	Previous values of a field record.	prev(fieldName)	✓
samples(X)	Count of non-null values of field records.	samples(fieldName)	✓
sdevp(X)	Returns the standard deviation of an entire population.	sdevp(numericField)	✓
sdevs(X)	Estimates standard deviation based on a sample.	sdevs(numericField)	✓
sum(X)	Sums a field across records.	sum(numericField)	✓
varp(X)	Returns the variance in an entire population.	varp(numericField)	✓
vars(X)	Returns the variance based on a sample.	vars(numericField)	✓
wavg(X)	Weighted average of a field across records. wavg(score,weight) Parameters: <ul style="list-style-type: none">score – Numeric field for score.weight – Numeric field for weightage.	wavg(marks,weight)	✓

NOTE

The following aggregates work with a time window (can subtract): count, samples, sum, sdev, var, and avg

Converting Timestamp to/from Integer

Allows you to convert Timestamp values to/from Integer which include the following examples:

- from posix to timestamp
- from posixmillis to timestamp
- from timestamp to posix
- from timestamp to posixmillis

The conversion uses the expression: `to([typename] , [expression] , [format])`

Examples:

- `to(int, timefieldname, 'POSIX')`

- `to(time, intfieldname, 'POSIX')`

Adding a Branch Operator

The branch operator will split a stream into one or more branches. The path for a stream is determined by a configured predicate within the branch operator.

The predicate expression will be evaluated for each incoming record. A record will be routed to the first branch with a matching predicate.

Steps:



1. On the *Application* page, click  and select **Branch**  in the *Add Operator* pane.

branch1



The **Branch** node  icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.

← Application1

⚠ + 🗑️ ⏪ Run ⏴ Save ⏮

input1
branch1

Settings	Metrics
Node ID	branch1
Inputs	input1
<input type="checkbox"/> Predicate	
<input type="checkbox"/>	
<input type="checkbox"/>	

Schema	INPUT/OUTPUT
Field Name	Type
Region	string (key, null)
Country	string (key, null)
Exchange	string (null)
Name	string (null)
Forex	string (null)
Symbol	string (null)
ISIN	string (null)
SEDOL	string (null)
Close_local	double (null)
Mcap_local	double (null)
Mcap_USD	double (null)
Industry	string (key, null)

The left (inbound) edge allows you connect to an input data or operator. The right (outbound) edges allow you to add more streams.

2. In the *Operator Settings* pane, define or select the following required properties:

Property	Description
Node ID	The ID of the branch operator.
Inputs	Automatically connects to the currently-selected operator. You can select another ID of the operator that will be the source of the data in the <i>Inputs</i> drop-down list. The preview of the data (INPUT/OUTPUT) is displayed in the <i>Schema</i> pane.
Predicate	A list of predicates. Each predicate contains an expression that will be evaluated for each record.

3. To add more predicates, click . A new predicate entry displays.

Enter at least two expressions.

<input type="checkbox"/> Predicate	 
<input type="checkbox"/> One_Day_Change < 0	
<input type="checkbox"/> One_Day_Change >= 0	

You can also:

- select the topmost check box to select all of the fields
- delete a field entry in the Field List by selecting its check box and clicking 

4. [Save](#) the changes.

Example

```

<branch>
  <id>Branch</id>
  <predicates>
    <!-- One_Day_Change < 0 -->
    <predicate>One_Day_Change < 0</predicate>
    <!-- One_Day_Change >= 0 -->
    <predicate>One_Day_Change >= 0</predicate>
  </predicates>
</branch>

```

Example 2

```
<streams>
  <stream>
    <source>Input</source>
    <sink>
      <operator>Branch</operator>
    </sink>
  </stream>
  <stream>
    <source>Branch</source>
    <port>1</port>
    <sink>
      <operator>Output1</operator>
    </sink>
  </stream>
  <stream>
    <source>Branch</source>
    <port>2</port>
    <sink>
      <operator>Output2</operator>
    </sink>
  </stream>
</streams>
```

Adding a Calculation Operator

The calculation operation will calculate a field and add the result as an additional field. Usually, input fields pass through an operation, but calculations can also be set to replace existing fields or simply remove them.

Steps:

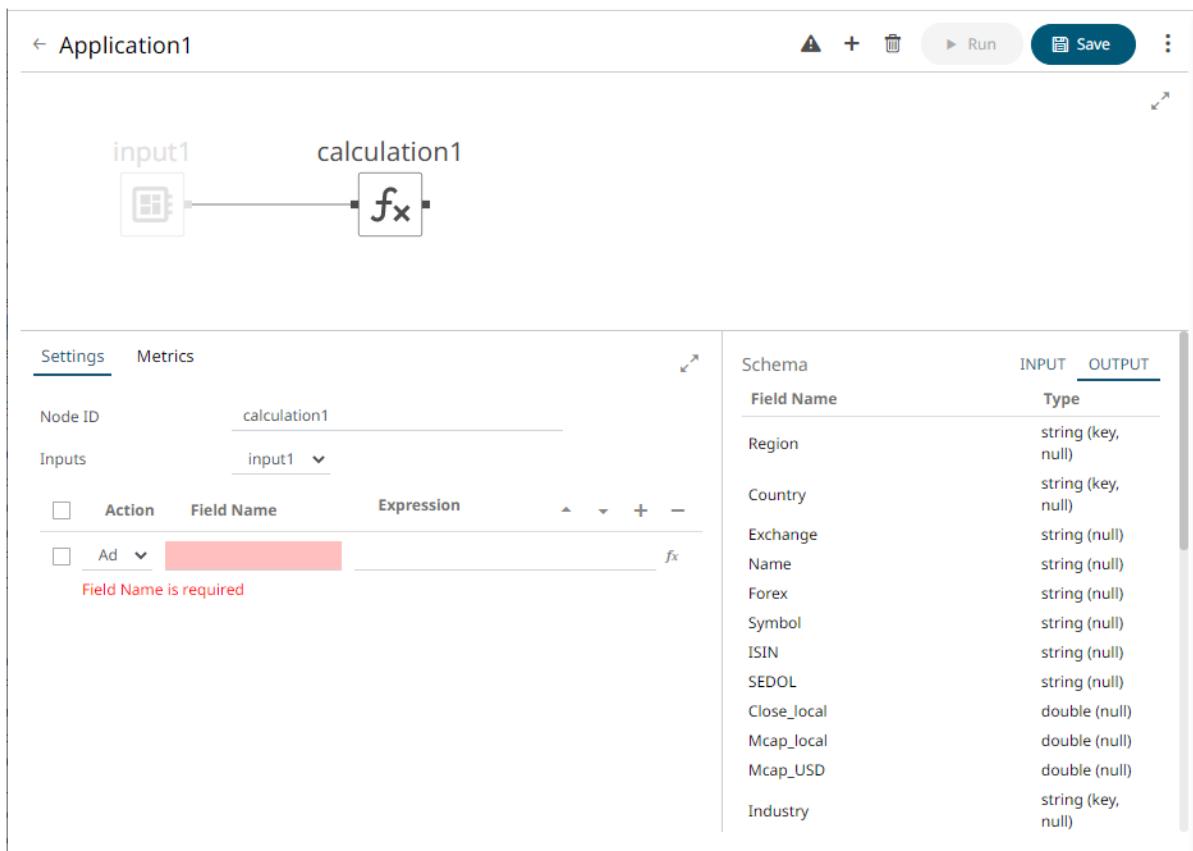


1. On the *Application* page, click  and select **Calculation** *Calculation* in the *Add Operator* pane.

calculation1



The **Calculation** node icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.



This operator has left (inbound) and right (outbound) edges that allow connection to other operators in the application.

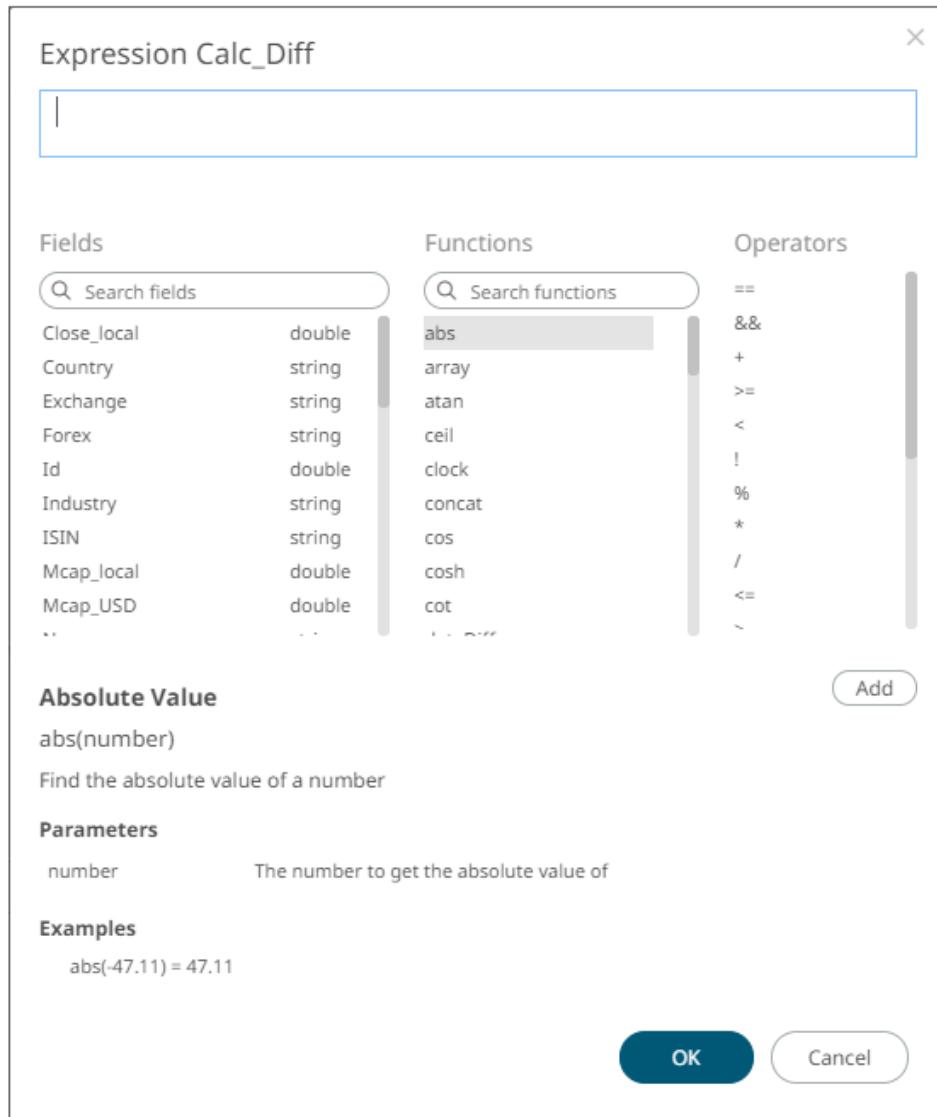
2. In the *Operator Settings* pane, define or select the following required properties:

Property	Description
Node ID	The ID of the calculation operator.
Inputs	Automatically connects to the currently-selected operator. You can select another ID of the operator that will be the source of the data in the <i>Inputs</i> drop-down list. The preview of the data (INPUT and OUTPUT) are displayed in the <i>Schema</i> pane.
Fields List	Set of fields (with actions and expressions). Enter at least one calculated field. Proceed to step 3.

3. Enter the *Field Name* and the *Expression* that will be evaluated for each incoming record.

To use the expression builder, click the **New Expression**  icon.

The *Expression <Field Name>* dialog displays.



Build an expression by double-clicking in the list of *Functions*, *Fields*, and *Columns*.

Add

You can also click on a function and operator then

To search for a particular column or function, enter it in the *Search Fields/Search Functions* box.

Functions

concat

concat

Or enter one or more characters/alphabets into the *Search Fields* box and the suggested list of columns that matched the entries will be displayed.

For example, after entering **Numeric**, the list will be displayed such as below:

Fields

Numeric

Numeric1	double
Numeric2	double

OK

Click **OK**. The new expression is added to the field instance.

<input type="checkbox"/>	Action	Field Name	Expression	▲	▼	+	-
<input type="checkbox"/>	Add	Calc_Diff	Numeric1 - Numeric2			fx	

4. Select any of the following actions: **Add**, **Replace**, or **Remove**.

<input type="checkbox"/>	Action	Field Name	Expression	▲	▼	+	-
<input type="checkbox"/>	Add	Calc_Diff	Numeric1 - Numeric2			fx	

Add

Remove

Replace

5. Click **+** to add a new field entry and repeat steps 3 and 4.

The OUTPUT schema is updated based on the added calculations.

Schema	INPUT	OUTPUT
Field Name	Type	
KeyField	string (key, null)	
Text1	string (null)	
Text2	string (null)	
Numeric1	double (null)	
Numeric2	double (null)	
Integer1	double (null)	
DateTime1	datetime (null)	
Bool1	string (null)	
DateTime2	datetime (null)	
Calc_Diff	double (not null)	
Calc_Mod	double (not null)	

You can also:

- select the topmost check box to select all of the fields
- change the order of the fields by selecting a field's check box and clicking either the  or  button
- delete a field entry in the Field List by selecting its check box and clicking 

6. [Save](#) the changes.

Supported Operators and Calculation Functions

This section lists the supported operators and calculation functions in Panopticon Streams.

NOTE

Panopticon Streams supports nullability where:

- a field may or may not allow null/empty/missing/NA values
- functions or operators may or may not allow null arguments (e.g., you can't divide seven by null)

Supported Operators

These are typically the operators that are used to create aggregation operator and calculation operator expressions.

Operator	Name	Description	Example	Nullability
!	Logical NOT	<p>! boolean</p> <p>Reverse arguments or results.</p> <p>Parameter:</p> <ul style="list-style-type: none"> boolean – A value of logical expression that can be evaluated as TRUE or FALSE. 		
!=	Not Equal To	Tests if one value is not equals to another.	12.3 != 47.11 = true	✓
%	Modulo	<p>number % divisor</p> <p>Gets the remainder from division.</p> <p>Parameters:</p> <ul style="list-style-type: none"> number – The Number to be divided. divisor – The number to divide with. 	17 % 5 = 2	
&&	Logical AND	Returns true only if both the conditions return true.	(7 > 5)&&(3 < 8) = true	
*	Multiplication	Multiply	9 * 5 = 45	
+	Addition	Adds two numbers or joins two or more text strings to produce a single piece of text.	47.11 + 9.7 = 56.81	
-	Subtraction	Subsctracts two numbers.	47 – 11 = 36	
/	Division	<p>number / divisor</p> <p>Parameters:</p> <ul style="list-style-type: none"> number – The number to be divided. divisor – The number to divide with. 	11 / 5 = 2	
<	Less than	Tests if one value is less (smaller) than another.	4 < 7 = true	
<=	Less Than or Equal To	Tests if one value is less than or equal to another.	7 <= 4 = false	
==	Equal To	Tests if one value is equals to another.	9 == 5 = false	✓
>	Greater Than	Tests if one value is greater (larger) than another.	7 > 4 = true	
>=	Greater Than or Equal To	Tests if one value is greater or equal to another.	7 >= 4 = true	
?:	Ternary IF	<p>Provides branching capability. If condition is true, then it returns exprIfTrue, else returns exprIfFalse.</p> <p>condition ? exprIfTrue : exprIfFalse</p> <p>Parameters:</p> <ul style="list-style-type: none"> condition – A value or logical expression that can be evaluated as true or false. exprIfTrue – The value to return when condition evaluates to true. 		✓

Operator	Name	Description	Example	Nullability
		<ul style="list-style-type: none"> exprIfFalse - The value to return when condition evaluates to false. 		
$^$	Exponentiation	Get the exponential value of the number.	$(2.0)^7.0 = 128.0$	
like	Like	Tests if the pattern exists in the text.	'olle' like pattern('.ll.') = true	
$ $	Logical OR	Returns true if any of the conditions return true.	$(7 > 5) (3 > 8) = \text{true}$	
-	Negation	Negation of the number.	$-(1 + 2) = -3$	
+	Positivity	Positivity of the number.		
()	Cast Operator	Explicit data type conversion. Valid casts are: <ul style="list-style-type: none"> 'int' 'real' 'text' 'time' 'bool' 	(int)sqr(Numeric1)	

Supported Calculation Functions

Operator	Name	Description	Example	Nullability
abs	Absolute Value	abs(number) Find the absolute value of a number. Parameter: <ul style="list-style-type: none"> number – The number to get the absolute number of. 	$\text{abs}(-47.11) = 47.11$	
array	Array	array(size,value) Create an array of specified size and initialize with the given value. Parameters: <ul style="list-style-type: none"> size – Size of the array. value – Array elements. 	array(5, 1.1)	✓
atan	Arc Tangent	atan(ordinate, abscissa) Get the inverse tangent of a number. Parameters: <ul style="list-style-type: none"> ordinate – The ordinate coordinate. abscissa – The abscissa coordinate (optional). 	$\text{atan}(1.0) = 0.785398$	
ceil	Ceiling	ceil(number) Returns the smallest double value that is greater than or equal to the argument and is equal to a mathematical integer.	$\text{ceil}(4.7) = 5.0$	

Operator	Name	Description	Example	Nullability
		<p>Parameter:</p> <ul style="list-style-type: none"> • number – The number to get the ceiling value of. 		
clock	Clock	Returns the current system Date/Time value.	clock()	
concat	Concatenate	<p>concat(text1,text2)</p> <p>Joins two text values.</p> <p>Parameters:</p> <ul style="list-style-type: none"> • text1 – First text to join. • text2 – Second text to join. 	concat('olle','pelle') = 'olle pelle'	
cos	Cosine	<p>cos(number)</p> <p>The natural logarithm (base e) of a real value.</p> <p>Parameter:</p> <ul style="list-style-type: none"> • number – A number to take the natural logarithm of . 	cos(0.0) = 1.0	
cosh	Hyperbolic Cosine	<p>cosh(number)</p> <p>Get the hyperbolic cosine of the number.</p> <p>Parameter:</p> <ul style="list-style-type: none"> • number – The number to get the hyperbolic cosine of. 	cosh(0.0) = 1.0	
cot	Cotangent	<p>cot(number)</p> <p>Get the cotangent of the number.</p> <p>Parameter:</p> <ul style="list-style-type: none"> • number – The number to get the cotangent of. 		
dateDiff	Date Difference	<p>dateDiff(startDatetime,endDatetime,unit)</p> <p>Calculates the difference in whole units between two time values, the return value is positive if endDatetime comes after startDatetime, and is negative if endDatetime comes before startDatetime.</p> <p>Valid units are 'DAYS', 'HOURS', 'MINUTES', 'SECONDS', "MILLISECONDS", 'MICROSECONDS', AND 'NANOSECONDS'.</p> <p>Parameters:</p> <ul style="list-style-type: none"> • startDatetime – The first (later) Date/Time value. • endDatetime – The second (earlier) Date/Time value. • unit – The time unit to use. 	dateDiff(#2019-06-17#,#2019-06-14#,'DAYS') = 3	
dateDiff2	Date Difference	Calculates the difference in fraction units between two time values, the return value is positive if endDatetime comes after startDatetime, and is negative if endDatetime comes before startDatetime.	dateDiff2(#2019-06-17T12:00:00#,#2019-06-14T00:00:00#,'DAYS') = 3.5	

Operator	Name	Description	Example	Nullability
		<p>Valid units are 'DAYS', 'HOURS', 'MINUTES', 'SECONDS', "MILLISECONDS", 'MICROSECONDS', AND 'NANOSECONDS'.</p> <p>Parameters:</p> <ul style="list-style-type: none"> • startDatetime – The first (later) time value. • endDatetime – The second (earlier) time value. • unit – The time unit to use. 		
datePart	Date Part	<p>Returns a specified part of a time value, the result is an integer value.</p> <p>Valid units are 'DAYS', 'HOURS', 'MINUTES', 'SECONDS', "MILLISECONDS", 'MICROSECONDS', AND 'NANOSECONDS'.</p> <p>Parameters:</p> <ul style="list-style-type: none"> • datetime – The Date/Time value. • part – The time part to get. 	datePart(#1973-07-23#,'YEARS') = 1973	
dateTrunc	Date Truncate	<p>dateTrunc(datetime,datetimePart)</p> <p>Truncates the specified time value to the accuracy specified by the time_part.</p> <p>Valid units are 'DAYS', 'HOURS', 'MINUTES', 'SECONDS', "MILLISECONDS", 'MICROSECONDS', AND 'NANOSECONDS'.</p> <p>Parameters:</p> <ul style="list-style-type: none"> • datetime – The Date/Time value. • datetimepart – The Date/Time part to be truncated. 	dateTrunc(#1973-07-23T12:34:56#,'YEARS') = #1973-01-01T00:00:00#	
exp	Exponential	<p>exp(number)</p> <p>Find the value of e raised to the power of a number.</p> <p>Parameter:</p> <ul style="list-style-type: none"> • number – The power that e is raised to. 	exp(0.0) = 1.0	
find	Find	<p>find(findText,withinText,startPosition)</p> <p>Returns the starting position of one string within another string, given a starting position.</p> <p>Parameters:</p> <ul style="list-style-type: none"> • findText – The text to find. • withinText – The text to search within. • startPosition – Start the search from this position (optional). 	find('ab','drabant') = 3	
floor	Floor	<p>floor(number)</p> <p>Returns the largest real value that is less than or equal to the argument and is equal to a mathematical integer.</p> <p>Parameter:</p> <ul style="list-style-type: none"> • number – The number to get the floor value of. 	floor(4.7) = 4.0	

Operator	Name	Description	Example	Nullability
get	Get	get(array,position) Gets the nth element of the array. Parameters: <ul style="list-style-type: none"> array – Array of items. position – Array element at this position. NOTE: Index of the array starts with 0.	get(array(5,1.1),1) = 1.1	✓
if	If	if(condition,exprIfTrue,exprIfFalse) The function provides branching capability. If condition is true, then it returns exprIfTrue, else it returns exprIfFalse. Parameters: <ul style="list-style-type: none"> condition – A value or logical expression that can be evaluated as true or false. exprIfTrue – The value to return when condition evaluates to true. exprIfFalse – The value to return when condition evaluates to false. 	if(a < b, a,b)	✓
ifNull	ifNull	ifNull(expression,altValue) The ifNull function returns the specified value if the expression is null, otherwise returns the expression. Parameters: <ul style="list-style-type: none"> expression – The expression to test whether it is null. altValue – The value to return if the expression is null. 	ifNull(null, 'b') = 'b'	✓
index	Index	index(array,text) Sorts the input array and outputs a lookup index. Parameters: <ul style="list-style-type: none"> array – Array of items. text – The order which the array should be sorted, the valid texts are 'asc' for ascending order or 'desc' for descending order. 	index(array(3,#1973-07-23#), 'asc')	
intpow	Integral Power	intpow(number,power) Raise a number to a power. Parameters: <ul style="list-style-type: none"> number – Number to raise a power. power - The power to raise a number to. 	intpow(2.0, 3.9) = 8.0	
invert	Invert	Inverts a lookup index. Since the index function returns an inverse permutation, you can apply the invert function which will turn it into a forward permutation (or rank).		
left	Left	left(text,numofChars)	left('olle', 3) = 'oll'	

Operator	Name	Description	Example	Nullability
		<p>Returns the leftmost characters from a string producing a new string.</p> <p>Parameters:</p> <ul style="list-style-type: none"> text – The text from which to extract characters. numofChars – Number of characters to be picked from the left. 		
len	Length	<p>len(value)</p> <p>Returns the number of characters in a string or the number of elements in an array.</p> <p>Parameter:</p> <ul style="list-style-type: none"> value – String or array to find the length of. 	len('olle') = 4	
ln	Logarithm	<p>ln(number)</p> <p>The natural logarithm (base e) of a real value.</p> <p>Parameter:</p> <ul style="list-style-type: none"> number – A number to take the natural logarithm of. 	ln(1) = 0	
log	Logarithm	<p>log(number)</p> <p>Logarithm with base 10.</p> <p>Parameter:</p> <ul style="list-style-type: none"> number – Number of which you want the logarithm. 	log(1000.0) = 3.0	
logn	Logarithm	<p>logn(number,logBase)</p> <p>Returns the Log Based N of Input.</p> <p>Parameters:</p> <ul style="list-style-type: none"> number – Number of which you want the logarithm. logBase – Base of the logarithm. 	logn(4711.0,4711.0) = 1.0	
lower	Lower	<p>lower(text)</p> <p>Convert text to lower case.</p> <p>Parameter:</p> <ul style="list-style-type: none"> text – Text to change case to lower. 	lower('OLLE') = 'olle'	✓
max	Maximum	Maximum of the two numbers.	max(11.0, 47.0) = 47.0	
mid	Mid	<p>mid(string,startPosition,numofChars)</p> <p>Returns the characters from the middle of a text string, given a starting position and length.</p> <p>Parameters:</p> <ul style="list-style-type: none"> string – The original string. startPosition – Starting position in string. numofChars – Length of the substring. 	mid('olle', 2,2) = 'll'	
min	Minimum	Minimum of the two numbers.	min(47.0, 11.0) = 11.0	

Operator	Name	Description	Example	Nullability
pow	Power	pow(number,power) Raise a number to a power. Parameters: <ul style="list-style-type: none"> • number – Number to raise a power. • power – The power to raise a number to. 	pow(-2.0, 7.0) = -128.0	
proper	Proper	proper(text) Converts a text to proper case; the first letter in each word in uppercase, and all other letters in the lower case. Parameter: <ul style="list-style-type: none"> • text – The text to make as a proper case. 	proper('olle asp') = 'Olle Asp'	✓
random	Random	random(minimumValue,maximumValue) Returns a random number with a positive sign. Parameters: <ul style="list-style-type: none"> • minValue – Minimum value or a random number (optional). • maxValue - Maximum value or a random number (optional). 	random(12.0) get a floating-point random number greater than or equal to 0.0 and less than 12.0	
replaceAll	Replace All	replaceAll(string,oldText,newText) Replaces all occurrences of the pattern with the replacement string. Parameters: <ul style="list-style-type: none"> • string – The original string. • oldText – The string to be replaced. • newText – The new replacement string. 	replaceAll('axa', 'a', 'b') = 'bx'	✓
replaceFirst	Replace First	replaceFirst(string,oldText,newText) Replaces the first occurrence of the pattern with the replacement string. Parameters: <ul style="list-style-type: none"> • string – The original string. • oldText – The string to be replaced. • newText – The new replacement string. 	replaceFirst('axa', 'a', 'b') = 'bxa'	✓
right	Right	right(text,numofChars) Returns the rightmost characters from a string producing a new string. Parameters: <ul style="list-style-type: none"> • text – The text from which to extract characters. • numofChars – Number of characters to be picked from the right. 	right('olle', 3) = 'lle'	
rnd	Rnd	rnd(scaleValue) Returns a random number with a positive or negative sign depending on scale value.		

Operator	Name	Description	Example	Nullability
		<p>Parameter:</p> <ul style="list-style-type: none"> • <code>scaleValue</code> – Positive <code>scaleValue</code> will result in a number that is maximum up to it and the negative <code>scaleValue</code> will result in a number that is minimum to it. 		
round	Round	<p>round(number,digits)</p> <p>Round a number to a given number of digits.</p> <p>Parameters:</p> <ul style="list-style-type: none"> • <code>number</code> – The number to round. • <code>digits</code> – The place at which number should be rounded (optional). 	round(47.11) = 47.0	
set	Set	<p>set(array,position,newvalue)</p> <p>Sets the nth element of the array.</p> <p>Parameters:</p> <ul style="list-style-type: none"> • <code>array</code> – Array of items. • <code>position</code> – Array element at this position. • <code>newvalue</code> – New value of the element. <p>NOTE: Index of first element starts with 0.</p>	set(array(5, 1.1), 1, 2.2)	✓
sign	Sign	<p>sign(number)</p> <p>Get the sign of a number, returns one if positive, negative one if negative, and zero if zero.</p> <p>Parameter:</p> <ul style="list-style-type: none"> • <code>number</code> – The number to get the sign of. 	sign(7) = 1	
sin	Sine	<p>sin(number)</p> <p>Get the sine of the number.</p> <p>Parameter:</p> <ul style="list-style-type: none"> • <code>number</code> – The number to the sine of. 		
sinh	Sinus Hyperbolic	<p>sinh(number)</p> <p>Get the sinus hyperbolic of the number.</p> <p>Parameter:</p> <ul style="list-style-type: none"> • <code>number</code> – The number to the sinus hyperbolic of. 	sinh(0.0) = 0.0	
sort	Sort	Applies a lookup index to an array.		
sqr	Square	<p>sqr(number)</p> <p>Returns square of the number.</p> <p>Parameter:</p> <ul style="list-style-type: none"> • <code>number</code> – The number to get the square of. 	sqr(3) = 9	
sqrt	Square Root	<p>sqrt(number)</p> <p>Returns square root of the number.</p> <p>Parameter:</p>	sqrt(9.0) = 3.0	

Operator	Name	Description	Example	Nullability
		<ul style="list-style-type: none"> number – The number to get the square root of. 		
tan	Tangent	tan(number) Get the tangent of the number. Parameter: <ul style="list-style-type: none"> number – The number to get the tangent of. 	tan(0.0) = 0.0	
trim	Trim	trim(text) Get the input text stripped of leading or following spaces. Parameter: <ul style="list-style-type: none"> text – The text to be stripped of leading or following spaces. 	trim(' olle ') = 'olle'	✓
trunc	Truncate	trunc(number,digits) Truncate a number to a given precision. Parameters: <ul style="list-style-type: none"> number – The number to truncate. digits – The precision of the truncation (optional and the default is 0). 	trunc(47.11) = 47.0	
upper	Upper	upper(text) Convert text to upper case. Parameter: <ul style="list-style-type: none"> text – Text to change case to upper. 	upper('olle') = 'OLLE'	

Example

```

<calculation>
  <id>Calculation</id>
  <fields>
    <field>
      <id>SquareRoot_ColumnA</id>
      <action>ADD</action>
      <expression>SquareRoot(ColumnA)</expression>
    </field>
  </fields>
</calculation>

```

Adding a Conflate Operator

The conflate operation is used to lower the frequency of updates. The conflate will retain the last records seen on the input and push them to the output stream on a fixed interval. For example, if the input is producing a high frequency data throughput, instead of processing all of these updates, a configured conflate will only push through a small set of records on a fixed interval.

Steps:

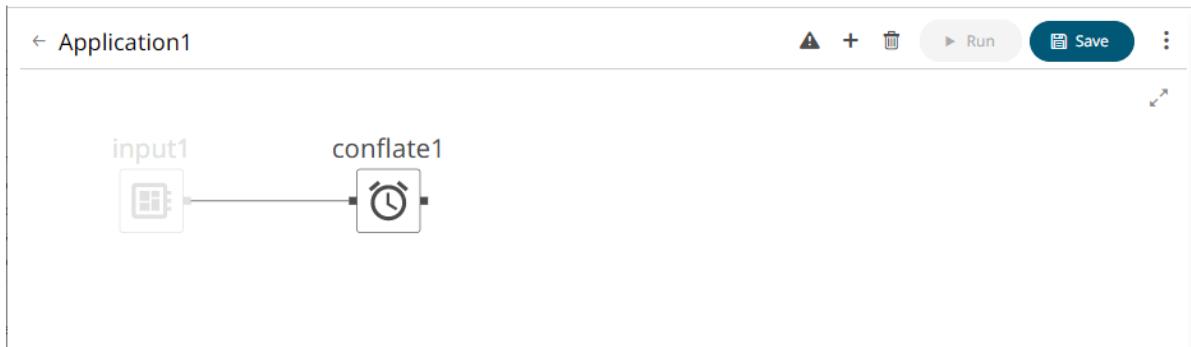


1. On the *Application* page, click  and select **Conflate**  in the *Add Operator* pane.

conflate1



The **Conflate** node icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.



The screenshot shows the *Application1* graph pane. An **input1** node (represented by a document icon) is connected to a **conflate1** node (represented by a clock icon). The *Operator Settings* pane is open, showing the following details:

Settings		Metrics	
Node ID	conflate1	Inputs	input1
Interval	1000	Timestamp	milliseconds
<input type="checkbox"/> Keep Records			

The *Schema* pane shows the data preview:

	INPUT	OUTPUT
Field Name	Type	
Region	string (key, null)	
Country	string (key, null)	
Exchange	string (null)	
Name	string (null)	
Forex	string (null)	
Symbol	string (null)	
ISIN	string (null)	
SEDOL	string (null)	
Close_local	double (null)	
Mcap_local	double (null)	
Mcap_USD	double (null)	
Industry	string (key, null)	

This operator has left (inbound) and right (outbound) edges that allow connection to other operators in the application.

2. In the *Operator Settings* pane, define or select the following properties:

Property	Description
Node ID	The ID of the conflate operator.
Inputs	<p>Automatically connects to the currently-selected operator.</p> <p>You can select another ID of the operator that will be the source of the data in the <i>Inputs</i> drop-down list.</p> <p>The preview of the data (INPUT and OUTPUT) are displayed in the <i>Schema</i> pane.</p>

Interval	The interval of which the data should be published to the output stream (in milliseconds).
Timestamp	The timestamp.
Keep Records	Check to retain or not remove flushed elements. This means the entire set of records will be flushed at each interval.

NOTE

Node ID, Inputs, and Interval are required properties.

3. [Save](#) the changes.

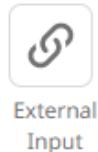
Example

```
<conflate>
  <id>Conflate </id>
  <interval>10000</interval>
</conflate>
```

Adding an External Input

Sources data directly from a Kafka topic.

Steps:



1. On the *Application* page, click  and select **External_input** in the *Add Operator* pane.

external_input1



The **External Input** node icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.

This operator serves as the initial source of the data in the application. The right (outbound) edge allows you to connect to other operators.

2. In the *Operator Settings* pane, define or select the following properties:

Property	Description
Node ID	The ID of the external input operator.
Input Type	Select the input type: STREAM , TABLE , or GLOBAL_TABLE .
Priority	Select the priority of the node's startup: <ul style="list-style-type: none"> APPLICATION – running and successful completion of the node is critical in the application startup. HIGHEST – highest priority but not critical. HIGH (Default) – high priority but not critical. STANDARD – standard priority. LOW – low priority.
Topic	The stream of records or input you will be subscribed to.
From Beginning	Select the check box to retrieve full history (from beginning to the latest) of the topic. If un-checked, only the latest messages after the application has started will be retrieved.
Partition Count	Enter the number of partitions for the Kafka topics that will be created for the External Input operator. Partitions allow you to parallelize a topic by splitting the data in a particular topic across multiple brokers wherein, each partition can be

placed on a separate machine to allow for multiple consumers to read from a topic in parallel.

NOTE: The External Input topic pulls the default partition count from the provided topic meta with generate schema call.

NOTE

Node ID, *Input Type*, and *Topic* properties are required.

3. Select the *Parser Plugin*:

- Avro

Parser Plugin	Avro	▼
Generate Output Schema	☰	
Key	+	-

- Protobuf

Parser Plugin	Protobuf	▼				
File Descriptor	No file selected	Browse				
Schema Name						
Type Name						
Generate Output Schema	☰					
<input type="checkbox"/> Source	Target	Index	Type	Date Format		
<table><tr><td>+</td><td>-</td></tr></table>					+	-
+	-					
Key	+	-				

Click **Browse** to select the **File Descriptor** (.desc file) in the *Open* dialog.

Then enter the *Schema Name* and *Type Name*.

Property	Description
Schema Name	The Protobuf schema.
Type Name	The message of Protobuf type that will be sent to Kafka.
File Descriptor	The <code>FileDescriptorSet</code> which: <ul style="list-style-type: none">• is an output of the protocol compiler.• represents a set of <code>.proto</code> files, using the <code>--descriptor_set_out</code> option.

- XML

Parser Plugin
Xml
▼

Record Xpath
(eg. //myroot/items/item)

Generate Output Schema
☰

<input type="checkbox"/>	Source	Target	XPath	Type	Date Format	
					+ -	
	Key					+ -

- JSON

Parser Plugin
Json
▼

Record Path
(eg. myroot.items.item)

Generate Output Schema
☰

<input type="checkbox"/>	Source	Target	Json Path	Type	Date Format	
					+ -	
	Key					+ -

- Text

If **Text** has been selected, confirm the **Column Delimiter** and **Text Qualifier**, and if the first row of the message includes column headings.

Parser Plugin
Text
▼

Column Delimiter
▼

Text Qualifier
▼

First Row Headings

Generate Output Schema
☰

<input type="checkbox"/>	Source	Target	Index	Type	Date Format	
					+ -	
	Key					+ -

4. You can also click the following icons:

Button	Description
	Fetch the schema of the output topic. This populates the list of columns, with the data type found from inspecting the first 'n' rows of the file.
	Add a new field entry.

	Select the check box of a field entry and click  to delete.
---	--

- Enter or select the following properties:

Property	Description
Source	The column name of the source schema.
Target	The column name of the target schema.
XPath/Json Path/Index	The column name of the target schema.
Type	The data type of the column. Can be: BOOLEAN, DATE, DATETIME, DOUBLE, FLOAT, INT, LONG, STRING, TIME.
Date Format	Date/Time format when the data type is DATE , DATETIME , or TIME .

- In the *Key* section, click  to add a key column from the data source schema. Repeat to add more.

You can also delete a key column in the list by selecting its check box and clicking .

- Save the changes.

Adding a Filter Operator

Used to filter a data source based on a predicate.

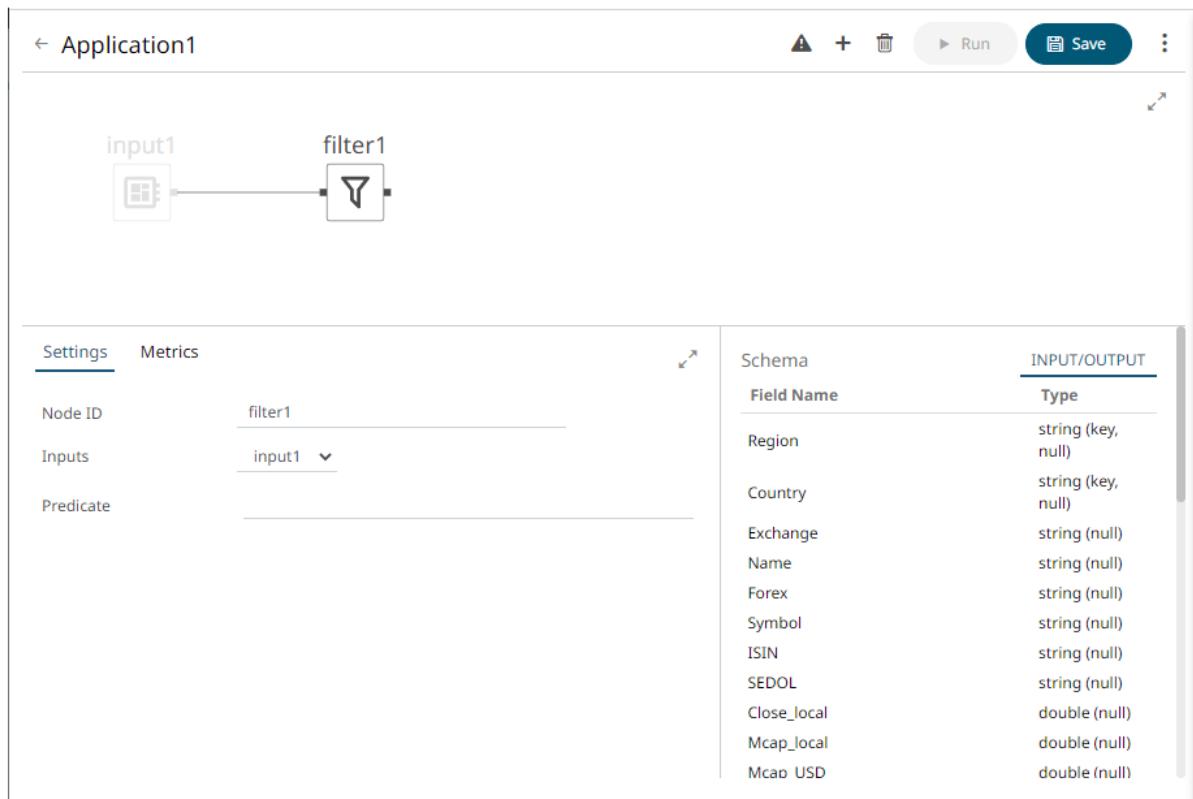
Steps:



- On the *Application* page, click  and select **Filter**  in the *Add Operator* pane.



The **Filter** node  icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.



This operator has left (inbound) and right (outbound) edges that allow connection to other operators in the application.

2. In the *Operator Settings* pane, define or select the following required properties:

Property	Description						
Node ID	The ID of the filter operator.						
Inputs	Automatically connects to the currently-selected operator. You can select another ID of the operator that will be the source of the data in the <i>Inputs</i> drop-down list. The preview of the data (INPUT/OUTPUT) is displayed in the <i>Schema</i> pane.						
Predicate	Determines whether the input record will be included or excluded. The records that will not match the predicate will be filtered out and will not be part of the output result. Example: <table border="1"> <tr> <td>Filter</td> <td>Filter</td> </tr> <tr> <td>Inputs</td> <td>FilterInput</td> </tr> <tr> <td>Predicate</td> <td>One_Day_Change >= 0</td> </tr> </table>	Filter	Filter	Inputs	FilterInput	Predicate	One_Day_Change >= 0
Filter	Filter						
Inputs	FilterInput						
Predicate	One_Day_Change >= 0						

3. [Save](#) the changes.

Example

```
<filter>
  <id>Filter</id>
  <!-- One_Day_Change >= 0 -->
  <predicate>One_Day_Change >= 0</predicate>
</filter>
```

Adding a Join Operator

Used to join data sources using common keys.

Steps:



1. On the Application page, click and select **Join** in the Add Operator pane.



The **Join** node icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.

The Graph pane shows a flow from 'input1' to 'join1', and 'input2' to 'join1'. The 'join1' node is highlighted.

Settings	Metrics	Schema	LEFT	RIGHT	OUTPUT
Node ID	join1	Field Name	Type		
Inputs	input2	Ticker	string (null)		
Time Window		Date	datetime (null)		
Join Type	INNER	Adj_Close	double (null)		
<input type="checkbox"/> Left Field	Right Field	Period_Change_proc	double (null)		
<input type="checkbox"/> Manually update output schema		Volume	double (null)		
		Turnover	double (null)		
		SP500_Change	double (null)		
		Relative_Change	double (null)		
		Holding	double (null)		

The left (inbound) edges allow you to select the input sources or operators that will be joined. The right (outbound) edge allows you to connect to other operators.

2. In the *Operator Settings* pane, define or select the following properties:

Property	Description
Node ID	The ID of the join operator.
Inputs	The left input automatically connects to the currently-selected operator. You can select another ID of the operator that will be the source of the data in the <i>Inputs</i> drop-down list. Then select the right input. The preview of the data (LEFT, RIGHT, and OUTPUT) are displayed in the <i>Schema</i> pane.
Time Window	The time window for the join operation (in milliseconds).
Join Type	The type of the join: INNER , LEFT , or OUTER
Left Field	The columns from the left source that will be used to join with.
Right Field	The columns from the right source that will be used to join with.

NOTE

Node ID, *Inputs*, *Join Type*, and *Left Field* with *Right Field* properties are required.

3. In the *Fields List* section, click . The key columns of the left and right sources are automatically displayed.



Node ID	join1
Inputs	input2 input1
Time Window	1000 milliseconds
Join Type	INNER
Left Field	<input type="checkbox"/> Ticker
Right Field	<input type="checkbox"/> Region
Manually update output schema	

Repeat step 3 to add more columns.

You can also:

- select the topmost check box to select all of the fields
- delete a field entry in the Field List by selecting its check box and clicking 

The LEFT, RIGHT, and OUTPUT schema are displayed.

Schema	LEFT	RIGHT	OUTPUT
Field Name	Type		
Ticker	string (key, null)		
Date	datetime (null)		
Adj_Close	double (null)		
Period_Change_proc	double (null)		
Volume	double (null)		
Turnover	double (null)		
SP500_Change	double (null)		
Relative_Change	double (null)		
Holding	double (null)		

Schema	LEFT	RIGHT	OUTPUT
Field Name	Type		
Region	string (null)		
Country	string (null)		
Exchange	string (null)		
Name	string (null)		
Forex	string (null)		
Symbol	string (null)		
ISIN	string (null)		
SEDOL	string (null)		
Close_local	double (null)		
Mcap_local	double (null)		
Mcap_USD	double (null)		
Industry	string (null)		

Schema	LEFT	RIGHT	OUTPUT
Field Name	Type		
Ticker	string (key, null)		
Date	datetime (null)		
Adj_Close	double (null)		
Period_Change_proc	double (null)		
Volume	double (null)		
Turnover	double (null)		
SP500_Change	double (null)		
Relative_Change	double (null)		
Holding	double (null)		
Country	string (null)		
Exchange	string (null)		
Name	string (null)		

4. [Save](#) the changes.

Example

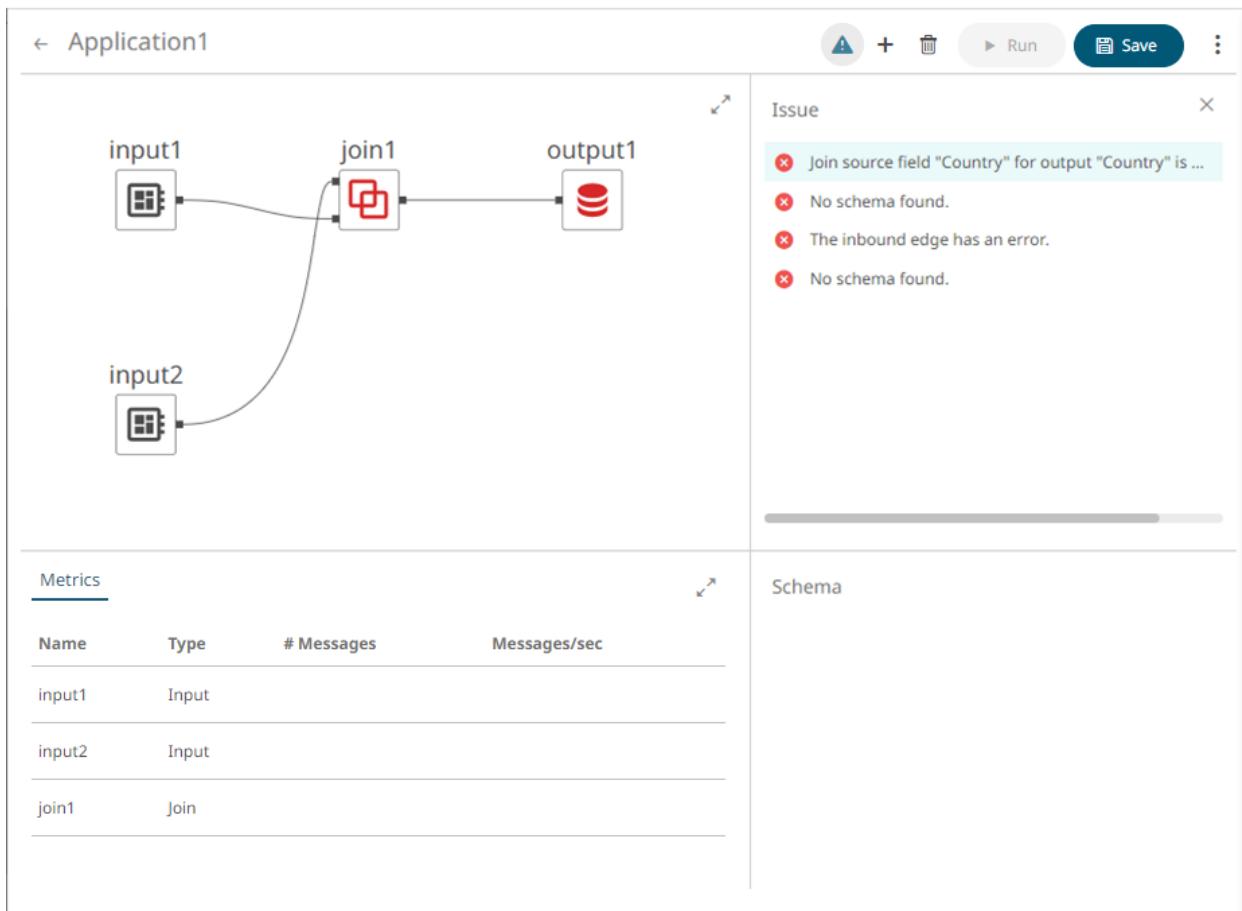
```

<join>
  <id>Join</id>
  <joinType>INNER</joinType>
  <left>
    <field>Ticker</field>
  </left>
  <right>
    <field>Ticker</field>
  </right>
  <timeWindow>1000</timeWindow>
</join>

```

Fixing Broken Joins

Changes in the input data sources may cause issues in the Join operator of an application like broken joins and output schema.



For example, if the original data source contains **Brand** and **Country** columns:

Schema	LEFT	RIGHT	OUTPUT
		Type	
Brand		string (key, null)	
Country		string (null)	

Schema	LEFT	RIGHT	OUTPUT
Field Name	Type		
Id	string (key, null)		
Make	string (null)		
Color	string (null)		

And if eventually the column **Country** is deleted in the data source, opening the application again will display:

Schema	LEFT	RIGHT	OUTPUT
Field Name	Type		
Brand	string (key, null)		

Schema	LEFT	RIGHT	<u>OUTPUT</u>
Field Name	Type		

Click **Manually Update Output Schema** to fix this error. Note that Country is no longer in the list of the Output schema.

Schema	LEFT	RIGHT	<u>OUTPUT</u>
Field Name	Type		
Id			string (null)
Make			string (null)
Color			string (null)
Brand			string (null)

Click  **Save** to apply the changes.

On the other hand, if new columns are added in the data source (e.g., if the **Country** column is added in the data source again), opening the application will display:

Schema	LEFT	RIGHT	OUTPUT
Field Name	Type		
Brand			string (key, null)
Country			string (null)

Schema	LEFT	RIGHT	OUTPUT
Field Name	Type		
Id			string (key, null)
Make			string (null)
Color			string (null)

Click **Manually Update Output Schema**. Note that Country is added in the Output schema.

Schema	LEFT	RIGHT	OUTPUT
Field Name	Type		
Id	string (null)		
Make	string (null)		
Color	string (null)		
Brand	string (null)		
Country	string (null)		



Click **Save** to apply the changes.

Adding a Metronome Input Operator

Similar with a synthetic input, this operator acts as a single timestamp field schema generator.

Steps:



1. On the *Application* page, click and select **Metronome** in the *Add Operator* pane.

metronome1



The **Metronome** node icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.

The right (outbound) edge allows you to connect to the other operators.

2. In the *Operator Settings* pane, define or select the following required properties:

Property	Description
Node ID	The ID of the metronome operator.
Topic	<p>The stream of records or input you will be subscribed to.</p> <p>Select the <i>Use Node ID</i> check box to use the value entered in the <i>Input ID</i>. Otherwise, uncheck the box and enter a new <i>Topic ID</i>.</p> <p>When adding Topic IDs, ensure they:</p> <ul style="list-style-type: none"> must be unique across an application must be specified must start with a letter (a to Z) or an underscore. Also, it can only contain letters (a to Z), numbers (0 to 9), and underscores
Partition Count	<p>Enter the number of partitions for the Kafka topics that will be created for the Metronome operator.</p> <p>Partitions allow you to parallelize a topic by splitting the data in a particular topic across multiple brokers wherein, each partition can be placed on a separate machine to allow for multiple consumers to read from a topic in parallel.</p>
Interval	The interval of which the data should be published to the output stream.
Name Field Id	The ID of the name field.
Name	The name of the ID.
Timestamp	The name of the new column that will include the timestamp.

The preview of the data (OUTPUT) is displayed in the *Schema* pane.

Schema	OUTPUT
Field Name	Type
id	string (key, not null)
timestamp	datetime (not null)

3. [Save](#) the changes.

Example

```
<metronome>
  <id>Metronome</id>
  <topic>Metronome</topic>
  <dataProducer>
    <bufferSize>0</bufferSize>
    <id>Metronome</id>
    <keyColumns/>
    <refreshPeriod>100</refreshPeriod>
  </dataProducer>
  <inputType>STREAM</inputType>
  <interval>100</interval>
  <name>Metronome</name>
  <nameFieldId>ID</nameFieldId>
  <timestampFieldId>Timestamp</timestampFieldId>
</metronome>
```

Adding a Python Transform Operator

A Python script can be executed as a data transformation step in the data pipeline.

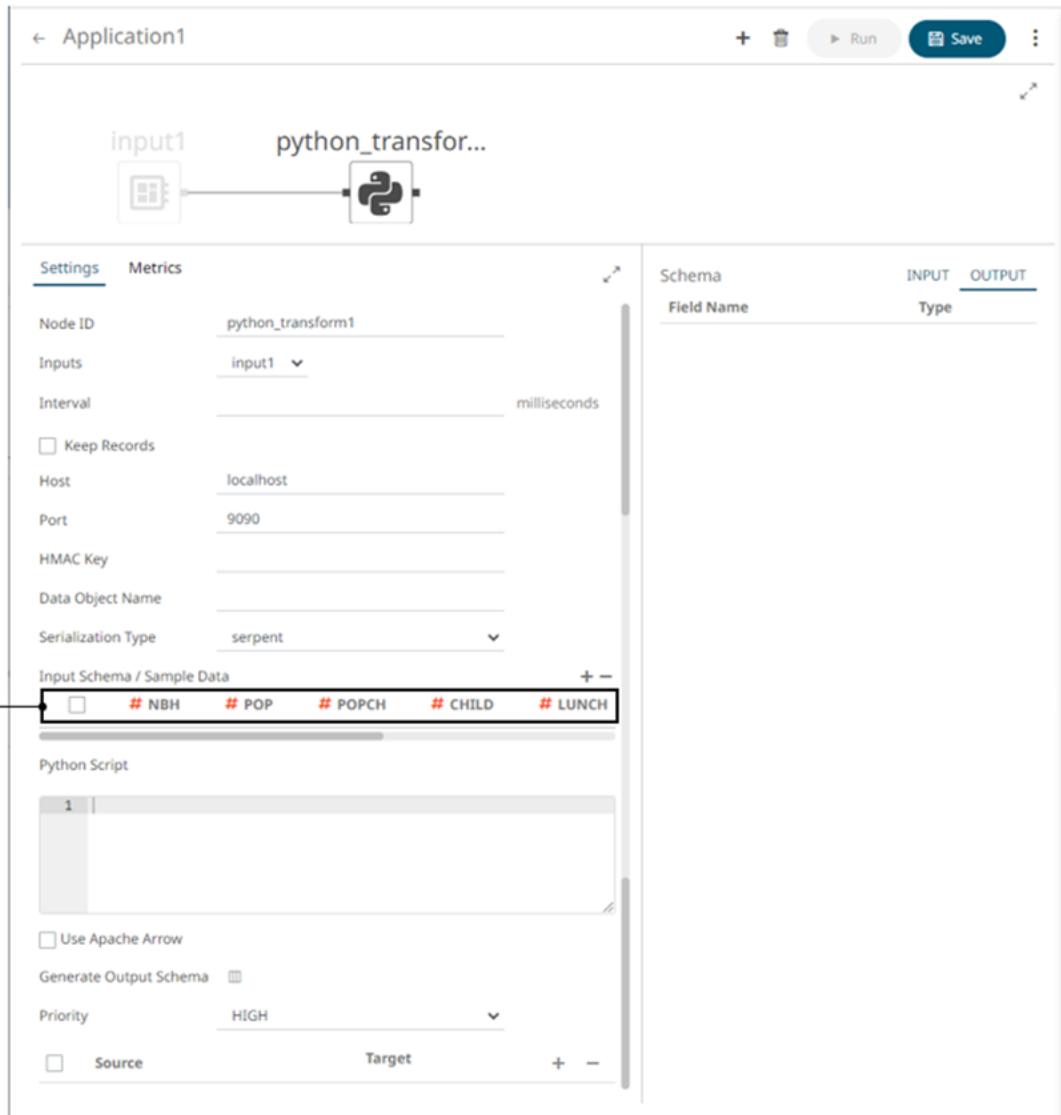
Steps:

1. On the *Application* page, click  and select **Python Transform** in the *Add Operator* pane.

python_transform...



The **Python Transform** node icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.



The right (outbound) edge allows you to connect to the other operators.

2. In the *Operator Settings* pane, define or select the following required fields:

Field	Description
Node ID	The ID of the Python Transform operator.
Inputs	The stream of records or input you will be subscribed to.
Interval	The interval of which the data should be published to the output stream (in milliseconds).
Keep Records	Check to retain or not remove flushed elements. This means the entire set of records will be flushed at each interval.
Host	Host of the Python Pyro instance.
Port	Port of the Python Pyro instance.
HMAC Key	The HMAC key that will be used to connect to the Python Pyro instance.

Data Object Name	The data structure (array of dictionaries) that Panopticon will produce, and then will be utilized by the Python Script.
Serialization Type	<p>The serialization type: Serpent or Pickle</p> <ul style="list-style-type: none"> simple serialization library based on <code>ast.literal_eval</code> faster serialization but less secure

NOTE

The *Host*, *Port*, *HMAC Key*, and *Serialization Type* fields will be hidden if their corresponding properties are set in the [Streams.properties](#) file.

Role	Corresponding Property in Streams.properties
Host	<code>connector.python.host</code>
Port	<code>connector.python.port</code>
HMAC Key	<code>connector.python.password</code>
Serialization Type	<code>connector.python.serializertype</code>

- Enter the required *Python Script* to execute on the active Pyro instance.
- Select the **Use Apache Arrow** check box to enable fast serialization of data frames in the Python transform.
- In the *Input Schema/Sample Data* section, the column names of the [Input](#) data source are displayed. In cases where there are no rows from the input data source and the Python script is not handling zero rows, you can add sample data to ensure transform is applied.

To add or manage the sample data, you can use the following icons:

Button	Description
	Add sample data for the input column names.
	Select the check box of a sample data row and click  to delete, or select the topmost check box and click  to delete all of the sample data rows.

- In the *Generate Output Schema* section, click **Generate Output Schema**  to fetch the schema of the output topic. This populates the list of columns, with the data type found from inspecting the first 'n' rows of the file.
- Select the *Priority* of the node's startup:

Priority	Description
APPLICATION	Running and successful completion of the node is critical in the application startup.
HIGHEST	Highest priority but not critical.
HIGH (Default)	High priority but not critical.
STANDARD	Standard priority.

LOW	Low priority.
-----	---------------

8. You can also click the following icons:

Button	Description
	Fetch the schema of the output topic. This populates the list of columns, with the data type found from inspecting the first 'n' rows of the file.
	Add a new field entry.
	Select the check box of a field entry and click  to delete.

9. [Save](#) the changes.

Example

```
<operators>
  <transform>
    <id>python_transform1</id>
    <transformPlugin>Python</transformPlugin>
    <transformPluginSettings/>
    <interval>1000</interval>
    <columns>
      <type>STRING</type>
    </columns>
    <maxRowsCount>0</maxRowsCount>
  </transform>
  <input>
    <id>input1</id>
    <topic>input1</topic>
    <globalTopic>UntitledApplication_0.input1</globalTopic>
    <dataProducer>
      <id>StreamSimulator_StocksStatic</id>
      <keyColumns>
        <field>Region</field>
      </keyColumns>
      <refreshPeriod>0</refreshPeriod>
    </dataProducer>
    <inputType>STREAM</inputType>
  </input>
  <output>
    <id>output1</id>
    <topic>output1</topic>
    <globalTopic>UntitledApplication_0.output1</globalTopic>
    <dataConsumer>TextOutput</dataConsumer>
  </output>
</operators>
<streams>
  <stream>
    <source>python_transform1</source>
    <sink>
      <operator>output1</operator>
    </sink>
  </stream>
  <stream>
    <source>input1</source>
    <sink>
      <operator>python_transform1</operator>
    </sink>
  </stream>
</streams>
```

Additional Best Practice Recommendations in Using Python with Panopticon

With a [Python transform](#) or the [Python connector](#) in Panopticon, it is fairly quick and easy to enter some short code snippet and use the result. However, as a project grows, and if a solution is moved into production and becomes business critical, you need more structure in your use of Python with Panopticon:

- ❑ Code should be made into functions, even if used only in one place and even if the code content is very brief. Thereby, the operations performed by each function will be contained and you avoid the risk of naming conflicts and contamination in the global environment.
- ❑ Ensure you handle exceptions in the code you write. For example, when applying a Python transform to data, you can do an initial check in your code to see if the dataset is either a zero-row or has any rows. In which case, you want to terminate and just return the empty dataset. You should also use try-except clauses, whereby in the event of an error, you could, for example, insert the error message into the designated column in your dataset

and then return it to Panopticon. As long as there is no error, the same column could contain a plain "OK" or similar as an indicator of a no-errors result.

- ❑ Functions should ideally be turned into a package. The benefit of that is mainly about the possibility of adding unit testing and automating dependency package imports.
- ❑ Your package should have unit tests that are run when building the package.
- ❑ Your package should import any other packages that you have a dependency on.
- ❑ Developing, Testing, and Debugging the package should happen in a proper IDE, where proper debugging tools and full error messages can be monitored easily. For testing and debugging, some boiler-plate code snippets and parameter input data can be prepared, to mimic the input which could come from Panopticon parameters when the code is used via Panopticon.
- ❑ In Panopticon, the code field of the transform or connector should contain an absolute minimum of code; perhaps as little as a single function call, where the function takes the necessary arguments coming from Panopticon parameters.

Adding a Rank Operator

Assign a rank number to records in the same group.

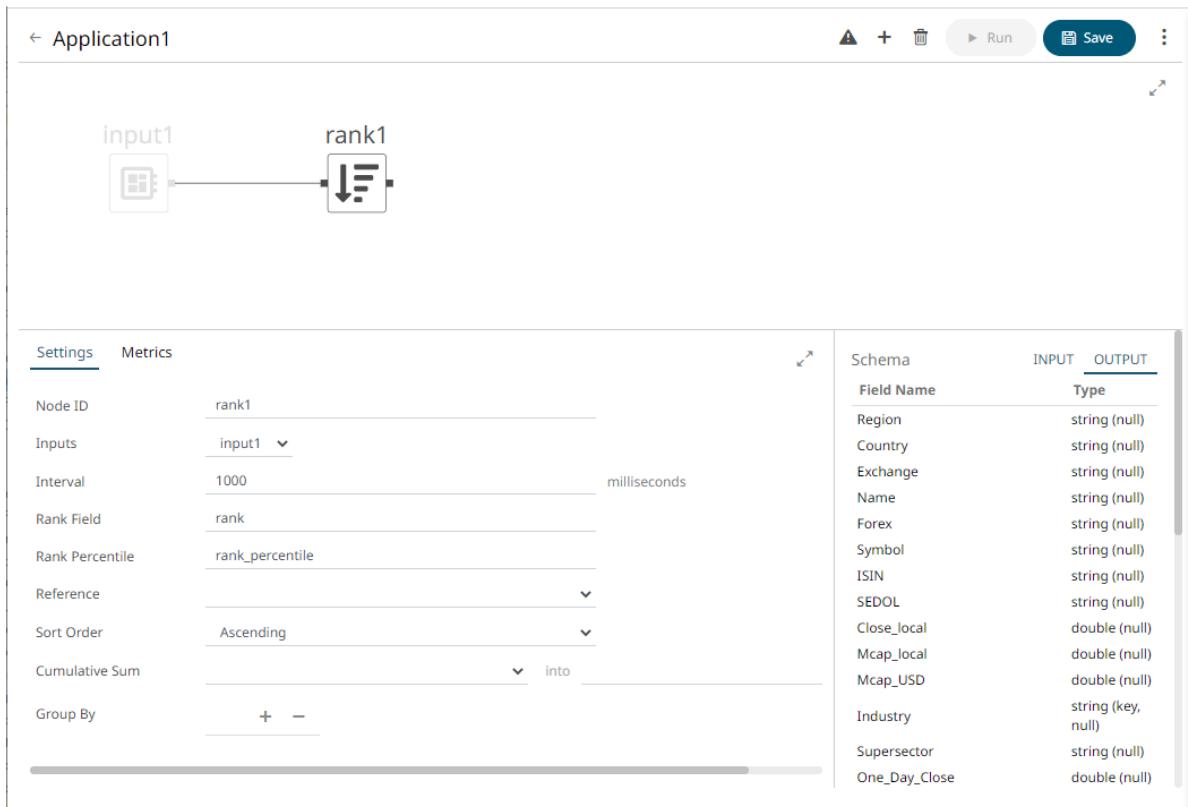
Steps:



1. On the *Application* page, click and select **Rank** in the *Add Operator* pane.



The **Rank** node icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.



The right (outbound) edge allows you to connect to the other operators.

2. In the *Operator Settings* pane, define or select the following required properties:

Property	Description
Node ID	The ID of the rank operator.
Inputs	The stream of records or input you will be subscribed to.
Interval	How often the collected data should be sorted, ranked, and output (in milliseconds)
Rank Field	The name of the rank number field in the output.
Rank Percentile	The name of the percentile field in the output. This is the rank number divided by the number of records in the group.
Reference	The input field to sort records on when ranking them.
Sort Order	The order to sort the records: ASCENDING (the lowest value gets rank one) or DESCENDING (the highest value gets rank one).
Cumulative Sum	<p>The cumulative sum based on the currently applied sort order for each Reference value.</p> <p>You can opt to specify a new value in the <i>Into</i> field. This column will be added in the Output schema.</p>
Group By	The name/IDs of the fields that the data will be grouped by. Records are ranked within each group. (Proceed to step 3.)

3. In the *Group By* section, click . A column is added in the list. Click the drop-down list to select the column that will be used to group the data.

The INPUT and OUTPUT schema are displayed.

Schema	INPUT	OUTPUT
Field Name	Type	
Region	string (key, null)	
Country	string (null)	
Exchange	string (null)	
Name	string (null)	
Forex	string (null)	
Symbol	string (null)	
ISIN	string (null)	
SEDOL	string (null)	
Close_local	double (null)	
Mcap_local	double (null)	
Mcap_USD	double (null)	
Industry	string (null)	

Schema	INPUT	OUTPUT
Field Name	Type	
Region	string (key, null)	
Country	string (null)	
Exchange	string (null)	
Name	string (null)	
Forex	string (null)	
Symbol	string (null)	
ISIN	string (null)	
SEDOL	string (null)	
Close_local	double (null)	
Mcap_local	double (null)	
Mcap_USD	double (null)	
Industry	string (null)	

Repeat step 3 to add more.

Two columns are added in the Output schema: <Rank Field> and the <Rank Percentile>. For example:

One_Day_Change	double (null)
One_Day_Change_USD	double (null)
One_Week_Change	double (null)
One_Week_Change_USD	double (null)
Two_Week_Change	double (null)
Two_Week_Change_USD	double (null)
One_Month_Change	double (null)
One_Month_Change_USD	double (null)
Two_Month_Change	double (null)
Two_Month_Change_USD	double (null)
Three_Month_Change	double (null)
Three_Month_Change_USD	double (null)
RecScore	double (null)
Id	double (null)
rank	int (null)
rank_percentile	double (null)

If you set a name for the *Cumulative Sum*, it will also be added in the Output schema.

One_Day_Change_USD	double (null)
One_Week_Change	double (null)
One_Week_Change_USD	double (null)
Two_Week_Change	double (null)
Two_Week_Change_USD	double (null)
One_Month_Change	double (null)
One_Month_Change_USD	double (null)
Two_Month_Change	double (null)
Two_Month_Change_USD	double (null)
Three_Month_Change	double (null)
Three_Month_Change_USD	double (null)
RecScore	double (null)
Id	double (null)
rank	int (null)
rank_percentile	double (null)
CumulativeSumSize	double (null)

You can also delete a column in the *Group By* list by selecting its check box and clicking  .

4. Save the changes.

Example

```
<rank>
  <id>rank1</id>
  <interval>1000</interval>
  <groupBy>
    <field>Region</field>
    <field>Country</field>
  </groupBy>
  <rankField>rank</rankField>
  <rankPercentileField>rank_percentile</rankPercentileField>
  <reference>Mcap_USD</reference>
  <sortOrder>ASCENDING</sortOrder>
  <cumSumSourceField>One_Week_Change</cumSumSourceField>
</rank>
```

Adding a Rekey Operator

Takes a stream data and changes its key. The new key can be any subset of fields from the stream.

NOTE

This operator can never be applied to a table since tables require keys to be unique and therefore, you need to specify how multiple records with the same key should be aggregated.

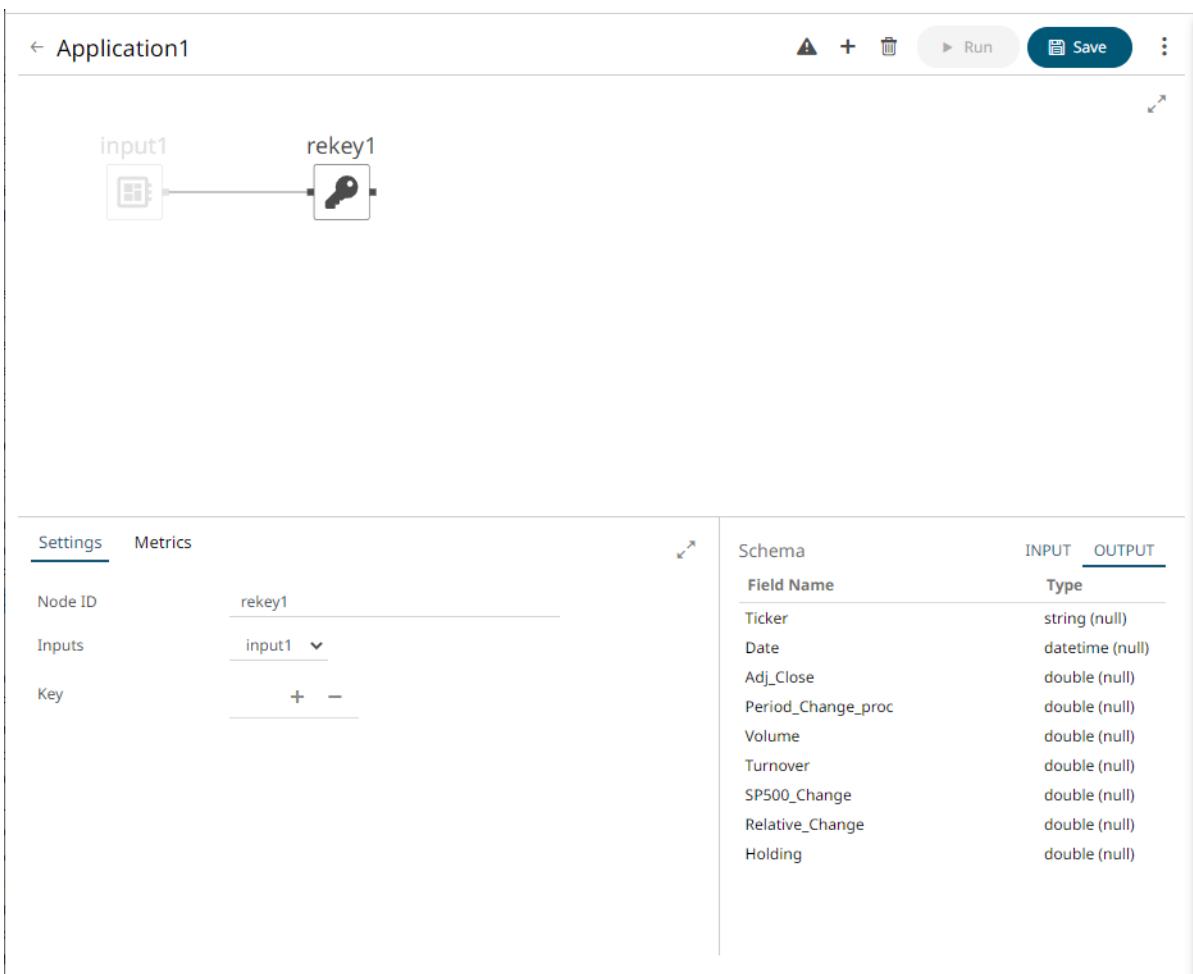
Steps:



1. On the *Application* page, click and select **Rekey** in the *Add Operator* pane.



The **Rekey** node icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.



The right (outbound) edge allows you to connect to the other operators.

2. In the *Operator Settings* pane, define or select the following required properties:

Property	Description
Node ID	The ID of the rekey operator.
Inputs	The stream of records or input you will be subscribed to.
Key	The key column. Proceed to step 3.

NOTE

Node ID, Inputs, and Key properties are required.

3. In the *Key* section, click to select the new key column in the drop-down list box from the data source schema. Repeat to add more.

You can also delete a key column in the list by selecting its check box and clicking .

The preview of the data (OUTPUT) is displayed in the *Schema* pane.

Schema	INPUT	OUTPUT
Field Name	Type	
Ticker		string (null)
Date		datetime (null)
Adj_Close		double (null)
Period_Change_proc		double (null)
Volume		double (null)
Turnover		double (null)
SP500_Change		double (null)
Relative_Change		double (null)
Holding		double (null)

4. [Save](#) the changes.

Example

```

<rekey>
  <id>rekey1</id>
  <key>
    <field>Ticker</field>
    <field>Volume</field>
  </key>
</rekey>

```

Adding a REST Transform Operator

Takes an input data frame, executes a REST call, and interprets the result which gets passed upstream.

Steps:



Rest

1. On the *Application* page, click  and select **Rest Transform** *Transform* in the *Add Operator* pane.

rest_transform1



The **REST Transform** node icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.

The right (outbound) edge allows you to connect to the other operators.

2. In the *Operator Settings* pane, define or select the following required properties:

Property	Description
Node ID	The ID of the REST Transform operator.
Inputs	The stream of records or input you will be subscribed to.
Interval	The interval of which the data should be published to the output stream (in milliseconds).
Keep Records	Check to retain or not remove flushed elements. This means the entire set of records will be flushed at each interval.

Property	Description
Authentication Type	<ul style="list-style-type: none"> Basic <div data-bbox="540 291 1237 523"> <p>Authentication Type Basic</p> <p>Url <input type="text"/></p> <p>User Id <input type="text"/></p> <p>Password <input type="text"/></p> </div> <p>Enter the <i>URL</i> of the REST API. Then enter the <i>User Id</i> and the <i>Password</i> that will be used to the connect to the REST API.</p> OAuth <div data-bbox="540 677 1286 1178"> <p>Authentication Type OAuth</p> <p>Token Url <input type="text"/></p> <p>Token Request Body <input type="text"/></p> <p>Add Access Token To Request Headers</p> <p>Url <input type="text"/></p> </div> <p>Then enter the following settings:</p> <ul style="list-style-type: none"> Token URL – The URL to retrieve the access token from. Token Request Body – The request body used for access token requests. Add Access Token To - The Access token retrieved from the <i>Token URL</i> can be added to headers, URL or request body, depending on how the REST endpoint needs the token. <div data-bbox="589 1396 861 1628"> <p>Request Headers</p> <ul style="list-style-type: none"> Request Headers Request Url Request Body </div> <ul style="list-style-type: none"> Request Header - A header is automatically added to the REST API request. Request URL - The URL needs to be manually parameterised with a <code>{access_token}</code> parameter, before calling the REST API, the parameter is replaced with the actual token. Request Body - The Request Body needs to be manually parameterised with a <code>{access_token}</code> parameter, before calling the REST API, the parameter is replaced with the actual token. URL – The URL of the REST API.

Property	Description
HTTP Method	<p>Select the appropriate HTTP method for the request from the following options:</p>  <ul style="list-style-type: none"> • GET – retrieve data • POST – add new data • PUT – replace existing data • DELETE – remove existing data
Timeout	The length of time to wait for the server response (10 to 300). Default is 10 .
Content Type	The required Content Type. Default is application/json .
Request Body	<p>The Request Body for HTTP POST.</p> <p>You can also click  to generate the Request Body.</p>

3. Select the **Multiple Records** check box to send unnamed array.
4. In the *Response Settings* section, click **Generate Output Schema**  to fetch the schema of the output topic. This populates the list of columns, with the data type found from inspecting the first 'n' rows of the file.
5. Select the *Response Type*:

- XML
- JSON

If **JSON** is selected, enter the *Record Path* which allows the identification of multiple records within the JSON document.

Response Type	<input type="text" value="Json"/>
Priority	<input type="text" value="HIGH"/>
Record Path	<input type="text" value=""/> (eg. myroot.items.item)

- Text

If **Text** is selected, confirm the **Column Delimiter**, **Text Qualifier**, and if the first row of the message includes column headings.

Response Type	Text	▼
Priority	HIGH	▼
Column Delimiter	Comma {,}	▼
Text Qualifier	<none>	▼
First Row Headings	<input checked="" type="checkbox"/>	

6. Select the *Priority* of the node's startup:

Priority	Description
APPLICATION	Running and successful completion of the node is critical in the application startup.
HIGHEST	Highest priority but not critical.
HIGH (Default)	High priority but not critical.
STANDARD	Standard priority.
LOW	Low priority.

7. Enter or select the following properties:

Property	Description
Source	The column name of the source schema.
Target	The column name of the target schema.
XPath/Json Path/Index	The column name of the target schema.
Type	The data type of the column. Can be: BOOLEAN, DATE, DATETIME, DOUBLE, FLOAT, INT, LONG, STRING, TIME.
Date Format	Date/Time format when the data type is DATE, DATETIME, or TIME.

8. You can also click the following icons:

Button	Description
+	Add a new field entry.
-	Select the check box of a field entry and click - to delete.

9. Save the changes.

Adding an R Transform Operator

An R script can be executed as a data transformation step in the data pipeline. Specifically:

- Data is retrieved from an underlying source.
- The returned data table is translated into an R data frame.
- The R data frame and supplied R Script are passed to an external R process running Rserve.
- The external Rserve process returns a resulting R data frame.
- The returned data frame is translated into a Panopticon table for visualization rendering.

For this to occur, both R and Rserve must be installed, and initialized.

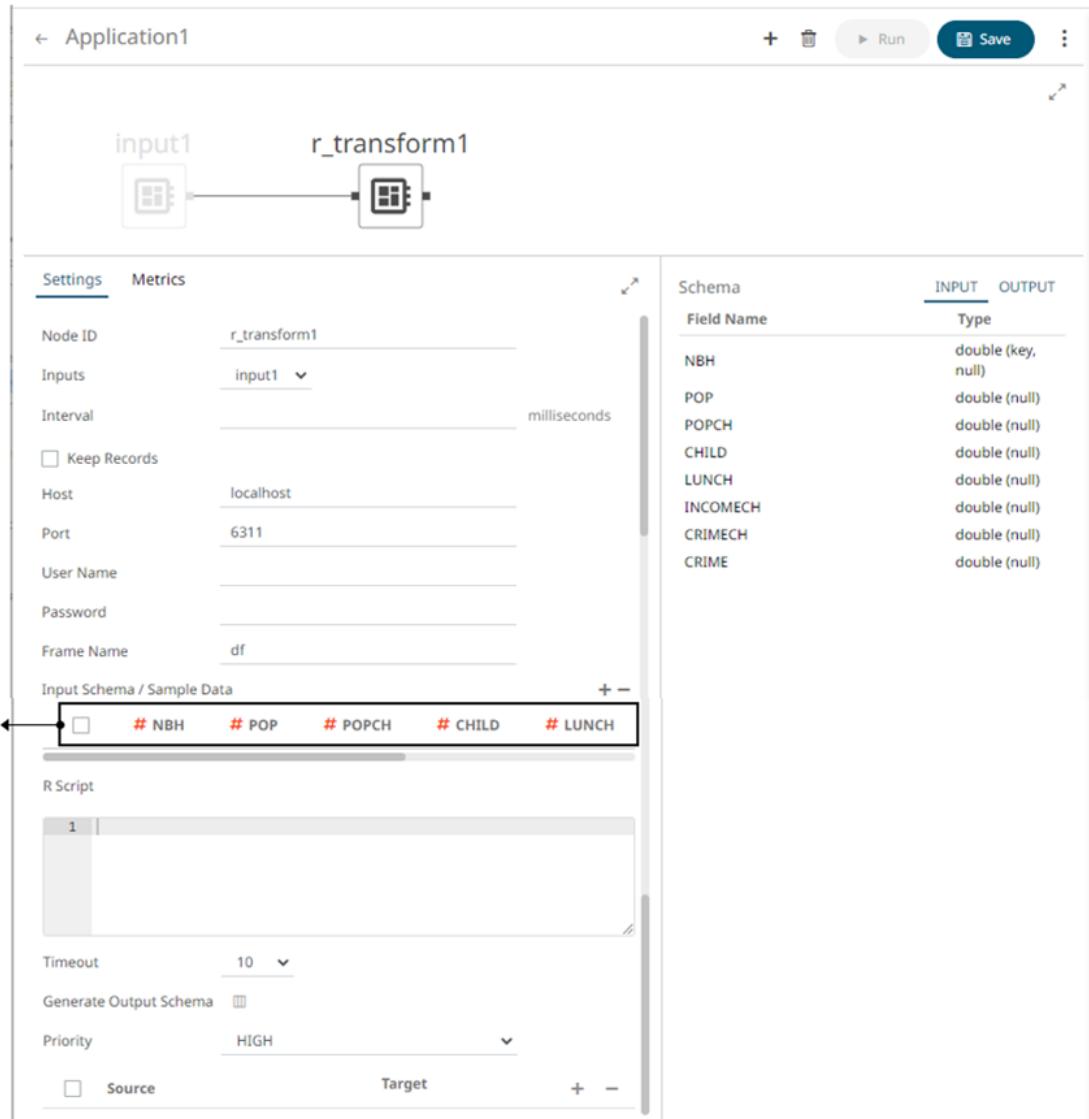
Steps:



1. On the *Application* page, click  and select **R Transform**  in the *Add Operator* pane.



The **R Transform** node icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.



The right (outbound) edge allows you to connect to the other operators.

2. In the *Operator Settings* pane, define or select the following required properties:

Property	Description
Node ID	The ID of the R Transform operator.
Inputs	The stream of records or input you will be subscribed to.
Interval	The interval of which the data should be published to the output stream (in milliseconds).
Keep Records	Check to retain or not to remove flushed elements. This means the entire set of records will be flushed at each interval.
Host	Host of the Rserve instance.
Port	Port of the Rserve instance.

User Name	The user Id if authentication is enabled on the Rserve process.
Password	The password if authentication is enabled on the Rserve process.
Frame Name	The <i>Frame Name</i> that Panopticon Streams will produce that will be utilized by the R scripts. The default is df .
R Script	The R script that reference the input frame name. Returns a data frame.
Timeout	The length of time to wait for the server response (10 to 300). Default is 10 .

3. On the *Input Schema/Sample Data* section, the column names of the [Input](#) data source are displayed. In cases where there are no rows from the input data source and the R script is not handling zero rows, you can add sample data to ensure transform is applied.

To add or manage the sample data, you can use the following icons:

Button	Description
	Add sample data for the input column names.
	Select the check box of a sample data row and click  to delete, or select the topmost check box and click  to delete all of the sample data rows.

4. In the *Generate Output Schema* section, click **Generate Output Schema**  to fetch the schema of the output topic. This populates the list of columns, with the data type found from inspecting the first 'n' rows of the file.
5. Select the *Priority* of the node's startup:

Priority	Description
APPLICATION	Running and successful completion of the node is critical in the application startup.
HIGHEST	Highest priority but not critical.
HIGH (Default)	High priority but not critical.
STANDARD	Standard priority.
LOW	Low priority.

6. You can also opt to click the following icons:

Button	Description
	Add a new field entry then enter/select the following properties: <ul style="list-style-type: none"> Source – the column of the source schema. Target – the column name of the target schema. Type - The data type of the column. Can be: BOOLEAN, DATE, DATETIME, DOUBLE, FLOAT, INT, LONG, STRING, TIME.
	Select the check box of a field entry and click  to delete.

7. Continue adding the necessary operators then [save](#) the changes in the application.

Additional Best Practice Recommendations in Using R with Panopticon

With an [R_transform](#) or the [Rserve](#) connector in Panopticon, it is fairly quick and easy to enter some short code snippet and use the result. However, as a project grows, and if a solution is moved into production and becomes business critical, you need more structure in your use of R and Rserve with Panopticon:

- ❑ Code should be made into functions, even if used only in one place and even if the code content is very brief. Thereby, the operations performed by each function will be contained and you avoid the risk of naming conflicts and contamination in the global environment.
- ❑ Ensure you handle exceptions in the code you write. For example, when applying an R transform to data, you can do an initial check in your code to see if the dataset is either zero-row or has any rows. In which case, you want to terminate and just return the empty dataset. You should also use tryCatch clauses, whereby in the event of an error or a warning, you could, for example, insert the error/warning message into the designated column in your dataset and then return it to Panopticon. As long as there is no error, the same column could contain a plain "OK" or similar as an indicator of a no-errors result.
- ❑ Functions should ideally be turned into a package. The benefit of that is mainly about the possibility of adding unit testing and automating dependency package imports.
- ❑ Your package should have unit tests that are run when building the package.
- ❑ Your package should import any other packages that you have a dependency on.
- ❑ Developing, Testing and Debugging the package should happen in a proper IDE, where proper debugging tools and full error messages can be monitored easily. For testing and debugging, some boiler-plate code snippets and parameter input data can be prepared, to mimic the input which could come from Panopticon parameters when the code is used via Panopticon.
- ❑ In Panopticon, the code field of the transform or connector should contain an absolute minimum of code; perhaps as little as a single function call, where the function takes the necessary arguments coming from Panopticon parameters.
- ❑ With R and Rserve, it should be configured to load (import) your packages on startup, which will avoid the overhead of repeated loading of the packages upon each call.

Adding a Scatter Operator

Given a record with array fields (must have the same length), the scatter operator will emit one record for each position in the array(s). This operator is similar with unpivot but on array positions instead of columns.

If the input record has an integer array field A of length N and text (non-array) field T, the operator will output N records with integer (non-array) field A and text (non-array) field T. For example, they will have values: { A[0], T }, { A[1], T }, ..., { A[N-1], T } (assuming zero-based indexing).

If the input has no array fields, the scatter operator is a no-op and will pass records through unchanged.

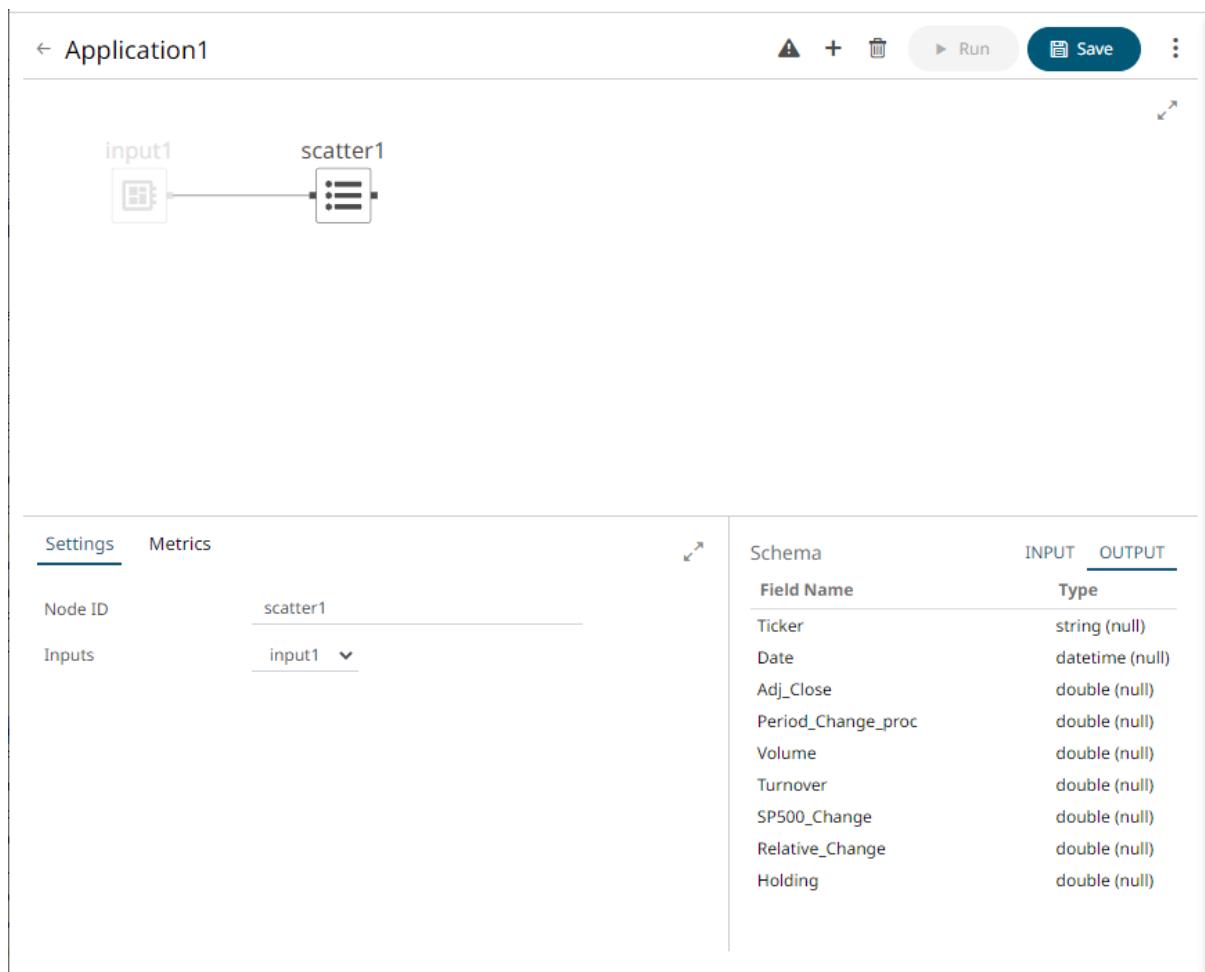
Steps:



1. On the *Application* page, click  and select **Scatter** in the *Add Operator* pane.



The **Scatter** node icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.



The right (outbound) edge allows you to connect to the other operators.

2. In the *Operator Settings* pane, define or select the following required properties:

Property	Description
Node ID	The ID of the scatter operator.
Inputs	The stream of records or input you will be subscribed to.

NOTE

Node ID and *Inputs* properties are required.

The preview of the data (OUTPUT) is displayed in the *Schema* pane.

Schema	INPUT	OUTPUT
Field Name	Type	
Ticker	string (key, null)	
Date	datetime (null)	
Adj_Close	double (null)	
Period_Change_proc	double (null)	
Volume	double (null)	
Turnover	double (null)	
SP500_Change	double (null)	
Relative_Change	double (null)	
Holding	double (null)	

3. [Save](#) the changes.

Example

```

<scatter>
    <id>scatter1</id>
</scatter>
</operators>
<streams>
    <stream>
        <source>input1</source>
        <sink>
            <operator>scatter1</operator>
        </sink>
    </stream>
</streams>

```

Adding a Table to Stream Operator

Aggregating on delta as a Table causes a change log, producing a single record. The Table to Stream operator morphs the single record back into stream.

Steps:

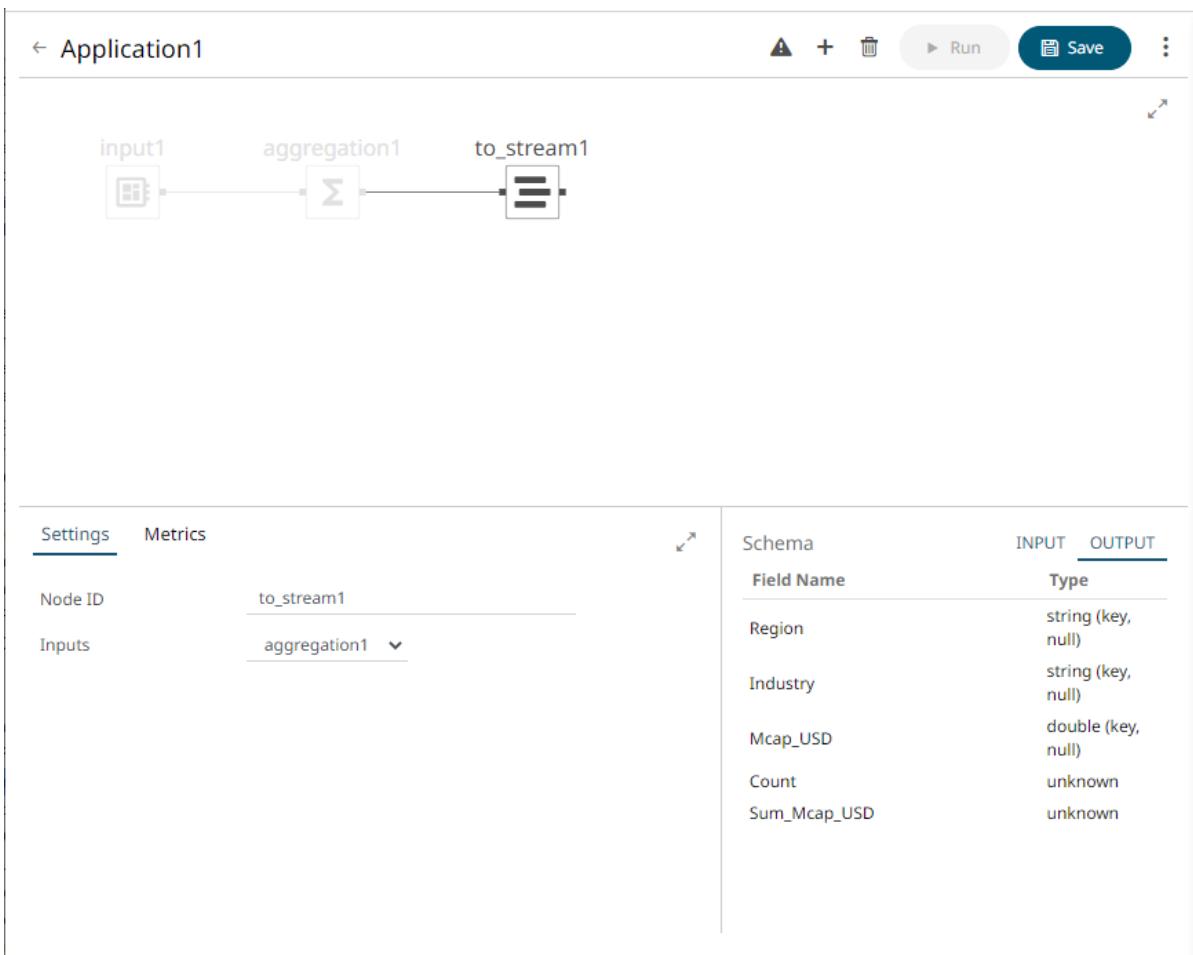


1. On the *Application* page, click  and select **To Stream** [To Stream](#) in the *Add Operator* pane.

to_stream1



The **To_stream** node icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.



The right (outbound) edge allows you to connect to the other operators.

2. In the *Operator Settings* pane, define or select the following required properties:

Property	Description
Node ID	The ID of the Table to Stream operator.
Inputs	<p>The left input automatically connects to the currently-selected operator. You can select another ID of the operator that will be the source of the data in the <i>Inputs</i> drop-down list. Ideally, this should be an aggregation operator.</p> <p>The preview of the data (LEFT, RIGHT, and OUTPUT) is displayed in the <i>Schema</i> pane.</p>

The preview of the data (OUTPUT) is displayed in the *Schema* pane.

Schema	INPUT	OUTPUT
Field Name	Type	
Region		string (key, null)
Industry		string (key, null)
Mcap_USD		double (key, null)
Count		unknown
Sum_Mcap_USD		unknown

NOTE

The data types of the aggregated columns are still unknown. The new data type will be applied once the application is saved.

For example:

Schema	INPUT	OUTPUT
Field Name	Type	
Region		string (key, null)
Industry		string (key, null)
Mcap_USD		double (key, null)
Count		long (not null)
Sum_Mcap_USD		double (not null)

3. [Save](#) the changes.

Example

```
<tostream>
  <id>to_stream1</id>
</tostream>
```

Adding a Union Operator

Used to perform a union of two streams. Both streams would need the same schema. Otherwise, the output would be the combination of both, with missing values returned as Null.

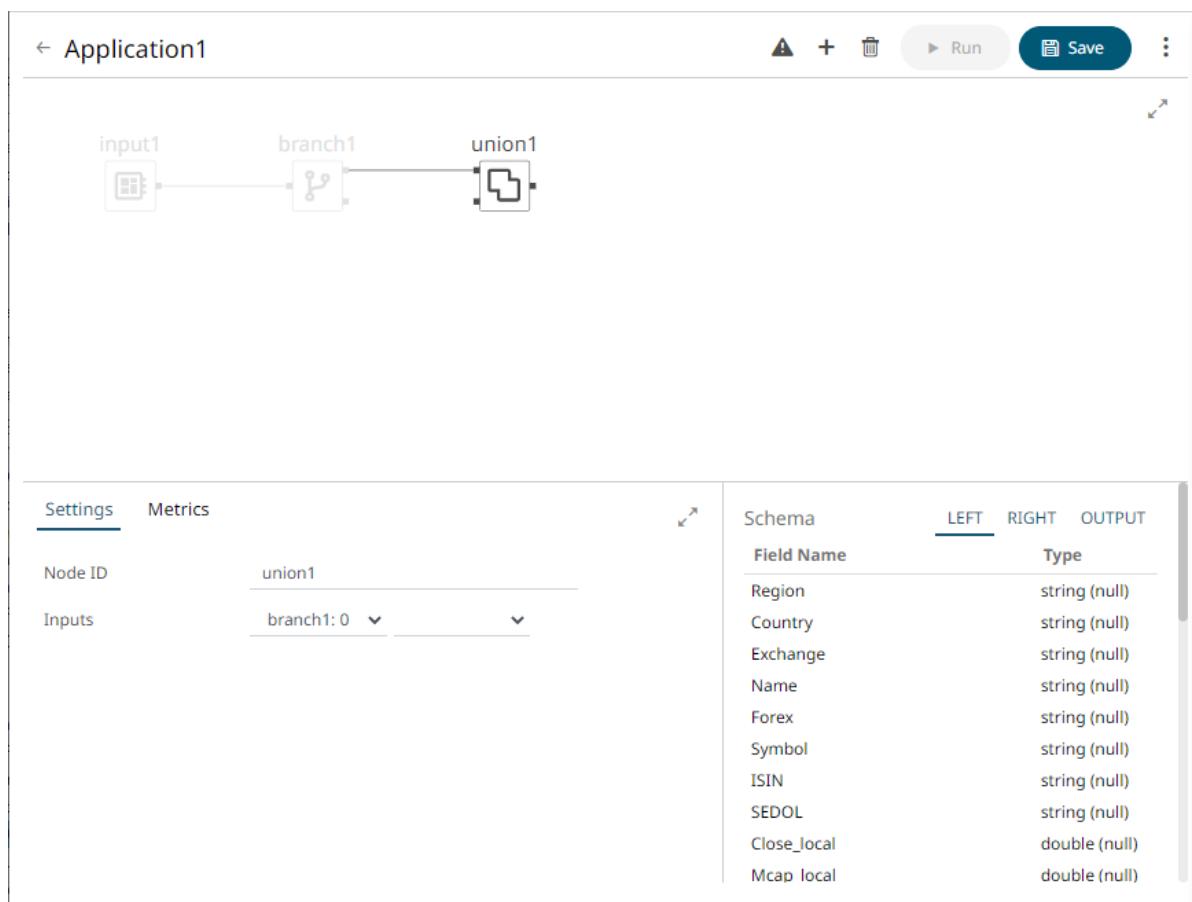
Steps:



1. On the *Application* page, click and select **Union** in the *Add Operator* pane.



The **Union** node icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.



The left (inbound) edges allow you to select the input streams. The right (outbound) edge allows you to connect to the other operators.

2. In the *Operator Settings* pane, define or select the following required properties:

Property	Description
Node ID	The ID the union operator.
Inputs	<p>The left input stream automatically connects to the currently-selected operator. You can select another ID of the operator that will be the source of the data in the <i>Inputs</i> drop-down list.</p> <p>Then select the right input stream.</p> <p>The preview of the data (LEFT, RIGHT and OUTPUT) is displayed in the <i>Schema</i> pane.</p>

Settings Metrics

Node ID union1

Inputs branch1: 0 branch1: 1

Schema	LEFT	RIGHT	OUTPUT
Field Name	Type		
Region	string (null)		
Country	string (null)		
Exchange	string (null)		
Name	string (null)		
Forex	string (null)		
Symbol	string (null)		
ISIN	string (null)		
SEDOL	string (null)		
Close_local	double (null)		
Mcap_local	double (null)		

3. [Save](#) the changes.

Adding an Output Operator

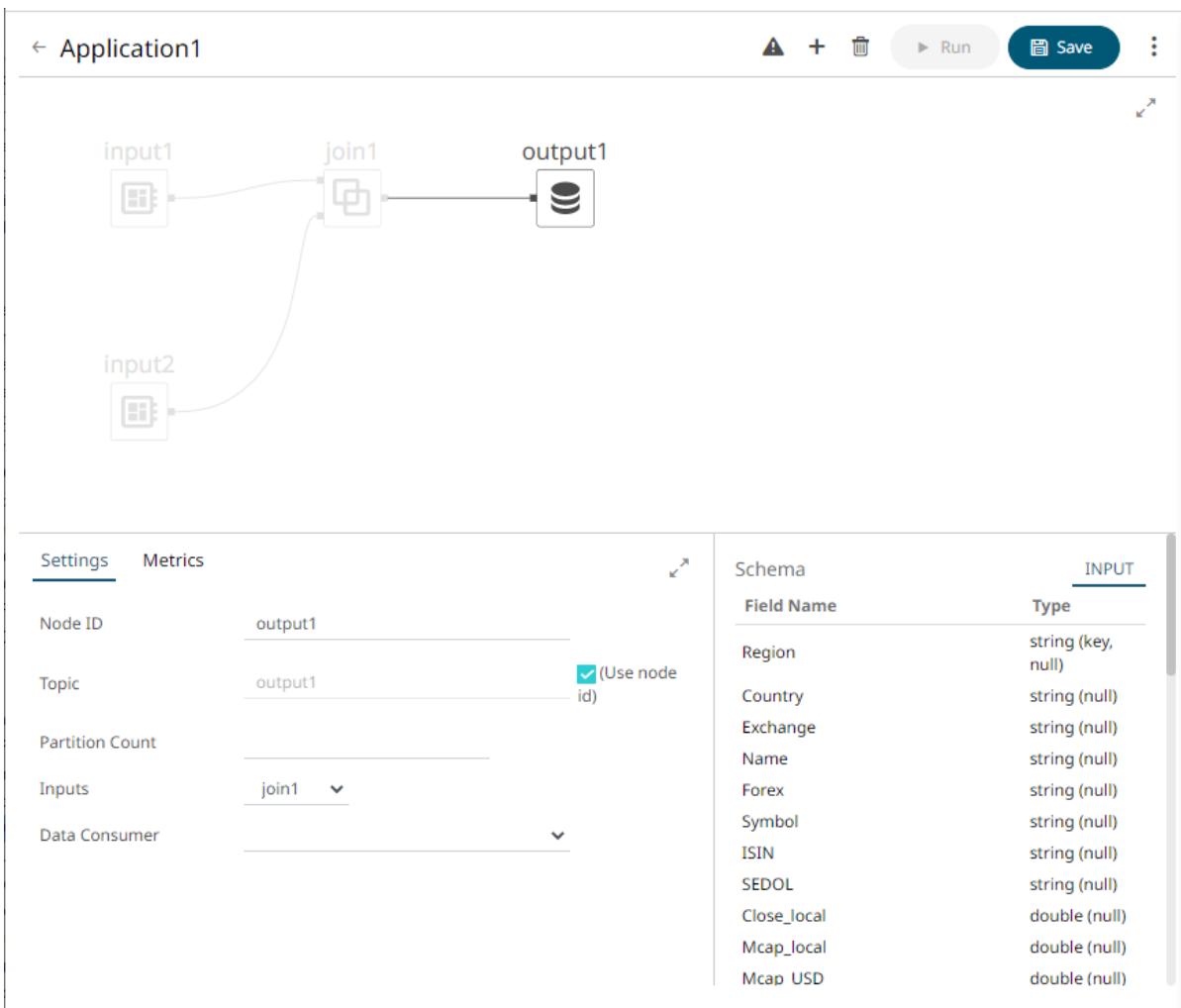
An output produces and publishes streams towards a Kafka topic or a data consumer.

Steps:

1. On the *Application* page, click  and select **Output** in the *Add Operator* pane.



The **Output** node  icon displays in the *Graph* pane, as well as the properties to be defined in the *Operator Settings* pane, and the preview of the data in the *Schema* pane.



The left (inbound) edge allows you to select the input source or operator.

2. In the *Operator Settings* pane, define or select the following properties:

Property	Description
Node ID	The ID of the output operator.
Topic	<p>The stream of records or output you will be subscribed to.</p> <p>Select the <i>Use Node ID</i> check box to use the value entered in the <i>Output ID</i>. Otherwise, uncheck the box and enter a new <i>Topic ID</i>.</p> <p>When adding Topic IDs, ensure they:</p> <ul style="list-style-type: none"> must be unique across an application must be specified must start with a letter (a to Z) or an underscore. Also, it can only contain letters (a to Z), numbers (0 to 9), and underscores
Partition Count	<p>Enter the number of partitions for the Kafka topics that will be created for the Output operator.</p> <p>Partitions allow you to parallelize a topic by splitting the data in a particular topic across multiple brokers wherein, each partition can be</p>

	placed on a separate machine to allow for multiple consumers to read from a topic in parallel.
Inputs	<p>The left input stream automatically connects to the currently-selected operator. You can select another ID of the operator that will be the source of the data in the <i>Inputs</i> drop-down list.</p> <p>The preview of the data (INPUT) is displayed in the <i>Schema</i> pane.</p>
Data Consumer	<p>Select the Data Consumer where the output will be produced or published.</p> <p>Currently, the following data consumers are supported:</p> <ul style="list-style-type: none"> • Text • JDBC Legacy • InfluxDB 1.x • Email • Kx kdb+ • REST • Apache Kafka

3. [Save](#) the changes.

ADDING APPLICATION-SPECIFIC PROPERTIES

Panopticon Streams properties can be viewed and configured in [Streams.properties](#). However, some of these server-wide properties can be overridden by adding and customizing them in an application.

Steps:

1. On the *Application* page, click  then select **Properties**.

The *Application Properties* dialog displays.

Application Properties

Key	Value	+	-
<input type="checkbox"/> output.retention.ms	900000		
<input type="checkbox"/> input.retention.ms	60000		
<input type="checkbox"/> table.retention.ms	86400000		

Close

NOTE

- Currently, the application properties are used to assign specific retention time (in milliseconds) for topic(s).
- *Partition Count* values that were added in operators in the application are displayed.

2. To add a property, click .

A new row for *Key* and *Value* entry displays.

NOTE

The *Keys* and *Values* must not be empty. Also, keys must be unique within the application property list.

3. Enter the *Key*. This is the application property to be configured.
4. Enter the corresponding *Value* of the key.

You can also opt to delete an application property entry by selecting its check box and clicking .

Close

5. Click .

Example

```
<properties>
    <!-- Keep tables alive one day -->
    <entry>
        <key>table.retention.ms</key>
        <value>86400000</value>
    </entry>
    <!-- Keep input and output streams for 1 second -->
    <entry>
        <key>input.retention.ms</key>
        <value>1000</value>
    </entry>
    <entry>
        <key>output.retention.ms</key>
        <value>1000</value>
    </entry>
    <!-- Custom retention time for InputStream topic -->
    <entry>
        <key>TimeSeries.retention.ms</key>
        <value>1111</value>
    </entry>
</properties>
```

Refer to **RetentionTimeExample** in the [Example Applications](#) section for more information.

SAVING AN APPLICATION

Saved applications (.app) are available in the PanopticonAppdata\CEP\Applications folder (i.e., c:\streamsserverdata\CEP\Applications).

Steps:

1. On the *Application* page, you can either click:

- the Save  icon

- the  icon. The context menu displays with three saving options:

- ◆  Save

Click to save the changes made in the application.

- ◆  Save as Copy

Click to make a duplicate of the application. The original name is appended with **_Copy**.

To change the *Application Name*, click on it to make it editable, then enter a new one and click .

NOTE

The *Name* or *ID* must start with a letter (a to Z) or underscore. Also, it can only contain letters (a to Z), numbers (0 to 9), and underscores.

- ◆  Revert to Saved

Click to revert to the previously-saved application settings.

NOTE

- After saving, you can start the execution of the application. You can do this either in the *Application* page or on the **Applications** tab.
- When saving an application, the color will indicate the status of the nodes:



Black – no issue



Yellow – no traffic on the topic



Red – there are definition issues. Refer to [Validating and Fixing Application Issues](#) for more information.

EDITING AN APPLICATION

NOTE

Applications that are started or running cannot be edited.

Steps:

1. On the **Applications** tab, click an application link to modify.

The *Application* page displays.

2. To change the *Application Name*, click on it to make it editable, then enter a new one and click .

NOTE

The *Name* or *ID* must start with a letter (a to Z) or underscore. Also, it can only contain letters (a to Z), numbers (0 to 9), and underscores.

3. You can also modify or add the following:

- [operators](#)
- [properties](#)

4. [Save](#) the changes.

To go back to the **Applications** tab, click  beside the application name.

NOTE

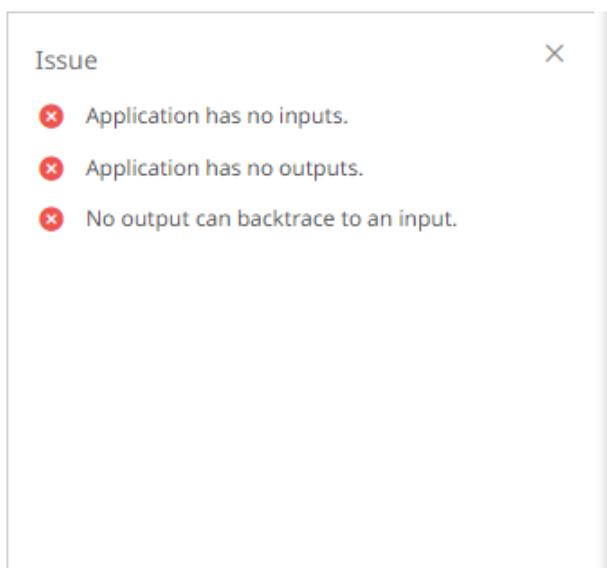
If there are changes that were not saved, a confirmation message displays asking if you will exit the *Application* page without saving. Click **Cancel** and then [save](#).

VALIDATING AND FIXING APPLICATION ISSUES

Panopticon Streams provides an error validation to help fix application issues.

Steps:

1. Click **Show Issues** . The list of *Issues* is displayed with the *Source* or operator with an error.



2. Click an issue to select.

Some possible issues include the following:

- for the input nodes, the data source is not available
- the application model parts are still not complete, or has invalid values
- for all nodes except inputs, there are faulty input definition or missing input connection
- for all nodes except outputs, there are faulty output definition or missing output connection

3. Apply the necessary changes and [save](#).

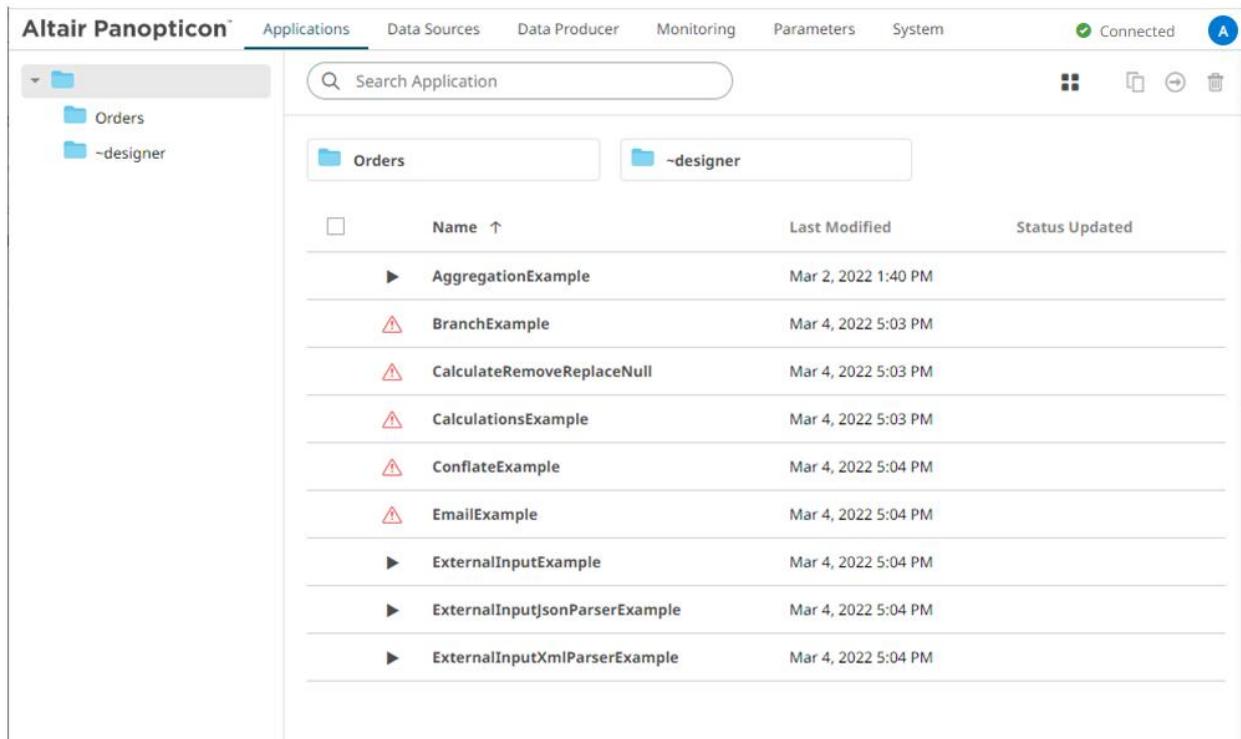
STARTING AN APPLICATION

NOTE

- Before starting an application, ensure:
 - the CEP engine has been started
 - the prerequisite [data sources are uploaded](#) on the **Data Sources** tab
 - the application model is defined correctly
- If the application is empty, the  icon is disabled. Refer to [Creating a New Application](#) for more information.

You can start an application either on the [Applications tab](#) or on the [Application page](#).

Starting an Application on the Applications Tab



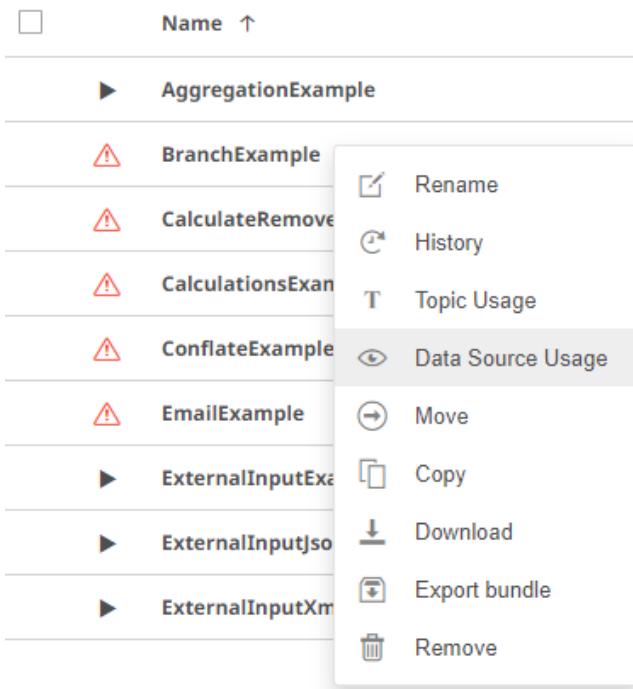
The screenshot shows the Altair Panopticon interface with the 'Applications' tab selected. The left sidebar shows a tree view with 'Orders' and '~designer' nodes. The main area displays a table of applications with columns for Name, Last Modified, and Status Updated. The table includes the following data:

	Name	Last Modified	Status Updated
▶	AggregationExample	Mar 2, 2022 1:40 PM	
⚠	BranchExample	Mar 4, 2022 5:03 PM	
⚠	CalculateRemoveReplaceNull	Mar 4, 2022 5:03 PM	
⚠	CalculationsExample	Mar 4, 2022 5:03 PM	
⚠	ConflateExample	Mar 4, 2022 5:04 PM	
⚠	EmailExample	Mar 4, 2022 5:04 PM	
▶	ExternalInputExample	Mar 4, 2022 5:04 PM	
▶	ExternalInputJsonParserExample	Mar 4, 2022 5:04 PM	
▶	ExternalInputXmlParserExample	Mar 4, 2022 5:04 PM	

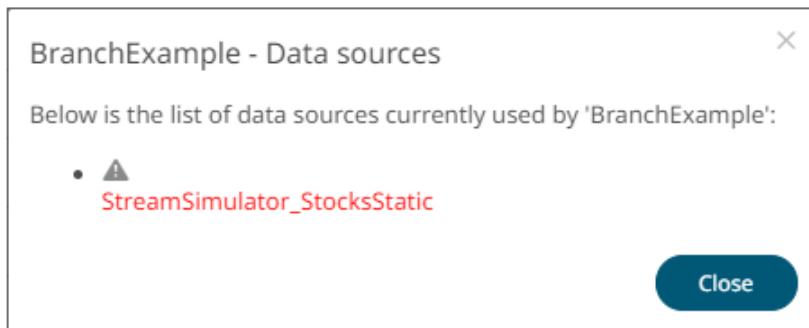
Steps:

1. To execute an application, ensure the  icon displays before the *Name*. This means the necessary data sources are already uploaded.

However, if  is displayed, right-click on the application and select **Data Source Usage** in the context menu.



The list of data sources used by the application is displayed. For example:

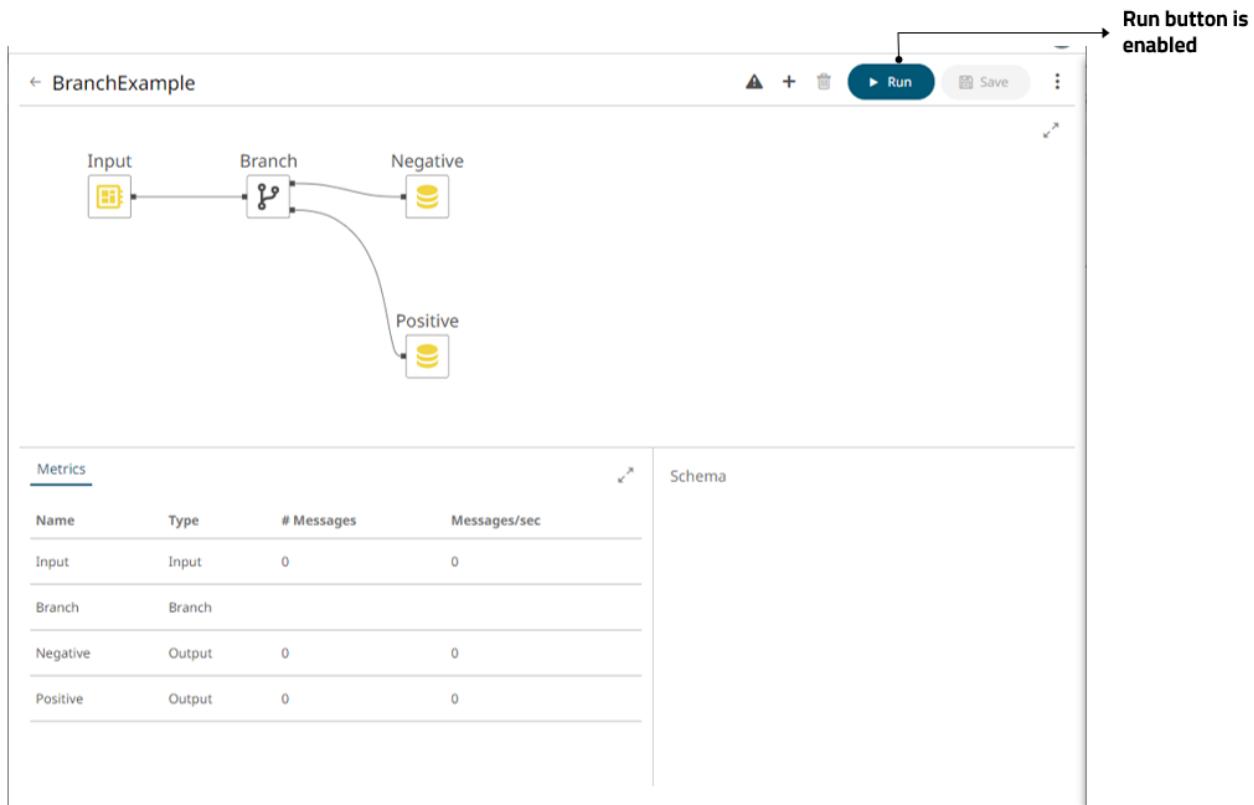


Refer to [Uploading Data Sources](#) or [Creating a Data Source](#) for more information.

2. Click . The icon changes to and the timestamp is displayed under the *Status Updated* column. The stream [topics](#) and the [data producer](#) are also generated.

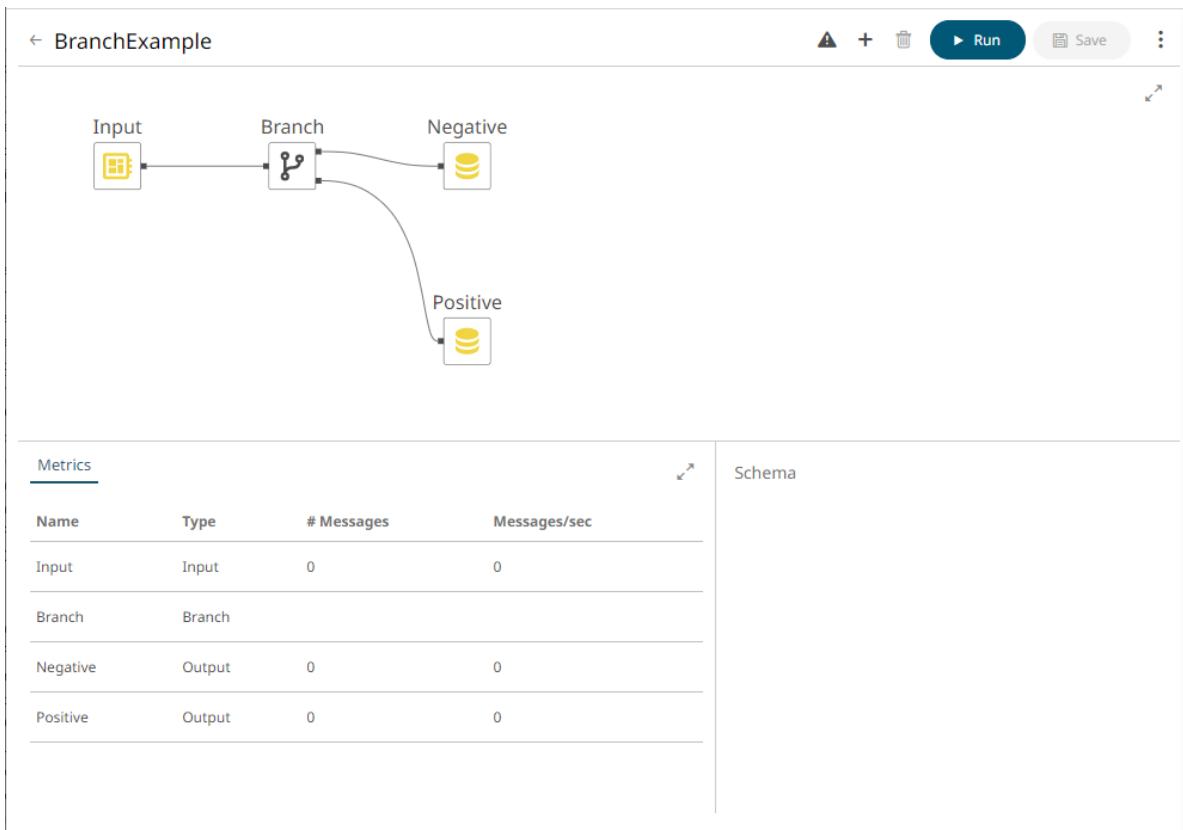
Starting an Application on the Application Page

Users with a Designer role have the ability to open and manage applications.



Steps:

1. On the **Applications** tab, click an application link to open and display it on the *Application* page.



2. Click  to run the application.

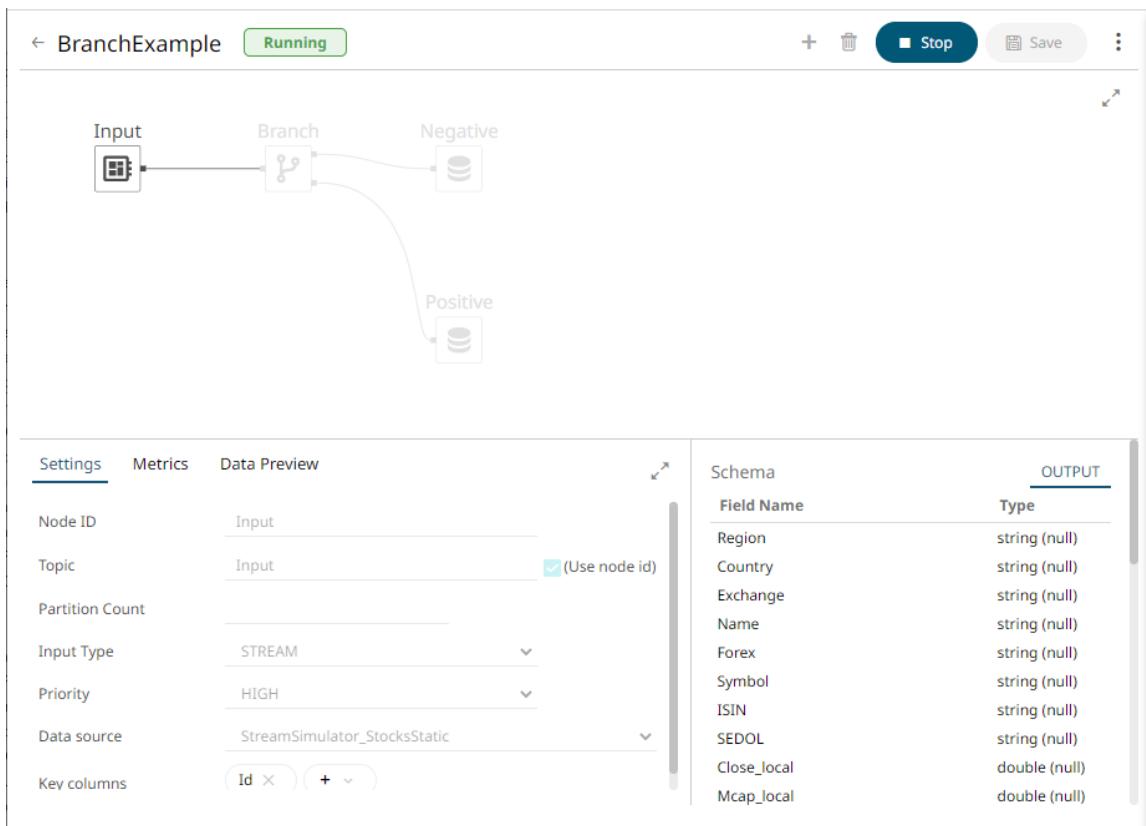
The  button changes to  and the **Running** status displays.



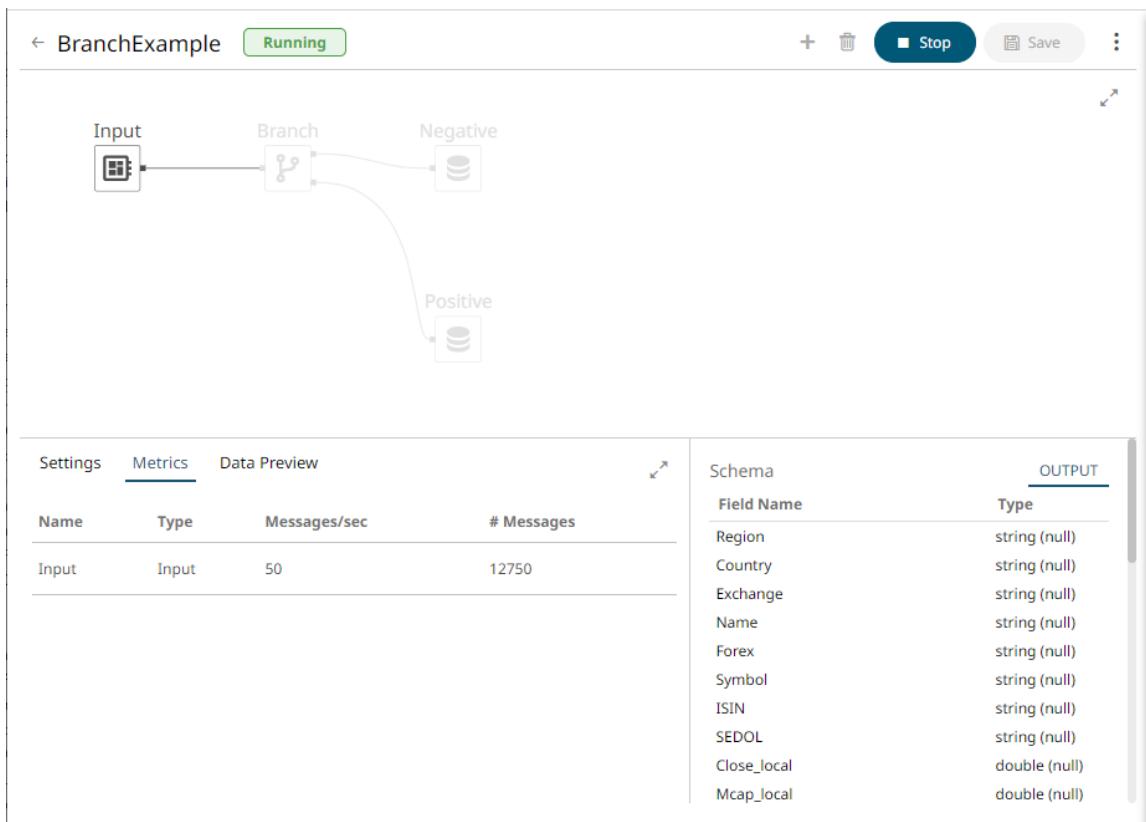
The stream [topics](#) and the [data producer](#) are generated.

3. You can also perform the following:

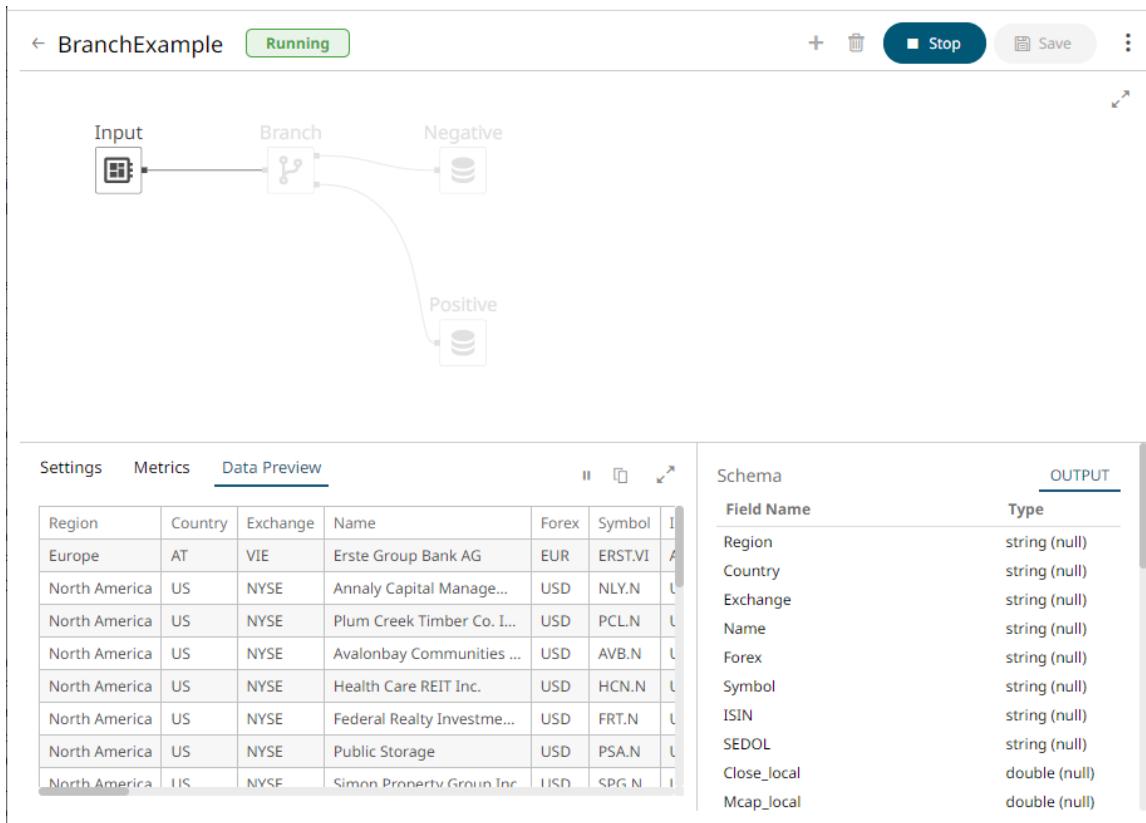
- click on a node in the *Graph* pane to display its *Operator Settings* as well as the preview of the data (OUTPUT) in the *Schema* pane



- click the **Metrics** tab to display the node's *Metrics* as well as the preview of the data (**OUTPUT**) in the *Schema* pane



- select an input or output node and click the **Data Preview** tab to display the data preview.



You can also click:

- ◆  to copy the data to a clipboard.
- ◆  to pause the update. To start the update, click .

STOPPING AN APPLICATION

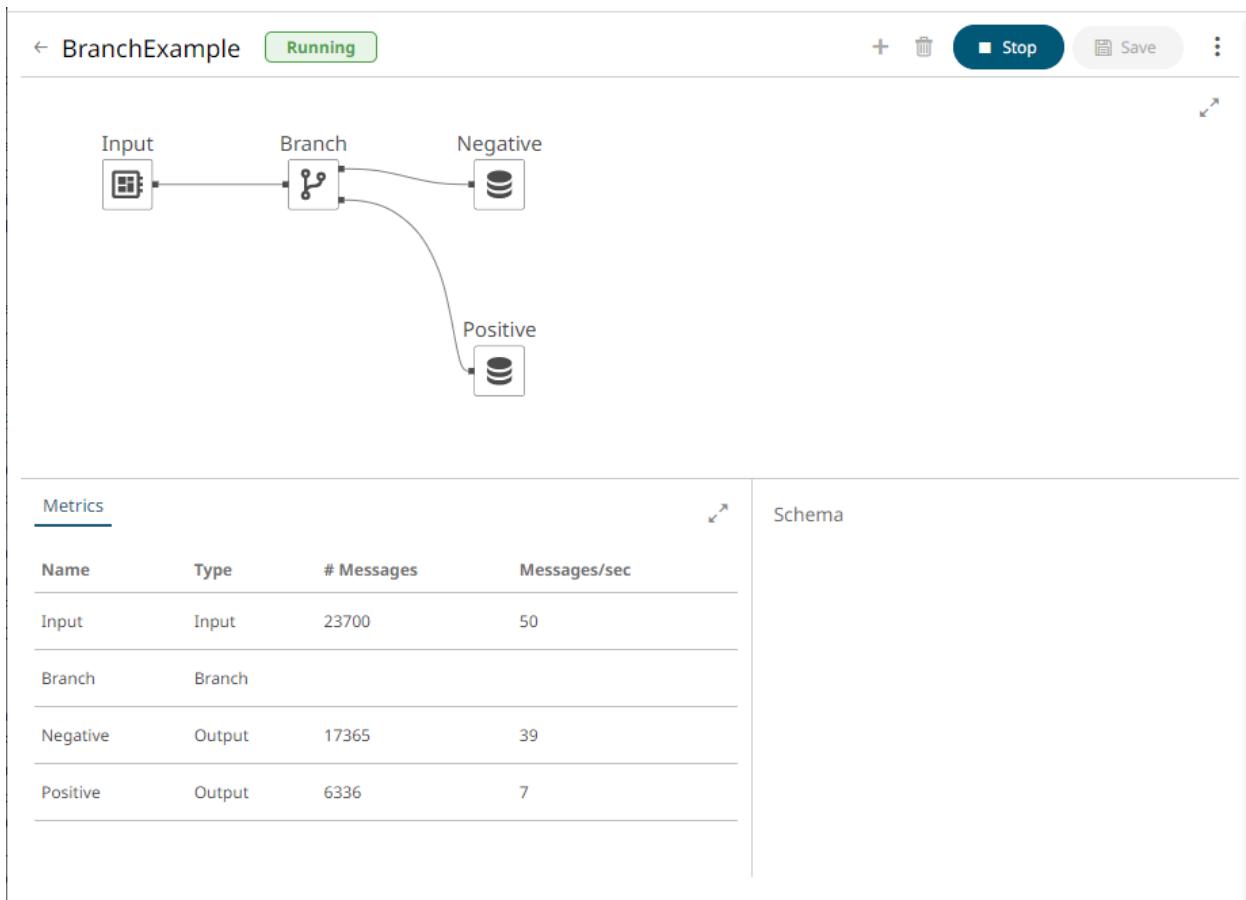
Stopping the execution of an application can either be done on the [Applications tab](#) or the [Application page](#).

Stopping an Application on the Applications Tab

 	BranchExample	Mar 4, 2022 5:24 PM	Mar 4, 2022 6:42 PM
---	---------------	---------------------	---------------------

Click  to stop the execution of the application. The icon is changed back to .

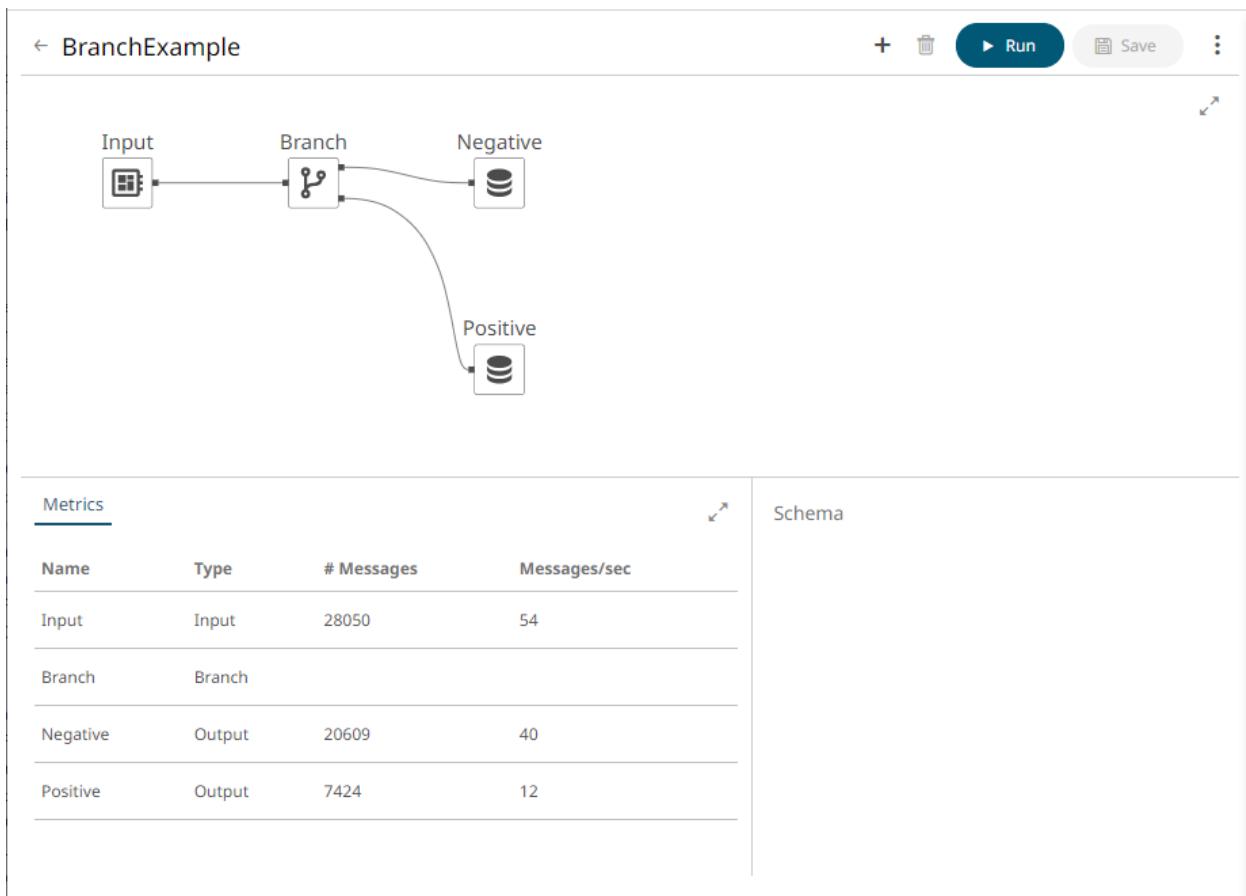
Stopping an Application on the Application Page



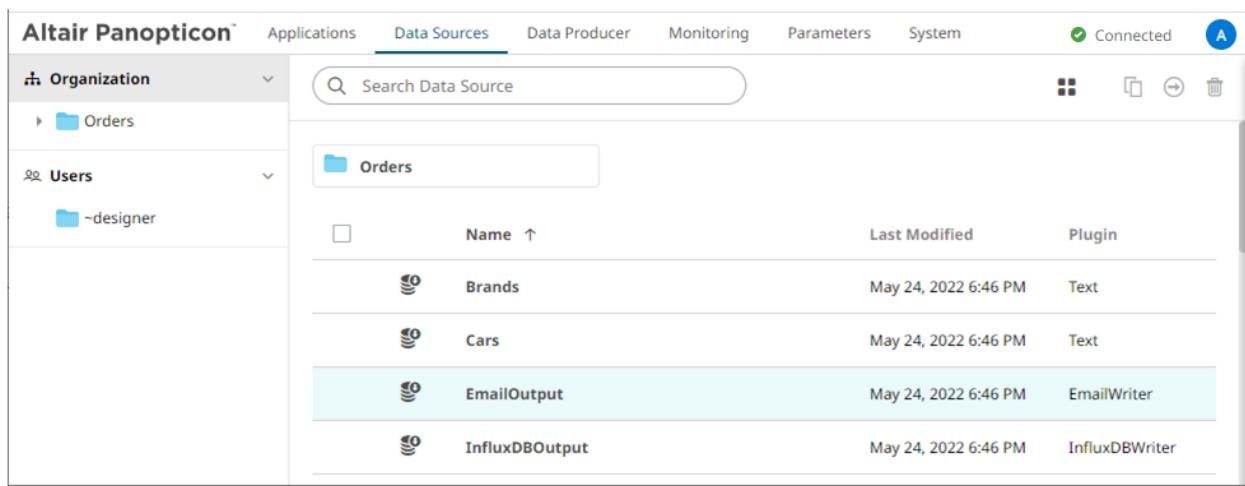
Click **Stop** to stop the execution of the application. The

changes to

.



[9] MANAGING DATA SOURCES



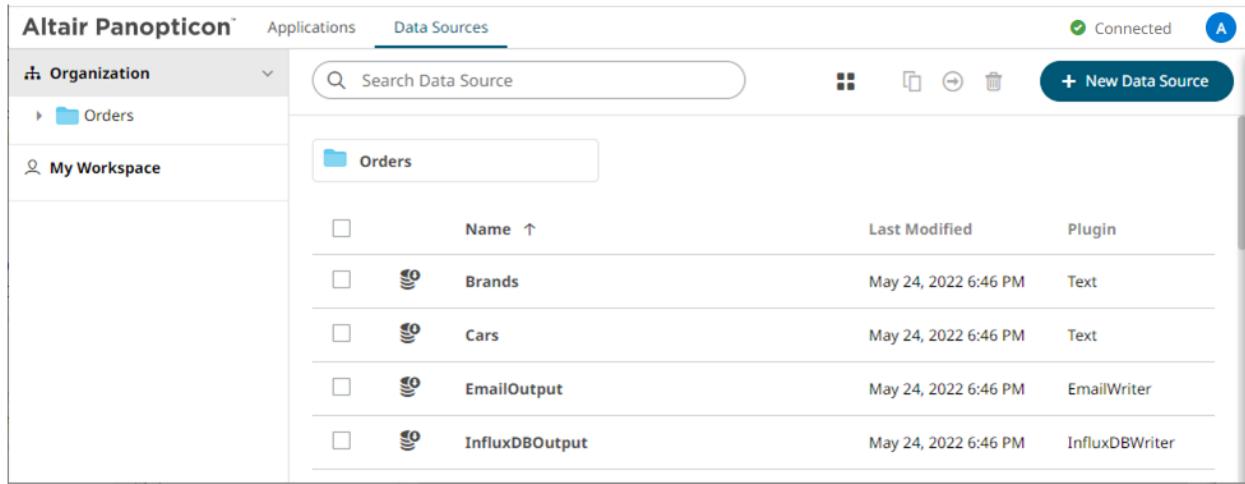
Name	Last Modified	Plugin
Brands	May 24, 2022 6:46 PM	Text
Cars	May 24, 2022 6:46 PM	Text
EmailOutput	May 24, 2022 6:46 PM	EmailWriter
InfluxDBOutput	May 24, 2022 6:46 PM	InfluxDBWriter

Figure 9-1. Data Sources page for the Administrator user role

On the **Data Sources** page, users with Administrator or Designer role can:

- [rename data sources](#)
- [View application usages](#)
- [move](#) or [copy](#) data sources to folders or subfolders to which the user has permission
- [download](#) a copy
- [remove](#) a data source

To [create](#) or [upload](#) a data source, a user must have a Designer role.



Name	Last Modified	Plugin
Brands	May 24, 2022 6:46 PM	Text
Cars	May 24, 2022 6:46 PM	Text
EmailOutput	May 24, 2022 6:46 PM	EmailWriter
InfluxDBOutput	May 24, 2022 6:46 PM	InfluxDBWriter

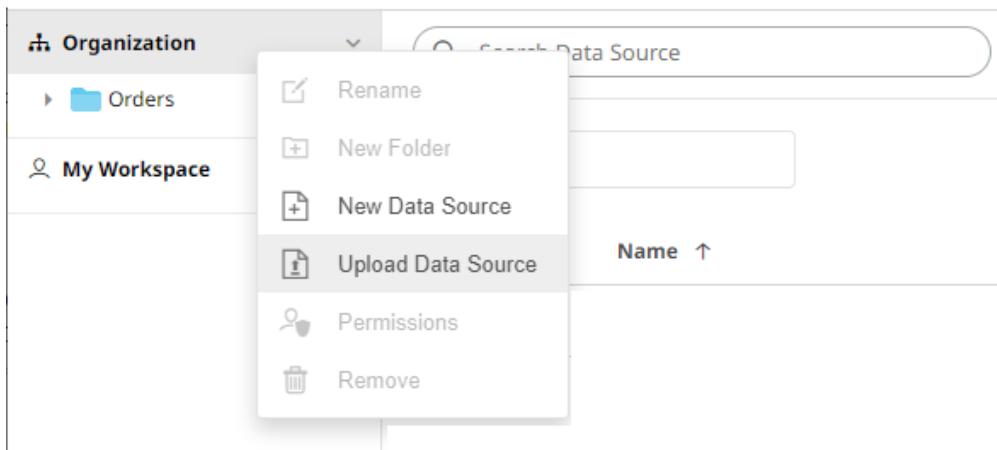
Figure 9-2. Data Sources page for the Designer user role

UPLOADING DATA SOURCES

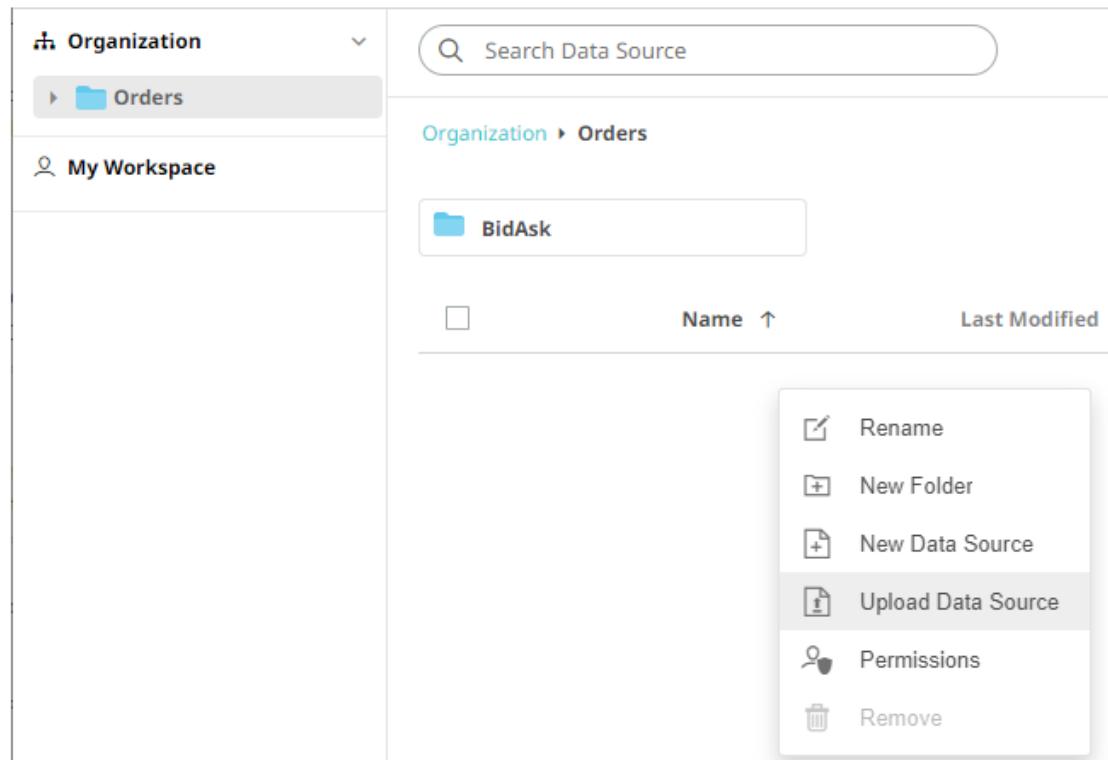
Users with a Designer role can upload data sources to folder or subfolders to which they have permission.

Steps:

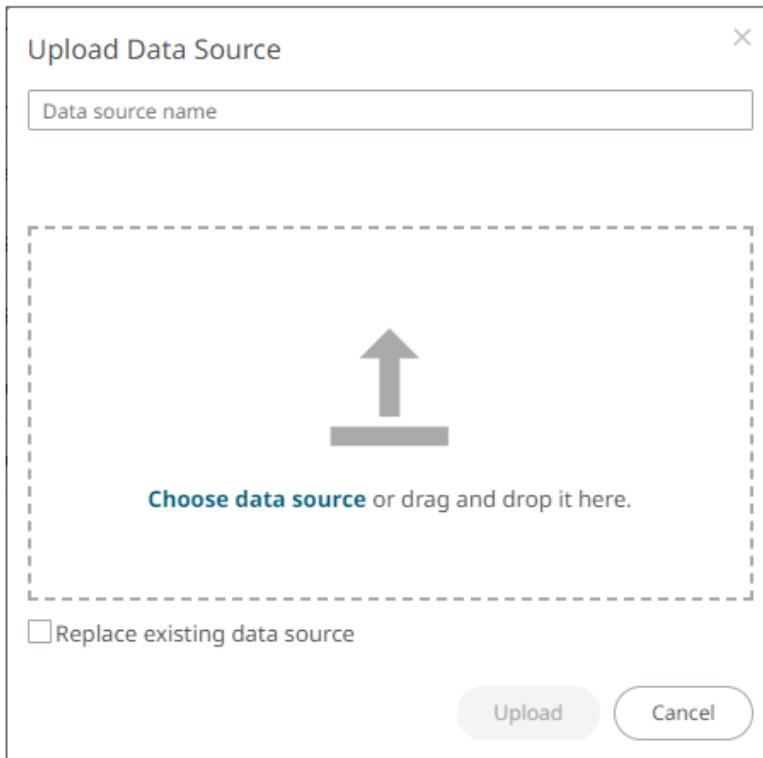
1. To upload data sources, you can either right-click a folder or subfolder then select **Upload Data Sources**:
 - on the expanded *Folder* hierarchy list



- or on the *Data Sources* pane



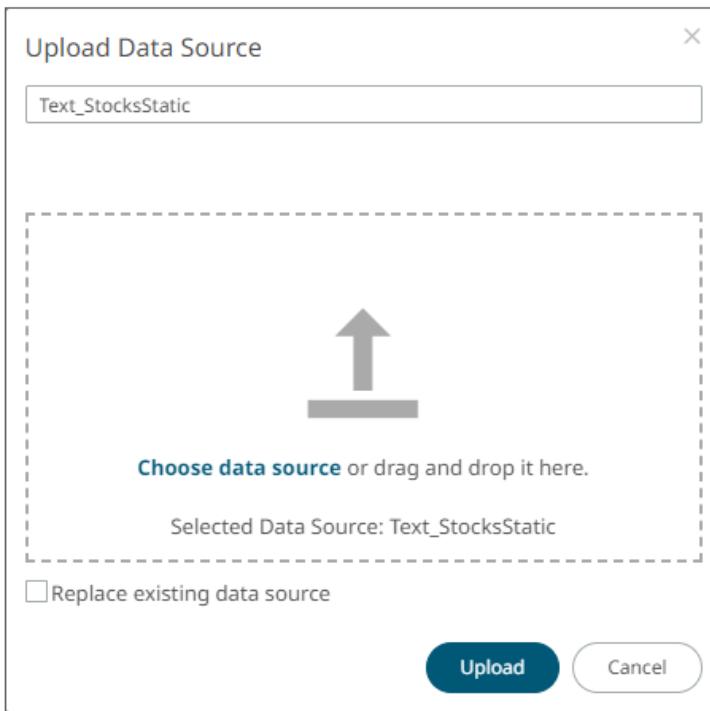
The *Upload Data Source* dialog displays.



2. To upload a data source, you can either:

- drag it from your desktop and drop in the dialog, or
- click **Choose Data Source** and select one in the *Open* dialog that displays.

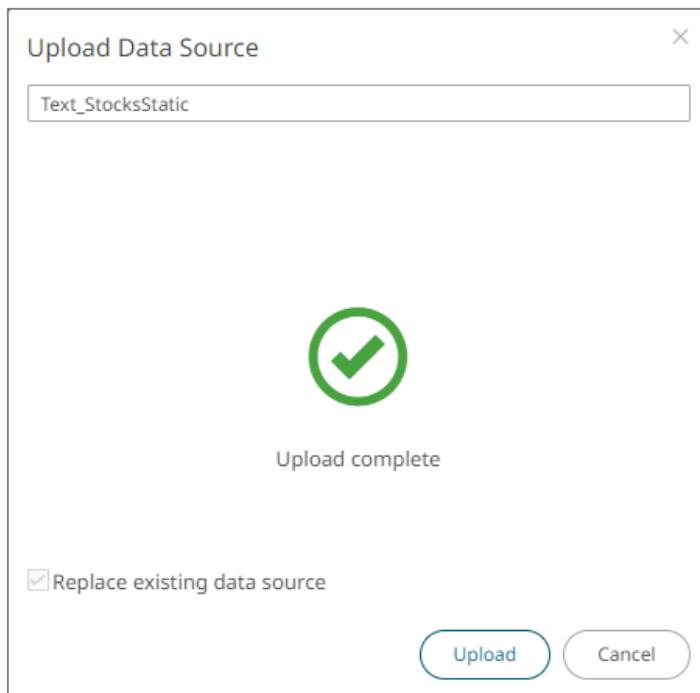
The name of the data source is displayed on the uploaded data source area and in the *Name* box.



3. You can opt to rename the data source.
4. To replace an existing data source, select the *Replace existing data source* check box.

5. Click **Upload**.

You will be notified when the data source has been uploaded.



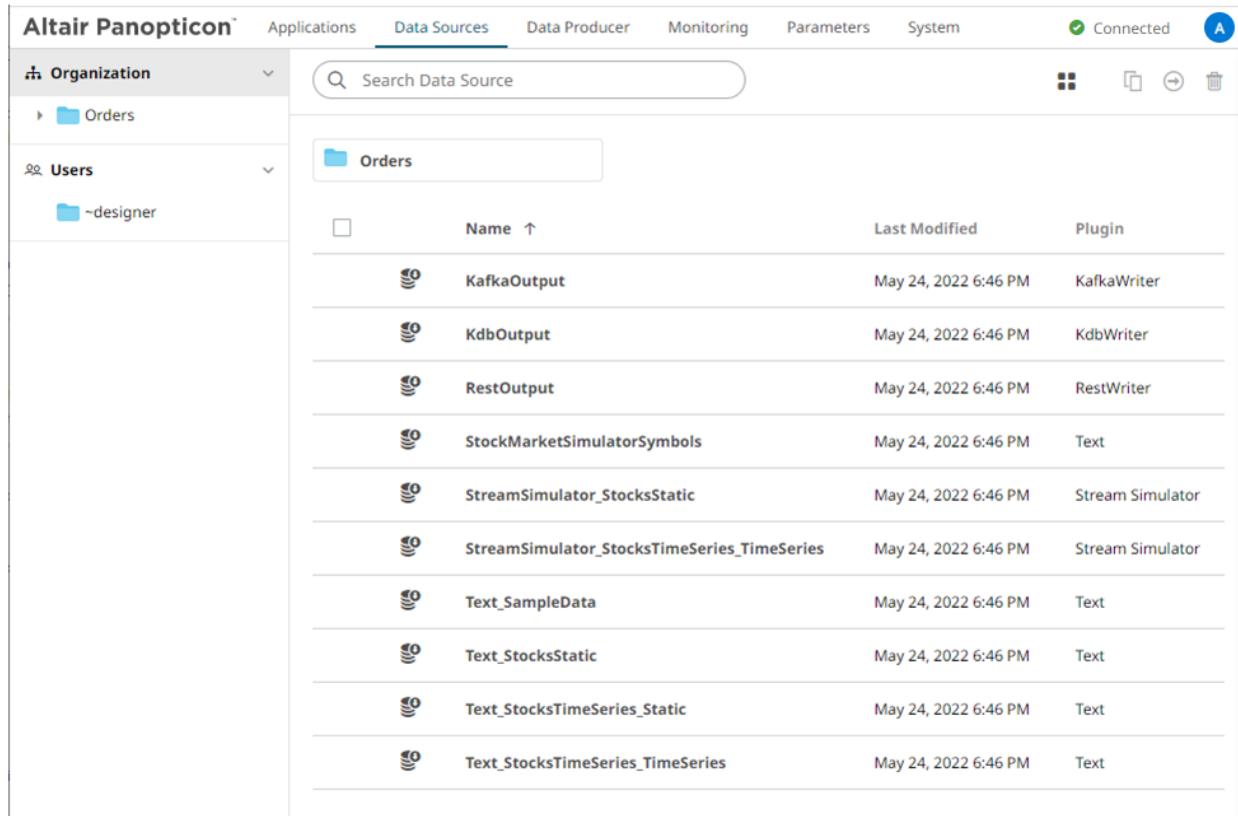
The data source is added and displayed on the **Data Sources** tab.

The screenshot shows the Altair Panopticon interface with the 'Data Sources' tab selected. The left sidebar shows a tree structure with 'Organization' expanded, showing 'Orders' and 'My Workspace'. The 'Orders' folder is selected. The main area displays a table of data sources. The table has columns for a checkbox, Name (sorted by Last Modified), Last Modified, and Plugin. One row is visible, showing 'Text_StocksStatic' as the name, 'May 24, 2022 8:03 PM' as the last modified date, and 'Text' as the plugin. At the top right of the main area, there are buttons for 'Connected' (green checkmark), 'New Data Source' (blue button), and other actions.

	Name ↑	Last Modified	Plugin
<input type="checkbox"/>	Text_StocksStatic	May 24, 2022 8:03 PM	Text

Folders and Data Sources Display View

Folders and data sources can be displayed either on a *List* or *Grid View*. By default, the folders and data sources are displayed in the *List View*.



	Name ↑	Last Modified	Plugin
1	KafkaOutput	May 24, 2022 6:46 PM	KafkaWriter
2	KdbOutput	May 24, 2022 6:46 PM	KdbWriter
3	RestOutput	May 24, 2022 6:46 PM	RestWriter
4	StockMarketSimulatorSymbols	May 24, 2022 6:46 PM	
5	StreamSimulator_StocksStatic	May 24, 2022 6:46 PM	Stream Simulator
6	StreamSimulator_StocksTimeSeries_TimeSeries	May 24, 2022 6:46 PM	Stream Simulator
7	Text_SampleData	May 24, 2022 6:46 PM	Text
8	Text_StocksStatic	May 24, 2022 6:46 PM	Text
9	Text_StocksTimeSeries_Static	May 24, 2022 6:46 PM	Text
10	Text_StocksTimeSeries_TimeSeries	May 24, 2022 6:46 PM	Text

On the toolbar, click **Grid View** . The folders and data sources are displayed as thumbnails.

NOTE

Data Sources are placed under their corresponding plugin.

The screenshot shows the Altair Panopticon interface with the 'Data Sources' tab selected. The left sidebar shows 'Organization' and 'Users' sections, with 'Orders' selected. The main area displays a list of data sources under 'Folders' (Orders) and several categories: KafkaWriter, TEXT, and StreamSimulator. A callout points to the 'TEXT' section with the label 'Plugins'.

- KafkaWriter**
 - KafkaOutput** (Modified an hour ago)
- TEXT**
 - StockMarketSimulatorSymbols** (Modified an hour ago)
 - Text_SampleData** (Modified an hour ago)
 - Text_StocksTimeSeries_Static** (Modified an hour ago)
- StreamSimulator**
 - StreamSimulator_StocksStatic** (Modified a few seconds ago)
 - StreamSimulator_StocksTimeS...** (Modified an hour ago)

Click **List View**  to return to the standard listing.

On either display view style:

- Clicking on a data source title or thumbnail displays the data source
- Right-clicking on a data source displays the context menu

The screenshot shows the Altair Panopticon interface with the 'Data Sources' tab selected. A context menu is open over the 'Text_StocksTimeSeries_Static' data source in the 'TEXT' section. The menu options are: Rename, Application Usage, Move, Copy, Download, and Remove.

SORTING THE LIST OF DATA SOURCES

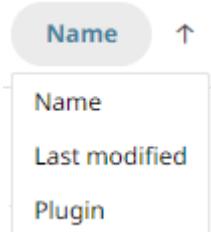
Sorting data sources can be done by *Name*, *Last Modified*, or *Plugin*.

Sort By	Default Sort Order
Name	Ascending
Last Modified	Descending
Plugin	Ascending

Steps:

On the *Folders and Data Sources Summary* layout, either:

- click the **Sort By** option on the *Toolbar* of the *Grid View*
By default, the sorting is by **Name**.



- Name
- Last Modified
- Plugin

Then click the *Sort Order*:

- Ascending
- Descending

- click on the **Name**, **Last Modified**, or **Plugin** column header of the *List View*

	Name	Last Modified	Plugin
	KafkaOutput	May 24, 2022 6:46 PM	KafkaWriter
	KdbOutput	May 24, 2022 6:46 PM	KdbWriter
	RestOutput	May 24, 2022 6:46 PM	RestWriter
	StockMarketSimulatorSymbols	May 24, 2022 6:46 PM	Text
	StreamSimulator_StocksStatic	May 24, 2022 6:46 PM	Stream Simulator
	StreamSimulator_StocksTimeSeries_TimeSeries	May 24, 2022 6:46 PM	Stream Simulator
	Text_SampleData	May 24, 2022 6:46 PM	Text
	Text_StocksStatic	May 24, 2022 6:46 PM	Text
	Text_StocksTimeSeries_Static	May 24, 2022 6:46 PM	Text
	Text_StocksTimeSeries_TimeSeries	May 24, 2022 6:46 PM	Text

Then click the *Sort Order*:

- Ascending
- Descending

SEARCHING FOR DATA SOURCES

To search for a particular data source, enter it in the *Search Data Source* box.

Searching data sources in the Grid View

You can also enter one of more characters into the **Search Data Source** box and the suggested list of data sources that matched the entries will be displayed.

Altair Panopticon™ Applications Data Sources Data Producer Monitoring Parameters System Connected A

Organization

Orders

Users

~designer

Output

Name ↑

Folders

Orders 7 data sources

EmailWriter

EmailOutput Modified a few seconds ago

InfluxDBWriter

InfluxDBOutput Modified a few seconds ago

JDBCWriter

JDBCOutput Modified a few seconds ago

KafkaWriter

KafkaOutput Modified 2 hours ago

KdbWriter

KdbOutput Modified a few seconds ago

TEXT

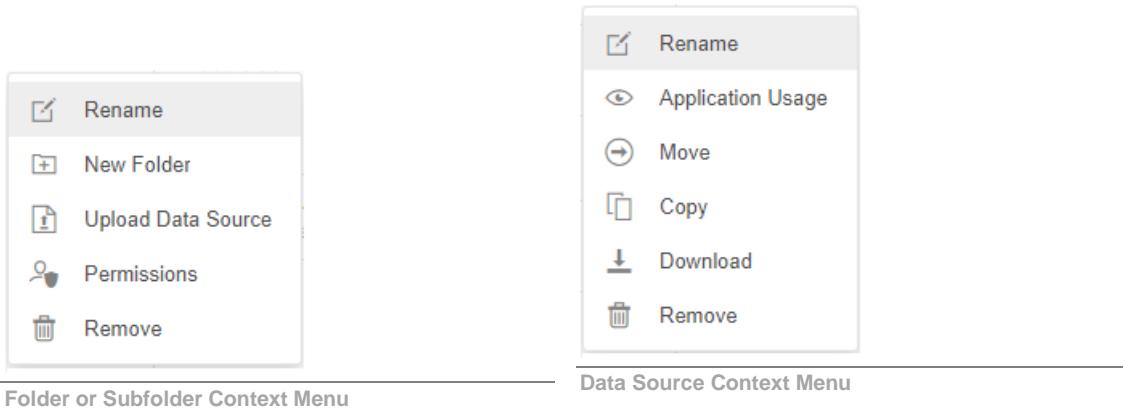
StreamSimulator

RENAMING DATA SOURCES OR FOLDERS

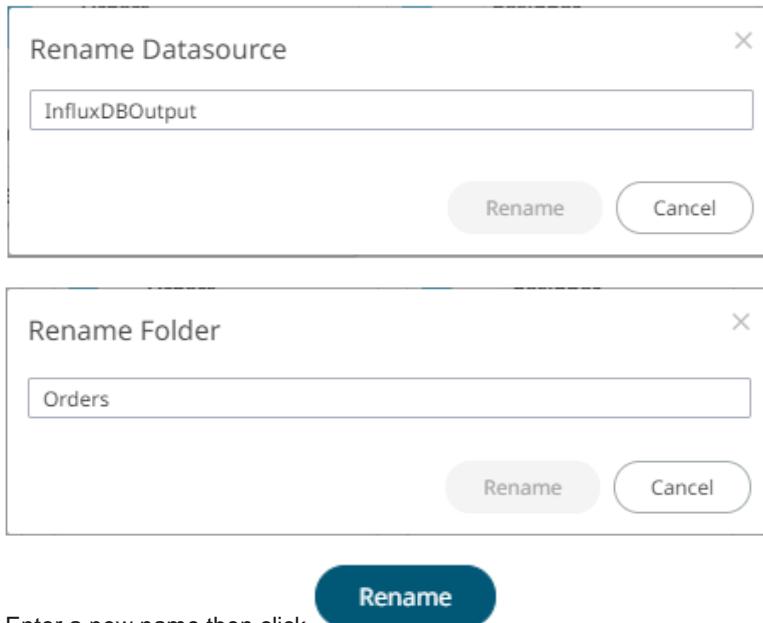
A user with an Administrator or Designer role can rename data sources and folders.

Steps:

1. Right-click on a data source or folder then select **Rename** in the context menu.



The *Rename Data Source* or *Rename Folder* dialog displays.



2. Enter a new name then click

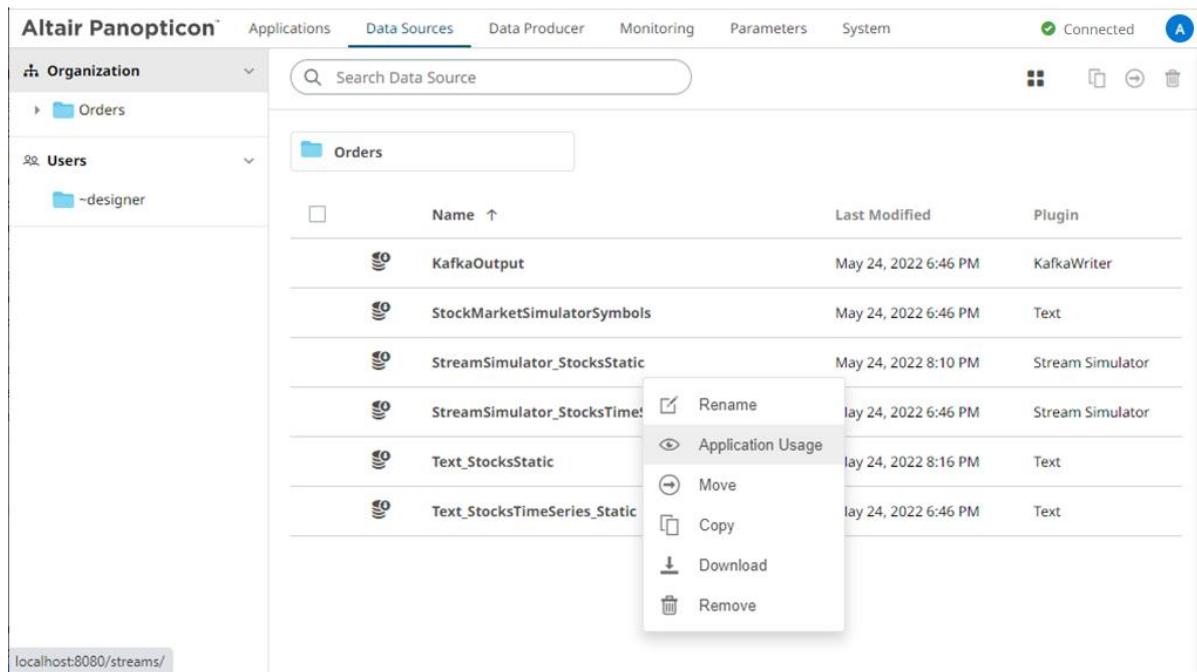
Rename

VIEWING APPLICATION USAGES

On the **Data Sources** tab, you can view the applications that currently use a data source.

Steps:

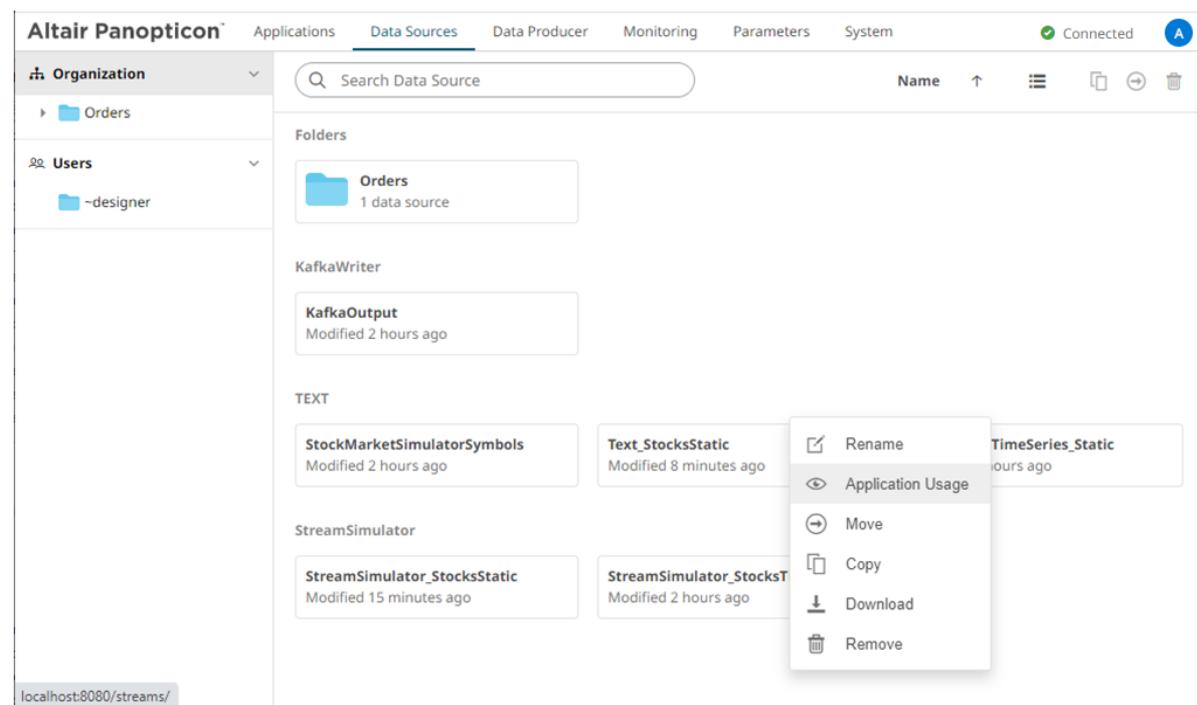
1. On the *List* view or *Grid* view, right-click on a data source and select **Application Usage**.



The screenshot shows the Altair Panopticon interface with the 'Data Sources' tab selected. On the left, a sidebar shows 'Organization' and 'Users' sections. The main area displays a list of data sources under the 'Orders' folder. A context menu is open over the 'Text_StocksTimeSeries_Static' entry, showing options: Rename, Application Usage, Move, Copy, Download, and Remove. The table lists the following data sources:

Name	Last Modified	Plugin
KafkaOutput	May 24, 2022 6:46 PM	KafkaWriter
StockMarketSimulatorSymbols	May 24, 2022 6:46 PM	Text
StreamSimulator_StocksStatic	May 24, 2022 8:10 PM	Stream Simulator
StreamSimulator_StocksTimeSeries	May 24, 2022 6:46 PM	Stream Simulator
Text_StocksStatic	May 24, 2022 8:16 PM	Text
Text_StocksTimeSeries_Static	May 24, 2022 6:46 PM	Text

localhost:8080/streams/

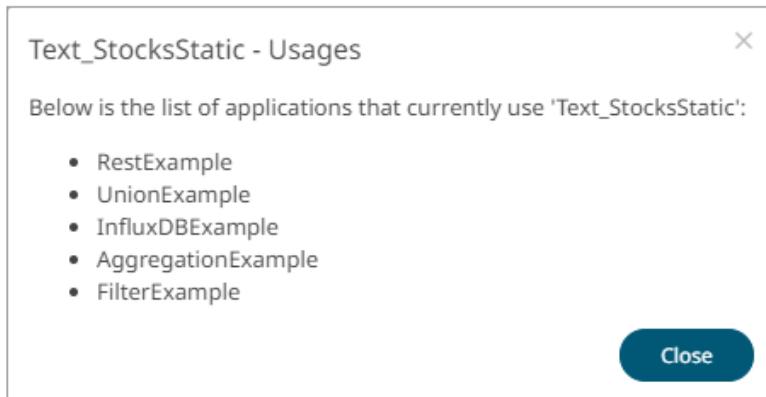


The screenshot shows the Altair Panopticon interface with the 'Data Sources' tab selected. On the left, a sidebar shows 'Organization' and 'Users' sections. The main area displays a list of applications using a data source. A context menu is open over the 'TimeSeries_Static' entry, showing options: Rename, Application Usage, Move, Copy, Download, and Remove. The applications listed are:

- Folders**
 - Orders (1 data source)
- KafkaWriter**
 - KafkaOutput (Modified 2 hours ago)
- TEXT**
 - StockMarketSimulatorSymbols (Modified 2 hours ago)
 - Text_StocksStatic (Modified 8 minutes ago)
 - TimeSeries_Static (Modified 2 hours ago)
- StreamSimulator**
 - StreamSimulator_StocksStatic (Modified 15 minutes ago)
 - StreamSimulator_StocksTimeSeries (Modified 2 hours ago)

localhost:8080/streams/

The list of applications that currently use the data source displays.



3. Click  .

MOVING DATA SOURCES

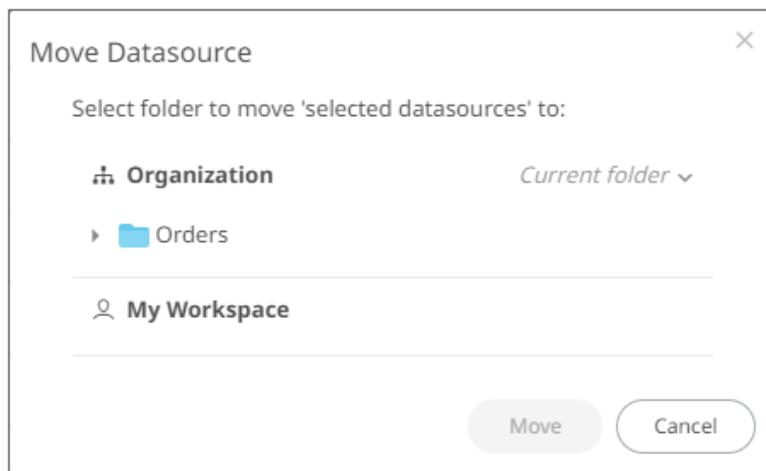
Users with a Designer role can move data sources to folders or subfolders to which they have permission.

Steps:

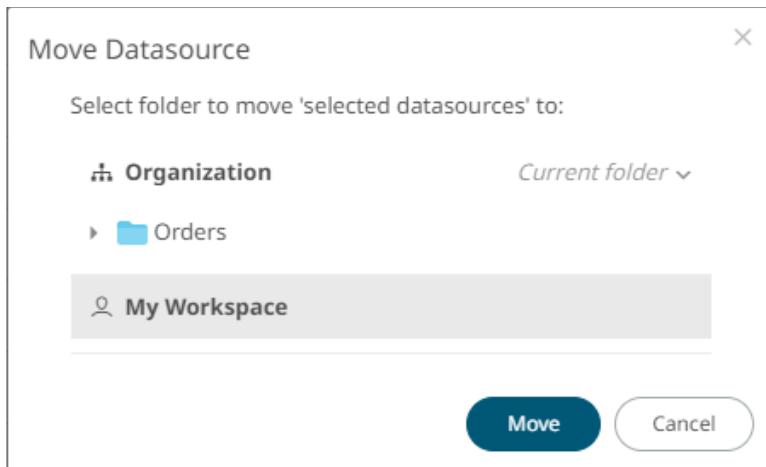
1. Select the check box of multiple data sources either on the *Grid View* or the *List View*.
2. Then select either:

- the **Move**  icon on the toolbar
- **Move** in the context menu

The *Move Data Source* dialog displays with the folder or subfolders to which the user has permission to move the data sources.



3. Select the folder or subfolder.



4. Click **Move**.

The data sources are moved and displayed on the selected folder.

The screenshot shows the 'Data Sources' tab of the Altair Panopticon interface. The left sidebar shows 'Organization' with 'Orders' and 'Users' sections. 'Users' has a sub-section '-designer' which is selected and highlighted with a gray background. The main area displays a table of data sources with columns: Name, Last Modified, and Plugin. Two entries are listed: 'KafkaOutput' (May 24, 2022 8:31 PM, KafkaWriter) and 'Text_StocksStatic' (May 24, 2022 8:31 PM, Text).

NOTE

If data sources with the same name are already in the selected folder, a copy of them will be moved.

COPYING DATA SOURCES

Users with a Designer role can copy data sources to folders or subfolders to which they have permission.

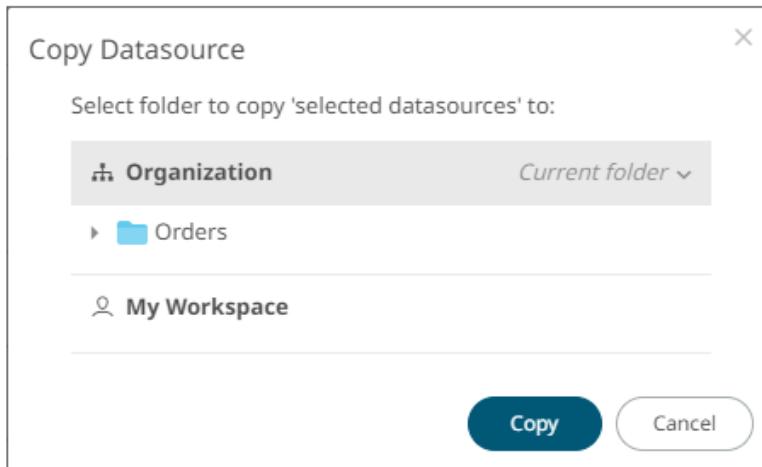
Steps:

1. Select the check box of one or multiple data sources either on the *Grid View* or *List View*.
2. Then select either:

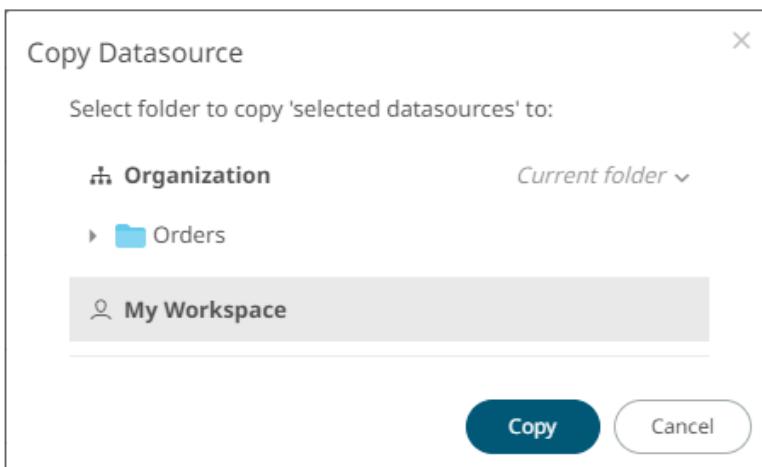


- the **Copy** icon on the toolbar
- **Copy** in the context menu

The *Copy Data Source* dialog displays with the folder or subfolders to which the user has permission to copy the data sources.



3. Select the folder or subfolder.



4. Click **Copy**.

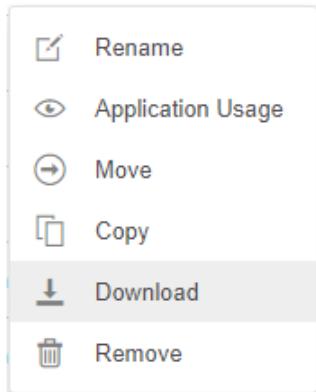
The data sources are copied and displayed on the selected folder.

NOTE

If data sources with the same name are already in the selected folder, it will be added as copy.

DOWNLOADING A DATA SOURCE

Users with Administrator or Designer role are allowed to download a copy of a data source by right-clicking on a data source and selecting **Download** in the context menu.



The data source is downloaded.

DELETING DATA SOURCES

Users with an Administrator or Designer role can delete data sources using the toolbar or context menu.

Steps:

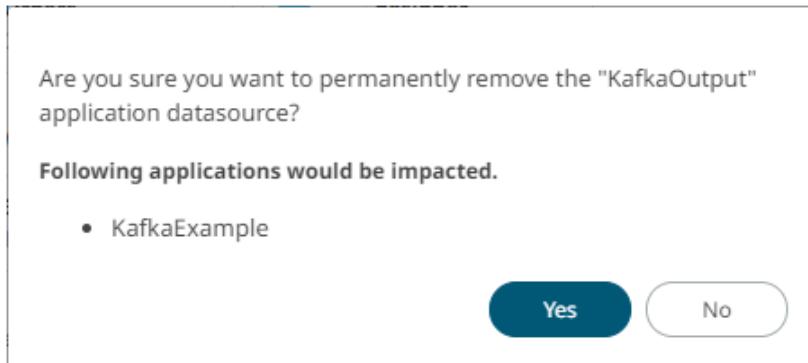
1. Select the check box of one or several data sources either on the *Grid View* or *List View*.

2. Then click either:

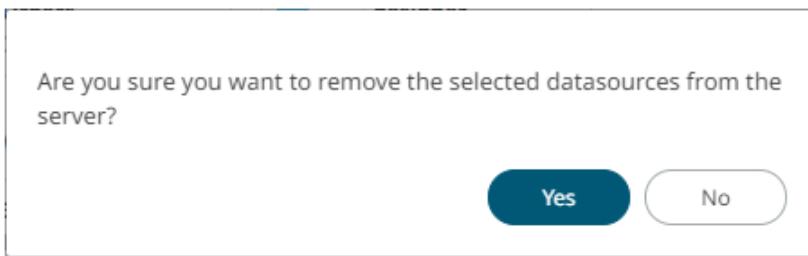
- the **Remove**  icon on the toolbar
- **Remove** in the context menu

A notification message displays.

For a data source, the corresponding applications that will be impacted is listed:



For several data sources:



3. Click  to remove.

CREATING A DATA SOURCE

Panopticon Streams supports the creation of data sources that can be used as inputs or outputs in the application model.

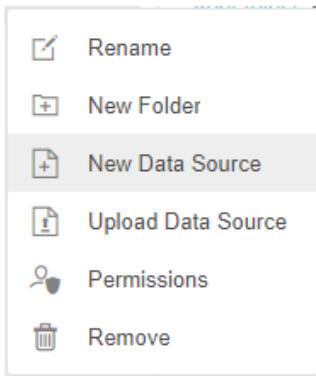
Steps:

1. On the **Data Sources** tab:

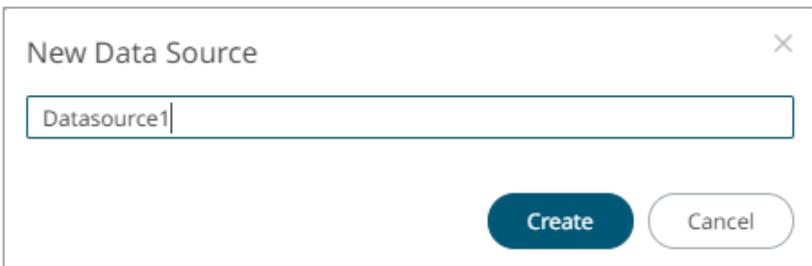


The image shows a dark blue rounded rectangle button with a white "New Data Source" label and a white plus sign icon to its left. Below the button, a list of steps is provided.

- click  on the toolbar, or
- right-click on a folder or subfolder and select **New Data Source**.

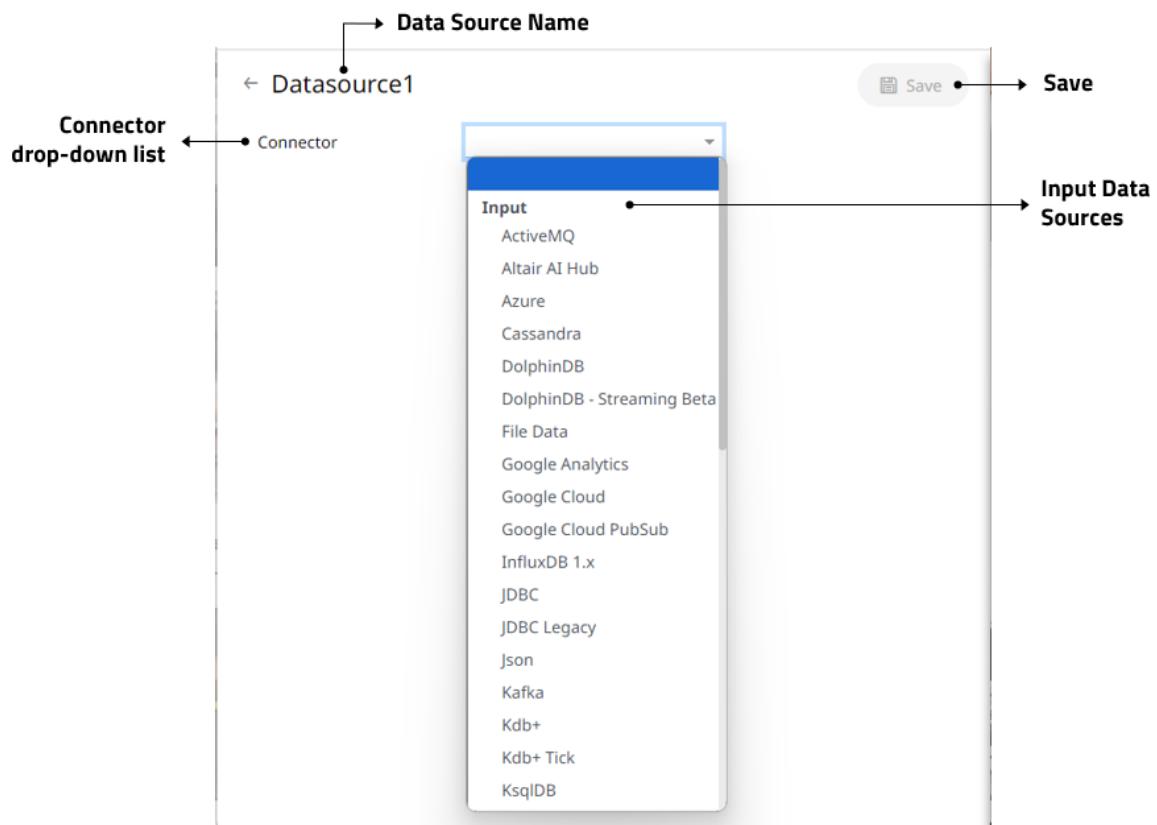


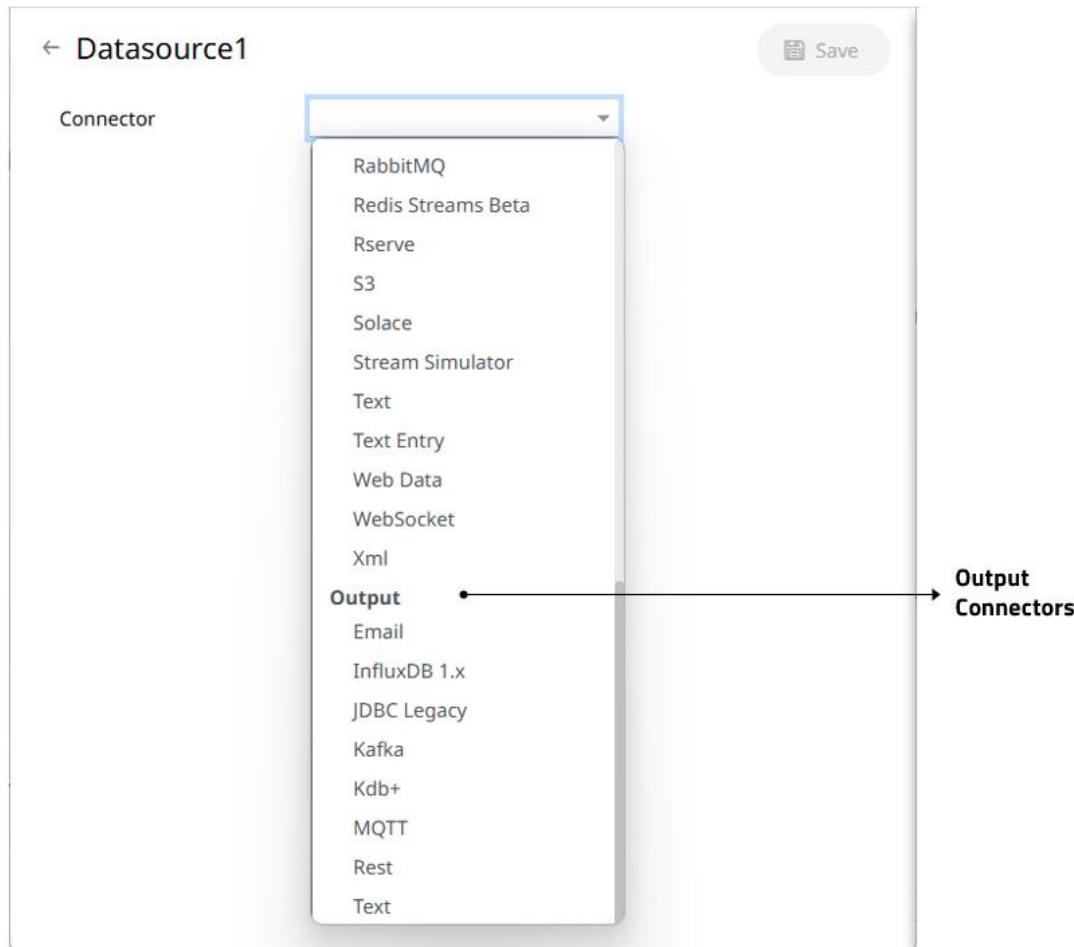
The *New Data Source* dialog displays.



2. Enter the *Name* of the data source and click

The **Data Source** tab displays the following sections:





Section/Pane	Description
Data Source Name	Name of the data source. Click the button to go back to the <i>Data Sources</i> listing page.
Connector drop-down list	Includes the input data sources and output connectors.
Save	Saves the changes made on the Data Sources tab.

NOTE

Inactive connectors are not displayed in the *Connector* drop-down.

3. Enter the *Name* of the data source. This should be unique and should only contain letters (a to Z), numbers (0 to 9), and underscores.
4. Click or press **Enter** to apply the name.
5. Select any of the following:
 - Output connectors

- ◆ [Email](#)
- ◆ [InfluxDB 1.x](#)
- ◆ [JDBC Legacy](#)
- ◆ [Apache Kafka](#)
- ◆ [Kx kdb+](#)
- ◆ [MQTT](#)
- ◆ [REST](#)
- ◆ [Text](#)
- Input data sources
 - ◆ [ActiveMQ](#)
 - ◆ [Altair AI Hub](#)
 - ◆ [Azure](#)
 - ◆ [Cassandra](#)
 - ◆ [DolphinDB](#)
 - ◆ [DolphinDB - Streaming](#)
 - ◆ [Google Analytics](#)
 - ◆ [Google Cloud](#)
 - ◆ [Google Cloud Pub/Sub](#)
 - ◆ [InfluxDB 1.x](#)
 - ◆ [JDBC Legacy](#)
 - ◆ [JDBC](#)
 - ◆ [JSON](#)
 - ◆ [Apache Kafka](#)
 - ◆ [Kx kdb+](#)
 - ◆ [Kx kdb+ Tick](#)
 - ◆ [ksqlDB](#)
 - ◆ [ksqlDB - Streaming](#)
 - ◆ [MongoDB](#)
 - ◆ [MQTT](#)
 - ◆ [MQTT Publisher](#)
 - ◆ [MS Excel](#)
 - ◆ [OneTick](#)
 - ◆ [OneTick CEP](#)
 - ◆ [Python](#)
 - ◆ [RabbitMQ](#)

- ◆ [Rserve](#)
- ◆ [S3](#)
- ◆ [Solace](#)
- ◆ [Stream Simulator](#)
- ◆ [StreamBase 7.1](#)
- ◆ [StreamBase LiveView](#)
- ◆ [Text](#)
- ◆ [Text Entry](#)
- ◆ [WebSocket](#)
- ◆ [Web Data](#)
- ◆ [XML](#)

The tab page changes depending on the selected connector. Refer to the sections below for more information.

Common Data Source Settings

Some of the data sources share the following settings or parts:

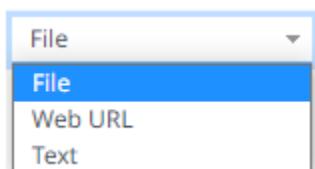
- [Data Connector File Source](#)
- [Message Type selection and definition](#)
- [Saving and loading of column definitions](#)
- [Data Source Toolbar](#)
- [Defining Real-time Settings](#)

Selecting and Defining the Data Connector File Source

Several connectors including [JSON](#), [Text](#), [XML](#), and [Stream Simulator](#), allow selection from a File, Web URL, or Text source.

Steps:

Select the connector file source:



- File

You can either:

- Upload a data source snapshot by clicking **Upload File**  then **Browse**  to browse to the file source.

After selecting the file, it is displayed with the timestamp of the snapshot.

JSON File Source	File	▼
Load Type	<input type="button" value="Upload File"/> <input type="button" value="Link To File"/>	
File	Books.json X	<input type="button" value="Browse"/>
	as of 2022-01-05 17:17:55	

To change the data source, click  then **Browse** to browse to a new file.

- Link to a data source file by clicking **Link to File** and entering a *File Path*.

Load Type	<input type="button" value="Upload File"/> <input type="button" value="Link To File"/>
JSON File Path	<input type="text" value=""/>
	(File Type: .json)

Ensure that in a cluster, you need to use a a shared path, or put it on every node and use a path that resolves on every node. You can update its contents whenever you want.

NOTE	<p>An error notification displays upon fetching sheets/column generation when the <i>File Path</i> is blank.</p> <div style="border: 1px solid #ccc; padding: 10px; width: 400px; height: 150px; margin-top: 10px;"> <p>Error!</p> <p> File path is required</p> <p style="text-align: right;"><input type="button" value="Close"/></p> </div> <p>Click Close and enter the file path.</p>
-------------	--

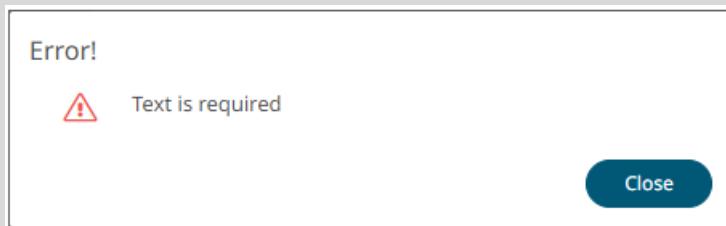
Text

Then enter the text block to be parsed.

Text

NOTE

An error notification displays upon column generation when the *Text* box is blank.

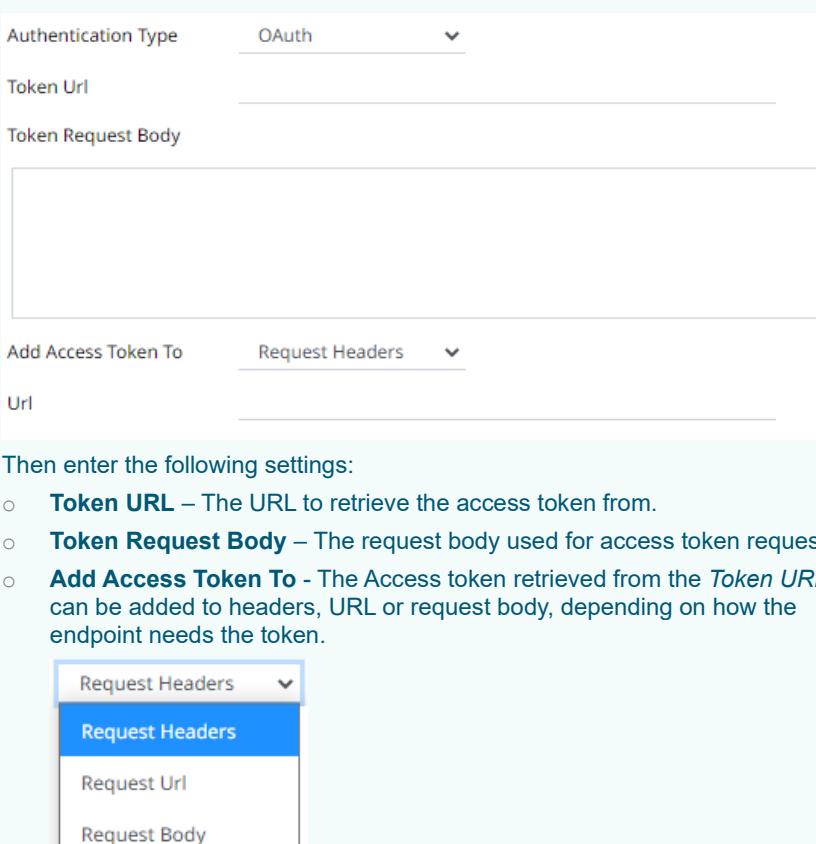
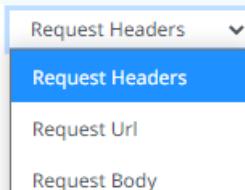


Click **Close** and enter the text block.

 Web URL

The dialog changes to allow specification of the following:

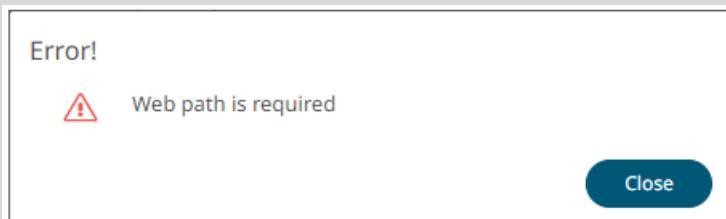
JSON File Source	Web URL
Authentication Type	Basic
Path	<input type="text"/>
Proxy Server URI	<input type="text"/>
Headers	<input type="text"/>
Content Encoding	None
User Id	<input type="text"/>
Password	<input type="text"/> <input type="checkbox"/> Show characters
Http Method	GET
Timeout	10
Request Body	<input type="text"/>
Content Type	application/x-www-form-urlencoded
Record Path	<input type="text"/> (eg. myroot.items.item)
Decimal Separator	Period {.}
File Encoding	<input type="text"/>

Property	Description
Authentication Type	<ul style="list-style-type: none"> Basic The basic authentication. OAuth  <p>Then enter the following settings:</p> <ul style="list-style-type: none"> Token URL – The URL to retrieve the access token from. Token Request Body – The request body used for access token requests. Add Access Token To - The Access token retrieved from the <i>Token URL</i> can be added to headers, URL or request body, depending on how the endpoint needs the token. <p></p> <ul style="list-style-type: none"> Request Header - A header is automatically added to the REST API request. Request URL - The URL needs to be manually parameterised with a {access_token} parameter, before calling the REST API, the parameter is replaced with the actual token. Request Body - The Request Body needs to be manually parameterised with a {access_token} parameter, before calling the REST API, the parameter is replaced with the actual token. <p>NOTES:</p> <ul style="list-style-type: none"> The given request body is posted to the Token URL as application/x-www-form-urlencoded, so the request body must be formatted like field1=value1&field2=value2, e.g., client_id=xxxx&client_secret=xxxx&grant_type=client_credentials. Not available in the Stream Simulator connector.
Path	The absolute path including the HTTP where the file is located.
Proxy Server URI	The HTTP Proxy setting that will allow the connector to reach the endpoint.
Headers	<ul style="list-style-type: none"> Headers are separated by a comma

	<ul style="list-style-type: none"> Each Header is entered as Name = Value, where <i>Name</i> and <i>Value</i> can be enclosed in double quotes to allow inclusion of any character except for double quotes <i>Name</i> and <i>Value</i> can also be left unquoted, in which case they may not include comma or equals characters
Content Encoding	Select the <i>Content Encoding</i> with the HTTP Header: None , GZip , Deflate , or GZip and Deflate
User Id	The user Id that will be used to connect to the connector's service.
Password	The password to connect to the connector's service. Select the Show Characters check box to display the entered characters.
HTTP Method	Select the appropriate HTTP method for the request from the following options: <ul style="list-style-type: none"> GET – retrieve data POST – add new data PUT – replace existing data DELETE – remove existing data
Timeout	The length of time to wait for the server response (10 to 300). Default is 10 .
Request Body	The Request Body for the HTTP POST.
Content Type	The required Content Type. Default is application/x-www-form-urlencoded
Record Path	The record path that will be queried by the connector's path (e.g., myroot.items.item).
File Encoding	Set the character encoding to use in text data. <ul style="list-style-type: none"> UTF-8 UTF-16 UTF-32 US-ASCII Windows-1252

NOTE

An error notification displays upon column generation when the *Path* is blank.



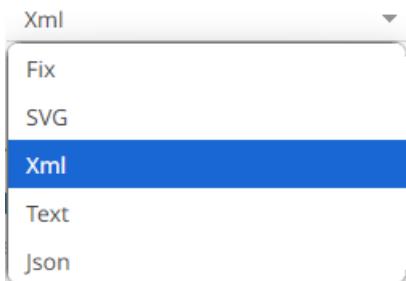
Click **Close** and enter the web path.

Defining the Message Type in Data Sources

You can select the message type that specifies the format of the data within the message in [ActiveMQ](#), [Google Cloud PubSub](#), [MQTT](#), [RabbitMQ](#), [Solace](#), and [WebSocket](#) connectors.

Steps:

1. Select the Message Type:



- FIX

Message Type	Fix					
Decimal Separator	Period {.}					
Generate Columns Save Load						
<input type="checkbox"/> Name	Fix Tag	Type	Date Format	Enabled	+	-

- JSON

If **JSON** is selected, enter the *Record Path* which allows the identification of multiple records within the JSON document (e.g., `myroot.items.item`) and select the [Array Handling](#) setting.

Message Type	Json					
Decimal Separator	Period {.}					
Record Path	(eg. myroot.items.item)					
Array Handling	Add Columns					
Generate Columns Save Load						
<input type="checkbox"/> Name	JsonPath	Type	Date Format	Enabled	+	-

- SVG

Message Type	SVG			
Decimal Separator	Period {.}			
Generate Columns				
Name	Type	Date Format	Enabled	Filter

NOTE

This parser is not available on the MQTT connector.

- Text

If **Text** has been selected, confirm the **Decimal Separator**, **Text Qualifier**, **Column Delimiter**, and if the first row of the message includes column headings.

Message Type	Text
Decimal Separator	Period {.}
Text Qualifier	None
Column Delimiter	Comma {,}
First Row Headings	<input checked="" type="checkbox"/>

Column Index controls the position of a column, Must be >= 0.

Generate Columns **Save** **Load**

<input type="checkbox"/> Name	Column Index	Type	Date Format	Filter	Enabled + -
-------------------------------	--------------	------	-------------	--------	----------------

- XML

Message Type	Xml
Decimal Separator	Period {.}

Prepend 'default:' for the elements falling under default namespace.

Generate Columns **Save** **Load**

<input type="checkbox"/> Name	XPath	Type	Date Format	Filter	Enabled + -
-------------------------------	-------	------	-------------	--------	----------------

- Define or set the columns that represent the sections of the message.

Property	Description
Name	The column name of the source schema.
Fix Tag/JsonPath/Column Index/XPath	The Fix Tag/JsonPath/Column Index/XPath of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Enabled	Determines whether the message field should be processed.

NOTE

To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of upper case S. There can be no additional characters following them.

For example: **yyyy-MM-dd HH:mm:ss.SSSSSS**

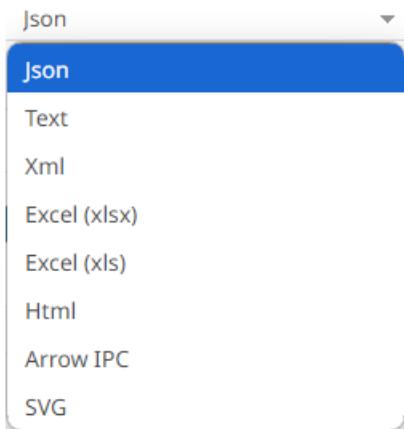
To delete a column, check its or all the column entries, check the topmost , then click .

Defining the Format in Data Sources

You can select the format to use in the connector for [Azure](#), [Google Cloud](#), [S3](#), and [Web Data](#) data sources.

Steps:

1. Select the *Data Type*:



- JSON

Fetch

If **JSON** is selected, click **Fetch** to fetch the *Record Paths* then select one. This allows the identification of multiple records within the JSON document. Then select the [Array Handling](#) setting.

Data Type	Json					
Decimal Separator	Period {.}					
Record Path	<input type="text"/> Fetch					
Array Handling	Add Columns					
Generate Columns <input type="button" value="Save"/> <input type="button" value="Load"/>						
<input type="checkbox"/> Name	JsonPath	Type	Date Format	Enabled	+	-

- Text

If **Text** is selected, confirm the **Decimal Separator**, **Text Qualifier**, **Column Delimiter**, and if the first row of the message includes column headings.

Data Type	Text
Decimal Separator	Period {.}
Text Qualifier	None
Column Delimiter	Comma {,}
First Row Headings	<input checked="" type="checkbox"/>

Column Index controls the position of a column, Must be ≥ 0 .

Generate Columns

<input type="checkbox"/> Name	Column Index	Type	Date Format	<input checked="" type="checkbox"/> Enabled	<input type="button" value="+"/>	<input type="button" value="-"/>
-------------------------------	--------------	------	-------------	---	----------------------------------	----------------------------------

- XML

If **XML** is selected, enter the *Record XPath* which allows the identification of multiple records within the XML document (e.g., `//myroot/items/item`) .

Data Type	Xml
Record XPath	(eg. <code>//myroot/items/item</code>)
Decimal Separator	Period {.}

Prepend 'default:' for the elements falling under default namespace.

Generate Columns

<input type="checkbox"/> Name	XPath	Type	Date Format	Enabled	<input type="button" value="+"/>	<input type="button" value="-"/>
-------------------------------	-------	------	-------------	---------	----------------------------------	----------------------------------

- Excel (xlsx) or Excel (xls)

For **Excel** file type, select the required sheet, decimal separator, and adjust the headers on first row, if needed.

Data Type	Excel (xlsx)
Sheet	<input type="button" value="Fetch Sheets"/>
Decimal Separator	Period {.}
Headers On First Row	Auto

Columns

Name	Type	Date Format	<input checked="" type="checkbox"/> Enabled
------	------	-------------	---

- Arrow IPC

Data Type

Arrow IPC

Decimal Separator

Period {.}

Generate Columns
Save
Load

Name

Type

Date Format

Enabled
+
-

- SVG

Data Type

SVG

Decimal Separator

Period {.}

NOTE

- The *Decimal Separator* setting is not available on the Web Data connector.
- In the SVG parser, columns listing is not displayed.

2. Define or set the columns that represent the sections of the file.

3. Property	Description
Name	The column name of the source schema.
JsonPath/Column Index/XPath	The JsonPath/Column Index/XPath of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Enabled	Determines whether the message field should be processed.

NOTE

To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of uppercase S. There can be no additional characters following them.

For example: **yyyy-MM-dd HH:mm:ss.SSSSSS**

To delete a column, check its or all the column entries, check the topmost , then click - .

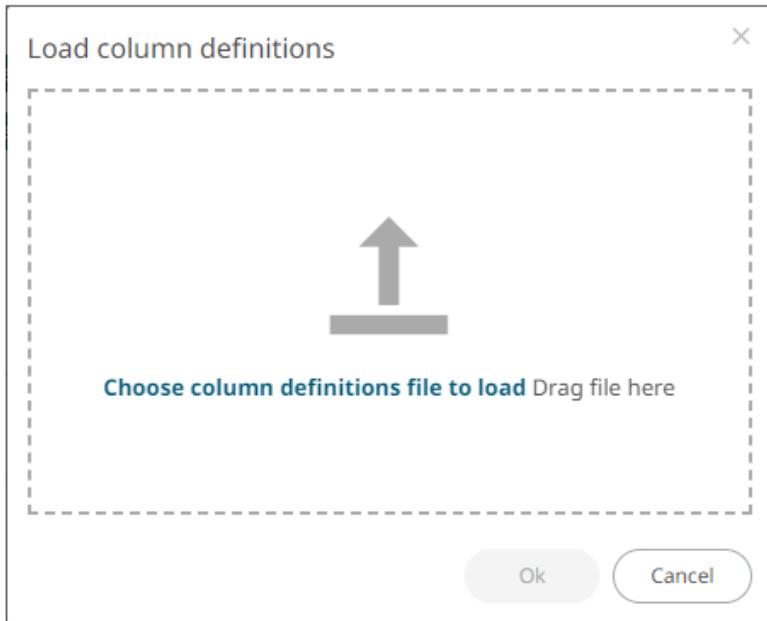
Saving or Loading Column Definitions in the Data Sources

Save or load column definitions in the data sources.

Steps:

1. Click Save to save a copy of a column definitions file (**.exs**).
2. Instead of generating columns done in step 8, click Load to load a column definitions (**.exs**) file.

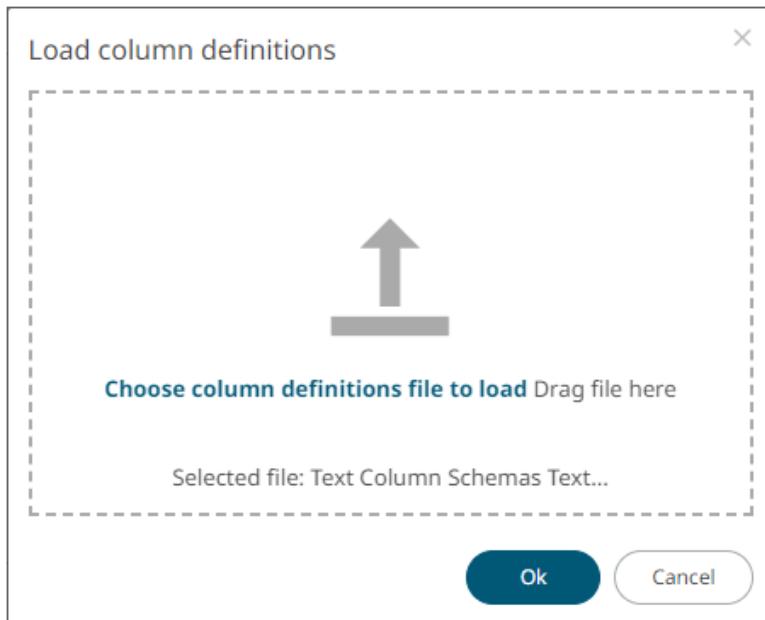
The *Load Column Definitions* dialog displays.



2.1. To load column definitions, you can either:

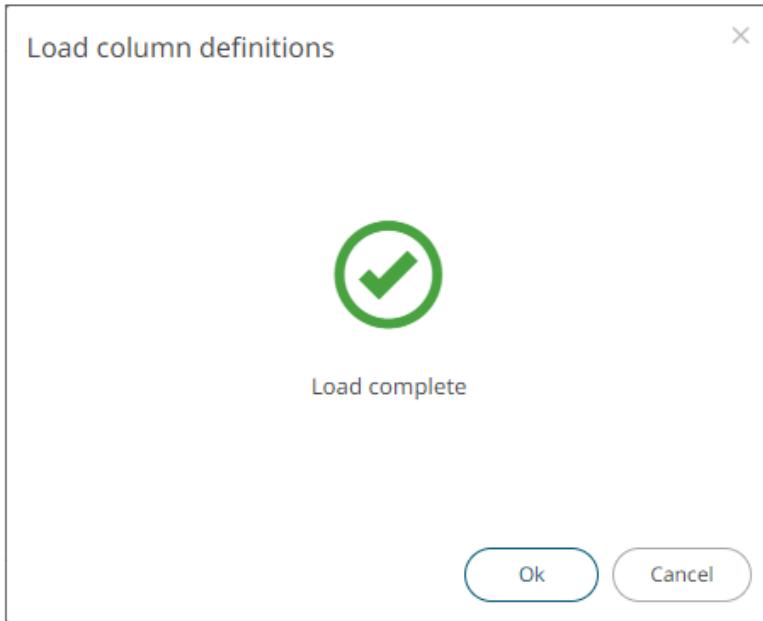
- ◆ drag it from your desktop and drop in the dialog, or
- ◆ click **Choose Column Definitions File to Load** and select one in the *Open* dialog that displays.

The name of the column definitions is displayed on the loaded column definitions area.



2.2. Click **Ok**.

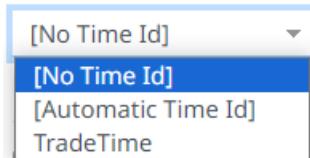
A notification displays when the file is loaded.



This populates the list of columns from the .exs file.

Defining Real-time Settings

Streaming connectors have a common section to specify the *Time Id Column* to generate the streaming time series window. The *Time Id Column* can be from the source dataset, or alternatively, automatically generated.



As new data arrives from the subscription, new time slices will automatically be added, and old ones will be deleted.

Steps:

1. Select either:

- Automatic Time Id

Time Id Column	[Automatic Time Id] <input type="button" value="▼"/>
Time Id Column Name	Automatic_Timestamp_Column

Then define the *Time Id Column Name*.

- Date/Time Id column either from the source data or automatically generated

Time Id Column	TradeTime <input type="button" value="▼"/>
Time Id Column Name	TradeTime

2. Select the **Reset Data on Reconnect** check box to flush out the stale data and reload data after reconnection.

Using the Data Source Toolbar

Several data sources have a toolbar:

Altair Panopticon™ Applications Data Sources

Connected A

← Datasource1

Save

Connector: InfluxDB

Url:

Port: 8086

Database:

User Id:

Password:

Measurement:

Source Target Type

Add Column

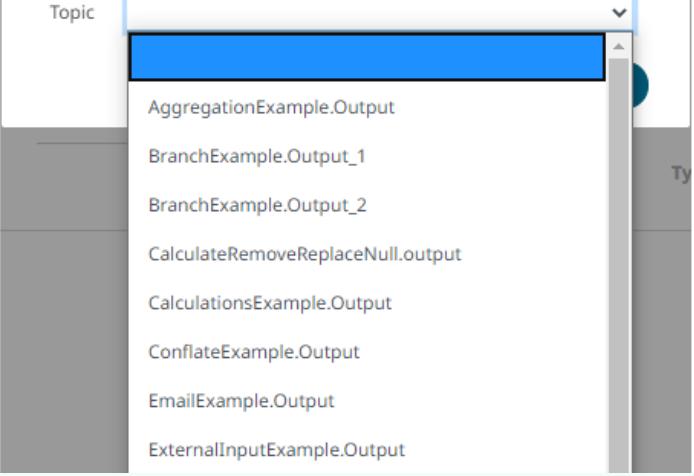
Delete Column

Fetch Schema

Test Connection

Click on any of the following icons:

Icon	Description																
	A new column entry displays. Enter or select the following properties: <table border="1"> <tr> <td><input type="checkbox"/></td> <td>Source</td> <td>Target</td> <td>Type</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td>Column_1</td> <td></td> <td>Text</td> <td colspan="4"></td> </tr> </table> <ul style="list-style-type: none"> Source The column name of the source schema. Target The column name of the target schema. Type The data type of the column. Can be a Text, Numeric, or Time. 	<input type="checkbox"/>	Source	Target	Type					<input type="checkbox"/>	Column_1		Text				
<input type="checkbox"/>	Source	Target	Type														
<input type="checkbox"/>	Column_1		Text														
<input type="checkbox"/>	Check the topmost <input type="checkbox"/> to select all data source column entries.																
	Check the <input type="checkbox"/> of a data source entry or check the topmost <input type="checkbox"/> to select all column entries and click to delete.																
	Allows you to select an output topic in the drop-down list.																

	<p>Select an output topic</p> <p>Topic <input type="text"/></p> <p><input type="button" value="Cancel"/> <input type="button" value="OK"/></p>
	<p>Select an output topic</p> <p>Topic <input type="text"/></p> 
	<p>Click OK. The schema of the selected output topic is displayed.</p>
	<p>Tests if the connection to the output connector is successful. If successful, a confirmation message displays.</p> <div style="border: 1px solid #ccc; padding: 10px; text-align: center;"> <p>Connection successful</p> <p><input type="button" value="OK"/></p> </div> <p>Click OK. Otherwise, an error prompt displays.</p> <div style="border: 1px solid #ccc; padding: 10px; text-align: center;"> <p>Error!</p> <p> Unable to parse url: /127.0.0.1:8086:8086</p> <p><input type="button" value="Close"/></p> </div> <p>Click Close and fix the connection error.</p>

Date/Time Key Elements

The key elements of the Date/Time format include:

Component	Format
Year	yyyy
Month	MM
Month as an abbreviation	MMM
Day	dd
Hour (24-hour clock)	HH
Minute	mm
Second	ss
Hour (12-hour clock; a.m./p.m.)	tt
Millisecond	SSS
Microsecond	SSSSSS
Nanosecond	SSSSSSSS
Space/separator (required if time is specified)	'T'
Zulu (Greenwich Mean Time)	'Z'
Time zone (ISO 8601 time zone)	X
UNIX Epoch time	POSIX
Milliseconds since UNIX Epoch time	POSIXMILLIS
Seconds since midnight	Seconds
Milliseconds since midnight	Millis
Microseconds since midnight	Micros
Nanoseconds since midnight	Nanos

NOTE

- To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of upper case S. There can be no additional characters following them.
For example: yyyy-MM-dd HH:mm:ss.SSSSSS
- The “Seconds”, “Millis”, “Micros”, and “Nanos” formats are used for parsing of the data in the data connectors and not for the display of the Date/Time columns.

Creating Email Output Connector

Steps:

1. On the **Data Source** tab, select **Output > Email** in the *Connector* drop-down list.

← EmailDataSource

Save

Connector	Email
Host	
Port	
Mail Security Mode	NONE
Sender Email Address	
Sender Password	
To Email Address	
CC Email Address	
BCC Email Address	
Subject	
Text	

2. Define or select the following properties:

Property	Description
Host	Email host address.
Port	Email host port.
Mail Security Mode	Select the email security mode: NONE , SSL , or TLS
Sender Email Address	Email address of the sender.
Sender Password	Password of the sender.

To Email Address	Email address of the recipient.
CC Email Address	Email address of the CC recipient.
BCC Email Address	Email address of the BCC recipient.
Subject	Subject of the email.
Text	Content of the email.

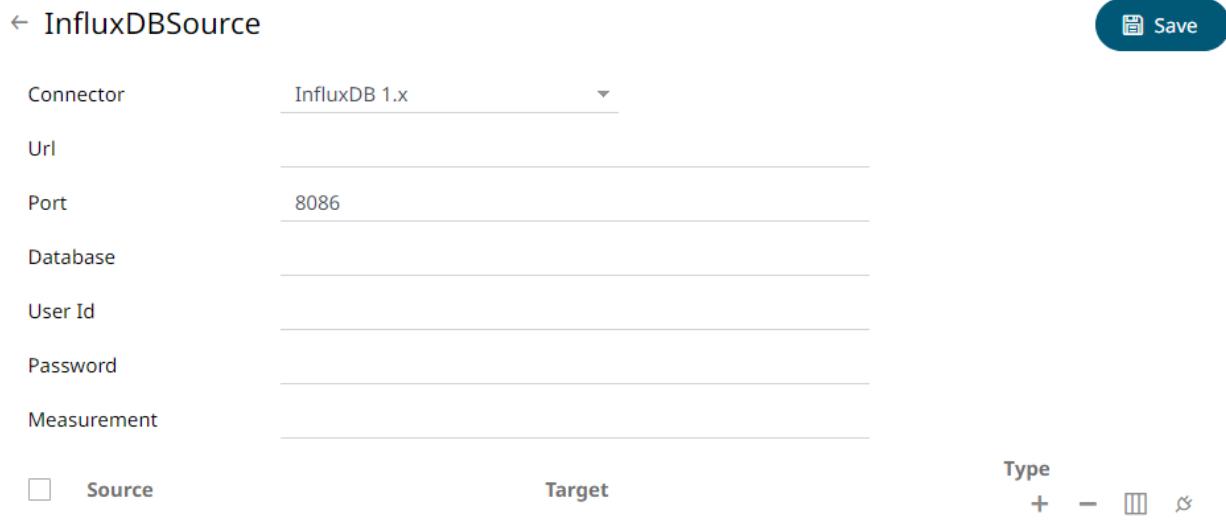
3. Click  . The new data source is added in the *Data Sources* list.

Creating InfluxDB 1.x Output Connector

Allows periodical dumping of data from a Kafka topic into a time series database such as InfluxDB 1.x.

Steps:

1. On the **Data Source** tab, select **Output > InfluxDB 1.x** in the *Connector* drop-down list.



← InfluxDBSource

Connector: InfluxDB 1.x

Url:

Port: 8086

Database:

User Id:

Password:

Measurement:

Type:

2. Define or select the following properties:

Property	Description
URL	URL of InfluxDB 1.x.
Port	The port running the InfluxDB 1.x HTTP service. Default is 8086 .
Database	The name of the database that will be communicate over the HTTP(S).
User Id	The user Id that will be used to connect to InfluxDB 1.x.
Password	The password that will be used to connect to InfluxDB 1.x.
Measurement	The table name that can be used as measurement.

3. You may opt to [use the toolbar](#) to complete the data source definition.



4. Click . The new data source is added to the *Data Sources* list.

Creating JDBC Legacy Output Connector

Allows periodical dumping of records from a Kafka topic into a JDBC database.

Steps:

1. On the **Data Source** tab, select **Output > JDBC Legacy** in the *Connector* drop-down list.

[JDBCLegacyOutput](#)

Connector	<input type="text" value="JDBC Legacy"/>	
JNDI Name	<input type="text"/>	
SqlDialect	<input type="text" value="AnsiSQL"/>	
Table	<input type="text"/>	
<input type="checkbox"/> Source	Target	Type

2. You can either select:

- JNDI Name

Enter the *JNDI resource name* to be used.

NOTE

The JNDI resource name needs to be on the form:

`java:/comp/env/jdbc/[resourcename]`

- URL

Enter the *URL* specific to the database's JDBC driver, the *Driver Class Name* specific to the driver, and the *User Id* and *Password*.

3. Select the appropriate *SQL Dialect* in the drop-down list to be able to generate the correct SQL for the required data repository.

You can select any of the following *SQL dialects*: AnsiSQL, MySQL, Oracle, SQL Server, SAP IQ, SAP ASE, Netezza, Vertica, SQLite, HadoopHive, DB2, PostgreSQL, Impala, Redshift, Informix, Teradata, dBase, SparkSQL.

4. Enter the source *Table* (can be parameterized).

5. You may opt to [use the toolbar](#) to complete the data source definition.



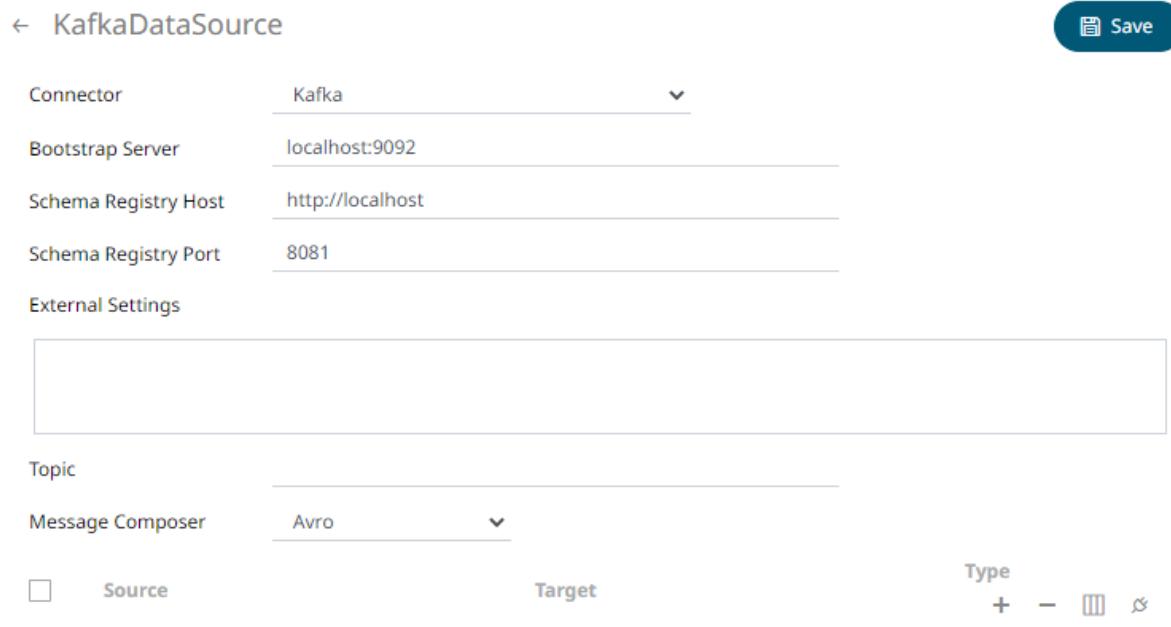
6. Click . The new data source is added to the *Data Sources* list.

Creating Apache Kafka Output Connector

Allows publishing of events to an external Kafka JSON or Avro topic. For Avro, ensure to point towards the schema registry used by the external Kafka cluster.

Steps:

1. On the **Data Source** tab, select **Output > Kafka** in the *Connector* drop-down list.



← KafkaDataSource Save

Connector	Kafka
Bootstrap Server	localhost:9092
Schema Registry Host	http://localhost
Schema Registry Port	8081
External Settings	
Topic	
Message Composer	Avro
<input type="checkbox"/> Source	<input type="checkbox"/> Target
Type + - grid refresh	

2. Enter or select the following properties:

Property	Description
Bootstrap Server	List of host/port pairs of Kafka servers used to bootstrap connections to a Kafka cluster. By default, the value is <code>localhost:9092</code> . However, this can be overridden by specifying another bootstrap server in the <i>External Settings</i> text box (as specified in step 3).
Schema Registry Host	Where the Schema Registry is located. This can be in a different location from the Kafka cluster.
Schema Registry Port	The port number of the schema registry which provides the serving layer for the metadata. Default is 8081 .

3. Enter the *External Settings* to support authentication (i.e., username and password). Note that if the bootstrap server is not secure, then there is no need to authenticate and you may leave this text box blank.

Below is an example of system settings for an SASL authentication:

```
bootstrap.servers=localhost:9093
sasl.jaas.config=\
    org.apache.kafka.common.security.plain.PlainLoginModule required \
```

4. Enter the *Topic* name.
5. Select the Message Composer: **Avro** or **JSON**.
6. You may opt to [use the toolbar](#) to complete the data source definition.

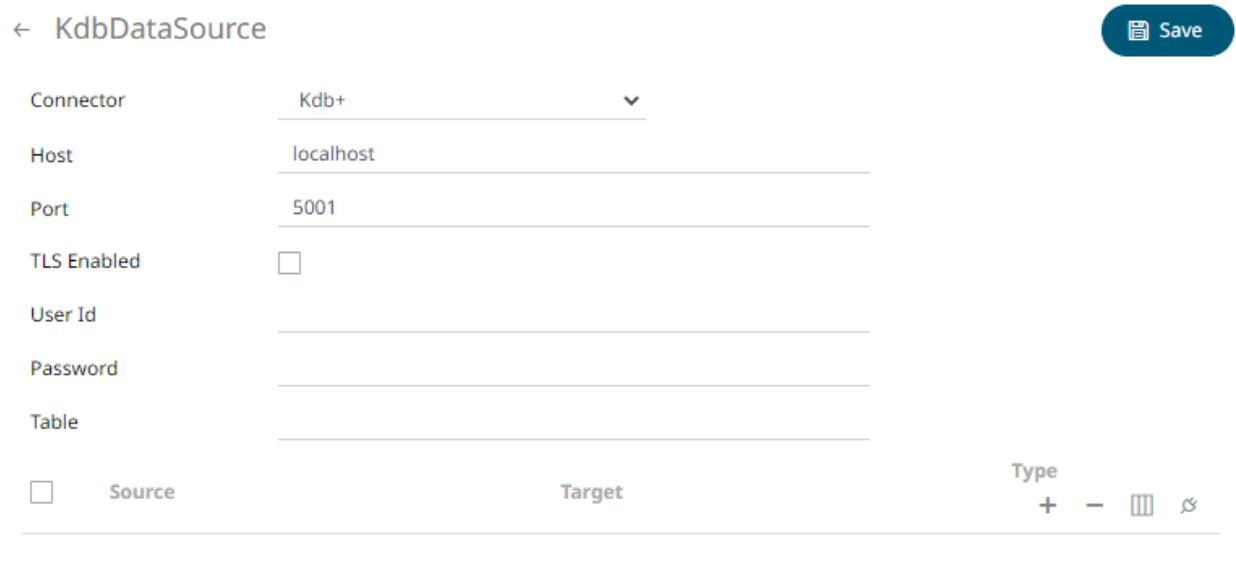
7. Click . The new data source is added in the *Data Sources* list.

Creating Kx kdb+ Output Connector

Allows periodical dumping of records from a Kafka topic into a Kx kdb+ connector.

Steps:

1. On the **Data Source** tab, select **Output > Kdb+** in the *Connector* drop-down list.



← KdbDataSource 

Connector	Kdb+
Host	localhost
Port	5001
TLS Enabled	<input type="checkbox"/>
User Id	
Password	
Table	
Source	Target
<input type="checkbox"/> Type + - ☰ ✖	

2. Define or select the following properties:

Property	Description
Host	Kx kdb+ host address.
Port	Kx kdb+ host port. Default is 5001 .
TLS Enabled	Ensure to check if you have started q with TLS only.
User Id	The user Id that will be used to connect to Kx kdb+.
Password	The password that will be used to connect to Kx kdb+.
Table	The source table.

NOTE

These properties can be parameterized.

3. You may opt to [use the toolbar](#) to complete the data source definition.



4. Click . The new data source is added in the *Data Sources* list.

Creating a MQTT Output Connector

Allows publishing of data to external MQTT topic.

Steps:

1. On the **Data Source** tab, select **Output > MQTT** in the *Connector* drop-down list.

← MQTTDataSource

Connector	MQTT
Broker URL	tcp://localhost:1883
Topic	
User Id	
Password	
Load Type	<input type="button" value="Upload File"/> <input type="button" value="Link To File"/>
CA Certificate	No file selected
Payload Template	

2. Define the following properties:

Property	Description
Broker URL	The location of the message broker. Default is tcp://localhost:1883
Topic	<p>The topic or the queue physical name. Can be parameterized. Example: level1/level2/level3/level4 etc.</p> <p>NOTES: You can also opt to use a wild card in the topic name specification.</p> <ul style="list-style-type: none"> The plus sign symbol (+) can be used as a wild card for any value at one specific level.

	<p>Example: level1/level2/+/level4</p> <ul style="list-style-type: none"> The hash sign symbol (#) can be used as a wild card for any values across more than one level. <p>Example: level1#/level4</p>
User Id	The user Id that will be used to connect to MQTT.
Password	The password that will be used to connect to MQTT.
Payload Template	The template that will be rendered to generate the payload. Can be parameterized with output schema columns.

3. To allow encrypted connections, you can either:

- Upload a CA Certificate file by clicking **Upload File**  then **Browse**  to browse to the file source.

After selecting the file, it is displayed with the timestamp.

<p>Load Type</p> <p>CA Certificate</p>	<p>Upload File  Link To File</p> <p>letsencrypt.cer  Browse</p> <p>as of 2022-08-17 11:13:14</p>
---	---

To change the certificate, click  then **Browse**  to browse to a new version of the file.

- Link to a CA Certificate file by clicking **Link to File**  and entering a *File Path*.

<p>Load Type</p> <p>CA Certificate</p>	<p>Upload File  Link To File</p> <p>(File Type: .crt,.cer,.der,.pem)</p>
---	---

4. You may opt to [use the toolbar](#) to complete the data source definition.

5. Click . The new data source is added in the *Data Sources* list.

Creating a REST Output Connector

Outputs an event to a REST API. This output connector can also be used as an alerting system.

Steps:

- On the **Data Source** tab, select **Output > Rest** in the *Connector* drop-down list.

Connector	Rest
Authentication Type	Basic
Url	<input type="text"/>
User Id	<input type="text"/>
Password	<input type="text"/>
Http Method	▼
Content Type	application/x-www-form-urlencoded
Timeout	10
Request Body	<input type="text"/>

2. Define or select the following properties:

Property	Description
Authentication Type	<ul style="list-style-type: none">• Basic Authentication Type: Basic Url: <input type="text"/> User Id: <input type="text"/> Password: <input type="text"/> Enter the <i>URL</i> of the REST API. Then enter the <i>User Id</i> and the <i>Password</i> that will be used to connect to the REST API.• OAuth

	<p>Authentication Type <input type="text" value="OAuth"/></p> <p>Token Url <input type="text"/></p> <p>Token Request Body <input type="text"/></p> <p>Add Access Token To <input type="text" value="Request Headers"/></p> <p>Url <input type="text"/></p> <p>Then enter the following settings:</p> <ul style="list-style-type: none"> ○ Token URL – The URL to retrieve the access token from. ○ Token Request Body – The request body used for access token requests. ○ Add Access Token To - The Access token retrieved from the <i>Token URL</i> can be added to headers, URL or request body, depending on how the REST endpoint needs the token. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> Request Headers Request Url Request Body </div> <ul style="list-style-type: none"> ▪ Request Header - A header is automatically added to the REST API request. ▪ Request URL - The URL needs to be manually parameterised with a {access_token} parameter, before calling the REST API, the parameter is replaced with the actual token. ▪ Request Body - The Request Body needs to be manually parameterised with a {access_token} parameter, before calling the REST API, the parameter is replaced with the actual token. ○ URL – The URL of the REST API. <p>HTTP Method Select the appropriate HTTP method for the request from the following options:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> GET POST PUT DELETE </div> <ul style="list-style-type: none"> ● GET – retrieve data ● POST – add new data
--	---

	<ul style="list-style-type: none"> • PUT – replace existing data • DELETE – remove existing data
Content Type	The required Content Type. Default is application/x-www-form-urlencoded
Timeout	The length of time to wait for the server response (10 to 300). Default is 10 .
Request Body	The Request Body for the HTTP POST.

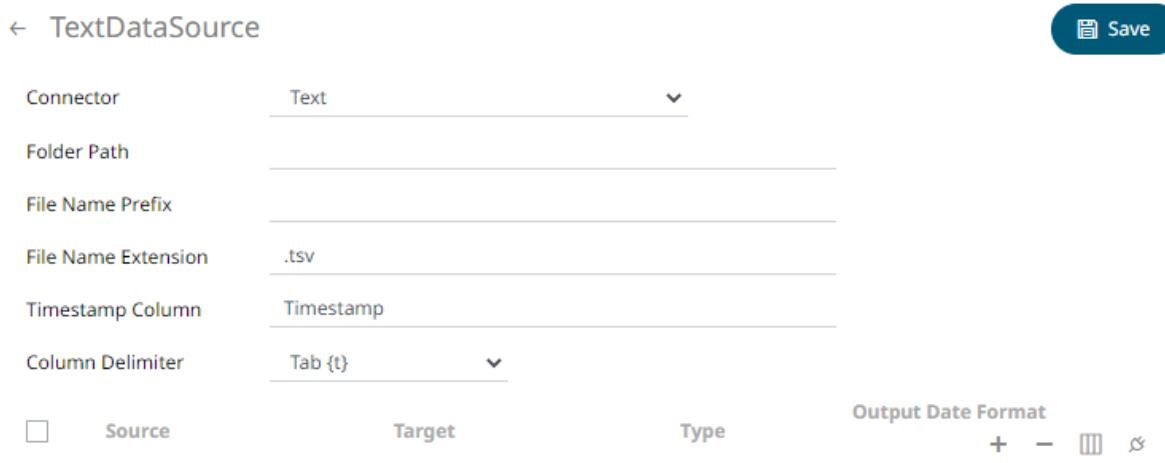
3. Click  . The new data source is added in the *Data Sources* list.

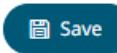
Creating Text Output Connector

Allows retrieval and processing of delimited Text files (such as CSV, TSV, etc.). The files produced can be consumed by the Text connector.

Steps:

1 On the **Data Source** tab, select **Output > Text** in the *Connector* drop-down list.



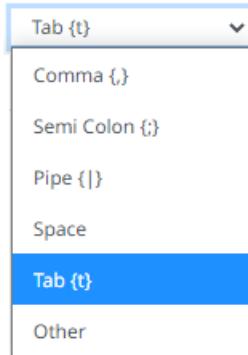
← TextDataSource 

Connector	Text		
Folder Path			
File Name Prefix			
File Name Extension	.tsv		
Timestamp Column	Timestamp		
Column Delimiter	Tab {t}		
Source	Target	Type	Output Date Format
			

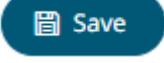
2 Define or select the following properties:

Property	Description
Folder Path	The path where the Text output will be placed.
File Name Prefix	<p>The prefix for the file name.</p> <p>This can be parameterized with field names. Consequently, each event can generate a new file in the given folder.</p> <p>For example, if the Text output connector is attached as the consumer to StockStatic, you can use “{Region}” inside the <i>File Name Prefix</i>, causing it to create three files for Asia Pacific, Europe, and North America.</p> <p>Note that partitioning file names with current date in “yyyyMMdd” format is still done automatically and can’t be controlled, at the moment. For the StockStatic example, if it was executed today, it would have created three files like Asia Pacific_20181219.tsv.</p>
File Name Extension	File name extension of the text output. Possible values are .tsv and .csv .
Timestamp Column	The name of the new column that will include the timestamp. Default is Timestamp .

3. Select the *Column Delimiter* from the drop-down list to be used when parsing the text file.



4. You may opt to [use the toolbar](#) to complete the data source definition.

5. Click . The new data source is added in the *Data Sources* list.

Creating ActiveMQ Input Data Source

Allows connection to Apache’s ActiveMQ message bus on a real-time streaming basis. Specifically, the connector allows Panopticon Streams to subscribe to XML, JSON or FIX based messages that are published on topics. The data format itself is arbitrary, and consequently, the connection includes the message definition.

Steps:

1. In the *New Data Source* page, select **Input > ActiveMQ** in the *Connector* drop-down list.

Connector	ActiveMQ
Broker	tcp://localhost:61616
User Id	
Password	
Topic	topic://topicname.*
Use durable subscription	<input checked="" type="checkbox"/>
Messages can contain partial data	<input type="checkbox"/>
Message Type	Xml
Decimal Separator	Period (.)

Prepend 'default:' for the elements falling under default namespace.

[Generate Columns](#) [Save](#) [Load](#)

<input type="checkbox"/> Name	XPath	Type	Date Format	Enabled	+	-
Real-Time Settings						
Time Id Column	[No Time Id]					
Time Id Column Name						
Reset Data on Reconnect	<input type="checkbox"/>					

2. Enter the following information:

Property	Description
Broker	The location of the message broker. Default is tcp://localhost:61616 .
User Id	The user Id that will be used to connect to the ActiveMQ service.
Password	The password to connect to the ActiveMQ service.
Topic	Accepts topic in <code>topic://topicname.*</code> format and also <code>topicname.*</code> . Therefore, <code>topic://pano.></code> and <code>pano.></code> both will work as topic value. Default is topic://topicname.*

3. Select/unselect the **Use durable subscription** check box.

NOTE

When connecting to a message bus, it is recommended to disable durable messaging. When it is enabled, this puts a heavier load to the server, and slows down the start and stop of subscriptions.

4. Select/unselect **Messages can contain partial data** check box.

5. Select the [Message Type](#).

6. Select either the period (.) or comma (,) as the *Decimal Separator*.

NOTE

Prepend 'default:' for the elements falling under default namespace.

7. Click **Generate Columns** to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the **Save** button is enabled.

8. You can also opt to [load or save](#) a copy of the column definition.

9. You can also opt to click  to add columns to the MQ connection that represent sections of the message. Then enter or select:

Property	Description
Name	The column name of the source schema.
Fix Tag/JsonPath/Text Column Index/XPath	The Fix Tag/JsonPath/Text Column Index/XPath of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Filter	Defined parameters that can be used as filter.
Enabled	Determines whether the message field should be processed.

NOTE

To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of upper case S. There can be no additional characters following them.

For example: yyyy-MM-dd HH:mm:ss.SSSSSS

If *Message Type* is set to **Fix**, the *Add Column* will display as:

<input type="checkbox"/>	Name	Fix Tag	Type	Date Format	Enabled		
<input type="checkbox"/>	Column_		Text				

If *Message Type* is set to **JSON**, the *Add Column* will display as:

<input type="checkbox"/>	Name	JsonPath	Type	Date Format	Filter	Enabled		
<input type="checkbox"/>	Colun		Text					

If *Message Type* is set to **Text**, the *Add column* will display as:

<input type="checkbox"/>	Name	Column Index	Type	Date Format	Filter	Enabled	+	-
<input type="checkbox"/>	Column 0		Text	▼	▼	▼	<input checked="" type="checkbox"/>	

If *Message Type* is set to **XML**, the *Add column* will display as:

<input type="checkbox"/>	Name	XPath	Type	Date Format	Filter	Enabled	+	-
<input type="checkbox"/>	Column		Text	▼	▼	▼	<input checked="" type="checkbox"/>	

To delete a column, check its or all the column entries, check the topmost , then click [-](#).

10. Define the [Real-time Settings](#).



11. Click [Save](#). The new data source is added in the *Data Sources* list.

Creating Altair AI Hub Input Data Source

The Altair AI Hub allows you to request data from endpoints created in Altair AI Hub.

Steps:

1. In the *New Data Source* page, select **Input > Altair AI Hub** in the *Connector* drop-down list.
2. Enter or select the following properties:

Property	Description
Endpoint	<p>Clicking Fetch Endpoints for the first time, or if the offline token is expired, you can enter it in the <i>Offline Token</i> dialog that displays.</p> <div style="border: 1px solid #ccc; padding: 10px; width: fit-content; margin: auto;"> <p>Offline token</p> <div style="border: 1px solid #ccc; height: 100px; margin-top: 10px;"></div> <div style="text-align: right; margin-top: 10px;"> OK Cancel </div> </div> <p>Consequently, the offline token is cached at the AppData repository for subsequent usage.</p>
URL	The Endpoint on Altair AI Hub or RTSA URL.
Authentication Type	<p>Select one from the following:</p> <ul style="list-style-type: none"> • None Can be used with Endpoints that have Consumer Permissions = "Public / anonymous". • Basic Can be used with Endpoints that have Consumer Permissions = "Basic authentication", and a User Id and Password are entered. User Ids can

	<p>be added to each specific Endpoint in Altair AI Hub, in the Deployment settings of the Endpoint.</p> <ul style="list-style-type: none"> • API Token Can be used with Endpoints that have Consumer Permissions = “Long-living API token”, and an API token is entered. API tokens can be created in Altair AI Hub, in the Deployment settings of the Endpoint.
Input Data	By default, this setting has a JSON object entered, with a single name “data” and a value which is a JSON array containing an empty JSON object. The array can contain some number of JSON objects, each being a single name-value pair of column names and column values. The required structure is further exemplified in the Test section of each Endpoint in Altair AI Hub.
Timeout	This is the time in seconds that will be allowed before Panopticon aborts the request. Possible values are one of a fixed set of seconds from 10 to 300.

3. Click **Generate Columns** to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first ‘n’ rows of the input data source is populated and the **Save** button is enabled.
4. You can also opt to [load or save](#) a copy of the column definition.
5. Click  to add columns to the Altair AI Hub connection that represent sections of the message. Then enter or select:

Property	Description
Name	The column name of the source schema.
JsonPath	The JsonPath of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Enabled	Determines whether the message field should be processed.

NOTE

To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of upper case S. There can be no additional characters following them.

For example: yyyy-MM-dd HH:mm:ss.SSSSSS

To delete a column, check its or all the column entries, check the topmost , then click .

6. Click **Save**. The new data source is added to the *Data Sources* list.

Creating Azure Input Data Source

Azure connector allows for retrieval of the file from an Azure blob storage. This connector allows JSON/XML/Text/Excel files to be read from the Azure blob storage.

Steps:

1. In the *New Data Source* page, select **Input > Azure** in the *Connector* drop-down list.

- Enter the following information:

Property	Description
Container	Azure container where the file resides.
Account Name	Azure storage account name.
Account Key	Azure storage account key. To test the connection, click Test Connection . If  Connection Failed displays, ensure the <i>Container</i> , <i>Account Name</i> , and <i>Account Key</i> values are correct. You can also hover on this message to view the connection error.

- Click the **Browse** tab. The available Azure blob files in the container you specified are displayed.
- Select the Azure blob file. The file name is displayed in the *File Path* field and the *Data Type* is updated with its related fields.
- Select the **Only Show Known File Types** checkbox to narrow down the list based on the selected data type.
- Select either the period (.) or comma (,) as the *Decimal Separator*.

NOTE

Prepend 'default:' for the elements falling under default namespace.

- Click **Generate Columns** to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the **Save** button is enabled.
- You can also opt to [load or save](#) a copy of the column definition.

NOTE

This option is not available for the **Excel** data type.

- You can also opt to click  to add columns to the Azure connection that represent sections of the message. Then enter or select:

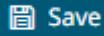
Property	Description
Name	The column name of the source schema.
JsonPath/Column Index/XPath	The JsonPath/Column Index/XPath of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Enabled	Determines whether the message field should be processed.

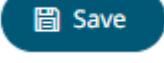
NOTE

To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of upper case S. There can be no additional characters following them.

For example: yyyy-MM-dd HH:mm:ss.SSSSSS

To delete a column, check its or all the column entries, check the topmost , then click  .



10. Click  . The new data source is added to the *Data Sources* list.

Creating AMPS Input Data Source

The AMPS connector allows connection to AMPS message bus on a real-time streaming basis. The connector allows Panopticon Streams to subscribe to the Native FIX and XML message support. The data format itself is arbitrary, and in turn the connection includes the message definition.

Steps:

1. In the *New Data Source* page, select **Input > AMPS** in the *Connector* drop-down list.

Connector	AMPS
Host	localhost
Port	9004
Protocol	Amps
Message Type	Fix
User Id	
Password	
Topic	
Filter	
Subscription Mode	SowAndDeltaSubscribe
Order By	(eg./orderDate DESC, /customerName ASC)
Options	oof,no_empty,
Batch Size	100
Timeout	5000
Decimal Separator	Period {.}

Prepend 'default:' for the elements falling under default namespace.

Generate Columns

<input type="checkbox"/>	Name	XPath	Type	Date Format	Filter	Enabled	+	-
Real-Time Settings								
<input type="checkbox"/>	Time Id Column	[No Time Id]						
Time Id Column Name								
<input type="checkbox"/>	Reset Data on Reconnect							

2. Enter the following information:

Property	Description
Host	AMPS host address.
Port	AMPS host port. Default is 9004 .
User Id	The user Id that will be used to connect to the AMPS service.
Password	The password to connect to the AMPS service.
Topic	The topic or queue physical name.
Filter	The filter expression.

3. Select the *Protocol*. This will specify the format of the headers:
 - Amps (default)
 - Fix
 - NvFix
 - XML
4. Select the [Message Type](#). This will specify the format of the data within the message:
5. Select from any of the following *Subscription Modes*:
 - Sow
 - SowAndSubscribe
 - SowAndDeltaSubscribe (default)
 - Subscribe
 - DeltaSubscribe
6. Enter the *Order By Statement* in order to limit the returned data. For example:

/orderDate DESC

/customerName ASC

7. Enter any of the following *Option/s* for the selected *Subscription Mode*:
 - cancel
 - live
 - no_empty
 - null
 - no_sowkey
 - oof
 - pause
 - replace
 - resume
 - send_keys
 - timestamp

NOTE

Leave the *Options* box blank if you selected the **Subscribe** subscription mode.

8. Enter the *Batch Size*. This is the number of messages that will be sent at a time as results are returned. Default is **100**.
9. Enter the *Timeout* for the length of time to wait for the Server response. Default is **5000**.

10. Select either the period (.) or comma (,) as the *Decimal Separator*.

NOTE

Prepend 'default:' for the elements falling under default namespace.

11. Click  to the fetch the schema based on the connection details. This populates the list of columns with the data type found from inspecting the first 'n' rows of the input data source.
12. You can also opt to click . This adds columns to the AMPS connection that will represent sections of the message.
13. Provide the following information:

Property	Description
Name	The column name of the source schema.
Fix Tag/XPath/Json Path	The Fix Tag/XPath/Json Path of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Filter	Defined parameters that can be used as filter. Only available for Fix, JSON, and XML message types.
Enabled	Determines whether the message field should be processed.

To delete a column, check its  or all the column entries, check the topmost  , then click .

14. Define the [Real-time Settings](#).



15. Click . The new data source is added in the *Data Sources* list.

Creating Cassandra Input Data Source

The Apache Cassandra connector allows connection to Apache and Datastax Cassandra instances, by executing a pre-defined CQL query, and retrieving the resulting data.

Steps:

1. On the *New Data Source* page, select **Cassandra** in the *Connector* drop-down list.

Connector	Cassandra	▼
Host	localhost	
Port	9042	
KeySpace		
User Id		
Password		
Enclose parameters in quotes	<input type="checkbox"/>	
CQL Query	<pre>CREATE TABLE users (user_id int, user_name text, user_email text, user_password text, PRIMARY KEY (user_id));</pre>	

2. Enter the following information:

Property	Description
Host	Apache Cassandra host address.
Port	Apache Cassandra host port. Default is 9042 .
KeySpace	Namespace that defines data replication in nodes.
User Id	The username used to connect to the Apache Cassandra service.
Password	The password used to connect to the Apache Cassandra service.

3. Select whether the parameters should be automatically enclosed in quotes, by selecting the **Enclose parameters in quotes** check box.

4. Enter the *CQL Query*, which can contain parameters in a similar manner to the database connector.

5. The time zone of input parameters and output data is by default, unchanged. Changing the time zone is supported by using the *Show in Timezone* drop-down list box based on the assumption that data are stored in UTC time and outputs are presented in the selected time zone.

6. Click  Save. The new data source is added in the *Data Sources* pane.

Creating DolphinDB Input Data Source

The DolphinDB connector allows you to connect and query tables using DolphinDB java API.

Steps:

1. On the *New Data Source* page, select **DolphinDB** in the *Connector* drop-down list.

← **DolphinDBInput** Save

Connector	DolphinDB
Host	localhost
Port	8848
User Id	
Password	<input type="password"/> Show characters
Query	<div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div>

2. Enter the following information:

Property	Description
Host	DolphinDB host address.
Port	DolphinDB host port. Default is 8848 .
User Id	The username used to connect to the DolphinDB service.
Password	The password used to connect to the DolphinDB service. Check the <i>Show Characters</i> box to display the entered password characters.

3. Enter the *Query*, which can contain parameters in a similar manner to the database connector.

4. Click Save. The new data source is added in the *Data Sources* pane.

Creating DolphinDB – Streaming Input Data Source

The DolphinDB streaming connector allows you to connect and subscribe streaming data using DolphinDB Java Streaming API.

Steps:

1. On the *New Data Source* page, select **DolphinDB – Streaming Beta** in the *Connector* drop-down list.

Connector	DolphinDB - Streaming Beta
Host	localhost
Port	8848
User Id	
Password	<input type="password"/> <input type="checkbox"/> Show characters
Table	
From Beginning	<input type="checkbox"/>
Real-Time Settings	
Time Id Column	[No Time Id]
Time Id Column Name	
Reset Data on Reconnect	<input type="checkbox"/>

2. Enter the following information:

Property	Description
Host	DolphinDB - Streaming host address.
Port	DolphinDB - Streaming host port. Default is 8848.
User Id	The user Id that will be used to connect to the DolphinDB - Streaming service.
Password	The password to connect to the DolphinDB - Streaming service. Check the <i>Show Characters</i> box to display the entered password characters.
Table	Table to subscribe against.

3. Check the *From Beginning* box to subscribe from the beginning to the latest messages.

From Beginning

If un-checked, you will only be subscribed to the latest messages.

4. Click **Fetch Schema** to retrieve the schema of the configured subscription.

This populates the *Id Column* with the set of columns from the schema of type `sym` and the text array such as Character/Boolean/GUID, etc. The selected *Id Column* can be used to select a key column to manage data updates and inserts.

NOTE: Every message definition needs a text column to be defined as the Id column. By default, only the latest data will be loaded into memory.

Furthermore, a streaming time series window can be generated by creating a compound key with the *Id Column*, plus a separately specified *Time ID* column. This *Time ID* column can be from the source dataset, or alternatively automatically generated.

If the *Time Id column* is selected, then a scrolling time window can be specified.

Time Id Column	[Automatic Time Id] ▾
Time Id Column Name	Automatic_Timestamp_Column

For **Automatic Time Id**, define the *Time Id Column Name*.

As new data arrives from the subscription, new time slices will automatically be added, and old ones will be deleted.

If a new Id is received, a new row is added to the in-memory data set representing the DolphinDB - Streaming topic subscription. While if an existing ID is received, an existing row is updated.

5. Modify the [Real-time settings](#) if further changes are required.



6. Click . The new data source is added in the *Data Sources* pane.

Creating Google Analytics Input Data Source

The Google Analytics connector allows you to track and report website traffic using the Google Analytics service. You will need the following to fetch Google Analytics data:

Field	Description
Service Account E-mail	The Service Account ID that is generated while creating credentials for the service account authentication.
Key File	The Key File (.p12) furnished by Google Analytics when you created the Service Account.
Profile ID	The Profile ID of the page you want to access in Google Analytics.

Steps:

1. In the *New Data Source* page, select **Input > Google Analytics** in the *Connector* drop-down list.

Connector	Google Analytics
Service Account Email	
Load Type	<input type="button" value="Upload File"/> <input type="button" value="Link To File"/>
File	No file selected <input type="button" value="Browse"/>
Profile Id	
Start Date	
End Date	
Fetch Dimensions And Metrics	
Filter By Categories	<input type="button" value="+"/>
Dimensions	<input type="button" value="+"/>
Metrics	<input type="button" value="+"/>

2. Enter the *Service Account Email* that was generated while creating credentials to the service account authentication.
3. Set the *Key File* that will be used to connect to Google Analytics in Panopticon. Do one of the following:

- Upload the *Personal Information Exchange* file by clicking **Upload File** then **Browse** to browse to the file.

After selecting the file, it is displayed with the timestamp.

Load Type	<input type="button" value="Upload File"/> <input type="button" value="Link To File"/>
File	My Project-4e22fbcadf9.p12 <input type="button" value="X"/> <input type="button" value="Browse"/>
as of 2022-06-15 15:30:01	

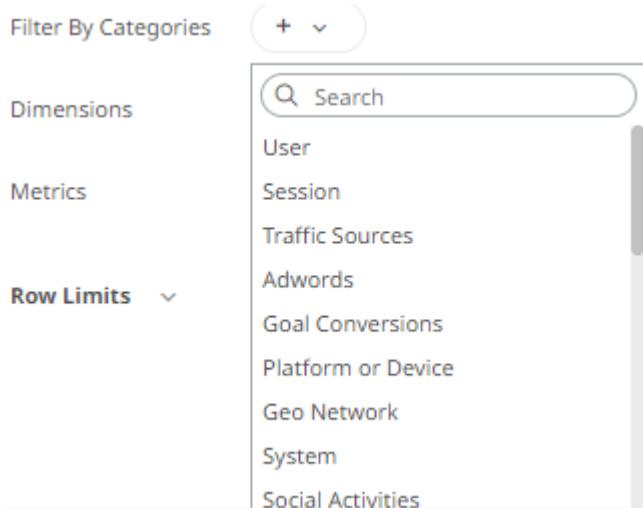
To change the key file, click then **Browse** to browse to a new version of the file.

- Link to a *Personal Information Exchange* file by clicking **Link to File** and entering a *File Path*.

Load Type	<input type="button" value="Upload File"/> <input type="button" value="Link To File"/>
Key File Path	(File Type: .p12)

4. Enter the *Profile ID* of the page you want to access in Google Analytics.
5. Enter the *Start Date* and *End Date*, if needed.
6. Click **Fetch Dimensions And Metrics** . This populates the *Filter By Categories*, *Dimensions*, and *Metrics* list boxes.

7. Click  then select any field from these list boxes.



Filter By Categories 

Dimensions

Metrics

Row Limits 

User

Session

Traffic Sources

Adwords

Goal Conversions

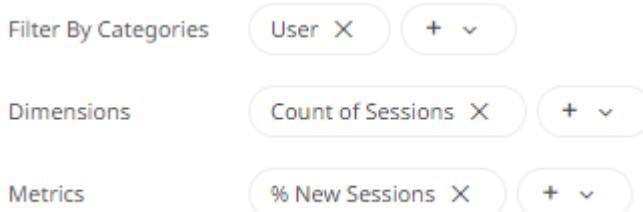
Platform or Device

Geo Network

System

Social Activities

For example:

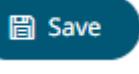


Filter By Categories  

Dimensions Count of Sessions  

Metrics % New Sessions  

Click  to remove a field.

8. Click . The new data source is added in the *Data Sources* list.

Creating Google Cloud Input Data Source

Google Cloud connector allows for retrieval of the file from Google Cloud storage. This connector allows JSON/XML/Text/Excel files to be read from the Google Cloud storage.

Steps:

1. In the *New Data Source* page, select **Input > Google Cloud** in the *Connector* drop-down list.

- Enter the following information:

Property	Description
Bucket	Google Cloud bucket where the file resides.
Access Key	Access key to your Google Cloud service account.
Secret Key	Secret key to your Google Cloud service account. To test the connection, click Test Connection . If  Connection Failed displays, ensure the <i>Bucket</i> , <i>Access Key</i> , and <i>Secret Key</i> values are correct. You can also hover on this message to view the connection error.

- Click the **Browse** tab. The available Google Cloud store files in the bucket you specified are displayed.
- Select the Google Cloud file. The file name is displayed in the *File Path* field and the *Data Type* is updated with its related fields.
- Select the **Only Show Known File Types** checkbox to narrow down the list based on the selected data type.
- Select either the period (.) or comma (,) as the *Decimal Separator*.

NOTE

Prepend 'default:' for the elements falling under default namespace.

- Click **Generate Columns** to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the **Save** button is enabled.
- You can also opt to [load or save](#) a copy of the column definition.

NOTE

This option is not available for the **Excel** data type.

- You can also opt to click  to add columns to the Google Cloud connection that represent sections of the message. Then enter or select:

Property	Description
Name	The column name of the source schema.
JsonPath/Column Index/XPath	The JsonPath/Column Index/XPath of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Enabled	Determines whether the message field should be processed.

NOTE

To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of upper case S. There can be no additional characters following them.

For example: yyyy-MM-dd HH:mm:ss.SSSSSS

To delete a column, check its or all the column entries, check the topmost , then click .

10. Click . The new data source is added in the *Data Sources* list.

Creating Google Cloud Pub/Sub Input Data Source

The Google Cloud Pub/Sub connector allows connection to Google Cloud Pub/Sub's message bus on a real-time streaming basis. Specifically, the connector allows Panopticon Streams to subscribe to XML, JSON, TEXT or FIX based messages that are published on particular topics. The data format itself is arbitrary, and consequently, the connection includes the message definition.

Steps:

1. In the *New Data Source* page, select **Input > Google Cloud PubSub** in the *Connector* drop-down list.

Connector

Google Cloud PubSub



Service Account Credential Json Text

Topic



Fetch

Subscription Name



Fetch

Message Type

Json



Decimal Separator

Period {.}



Record Path

(eg. myroot.items.item)

Generate Columns

Save

Load

<input type="checkbox"/>	Name	JsonPath	Type	Date Format	Enabled	+	-
<input type="checkbox"/>	TradeTi		Time		<input checked="" type="checkbox"/>		

Attribute Columns

<input type="checkbox"/>	Name	Attribute Name	Enabled	+	-

Real-Time SettingsTime Id Column Time Id Column Name Reset Data on Reconnect

2. Enter the *Service Account Credential JSON Text* with the generated JSON key (contains the private key) in the following format:

```
{  
  "type": "service_account",  
  "project_id": "project-id",  
  "private_key_id": "some_number",  
  "private_key": "-----BEGIN PRIVATE KEY-----\n....  
=\n-----END PRIVATE KEY-----\n",  
  "client_email": "<api-name>api@project-id.iam.gserviceaccount.com",  
  "client_id": "...",  
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",  
  "token_uri": "https://accounts.google.com/o/oauth2/token",  
  "auth_provider_x509_cert_url":  
  "https://www.googleapis.com/oauth2/v1/certs",  
  "client_x509_cert_url": "https://www.googleapis.com/...<api-  
name>api%40project-id.iam.gserviceaccount.com"  
}
```

NOTE

Ensure that when parameterizing the values in the Credential JSON Text, there is no white space as a single line content.

3. Click **Fetch** to populate the *Topic* drop-down list. Initially, the first topic in the list is displayed in the *Topic* drop-down box.

Select a topic.

4. Click **Fetch** to populate the *Subscription Name* drop-down list and select a subscription name.

You can also opt to create a subscription by manually entering the value into the *Subscription Name* list box.

NOTE

- A subscription name will be automatically generated when it is not entered or selected in the drop-down list.

This subscription will be created for connection and will be deleted as soon as its work is done. For example, when starting a presentation mode, a subscription will be created. Upon quitting the presentation mode, the subscription will then be deleted.
- Pub/Sub can automatically delete inactive subscriptions. This can be done by configuring the minimum required time of inactivity to schedule a subscription for deletion. This time must be longer than the message retention duration.

5. Select the [Message Type](#).

- elect either the period (.) or comma (,) as the *Decimal Separator*.

NOTE

Prepend 'default:' for the elements falling under default namespace.

- Click **Generate Columns** to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the Save button is enabled.
- You can also opt to [load or save](#) a copy of the column definition.
- Click . This adds columns to the Google Cloud Pub/Sub connection that will represent sections of the message.
- Provide the following information:

Property	Description
Name	The column name of the source schema.
Fix Tag/XPath/Json Path	The Fix Tag/XPath/Json Path of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Filter	Defined parameters that can be used as filter. Only available for JSON, Text, and XML message types.
Enabled	Determines whether the message field should be processed.

To delete a column, check its  or all the column entries, check the topmost , then click .

- Google Cloud Pub/Sub messages can have additional metadata as custom attributes.

Panopticon Google Cloud Pub/Sub connector supports reading these attributes as column values. The generate column logic automatically checks and generates attribute columns if messages received contain attributes.

Additionally, like columns from message data, you can manually add them by clicking . A new entry displays.

Attribute Columns

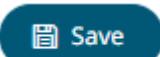
<input type="checkbox"/> Name	Attribute Name	Enabled		
<input type="checkbox"/> Attribute_1	Attribute_1	<input checked="" type="checkbox"/>		

Name can be any unique column name within the data source. The attribute name must match to an attribute name in message otherwise it will be treated as null value. Currently all attribute columns are treated as Text columns, we can't change column type.

Select the *Enabled* check box to enable an attribute column.

To delete an attribute column, check its  or all the column entries, check the topmost , then click .

12. Define the Real-time Settings.



13. Click . The new data source is added in the *Data Sources* list.

Creating an InfluxDB 1.x Input Data Source

The InfluxDB 1.x connector allows for the retrieval of a JSON data set from InfluxDB 1.x. The database communicates over HTTP(S) where you can define a query in the URL to return the desired data.

Steps:

1. In the *New Data Source* page, select **Input > InfluxDB 1.x** in the *Connector* drop-down list.

← InfluxDBInput

 Save

Connector	InfluxDB 1.x	
Url	<input type="text"/>	
Port	8086	<input type="text"/>
User Id	<input type="text"/>	
Password	<input type="password"/>	<input type="checkbox"/> Show characters
Database	<input type="text"/>	
Time out (Secs)	10	<input type="text"/>
Query	<input type="text"/>	

2. Enter the following information:

Property	Description
URL	InfluxDB 1.x host address.
Port	InfluxDB 1.x host port. Default is 8086 .
User Id	The user Id that will be used to connect to the InfluxDB 1.x service.
Password	The password to connect to the InfluxDB 1.x service. Select the Show Characters check box to display the entered characters.
Database	The name of the database that will communicate over the HTTP(S).
Time out (Secs)	The time out period applied to both the TCP socket and for individual read IO operations. Default is 10 .

3. Enter an SQL-like query language into the *Query* box.

 Save

4. Click . The new data source is added to the *Data Sources* list.

Creating JDBC Legacy Input Data Source

The JDBC Legacy connector allows the retrieval and processing of data from virtually any database that has a JDBC driver.

IMPORTANT

For DolphinDB, the query builder is not supported, only the query mode.

Steps:

1. In the *New Data Source* page, select **Input > JDBC Legacy** in the *Connector* drop-down list.

← JDBCLegacyInput  Save

Connector	JDBC Legacy	
JNDI Name	<small>(JNDI resource name as defined inside Context eg. jdbc/MyDB)</small>	
SqlDialect	AnsiSQL	
Timeout	60	
Enclose parameters in quotes	<input checked="" type="checkbox"/>	
Allow In-Memory parameter filtering	<input type="checkbox"/>	
Use data modification query	<input type="checkbox"/>	
<input checked="" type="radio"/> Table		
Table	<input type="button" value="Load"/>	
Search Tables		
Join Table	Left Column	Right Column
<input type="button" value="Generate Columns"/>		
<input type="checkbox"/> Column	<input type="checkbox"/> Parameterize	<input type="checkbox"/> Aggregate
Date Time	<input type="button" value="From"/>	<input type="button" value="To"/>
<input type="checkbox"/> Constrain By Date Time		
<input type="radio"/> Query		

2. You can either select:

- JNDI Name

JNDI Name	(JNDI resource name as defined inside Context eg. jdbc/MyDB)
-----------	--

Enter the *JNDI resource name* to be used.

NOTE	<p>The JNDI resource name needs to be on the form:</p> <p>jdbc/ [resourcename]</p>
-------------	--

- URL

URL	<input type="text"/>
Driver Class Name	<input type="text"/>
User Id	<input type="text"/>
Password	<input type="password"/> <input type="checkbox"/> Show characters

Enter the *URL* specific to the database's JDBC driver, the *Driver Class Name* specific to the driver, and the *User Id* and *Password*.

Select the **Show Characters** check box to display the entered characters.

3. Select the appropriate *SQL Dialect* in the drop-down list to be able to generate the correct SQL for the required data repository.

You can select any of the following *SQL dialects*: AnsiSQL, MySQL, Oracle, SQL Server, SAP IQ, SAP ASE, Netezza, Vertica, SQLite, HadoopHive, DB2, PostgreSQL, Impala, Redshift, Informix, Teradata, dBase, SparkSQL.

Default is **AnsiSQL**.

4. Enter the *Timeout*. This is the length of time to wait for the server response. Default is **60**.

5. Check any of the following options when building the query:

- Enclose parameters in quotes

By default, this option is checked, as the common use case for parameters is a filter `WHERE` clause.

- Allow in-memory parameter filtering

Allows the whole dataset to be returned, and then filtered in memory. This process is much less efficient than adding the parameter as a `WHERE` clause of the SQL query; however, it may be efficient in cases where small sets of records are returned on a very frequent basis.

- Use data modification query

Signals that the table is created for writing data. This property is also used for filtering out target data tables for further data update action configuration

6. When **Table** is selected, the section below is enabled:

Table

Table **Load**

Search Tables

Join Table	Left Column	Right Column
<input type="checkbox"/> Column	<input type="checkbox"/> Parameterize	<input type="checkbox"/> Aggregate
Date Time	or	+
<input type="checkbox"/> Constrain By Date Time	From <input type="text"/>	To <input type="text"/>

7. On the *Table* field, click **Load** to populate the drop-down list with tables. Select a table.

The list of tables that you can join is displayed. Also, the SQL query is generated and displayed in the *Query* text box.

Table Table

Table **Load**

Search Tables

Join Table	Left Column	Right Column
<input type="checkbox"/> public.forex	<input type="text"/>	<input type="text"/>
<input type="checkbox"/> public.industry	<input type="text"/>	<input type="text"/>

Generate Columns

<input type="checkbox"/> Column	<input type="checkbox"/> Parameterize	<input type="checkbox"/> Aggregate
Date Time	or	+
<input type="checkbox"/> Constrain By Date Time	From <input type="text"/>	To <input type="text"/>

Query

```
1 SELECT * FROM "public"."stocks"
```

Use *Search Tables* to filter the list.

forex

Join Table	Left Column	Right Column
<input type="checkbox"/> public.forex	<input type="text"/>	<input type="text"/>

8. Perform a join by checking one or more tables in the list.

The *Left Column* and *Right Column* fields are automatically filled out with the common fields.

Table

Table	public.stocks	Load
Search Tables		
Join Table	Left Column	Right Column
<input checked="" type="checkbox"/> public.forex	id	id
<input type="checkbox"/> public.industry		

You can also opt to select other common fields.

The SQL query is generated and displayed in the *Query* text box.

Search Tables		
Join Table	Left Column	Right Column
<input checked="" type="checkbox"/> public.forex	forex	forex
<input type="checkbox"/> public.industry		
Generate Columns		
<input type="checkbox"/> Column	<input type="checkbox"/> Parameterize	<input type="checkbox"/> Aggregate
Date Time	or	+
<input type="checkbox"/> Constrain By Date Time	From	To
<input type="radio"/> Query		
<pre>1 SELECT * FROM ("public"."stocks" LEFT JOIN "public"."forex" on "stocks"."forex" = "forex"."forex")</pre>		

9. Click **Generate Columns**. The columns populate the *Output Column* section.

<input type="checkbox"/> Column	<input type="checkbox"/> Parameterize	<input type="checkbox"/> Aggregate
<input type="checkbox"/> stocks.id		Sum
<input type="checkbox"/> stocks.region		Group By
<input type="checkbox"/> stocks.country		Group By
<input type="checkbox"/> stocks.forex		Group By
<input type="checkbox"/> stocks.mcaplocal		Group By
<input type="checkbox"/> forex.id		Sum
<input type="checkbox"/> forex.forex		Group By
<input type="checkbox"/> forex.exchange		Group By

10. Individual columns can be added by selecting the corresponding *Column* check box in the *Output Column* listing. To select all of the columns, select the topmost check box.

The SQL query is generated and displayed in the *Query* text box.

11. If the data returned is to be aggregated, then the **Aggregate** box should be checked. For each selected column, the possible aggregation methods are listed including:

- Text Columns: Last, First, Count, Group By
- Date Columns: Count, Min, Max, Group By
- Numeric Columns: Last, First, Sum, Count, Min, Max, Mean, Group By

The SQL query is generated and displayed on the *Query* text box.

12. Select the **Parameterize** check box and match the parameter to the appropriate column. By default, they will be matched by name.

The appropriate SQL Query is updated in the *Query* text box. This shows the default parameter value for the preview, and at run time the SQL will be updated to whatever the parameter value is.

13. If the data is to be filtered or aggregated on Date/Times, then a valid *Date Time* field needs to be selected from either a single Date/Time field, or a compound column created from a selected *Date* and a selected *Time* column.

 or   

14. Select the **Constrain by Date Time** check box and enter *From* and *To* Date/Time constraints.

15. Click the **Query** radio button to enable the text box and modify the SQL-like query language.

 Save

16. Click . The new data source is added to the *Data Sources* list.

Creating JDBC Input Data Source

JDBC connector is the new version of JDBC Legacy connector and is the recommended connector for any new JDBC connectivity for better performance and configuration UI. Just like JDBC Legacy connector, it also allows the retrieval and processing of data from virtually any database, that has a JDBC driver.

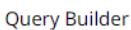
Steps:

1. In the *New Data Source* page, select **Input > JDBC** in the *Connector* drop-down list.

 JDBCInput

 Save

Connector 

 Connection  Query Builder  SQL  Advanced

JDBC Driver 

URL 

User Id 

Password  Show characters

Timeout  60

2. On the **Connection** tab, set either of the following connection settings:

- URL

URL

Driver Class Name

User Id

Password

Show characters

Select the *JDBC Driver* then enter its specific *URL*, and the *User Id* and *Password*.

Select the **Show Characters** checkbox to display the entered characters.

- JNDI Name

JNDI Name

(JNDI resource name as defined inside Context eg. jdbc/MyDB)

Enter the *JNDI resource name* to be used.

NOTE

The JNDI resource name needs to be on the form:

`jdbc/ [resourcename]`

3. Adjust the *Timeout*, if needed. Default is **60**.
4. Query definition and execution can be done, using either the query builder or freeform SQL. To use the query builder, select the **Query Builder** tab. Otherwise, proceed to step 5.

The **Use Query Builder** option is turned on by default.

Connection
Query Builder
SQL
Advanced

Use Query Builder

SqlDialect

Table

Load

Join Tables

Generate Columns

Column

Parameterize

Aggregate

Date Time

or

+

Constrain By Date Time

From

To

Preview Query

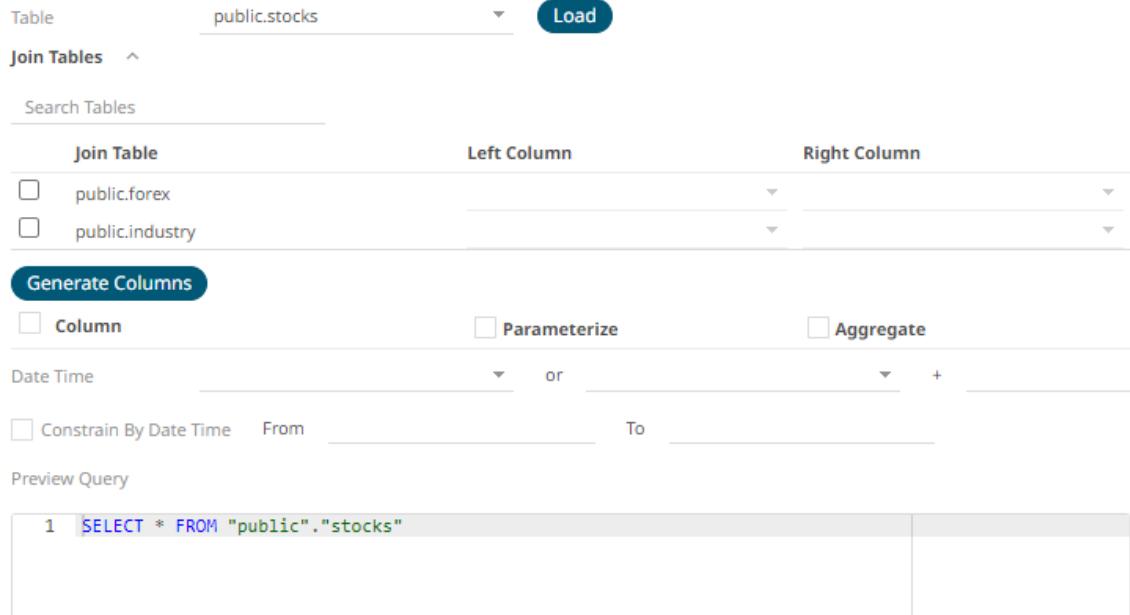
```
1 SELECT * FROM
```

- 4.1. Select the appropriate *SQL Dialect* in the drop-down list to be able to generate the correct SQL for the required data repository.

4.2. In the *Table* field, click  to populate the drop-down list with tables. Select a table.

The SQL query is generated and displayed in the *Preview Query* text box.

Also, expanding the *Join Tables* displays the list of tables that you can join.



Table 

Join Tables 

Search Tables

Join Table	Left Column	Right Column
<input type="checkbox"/> public.forex		
<input type="checkbox"/> public.industry		

Generate Columns

Column Parameterize Aggregate

Date Time  or  

Constrain By Date Time From To

Preview Query

```
1 SELECT * FROM "public"."stocks"
```

Use *Search Tables* to filter the list.



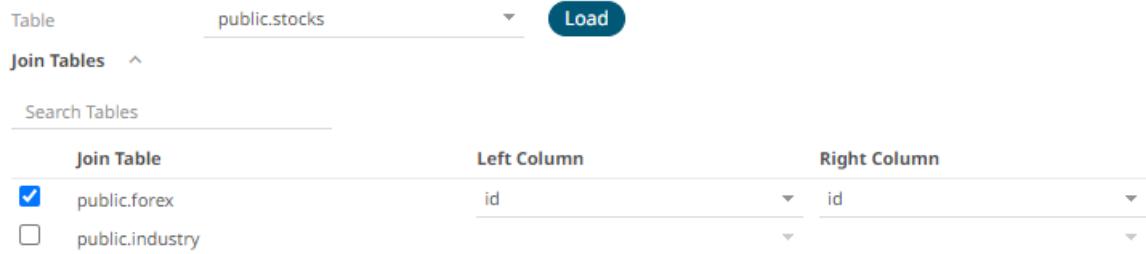
forex

Table 

Join Tables 

Join Table	Left Column	Right Column
<input type="checkbox"/> public.forex		

4.3. Perform a join by checking one or more tables in the list.



Table 

Join Tables 

Search Tables

Join Table	Left Column	Right Column
<input checked="" type="checkbox"/> public.forex	id	id
<input type="checkbox"/> public.industry		

You can also opt to select other common fields.

The SQL query is generated and displayed in the *Preview Query* text box.

Join Tables ^

Search Tables

Join Table	Left Column	Right Column
<input checked="" type="checkbox"/> public.forex	forex	forex
<input type="checkbox"/> public.industry		

Generate Columns

Column Parameterize Aggregate

Date Time or +

Constrain By Date Time From _____ To _____

Preview Query

```
1 | SELECT * FROM ("public"."stocks" LEFT JOIN "public"."forex" on "stocks"."forex" = "forex"."forex")
```

4.4. Click **Generate Columns**. The columns populate the *Output Column* section.

Generate Columns

Column Parameterize Aggregate

<input type="checkbox"/> stocks.id	Sum
<input type="checkbox"/> stocks.region	Group By
<input type="checkbox"/> stocks.country	Group By
<input type="checkbox"/> stocks.forex	Group By
<input type="checkbox"/> stocks.mcaplocal	Group By
<input type="checkbox"/> forex.id	Sum
<input type="checkbox"/> forex.forex	Group By
<input type="checkbox"/> forex.exchange	Group By

4.5. Individual columns can be added by selecting the corresponding *Column* check box in the *Output Column* listing. To select all of the columns, select the topmost check box.

The SQL query is generated and displayed in the *Preview Query* text box.

4.6. If the data returned is to be aggregated, then the **Aggregate** box should be checked. For each selected column, the possible aggregation methods are listed including:

- ◆ Text Columns: Count, Group By
- ◆ Date Columns: Count, Group By
- ◆ Numeric Columns: Sum, Count, Min, Max, Group By

4.7. Select the **Parameterize** check box and match the parameter to the appropriate column. By default, they will be matched by name.

The appropriate SQL Query is updated in the *Preview Query* text box.

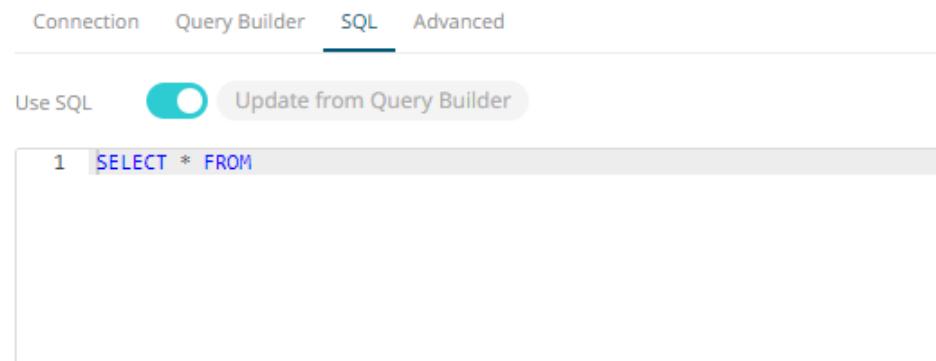
4.8. If the data is to be filtered or aggregated on Date/Times, then a valid *Date Time* field needs to be selected from either a single Date/Time field, or a compound column created from a selected *Date* and a selected *Time* column.

Date Time or +

4.9. Select the **Constrain by Date Time** check box, and enter *From* and *To* Date/Time constraints that are assumed to be in this time zone for incorporation into the query.

If the query is to filter/constrain the results on Date/Time, the constrain sections are completed.

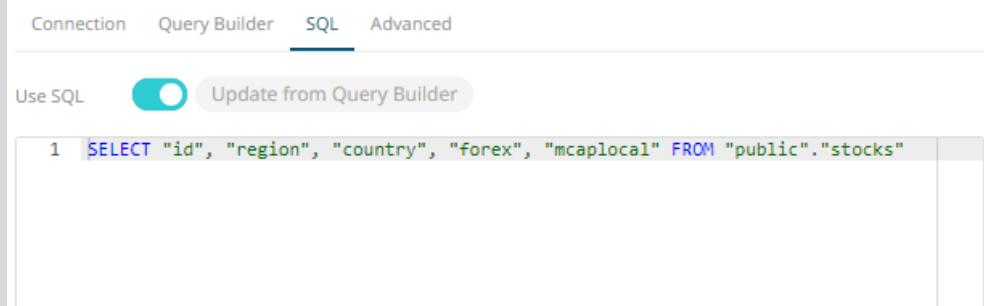
5. To use freeform SQL, select the **SQL** tab and turn on **Use SQL** toggle button.



5.1. Modify the SQL-like query language in the *User Query* text box.

NOTE

- If you initially used the Query Builder then switched to the freeform SQL option, the content of *Preview Query* is copied to the *User Query* text box.

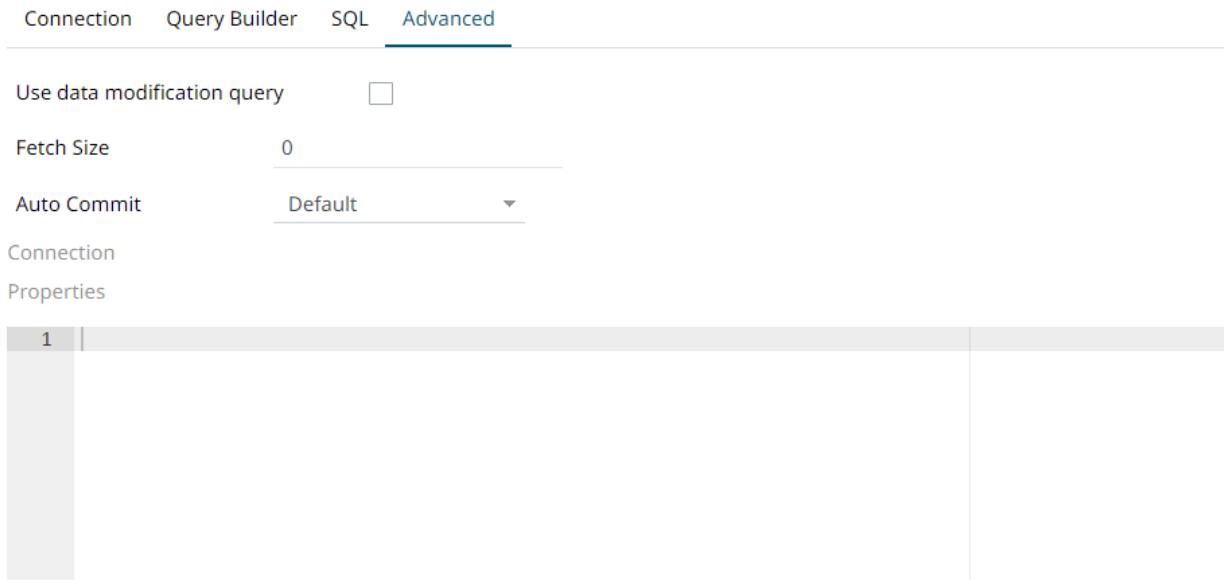


- Switching back to the Query Builder, the *Preview Query* is updated, keeping the *User Query* unmodified.

- The 'Update from Query Builder' button is enabled when *User Query* is non-empty and different from *Preview Query*.

Click this button to update the *User Query* from the query builder.

6. Select the **Advanced** tab.



Set the following property, if needed:

Property	Description
Use data modification query	Signals that the table is created for writing data. This property is also used for filtering out target data tables for further data update action configuration.
Fetch Size	Sets the number of rows to fetch per iteration.
Auto Commit	Postgres ignores fetch size if auto commit is not set to False . You would need to explicitly set it to force when using fetch size.
Connection Properties	NOTE: This property is only applicable for URL connection. Enter Java-style properties format which can consist of a series of lines (terminated by CRLF , CR or LF) where each is a key-value pair, a comment, or a blank line.

7. Click  . The new data source is added to the *Data Sources* list.

Creating a JSON Input Data Source

The JSON connector allows the retrieval and processing of JSON files, either from a disk, a Text, or from a defined URL.

Steps:

1. In the *New Data Source* page, select **Input > Json** in the *Connector* drop-down list.

Connector	Json					
JSON File Source	File					
Load Type	Upload File Link To File					
JSON File Path	No file selected Browse					
Decimal Separator	Period (.)					
File Encoding						
Record Path	Fetch					
Array Handling	Add Rows					
Generate Columns Save Load						
<input type="checkbox"/> Name	JsonPath	Type	Date Format	Enabled	+	-

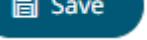
2. Select the JSON [File Source](#).
3. Select either the period (.) or comma (,) as the *Decimal Separator*.
4. Set the *File Encoding* to use:
 - UTF-8
 - UTF-16
 - UTF-32
 - US-ASCII
 - Windows-1252
5. Click [Fetch](#) to fetch the *Record Paths* then select one.
6. Set the *Array Handling* setting to control how the data table will be created and to accommodate the array of values. Can be any of the following:
 - Add Rows (default) - one row will be created for each value in the array
 - Add Columns - one column will be created for each value in the array
 - Concatenate - one column will be created for all values in the array, concatenated with a delimiter character
7. Click [Generate Columns](#) to fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the Save button is enabled.
6. You can also opt to [load or save](#) a copy of the column definition.
7. Click [+](#) to add columns and specify their properties:

Property	Description
----------	-------------

Name	The column name of the source schema.
Json Path	The Json Path of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Enabled	Determines whether the message field should be processed.

<input type="checkbox"/>	Name	JsonPath	Type	Date Format	Enabled	+	-
<input type="checkbox"/>	Column_1		Text	▼	<input checked="" type="checkbox"/>	-	+

To delete a column, check its or all the column entries, check the topmost , then click [-](#) .

8. Click  . The new data source is added to the *Data Sources* list.

Creating Apache Kafka Input Data Source

Allows Panopticon Streams to subscribe to Kafka topics on an external cluster.

Steps:

1. In the *New Data Source* page, select **Input > Kafka** in the *Connector* drop-down list.

Connector

Bootstrap Server

Schema Registry Host

Schema Registry Port

External Settings

Topic Fetch Topics Hide internal topics

From Beginning

Message Type

Decimal Separator

Record Path (eg. myroot.items.item)

Generate Columns Save Load

<input type="checkbox"/> Name	JsonPath	Type	Date Format	Filter	Enabled	+	-

Real-Time Settings

Time Id Column

Time Id Column Name

Reset Data on Reconnect

2. Enter the connection details:

Property	Description
Bootstrap Server	List of host/port pairs of Kafka servers used to bootstrap connections to a Kafka cluster. By default, the value is <code>localhost:9092,broker:29092</code> . However, this can be overridden by specifying another bootstrap server in the <i>External Settings</i> text box (as specified in step 3).
Schema Registry Host	Where the Schema Registry is located. This can be in a different location from the Kafka cluster.
Schema Registry Port	The port number of the schema registry which provides the serving layer for the metadata. Default is 8081 .

3. Enter the *External Settings* to support authentication (i.e., username and password). Note that if the bootstrap server is not secure, then there is no need to authenticate and you may leave this text box blank.

Below is an example of system settings for an SASL authentication:

```
bootstrap.servers=localhost:9093
sasl.jaas.config=\
  org.apache.kafka.common.security.plain.PlainLoginModule
required \
  username="dwchuser" \
  password="dwchpwd";
```

4. Click **Fetch Topics**. The first topic in the *Topic* drop-down list is selected and the schema is displayed.

By default, the **Hide Internal Topics** toggle button is enabled and the **Avro** message type is selected.

Name	Type	Enabled	Filter
Industry		<input checked="" type="checkbox"/>	
Count	Numeric	<input checked="" type="checkbox"/>	
_a1	Numeric	<input checked="" type="checkbox"/>	
_a2	Numeric	<input checked="" type="checkbox"/>	
Sum_Mcap_USD	Numeric	<input checked="" type="checkbox"/>	
First_Close_local	Numeric	<input checked="" type="checkbox"/>	
Last_Close_local	Numeric	<input checked="" type="checkbox"/>	
Min_One_Day_Change	Numeric	<input checked="" type="checkbox"/>	

Tap the slider to turn it off. The internal Kafka topics are also displayed in the drop-down list.

Name	Type	Enabled	Filter
Industry		<input checked="" type="checkbox"/>	
Count	Numeric	<input checked="" type="checkbox"/>	

Click the drop-down list to search and select the desired topic.

For non-Avro topics, select the *Message Type*: **Fix**, **XML**, **Text**, **JSON**, or **Protobuf**.

- If **Text** is selected, confirm the **Text Qualifier**, **Column Delimiter**, and if the first row of the message includes column headings.

Message Type	Text
Decimal Separator	Period {.}
Text Qualifier	None
Column Delimiter	Comma {,}
First Row Headings	<input checked="" type="checkbox"/>

Column Index controls the position of a column, Must be ≥ 0 .

Property	Description
Text Qualifier	Specifies if fields are enclosed by text qualifiers, and if present to ignore any column delimiters within these text qualifiers.
Column Delimiter	Specifies the column delimiter to be used when parsing the text file.
First Row Headings	Determines if the first row should specify the retrieved column headings, and not be used in data discovery.

- If **JSON** is selected, enter the *Record Path* which allows the identification of multiple records within the JSON document (e.g., **myroot.items.item**).

Message Type	Json
Decimal Separator	Period {.}
Record Path	(eg. myroot.items.item)

Property	Description
Record Path	The record path that will be queried by the connector's path (e.g., myroot.items.item).

- If **Protobuf** is selected, confirm the **Decimal Separator**, and enter the *Schema Name* and *Type Name*.

Then click **Browse** to select the **File Descriptor** (.desc file) in the *Open* dialog.

Message Type	Protobuf
Decimal Separator	Period {.}
File Descriptor	No file selected Browse
Schema Name	
Type Name	

Property	Description
Schema Name	The Protobuf schema.
Type Name	The message of Protobuf type that will be sent to Kafka.
File Descriptor	<p>The <code>FileDescriptorSet</code> which:</p> <ul style="list-style-type: none"> is an output of the protocol compiler. represents a set of <code>.proto</code> files, using the <code>--descriptor_set_out</code> option.

- Select the **From Beginning** check box to subscribe from the beginning to the latest messages.

If un-checked, you will only be subscribed to the latest messages.

- Select either the period (.) or comma (,) as the *Decimal Separator*.

NOTE

Prepend 'default:' for the elements falling under default namespace.

- Click **Generate Columns** to fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the **Save** button is enabled.
- For non-Avro message types, except **Protobuf**, click  to add columns to the Kafka connection that represent sections of the message. Then enter or select:

Property	Description
Name	The column name of the source schema.
Fix Tag/JsonPath/Text Column Index/XPath	The Fix Tag/JsonPath/Text Column Index/XPath of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The <u>format</u> when the data type is Time .
Filter	Defined parameters that can be used as filter. Only available for Avro, JSON, Text, and XML message types.
Enabled	Determines whether the message field should be processed.

NOTE

To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of upper case S. There can be no additional characters following them.

For example: yyyy-MM-dd HH:mm:ss.SSSSSS

- You can also opt to [load or save](#) a copy of the column definition.

10. Define the [Real-time Settings](#).



11. Click . The new data source is added in the *Data Sources* list.

Creating Kx kdb+ Input Data Source

The Kx kdb+ input data source allows connection to the Kx kdb+ databases on a polled basis.

Steps:

1. In the *New Data Source* page, select **Input > Kdb+** in the *Connector* drop-down list.

Connector: Kdb+ 

Host: localhost

Port: 5001

TLS Enabled:

User Id:

Password:

Timeout: 30

Retry count: 0

Table

Namespace: 

Table: 

[Generate Columns](#)

Column Parameterize Aggregate

Date Time: or  

Constrain By Date Time From To

Period Seconds 

Query

2. Enter the following properties:

Property	Description
Host	Kx kdb+ host address.
Port	Kx kdb+ host port. Default is 5001 .
TLS Enabled	Ensure to check if you have started q with TLS only.
User Id	The user Id that will be used to connect to Kx kdb+.

Password	The password that will be used to connect to Kx kdb+.
Timeout	The length of time to wait for the server response in seconds. Default is 30 .
Retry Count	For long running queries, a query timeout can be specified to prevent the server from locking up. Default is 0 .

3. When **Table** is selected, the section below is enabled:

The Namespace drop-down is an editable combo box.

The *Namespace* drop-down is an editable combo box.

You can either:

- click **Load** and select a namespace from the list of all root level namespaces. By default, the selected namespace will be root (backtick `).
- For nested namespaces, enter them in the *Namespace* box (e.g., panopticon.test) to get the tables that were created under these namespaces.

4. On the *Table* field, click **Load** to populate the drop-down list with tables and views. Select a table or view.

5. Click **Generate Columns**. The columns of the selected table or view populates the *Output Column* section.

6. Individual columns can be added by selecting the corresponding *Column* check box in the *Output Column* listing.

7. If the data returned is to be aggregated, then the **Aggregate** checkbox should be selected. For each selected column, the possible aggregation methods are listed including:

- Text Columns: Group By
- Date Columns: Count, Min, Max, Group By
- Numeric Columns: Sum, Count, Min, Max, Group By

In addition, the qSQL query is generated and displayed on the *Query* text box.

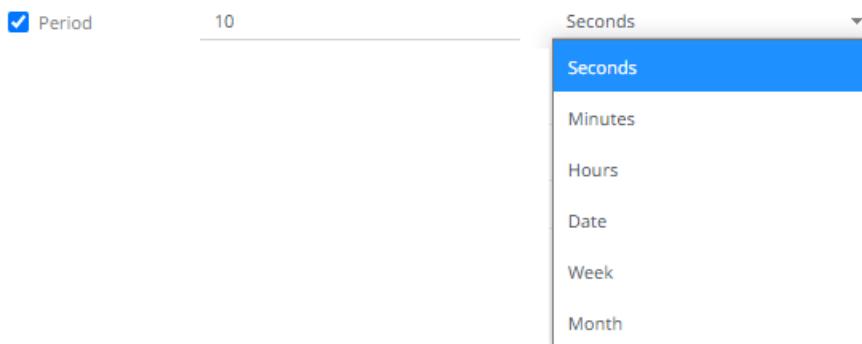
8. Select the **Parameterize** check box and match the parameter to the appropriate column. By default, they will be matched by name.

The appropriate qSQL query is updated on the *Query* text box. This shows the default parameter value for the preview, and at run time the qSQL will be updated to whatever the parameter value is.

9. If the data is to be filtered or aggregated on Date/Times, then a valid *Date Time* field needs to be selected from either a single Date/Time field, or a compound column created from a selected *Date* and a selected *Time* column.



10. Select the **Constrain by Date Time** check box and enter *From* and *To* Date/Time constraints.
11. In Kx kdb+, you can modify the query to regroup the aggregated data per time units (i.e., Seconds, Minutes, Hours, Date, Week, Month). Select the **Period** check box, enter the time duration and click  then select the time unit.



12. Click the **Query** radio button to enable the text box and modify the qSQL query language.

13. Select the *Flatten List Limit*.

This allows retrieval of the first 'n' items in the list and produce new columns in the output schema with a dot notation.

For example, if there are two nested fields (BidPrices and OfferPrices) and the flatten list limit selected is five, then the output schema will be:

BidPrices.1, BidPrices.2, BidPrices.3, BidPrices.4, BidPrices.5, OfferPrices.1, OfferPrices.2, OfferPrices.3, OfferPrices.4, OfferPrices.5

If there are less than five items in the list, then the values will be null.

NOTE

Currently, this feature works for the **Service** subscription type. Also, it only flattens numeric columns.

14. Select **Pass to function** check box to activate a connection to a server using a proxy. Enter the value.
15. You may also define a [Deferred Sync Query](#).

16. Click . The new data source is added in the *Data Sources* list.

Kx kdb+ - Deferred Sync Query

The Deferred Sync Query feature allows the Kx kdb+ connector to support synchronous and asynchronous reads. The advantage of using this option is that there is no queue on the Kx kdb+ server side, queries are farmed out to slaves and returned to asynchronous instead.

Deferred Sync Query (use {Query} parameter here as a place holder for the target query)

```
{@[neg .z.w;@[value;x;"failed to run query"];;"failed to post back"]}{["{Query}"]}
```

Selecting the *Deferred Sync Query* check box would enable the query box:

Deferred Sync Query (use {Query} parameter here as a place holder for the target query)

```
{@[neg .z.w;@[value;x;"failed to run query"];;"failed to post back"]}{["{Query}"]}
```

The {Query} parameter is used as a place holder for the target query that is defined in the *Query* builder.

Creating Kx kdb+Tick Input Data Source

The Kx kdb+tick input data source allows connection to a Kx kdb+ ticker plant on a real-time streaming basis.

Specifically, it allows Panopticon Streams to subscribe to Kx kdb+tick through the definition of *Service*, *Table*, *Symbol*, or directly through *Functional Subscription*.

Steps:

1. In the *New Data Source* page, select **Input > KDB+ Tick** in the *Connector* drop-down list.

Connector	Kdb+ Tick
Host	localhost
Port	5010
TLS Enabled	<input type="checkbox"/>
User Id	
Password	
Subscription Type	<input checked="" type="radio"/> Service <input type="radio"/> Functional Subscription
Subscription Name	.u.sub
Table	
Symbol	

Multiple symbols should be separated by comma.

Fetch Schema

Constrain subscription to matching symbols [Id Column]

Initialize with historic data

Host	localhost
Port	5010
TLS Enabled	<input type="checkbox"/>
User Id	
Password	
Timeout	30
Query	

Deferred Sync Query (use {Query} parameter here as a place holder for the target query)

`{@[neg .z.w;@[value;x;'"failed to run query"';'"failed to post back"]]}["{Query}"]`

Flatten List Limit

Real-Time Settings

Time Id Column	[No Time Id]
Time Id Column Name	
Reset Data on Reconnect	<input type="checkbox"/>

2. Enter the following properties:

Property	Description
Host	Kx kdb+tick host address.
Port	Kx kdb+tick host port. Default is 5010 .
TLS Enabled	Ensure to check if you have started q with TLS only.
User Id	The user Id that will be used to connect to Kx kdb+tick.
Password	The password that will be used to connect to Kx kdb+tick.

NOTE

These properties can be parameterized.

3. Select either Subscription Type:

- Service

Enter the following properties:

- Subscription Name (e.g., **.u.sub**)

NOTE

Instead of entering the table and symbol to subscribe against in the Table and Symbol text boxes, you can specify the full subscription syntax in the Subscription Name text box. For example:

`.u.sub[table;`symbol]`

To subscribe to the trade table and AAPL, AIG, and DOW symbols, enter this in the Subscription Name text box:

`.u.sub[`trade;`AAPL`AIG`DOW]`

- Table to subscribe against (e.g., **trade**)

NOTE

- You may use just a back tick for the table name, intending to subscribe to all available tables.
- When a table name is not entered in the Table text box, then the Symbol text box is disabled meaning it will not be used while doing subscription.

- ◆ Symbol to subscribe against (e.g., **AAPL**)

NOTE

Multiple symbols should be separated by a comma.

- Functional Subscription

Enter the functional subscription that needs to be issued (e.g., `.u.sub["trade"]`)

4. Click **Fetch Schema** to retrieve the schema of the configured subscription.

This populates the *Id Column* with the set of columns from the schema of type sym and the text array such as Character/Boolean/GUID, etc.

5. Check *Constrain subscription to matching symbols* to select the column which contains specific symbols. Otherwise, the filtering against these symbols will not take place.

NOTE

The *Constrain subscription to matching symbols* only lists sym fields. Therefore, if you select a non sym type in the *Id Column*, it is not recommended to select the default value [*Id Column*] in the *Constrain subscription to matching symbols* drop-down list.

6. Activate or deactivate *Initialize with historic data*. If unchecked, the data source will only be populated with streaming updates that are subscribed against. If checked, the data source is first initialized against a store of data, after which subscribed streaming updates are then applied.

7. Enter the following information:

- Host
- Port
- User Id
- Password
- Timeout
- Query

These entries can be parameterized.

8. Select *Deferred Sync Query* check box to allow the Kxkdb+tick data source to support synchronous and asynchronous reads. The advantage of using this option is that there is no queue on the Kx kdb+tick server side, queries are farmed out to slaves and returned to asynchronous instead.

The {Query} parameter is used as a place holder for the target query that is defined in the Query builder.

9. Select the *Flatten List Limit*.

This allows retrieval of the first 'n' items in the list and produce new columns in the output schema with a dot notation.

For example, if there are two nested fields (BidPrices and OfferPrices) and the flatten list limit selected is five, then the output schema will be:

`BidPrices.1, BidPrices.2, BidPrices.3, BidPrices.4, BidPrices.5, OfferPrices.1, OfferPrices.2, OfferPrices.3, OfferPrices.4, OfferPrices.5`

If there are less than five items in the list, then the values will be null.

NOTE

Currently, this feature works for the Service subscription type. Also, it only flattens numeric columns.

10. Define the [Real-time Settings](#).



11. Click **Save**. The new data source is added in the *Data Sources* list.

Creating ksqlDB Input Data Source

The ksqlDB connector allows executing ksqlDB pull queries and terminating push queries.

NOTE

Pull queries fetch the current state of a materialized view which is incrementally updated as new events arrive.

Steps:

1. In the *New Data Source* page, select **Input > ksqlDB** in the *Connector* drop-down list.

← [ksqlDBInput](#)



Connector	<input type="text" value="KsqlDB"/>			
Server Url	<input type="text" value="http://localhost:8088"/>			
Username	<input type="text"/>			
Password	<input type="text"/>			
<input type="checkbox"/> Collection	<input type="text" value="Stream"/>			
Query	<input type="text"/>			
From Beginning	<input type="checkbox"/>			
Timeout	<input type="text" value="5"/> seconds			
Decimal Separator	<input type="text" value="Period (.)"/>			
Generate Columns	Save Load			
<input type="checkbox"/> Name	Type	Date Format	Enabled	+ -

2. Enter the following properties:

Property	Description
Server URL	ksqlDB host address.

Username	User Id that will be used to connect to ksqlDB.
Password	Password that will be used to connect to ksqlDB.

3. Select the **Collection** check box to enable and select either:

- [Stream](#)

Immutable and append-only collections which are useful for representing a series of historical facts. Adding multiple events with the same key allows these events to be appended to the end of the stream.

- [Table](#)

Mutable collections. Adding multiple events with the same key allows the table to only keep the value for the last key. This collection is helpful in modeling change over time and often used to represent aggregations.

4. Click **Fetch** to populate the drop-down list. Select the collection.

5. Enter an SQL-like query language into the *Query* box.

6. Select the *From Beginning* check box to subscribe from the beginning to the latest messages.

From Beginning

If un-checked, you will only be subscribed to the latest messages.

7. Enter the *Timeout*. Default is 5 (in seconds).

8. Select either the dot (.) or comma (,) as the *Decimal Separator*.

9. Click **Generate Columns** to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the **Save** button is enabled.

10. You can also opt to [load or save](#) a copy of the column definition.

11. Click . A new column entry displays. Enter or select the following properties:

Property	Description
Name	The column name of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Enabled	Determines whether the message should be processed.

To delete a column, check its or all the column entries, check the topmost , then click .

12. Click **Save**. The new data source is added in the *Data Sources* list.

Creating ksqlDB - Streaming Input Data Source

The ksqlDB - Streaming connector allows executing ksqlDB push queries.

Steps:

1. In the *New Data Source* page, select **Input > ksqlDB - Streaming** in the *Connector* drop-down list.

← [ksqlDBStreamingInput](#) Save

Connector	KsqlDB - Streaming				
Server Url	http://localhost:8088				
Username					
Password					
<input type="checkbox"/> Collection	Stream				
Query					
<input type="checkbox"/> From Beginning					
Timeout	5 seconds				
Decimal Separator	Period {.}				
Generate Columns Save Load					
<input type="checkbox"/> Name	Type	Date Format	Enabled	+	-
<input type="checkbox"/> TradeTime	Time		<input checked="" type="checkbox"/>		
Real-Time Settings					
Time Id Column	[No Time Id]				
Time Id Column Name					
Reset Data on Reconnect	<input type="checkbox"/>				

2. Enter the following properties:

Property	Description
Server URL	ksqlDB - Streaming host address.
Username	User Id that will be used to connect to ksqlDB - Streaming.
Password	Password that will be used to connect to ksqlDB - Streaming.

3. Select the **Collection** check box to enable and select either:

- [Stream](#)

Immutable and append-only collections which are useful for representing a series of historical facts. Adding multiple events with the same key allows these events to be appended to the end of the stream.

- [Table](#)

Mutable collections. Adding multiple events with the same key allows the table to only keep the value for the last key. This collection is helpful in modeling change over time and often used to represent aggregations.

4. Click **Fetch** to populate the drop-down list. Select the collection.
5. Enter an SQL-like query language into the *Query* box.
6. Select the *From Beginning* check box to subscribe from the beginning to the latest messages.

From Beginning

If un-checked, you will only be subscribed to the latest messages.

7. Enter the *Timeout*. Default is 5 (in seconds).
8. Select either the dot (.) or comma (,) as the *Decimal Separator*.
9. Click **Generate Columns** to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the **Save** button is enabled.
10. You can also opt to [load or save](#) a copy of the column definition.

11. Click . A new column entry displays. Enter or select the following properties:

Property	Description
Name	The column name of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Enabled	Determines whether the message should be processed.

To delete a column, check its or all the column entries, check the topmost , then click .

12. Define the [Real-time Settings](#).

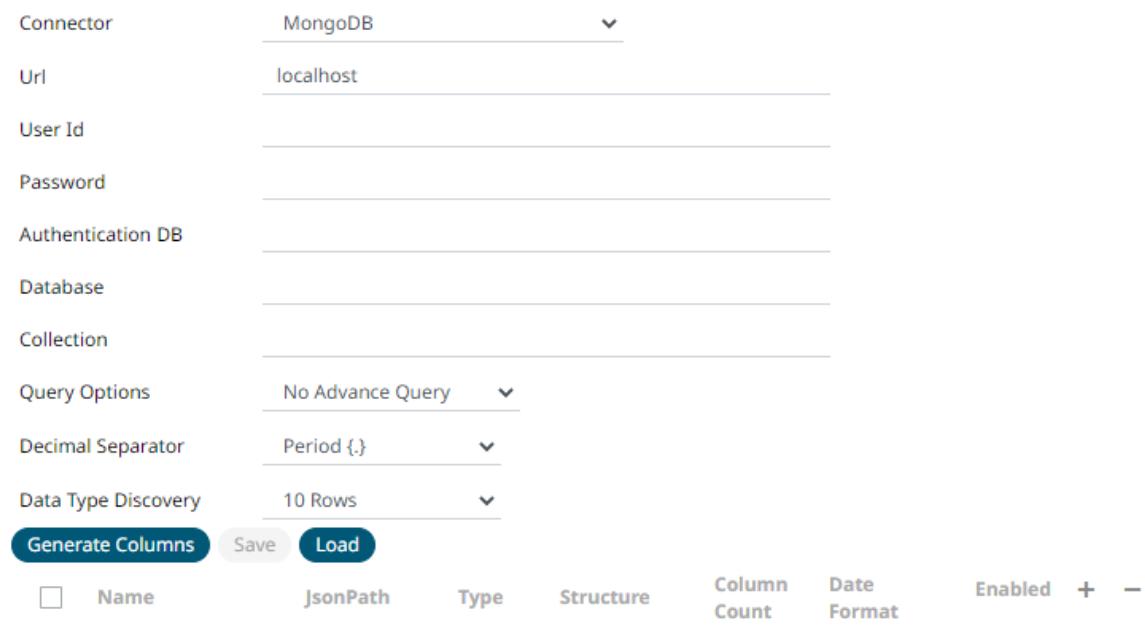
13. Click **Save**. The new data source is added in the *Data Sources* list.

Creating MongoDB Input Data Source

The MongoDB connector is an interface used to import MongoDB's schema-less BSON documents into a table schema that Panopticon Streams can interpret and analyze. It uses many BSON structure types and MongoDB query features.

Steps:

1. In the *New Data Source* page, select **Input > MongoDB** in the *Connector* drop-down list.



Connector: MongoDB

Url: localhost

User Id: (empty)

Password: (empty)

Authentication DB: (empty)

Database: (empty)

Collection: (empty)

Query Options: No Advance Query

Decimal Separator: Period (.)

Data Type Discovery: 10 Rows

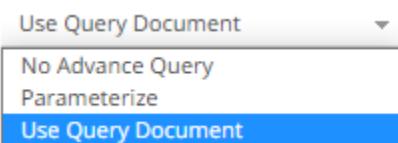
Buttons: Generate Columns, Save (highlighted), Load

Name	JsonPath	Type	Structure	Column Count	Date Format	Enabled
<input type="checkbox"/>						+ -

2. Enter the following properties:

Property	Description
URL	Enter either: <ul style="list-style-type: none"> • localhost if the database resides on the same computer, or • enter the IP address and port of the computer where MongoDB is installed (e.g., 192.168.1.1:27017). If no port is specified, the default is 27017.
User Id	The user Id that will be used to connect to MongoDB.
Password	The password that will be used to connect to MongoDB.
Authentication DB	The database where the user is created.
Database	The database that will be used.
Collection	The collection that will be used.

3. To make the Query Document feature of MongoDB available in Panopticon, select **Use Query Document**.



The *Plugin Settings* pane updates to display the settings of this query option.

Query Options	Use Query Document
Method	Find
JSON Query	<input type="text"/>
Sort	<input type="text"/> (eg: {"Column1":1})
Projection	<input type="text"/> (eg. {"Column1": 1, "Column2": 0})

For the *Method* option, select any of the following values:

- **Find** (Default)

Allows you to fetch a document from a MongoDB collection.

Two more configurable settings are available:

- ◆ **Sort**

Provide a JSON object that defines the sort criteria, then set the order to either **1** for ascending or **-1** for descending (e.g., {"**address.building**":1}).

- ◆ **Projection**

Provide a JSON object to include or exclude from the result of the **Find** query.

For example, if a document has 9 documents and you only need to display 5, you can either:

- select 5 JSON objects, then set the limit value to **1** to display

{"cuisine":1, "grades":1, "restaurant_id":1, "name":1, "borough":1}

- select 4 JSON objects then set the limit value to **0** to hide

{"address.zipcode":0, "address.coord":0, "address.street":0, "address.building":0}

NOTE

_id field is always displayed while executing the **Find** method. You can opt to set this field to **0** to hide it (e.g., {"**_id**":0}).

- **Aggregate**

Allows you to specify aggregation pipeline using multiple stages.

Python format of the pipeline query can be used, see [Getting Started with Aggregation Pipelines in Python](#) for more information. In addition, JSON style syntax with query contained in a JSON object can also be used.

See [Query Documents](#) for more information on the Query Documents feature on MongoDB.

Query Options [Use Query Document](#)

Method [Aggregate](#)

JSON Query

```
{"borough": "Bronx"}
```

4. Instead of using **Use Query Document**, select the **Parameterize** query option.

Query Options [Parameterize](#)

Parameter [Fetch Parameters](#)

Filter By

Click **Fetch Parameters** to populate the *Parameter* drop-down and select a value. Then select what column to filter on in the *Filter By* drop-down.

5. Select either the period (.) or comma (,) as the *Decimal Separator*.

NOTE

Prepend 'default:' for the elements falling under default namespace.

6. Select the *Data Type Discovery*. This property specifies how many rows to fetch from the input data source, when auto generating the schema after clicking [Generate Columns](#).

Data Type Discovery

[Generate Columns](#)

10 Rows

1 Row

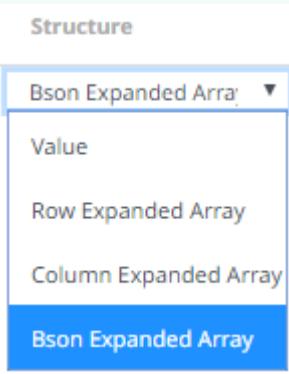
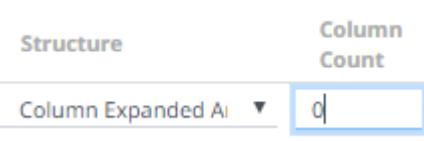
10 Rows

50 Rows

7. You can also opt to [load or save](#) a copy of the column definition.

8. You can also opt to click . A new row displays in the JSON list box. Enter the necessary information for each column.

Property	Description
Name	The column name of the source schema. NOTE: It is recommended to name the column the same as its JSON path for clarity and uniformity.
JsonPath	The JsonPath of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Structure	Used for more advanced features and are covered in the Row-Wise Array Expansion , Column-Wise Array Expansion , and Bson-Wise Array Expansion

	<p>sections. Value is the default structure and will always display data regardless of actual structure.</p> 
Column Count	<p>Enabled when Column-Expanded Array structure is selected.</p>  <p>Enter the number of columns for the plugin to generate as columns for that array.</p>
Date Format	<p>The format when the data type is Time.</p> <p>NOTE: To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of upper case S. There can be no additional characters following them.</p> <p>For example: yyyy-MM-dd HH:mm:ss.SSSSSS</p>
Enabled	Determines whether the message field should be processed.

To delete a column, check its or all the column entries, check the topmost , then click  .

9. Click  . The new data source is added in the *Data Sources* list.

Row-Wise Array Expansion

MongoDB's BSON document structure can store array data types. In order to interpret that data, the user has to decide how they want those multi-value fields to be displayed.

Row-wise array expansion takes an array of values and expands them in a single column creating a new row for each value in the array. If there are multiple row-expanded arrays in the same document, then the number of rows generated is equal to the largest array size. Additionally, an *Automatic x-axis* column is automatically generated for use as an x-axis value for visualizations using array data.

To use the row-wise array expansion feature, select **Row-Expanded Array** from the *Structure* drop-down box.

This feature will only work for an array data type. If the actual data type in MongoDB is not array or the array is empty, the column will not populate.

Column-Wise Array Expansion

MongoDB's BSON document structure can store array data types. In order to interpret that data, the user has to decide how they want those multi-value fields to be displayed.

Column-wise array expansion takes an array of values and expands them into multiple table columns creating a number of columns equal to an array specific number set by the user. If there are multiple column-expanded arrays in the same document, the combined number of new columns is appended to the end of the table with their respective array indices and the original columns are removed.

To use the column-wise expansion feature, select **Column-Expanded Array** in the *Structure* drop-down box.

The corresponding *Column Count* text box will be enabled and the user can enter the number of columns for the plugin to generate as columns for that array.

Bson-Wise Array Expansion

MongoDB's BSON document structure can store array data types. In order to interpret that data, the user has to decide how they want those multi-value fields to be displayed.

Bson-wise array expansion allows parsing of all the fields of a nested hierarchy in a sub document of a JSON array. During data retrieval, the column value is converted to JSON, and nested columns are flattened based on a JSON parser logic.

To use the Bson-wise expansion feature, select **Bson-Expanded Array** in the *Structure* drop-down box.

Creating MQTT Input Data Source

The MQTT connector allows:

- Connection to MQTT's message bus on a real-time streaming basis.
- Panopticon to subscribe to FIX, JSON, Text, or XML based messages that are published on topics. The data format itself is arbitrary, and consequently, the connection includes the message definition.
- Encrypted/SSL/TLS connections using a CA certificate file, Client Certificate File, and Client Key File.

Steps:

1. In the *New Data Source* page, select **Input > MQTT** in the *Connector* drop-down list.

Connector	MQTT					
Broker URL	tcp://localhost:1883					
Topic						
User Id						
Password						
Load Type	<input type="button" value="Upload File"/> <input type="button" value="Link To File"/>					
CA Certificate	No file selected <input type="button" value="Browse"/>					
Topic Level Separator	/					
Message Type	Json					
Decimal Separator	Period {.}					
Record Path	(eg. myroot.items.item)					
<input type="button" value="Generate Columns"/> <input type="button" value="Save"/> <input type="button" value="Load"/>						
<input type="checkbox"/> Name	JsonPath	Type	Date Format	Enabled	<input type="button" value="+"/>	<input type="button" value="-"/>
Topic Columns						
<input type="checkbox"/> Name	Level	Enabled <input type="button" value="+"/> <input type="button" value="-"/>				
Real-Time Settings						
Time Id Column	[No Time Id]					
Time Id Column Name						
Reset Data on Reconnect	<input type="checkbox"/>					

2. Enter the following properties:

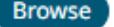
Property	Description
Broker URL	The location of the message broker. Default is tcp://localhost:1883 .
Topic	<p>The topic or the queue physical name. Example: level1/level2/level3/level4 etc.</p> <p>NOTES:</p> <p>You can also opt to use a wild card in the topic name specification.</p> <ul style="list-style-type: none"> The plus sign symbol (+) can be used as a wild card for any value at one specific level. Example: level1/level2/+/level4 The hash sign symbol (#) can be used as a wild card for any values across more than one level. Example: level1/#/level4
User Id	The user Id that will be used to connect to MQTT.

Password	The password that will be used to connect to MQTT.
----------	--

3. To allow encrypted connections, you can either:

- Upload a CA Certificate file by clicking **Upload File**  then **Browse**  to browse to the file source.

After selecting the file, it is displayed with the timestamp.

Load Type	Upload File  Link To File 
CA Certificate	letsencrypt.cer  Browse 
	as of 2022-08-17 11:13:14

To change the certificate, click  then **Browse**  to browse to a new version of the file.

- Link to a CA Certificate file by clicking **Link to File**  and entering a *File Path*.

Load Type	Upload File  Link To File 
CA Certificate	(File Type: .crt,.cer,.der,.pem)

4. In MQTT, a topic consists of one or more topic levels. Enter the *Topic Level Separator* to use. Default is / (forward slash).

5. Select the [Message Type](#).

6. Select either the period (.) or comma (,) as the *Decimal Separator*.

NOTE

Prepend 'default:' for the elements falling under default namespace.

7. Click **Generate Columns**  to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the Save button is enabled.

8. You can also opt to [load or save](#) a copy of the column definition.

9. You can opt to click  to add columns to the MQTT connection that represent sections of the message. Then enter or select:

Property	Description
Name	The column name of the source schema.
XPath/JsonPath/Fix Tag/Column Index	The XPath/JsonPath/Fix Tag/Column Index of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time

Date Format	The format when the data type is Time . NOTE: To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of upper case S. There can be no additional characters following them. For example: yyyy-MM-dd HH:mm:ss.SSSSSS
Filter	Defined parameters that can be used as filter. Only available for JSON, Text, and XML message types.
Enabled	Determines whether the message field should be processed.

To delete a column, check its or all the column entries, check the topmost , then click .

10. Text for topic levels can be consumed as additional columns into the data table.

The *Topic Columns* section shows and allows defining data table columns and mapping them to topic hierarchy levels (index based from left, 0 based).

Like columns from message data, manually add them by clicking . A new entry displays.

Topic Columns			
<input type="checkbox"/>	Name	Level	Enabled  
<input type="checkbox"/>	Level_1	0	<input checked="" type="checkbox"/>

Name can be any unique topic level within the topic name. The *Level* is the hierarchy level of the topic column.

Select the *Enabled* check box to enable a topic column.

To delete a topic column, check its or all the topic column entries, check the topmost , then click .

11. Define the [Real-time Settings](#).

12. Click . The new data source is added in the *Data Sources* list.

Creating MQTT Publisher Input Data Source

The MQTT Publisher connector allows writing message to a MQTT topic.

Steps:

1. Enter the following properties:

Property	Description
Broker URL	The location of the message broker. Default is <code>tcp://localhost:1883</code>
Topic	The topic or the queue physical name. Example:

	<p>level1/level2/level3/level4 etc.</p> <p>NOTES:</p> <p>You can also opt to use a wild card in the topic name specification.</p> <ul style="list-style-type: none"> • The plus sign symbol (+) can be used as a wild card for any value at one specific level. Example: level1/level2/+/level4 • The hash sign symbol (#) can be used as a wild card for any values across more than one level. Example: level1/#/level4
User Id	The user Id that will be used to connect to MQTT Publisher.
Password	The password that will be used to connect to MQTT Publisher.

2. To allow encrypted connections, select the *CA Certificate*, *Client Certificate*, and *Client Key* by clicking **Browse**  to browse to the file sources.
3. For *Payload*, enter the data that you want to send as message. The format can be anything supported by MQTT broker, typically JSON.

NOTE

This property can also be parameterized.

Creating MS Excel Input Data Source

This is the most commonly used data connector when prototyping and is used for retrieving data from MS Excel workbooks or spreadsheets, where for each selected sheet, the first row contains the field/column names, and subsequent rows contain the data.

The MS Excel connector supports reading data files stored in either the legacy XLS, and the newer XLSX format. The XLSX format is read on a row-by-row basis, allowing for better performance and less memory consumption compared to the XLS format.

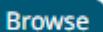
NOTE

In production use, it is not advised to use a single Excel file as multiple Panopticon data sources. This is because, when using the same Excel file with the data on several sheets, conflicts may occur in reading the file.

Steps:

1. On the *New Data Source* page, select **MS Excel** in the *Connector* drop-down list.

Connector	MS Excel																														
Excel File Path	No file selected																														
Sheet	<input type="text"/>																														
Headers On First Row	Auto																														
Columns	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Date Format</th> <th></th> </tr> </thead> <tbody> <tr><td>Item</td><td>Text</td><td></td><td><input checked="" type="checkbox"/></td></tr> <tr><td>isodatetime</td><td>Time</td><td></td><td><input checked="" type="checkbox"/></td></tr> <tr><td>ask_price</td><td>Numeric</td><td></td><td><input checked="" type="checkbox"/></td></tr> <tr><td>ask_volume</td><td>Numeric</td><td></td><td><input checked="" type="checkbox"/></td></tr> <tr><td>bid_price</td><td>Numeric</td><td></td><td><input checked="" type="checkbox"/></td></tr> <tr><td>bid_volume</td><td>Numeric</td><td></td><td><input checked="" type="checkbox"/></td></tr> </tbody> </table>			Name	Type	Date Format		Item	Text		<input checked="" type="checkbox"/>	isodatetime	Time		<input checked="" type="checkbox"/>	ask_price	Numeric		<input checked="" type="checkbox"/>	ask_volume	Numeric		<input checked="" type="checkbox"/>	bid_price	Numeric		<input checked="" type="checkbox"/>	bid_volume	Numeric		<input checked="" type="checkbox"/>
Name	Type	Date Format																													
Item	Text		<input checked="" type="checkbox"/>																												
isodatetime	Time		<input checked="" type="checkbox"/>																												
ask_price	Numeric		<input checked="" type="checkbox"/>																												
ask_volume	Numeric		<input checked="" type="checkbox"/>																												
bid_price	Numeric		<input checked="" type="checkbox"/>																												
bid_volume	Numeric		<input checked="" type="checkbox"/>																												

2. Upload a data source snapshot by clicking  and browse to the file source.

When a file is selected, the MS Excel connector will automatically select the first available sheet, set the first row as headers, and populate available columns.

Connector	MS Excel																														
Excel File Path	bidoffertrade.xlsx																														
Sheet	Price																														
Headers On First Row	Auto																														
Columns	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Date Format</th> <th></th> </tr> </thead> <tbody> <tr><td>Item</td><td>Text</td><td></td><td><input checked="" type="checkbox"/></td></tr> <tr><td>isodatetime</td><td>Time</td><td></td><td><input checked="" type="checkbox"/></td></tr> <tr><td>ask_price</td><td>Numeric</td><td></td><td><input checked="" type="checkbox"/></td></tr> <tr><td>ask_volume</td><td>Numeric</td><td></td><td><input checked="" type="checkbox"/></td></tr> <tr><td>bid_price</td><td>Numeric</td><td></td><td><input checked="" type="checkbox"/></td></tr> <tr><td>bid_volume</td><td>Numeric</td><td></td><td><input checked="" type="checkbox"/></td></tr> </tbody> </table>			Name	Type	Date Format		Item	Text		<input checked="" type="checkbox"/>	isodatetime	Time		<input checked="" type="checkbox"/>	ask_price	Numeric		<input checked="" type="checkbox"/>	ask_volume	Numeric		<input checked="" type="checkbox"/>	bid_price	Numeric		<input checked="" type="checkbox"/>	bid_volume	Numeric		<input checked="" type="checkbox"/>
Name	Type	Date Format																													
Item	Text		<input checked="" type="checkbox"/>																												
isodatetime	Time		<input checked="" type="checkbox"/>																												
ask_price	Numeric		<input checked="" type="checkbox"/>																												
ask_volume	Numeric		<input checked="" type="checkbox"/>																												
bid_price	Numeric		<input checked="" type="checkbox"/>																												
bid_volume	Numeric		<input checked="" type="checkbox"/>																												

By default, all the generated columns are enabled. You can uncheck the **Select All** box, then check the boxes of the columns that will be enabled.

3. Adjust *Sheet* selection, if required. Selecting a new sheet will re-populate the *Columns* list.

4. Adjust the **Headers On First Row** if needed. By default, the connector will pick up headers from the first row if all cells on the first row contain text data.

You can opt to select one of the following:

- Leave headers on first row as **Auto** if you want the connector to automatically pick up column names from sheet.
- Select **Yes** to force picking first row as headers.

- Select **No** to force not picking first row as headers. This will auto generate all column names.

Headers On First Row No

Columns

Name	Type	Date Format	Enabled
Column1	Text		<input checked="" type="checkbox"/>
Column2	Text		<input checked="" type="checkbox"/>
Column3	Text		<input checked="" type="checkbox"/>
Column4	Text		<input checked="" type="checkbox"/>
Column5	Text		<input checked="" type="checkbox"/>
Column6	Text		<input checked="" type="checkbox"/>

- Adjust column *Type* or *Date Format* to adjust data interpretation.

 **Save**

Click . The new data source is added in the *Data Sources* pane.

Creating OneTick Input Data Source

The OneTick connector allows connection to OneMarketData OneTick tick history databases on a polled basis. In general, it is used to retrieve conflated time series data sets. The connector supports either:

- Execution of a specified OTQ
- Execution of a specified parameterized OTQ
- Execution of a custom SQL Query

Steps:

- In the *New Data Source* page, select **Input > OneTick** in the *Connector* drop-down list.

Connector	OneTick
Context	REMOTE
Show local OTQs	<input checked="" type="checkbox"/>
Show remote OTQs	<input type="checkbox"/>
<input checked="" type="radio"/> OTQs	
Selected OTQ:	<input type="text"/> 
Symbol list	<input type="text"/>
From	<input type="text"/>
To	<input type="text"/>
<input type="radio"/> Query	
	

Separate DB Name

Show per-symbol errors as warnings

2. Enter the *Context* (for example, **REMOTE**).
3. You can either check:
 - **Show Local OTQs** box to display the local OTQs in the *Selected OTQ* drop-down list.
 - **Show Remote OTQs** box to display the remote OTQs in the *Selected OTQ* drop-down list.

An OTQ can be specified for execution, or a custom SQL query can be executed, through selection of the appropriate radio button:

- OTQs
- Query

4. Click **Load**  to populate the *Selected OTQ* drop-down list. Select an OTQ.

The list of input parameters that the OTQ expects is displayed. In addition, the basic SQL query is generated allowing the OTQ to be executed and the input parameters specific to the selected OTQ. The following are generic to all OTQs:

- Symbol List

NOTE

This property will accept comma-separated values either hardwired or parameterized.

- From
- To

These add additional filter criteria such as symbol, and time window onto the basic OTQ.

5. Check the **Separate DB Name** box to generate a separate field for the database name.
6. Check the **Show per symbol errors as warnings** box to proceed with warnings in the log if symbol errors are returned.

The result is a fully generated OneTick SQL query. This can be edited as required.

7. Click  . The new data source is added in the *Data Sources* list.

Creating OneTick CEP Input Data Source

The OneTick CEP connector allows connection to OneMarketData OneTick tick history databases on a streaming subscription basis. The connector supports either:

- Execution of a specified OTQ
- Execution of a specified parameterized OTQ
- To use the OneTick CEP connector, it requires a JAR file to be added and some configurations to be performed. Further details are provided in the [Panopticon Real Time Installation and Troubleshooting Guide](#).

Steps:

1. In the *New Data Source* page, select **Input > OneTick CEP** in the *Connector* drop-down list.

← OneTickCEPInput

 Save

Connector	OneTick CEP
Context	REMOTE
Show local OTQs	<input checked="" type="checkbox"/>
Show remote OTQs	<input type="checkbox"/>
OTQs	<input checked="" type="radio"/>
Selected OTQ:	<input type="text"/> 
Separate DB Name	<input type="checkbox"/>
Symbol List	<input type="text"/>
From	<input type="text"/>
To	<input type="text"/>
	

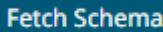
Real-Time Settings

Time Id Column	<input type="text"/> [No Time Id]
Time Id Column Name	<input type="text"/>
Reset Data on Reconnect	<input type="checkbox"/>

2. Enter the *Context* (for example, **REMOTE**).
3. You can either check:
 - **Show Local OTQs** box to display the local OTQs in the *Selected OTQs* drop-down list.
 - **Show Remote OTQs** box to display the remote OTQs in the *Selected OTQs* drop-down list.
4. Click  to populate the *Selected OTQ* drop-down list. Select an OTQ.

The *OTQ Parameters* section displays with the list of input parameters based on the selected OTQ.

OTQ Parameters	
Name	Value
filename	<input type="text"/>

5. Select/unselect the *Separate DB Name* box.
6. Click  to populate the *Id Column* list box.
7. From this list box select the field which will define a unique data record to subscribe against.

The following are generic to all OTQs

- Symbol List
- From

- To

These add additional filter criteria such as symbol, and time window onto the basic OTQ.

8. Define [Real-time Settings](#).



9. Click . The new data source is added in the *Data Sources* list.

Creating Python Input Data Source

The Python connector lets you load data through Python. This can be useful for example when connecting to unusual data sources for which there is a client library in Python, or when you want to apply custom data transforms to the data in Python.

A requirement for using the Python connector is that your Panopticon server has been configured to integrate with a running Python environment, as described in the [Real Time Installation and Reference Guide](#).

The Python connector has settings that may be pre-configured by the Panopticon server administrator, and thereby not exposed to the users of the connector. If the settings are not pre-configured, you will see the following settings:

When using Pyro4 integration:

Setting	Description
Host	The hostname or IP-number where Python is running.
Port	The port number where Pyro4 is listening.
HMAC Key	A secret password set in the integration configuration that must be matched.
Serialization Type	Can be either serpent or pickle .

When using FAST API integration (Linux servers only):

Setting	Description
Host	The hostname or IP-number where Python is running.
Port	The port number where FAST API is listening.

You will also see the following setting:

Setting	Description
Python Script (checkbox)	Whether Apache Arrow serialization should be applied to the data during transfer from Python to Panopticon. This will make the data transfer significantly faster.

NOTE

If your Panopticon server has not been configured for integration with a Python environment, you will not be able to use the Python connector, regardless of what settings you make in the connector.

Creating RabbitMQ Input Data Source

The RabbitMQ connector allows connection to RabbitMQ's message bus on a real-time streaming basis. Specifically, the connector allows Panopticon Streams to subscribe to XML, JSON, Text or FIX based messages that are published on particular topics.

Steps:

1. In the *New Data Source* page, select **Input > RabbitMQ** in the **Connector** drop-down list.

← RabbitMQInput

Connector: RabbitMQ

Broker: (empty)

User Id: (empty)

Password: (empty)

Exchange Type: topic

Exchange: (empty)

Durable

Auto Delete

Routing Key: (empty)

Explicit Queue

Properties: (empty)

Durable

Auto Delete

Message Type: Json

Decimal Separator: Period (.)

Record Path: (eg. myroot.items.item)

Generate Columns Save Load

<input type="checkbox"/> Name	JsonPath	Type	Date Format	Enabled	+	-
<input type="checkbox"/> TradeTin		Time		<input checked="" type="checkbox"/>		

Real-Time Settings

Time Id Column: [No Time Id]

Time Id Column Name: (empty)

Reset Data on Reconnect:

Save

2. Enter the connection details including:

Property	Description
Broker	The location of the message broker.

User Id	The user Id that will be used to connect to RabbitMQ.
Password	The password that will be used to connect to RabbitMQ.

3. Select any of the following *Exchange Types*:

Exchange Type	Description
Default	<p>A direct exchange with no name that is pre-declared by the broker. Selecting this exchange type disables the <i>Exchange</i> section (<i>Exchange</i> and <i>Routing Key</i> properties).</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Exchange Type <input type="text" value="default"/> <input type="button" value="▼"/></p> <p>Exchange <input type="text"/></p> <p><input type="checkbox"/> Durable</p> <p><input type="checkbox"/> Auto Delete</p> <p>Routing Key <input type="text"/></p> <p><input checked="" type="checkbox"/> Explicit Queue <input type="text"/></p> </div>
Fanout	Broadcasts all of the messages it receives to all of the queues it knows and the routing key is ignored (the <i>Routing Key</i> field is disabled).
Direct	Delivers messages to queues based on a message routing key. It is ideal for the unicast routing of messages, although it can be used for multicast routing as well.
Topic	A message sent with a particular routing key will be delivered to all of the queues that are bound with a matching binding key.
Headers	Exchanges routed based on arguments containing headers and optional values.

4. Depending on the selected *Exchange Type*, select or define the following:

Exchange Type Property	Description
Exchange	Name of the exchange.
Durable	Enable so the exchange can survive a broker restart.
Auto Delete	Enable so the exchange is deleted when the last queue is unbound from it.
Routing Key	The routing key used to deliver messages to queues.
Headers	<p>This field is only available when the message type is Header.</p> <p>Binding a queue to a Headers exchange is possible using more than one header for matching. Setting <i>x-match</i> to any, means just one matching value is sufficient. Setting it to all means that all values must match. Default is x-match=all.</p>

- Select the *Explicit Queue* check box and enter the custom queue name. Then enter or enable the following properties:

Queue Property	Description
Properties	The custom queue property.
Durable	Enable so the queue can survive a broker restart.
Auto Delete	Enable so the queue that had the least consumer will be deleted when that connection closes.

- Select the [*Message Type*](#).
- Select either the period (.) or comma (,) as the *Decimal Separator*.

NOTE

Prepend 'default:' for the elements falling under default namespace.

- Click **Generate Columns** to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the Save button is enabled.
- You can also opt to [load or save](#) a copy of the column definition.
- You can opt to click  to add columns to the RabbitMQ connection that represent sections of the message. Then enter or select:

Property	Description
Name	The column name of the source schema.
Fix Tag/Json Path/Text Column Index/Xpath	The Fix Tag/Json Path/Text Column Index/Xpath of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The <u>format</u> when the data type is Time . NOTE: To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of upper case S. There can be no additional characters following them. For example: yyyy-MM-dd HH:mm:ss.SSSSSS
Filter	Defined parameters that can be used as filter. Only available for JSON, Text, and XML message types.
Enabled	Determines whether the message field should be processed.

To delete a column, check its  or all the column entries, check the topmost  , then click .

- Define the [*Real-time Settings*](#).



12. Click **Save**. The new data source is added in the *Data Sources* list.

Creating Amazon S3 Input Data Source

The S3 connector allows for retrieval of the file from an S3 storage location. This connector allows JSON/XML/Text/Excel files to be read from the S3 storage. This connector will work with any S3 compliant storage providers.

Steps:

1. In the *New Data Source* page, select **Input > S3** in the *Connector* drop-down list.
2. Enter the following information:

Property	Description
URL	URL where the S3 bucket can be accessed. Default is https://s3.amazonaws.com .
Bucket	S3 bucket where the file resides.
Access Key	Access key to your S3 service account.
Secret Key	Secret key to your S3 service account. To test the connection, click Test Connection . If Connection Failed displays, ensure the <i>Bucket</i> , <i>Access Key</i> , and <i>Secret Key</i> values are correct. You can also hover on this message to view the connection error.
File Path	Path of the on the S3 bucket.

3. Select the [Data Type](#).
4. Select either the period (.) or comma (,) as the *Decimal Separator*.

NOTE

Prepend 'default:' for the elements falling under default namespace.

5. Click **Generate Columns** to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the **Save** button is enabled.
6. You can also opt to [load or save](#) a copy of the column definition.

7. You can also opt to click  to add columns to the S3 connection that represent sections of the message. Then enter or select:

Property	Description
Name	The column name of the source schema.
JsonPath/Column Index/XPath	The JsonPath/Column Index/XPath of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Enabled	Determines whether the message field should be processed.

NOTE

To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of upper case S. There can be no additional characters following them.

For example: yyyy-MM-dd HH:mm:ss.SSSSSS

To delete a column, check its or all the column entries, check the topmost , then click .

8. Click . The new data source is added in the *Data Sources* list.

Creating Rserve Input Data Source

The Rserve connector allows the retrieval of an output data frame from a running Rserve process.

For R connectivity, R must be first installed, together with the Rserve library. In addition, R must be open, and the Rserve library must be loaded and initialized.

Steps:

1. In the *New Data Source* page, select **Input > Rserve** in the *Connector* drop-down list.

Connector	Rserve	▼
Host	localhost	
Port	6311	
User Id		
Password		
R Script	<input type="checkbox"/> Enclose Parameters in Quotes <div style="border: 1px solid #ccc; padding: 5px; height: 300px; width: 100%;"></div>	
Timeout	10	seconds

2. Enter the following properties:

Property	Description
Host	Rserve host address.
Port	Rserve host port. Default is 6311 .
User Id	The user Id that will be used to connect to the Rserve service.
Password	The password that will be used to connect to the Rserve service.

3. Enter the required *R script* to execute on the active Rserve instance.
 4. The Timeout is set to **10** seconds by default to ensure that slow running R scripts do not impact other areas of the product. You can opt to enter a new value.
 5. Select whether the parameters should be automatically enclosed in quotes by selecting the *Enclose parameters in quotes* check box.

6. Click  Save. The new data source is added in the *Data Sources* list.

Creating Solace Input Data Source

The Solace connector allows connection to Solace's message bus on a real time streaming basis. Specifically, the connector allows Panopticon Streams to subscribe to messages that are published in particular topics in Solace and consequently, perform operational analytics.

Steps:

1. In the *New Data Source* page, select **Input > Solace** in the *Connector* drop-down list.

← SolaceInput Save

Connector	Solace
Host	
VPN Name	default
User Id	
Password	
Topic	
Message Type	Text
Decimal Separator	Period (.)
Text Qualifier	<none>
Column Delimiter	Comma (,)
First Row Headings	<input checked="" type="checkbox"/>

Column Index controls the position of a column, Must be ≥ 0 .

Generate Columns

<input type="checkbox"/> Name	Column Index	Type	Date Format	Filter	Enabled	+	-

Timestamp Name:

Date:

Time:

Real-Time Settings

Time Id Column: [No Time Id]

Time Id Column Name:

Reset Data on Reconnect:

2. Enter the connection details including:

Property	Description
Host	Solace host address.
VPN Name	Message VPN name. Default is default .

User Id	The user Id that will be used to connect to Solace.
Password	The password that will be used to connect to Solace.

3. Enter the *Topic* or the queue physical name.
4. Select the [Message Type](#). This will specify the format of the data within the message.

Aside from the **JSON**, **Text**, and **XML** message types, **Protobuf** and **SDTMap** are also supported in Solace.

If **Protobuf** is selected, confirm the **Decimal Separator**, and enter the **Schema Name** and **Type Name**.

Then click  to select the **File Descriptor** (.desc file) in the *Open* dialog.

Message Type	Protobuf
Decimal Separator	Period {.}
Schema Name	
Type Name	
File Descriptor	<input type="text" value="No file selected"/> 

Property	Description
Schema Name	The Protobuf schema.
Type Name	The message of Protobuf type that will be sent to Kafka.
File Descriptor	<p>The <code>FileDescriptorSet</code> which:</p> <ul style="list-style-type: none"> • is an output of the protocol compiler. • represents a set of <code>.proto</code> files, using the <code>--descriptor_set_out</code> option.

For **SDTMap**, confirm the **Decimal Separator**.

Message Type	SDTMap
Decimal Separator	Period {.}

5. Select either the period (.) or comma (,) as the *Decimal Separator*.

NOTE

Prepend 'default:' for the elements falling under default namespace.

6. Click  to the fetch the schema based on the connection details. This populates the list of columns with the data type found from inspecting the first 'n' rows of the input data source.
7. You can opt to click  to add columns to the Solace connection that represent sections of the message. Then enter or select:

Property	Description
Name	The column name of the source schema.
JsonPath/Column Index/XPath	The JsonPath/Column Index/XPath of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The <u>format</u> when the data type is Time . NOTE: To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of upper case S. There can be no additional characters following them. For example: yyyy-MM-dd HH:mm:ss.SSSSSS
Filter	Defined parameters that can be used as filter. Only available for JSON, Text, and XML message types.
Enabled	Determines whether the message field should be processed.

To delete a column, check its  or all the column entries, check the topmost , then click .

8. To create a new Timestamp field, enter a new *Timestamp Name* and then select the valid Date/Time from either a single *Date* or *Time* field, or a compound column created from *Date* and *Time* fields.
9. Define the [Real-time Settings](#).

10. Click . The new data source is added in the *Data Sources* list.

Creating Stream Simulator Input Data Source

The Stream Simulator connector is very similar to the Text connector with the addition of the time windowing of message queue connectors.

Creating the Stream Simulator input data source includes setting for how fast and how many messages are pushed through in each batch.

Steps:

1. In the *New Data Source* page, select **Input > Stream Simulator** in the *Connector* drop-down list.

Connector	Stream Simulator
Text File Source	File
Load Type	<input type="button" value="Upload File"/> <input type="button" value="Link To File"/>
Text File Path	No file selected <input type="button" value="Browse"/>
Skip First n Rows	0
Data Type Discovery	10 Rows
Decimal Separator	Period (.)
Text Qualifier	None
Column Delimiter	Comma (,)
First Row Headings	<input checked="" type="checkbox"/>

Column Index controls the position of a column, Must be ≥ 0 .

<input type="checkbox"/> Name	Column Index	Type	Date Format	Enabled
Simulation Type	<input checked="" type="radio"/> Record <input type="radio"/> Time			<input type="button" value="+"/> <input type="button" value="-"/>
Sort Order	Use file sort order			
Sorted By Column				
Playback Set Size				
Start Up Set Size				
Playback Interval (ms)	1000			
Loop	<input type="checkbox"/>			
Real-Time Settings				
Time Id Column	[No Time Id]			
Time Id Column Name				
Real-time Limit (ms)	1000			
Reset Data on Reconnect	<input type="checkbox"/>			

2. Select the Text [File Source](#).

The standard settings controlling how the text file is parsed, is listed.

These include:

Property	Description
Skip First N Rows	Specifies the number of rows that will be skipped.
Data Type Discovery	Specifies how many rows from the text file should be used when automatically determining the data types of the resulting columns.
Decimal Separator	Select either the period (.) or comma (,) as the decimal separator.
Text Qualifier	Specifies if fields are enclosed by text qualifiers, and if present, to ignore any column delimiters within these text qualifiers. Can be any of the following options: <ul style="list-style-type: none">• None

	<ul style="list-style-type: none"> • Single Quote • Double Quote
Column Delimiter	Specifies the column delimiter to be used when parsing the text file.
First Row Headings	Determines if the first row should specify the retrieved column headings, and not be used in data discovery.

3. Click **Generate Columns** to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the **Save** button is enabled.

4. You can also opt to [load or save](#) a copy of the column definition.

5. You can opt to click . A new column entry displays. Enter or select the following properties:

Property	Description
Name	The column name of the source schema.
Column Index	The column index controls the position of a column. Must be ≥ 0 .
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time.
Enabled	Determines whether the message should be processed.

To delete a column, check its  or all the column entries, check the topmost  , then click  .

6. Select the *Simulation Type*:

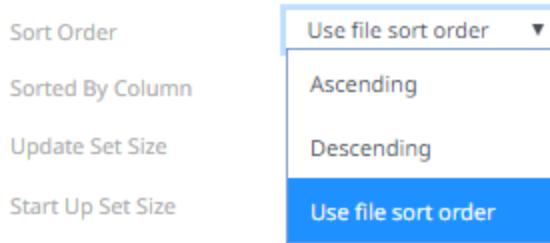
- Record

Sends the number of records for each interval of time. By default, records are sent in the same order of the source.

Simulation Type	<input checked="" type="radio"/> Record	<input type="radio"/> Time
Sort Order	Use file sort order 	
Sorted By Column		
Playback Set Size		
Start Up Set Size		
Playback Interval (ms)	1000	

This simulation type allows the specification of the following:

- ♦ Sort Order



When you select the **Use file sort order**, it will use the default sorting order of the file.

When you either select **Ascending** or **Descending** as the Sort Order, this enables the *Sorted by Column* drop down list.

Select the column that will be used for the sorting.



- ◆ Playback Set Size

The number of records set to be updated during simulate/playback.

- ◆ Start Up Set Size

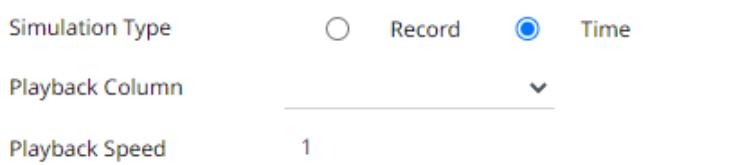
The number of records set to be published initially (on start-up).

- ◆ Playback Interval (ms)

The update interval period for the record-based playback. Default is **1000 (ms)**.

- Time

Simulates records as they occur in real-time.



This simulation type allows the specification of the following:

- ◆ Playback Column

The playback column which is a Date/Time type.

- ◆ Playback Speed

A multiplier which to either speed up or slow down the playback. Default is **1**.

- If $0 < \text{value} < 1$ slow down
- If $\text{value} = 1$ records will be published as they occur
- if $\text{value} > 1$ speed up

NOTE

For time-based simulation, if the Date/Time column have improper dates, it will fail and stop.

7. Select the **Loop** check box to enable looping through the file.
8. Define the [Real-time Settings](#).
9. Modify the *Real-time Limit* to vary the data throttling. This defaults to **1000** milliseconds.



10. Click **Save**. The new data source is added in the *Data Sources* list.

Creating StreamBase Input Data Source

The StreamBase 7.1 connector allows connection to the StreamBase CEP engine instance on a real-time streaming basis.

To use the StreamBase connector, Streambase 7.1 redistributable must be installed.

Refer to <http://www.streambase.com/products/streambasecep/download-streambase/> for more information in downloading StreamBase products.

Steps:

1. In the *New Data Source* page, select **Input > StreamBase 7.1** in the *Connector* drop-down list.

← StreamBaseInput

Connector: StreamBase 7.1

Primary URL: sb://localhost:10000

Secondary URL:

User Id:

Password:

Stream:

Predicate:

Show in Timezone:

Enclose parameters in quotes

Real-Time Settings

Time Id Column: [No Time Id]

Time Id Column Name:

Reset Data on Reconnect

2. Enter the following properties:

Property	Description
Primary URL	Primary URL of the StreamBase 7.1. Default is sb://localhost:100000.
Secondary URL	Secondary URL of the StreamBase 7.1. NOTE: More than two StreamBase server URLs can be specified by comma separation.
User Id	User Id that will be used to connect to StreamBase 7.1.
Password	Password that will be used to connect to StreamBase 7.1.

- Click **Fetch Streams** to return a list of updated streams. Selection of a stream returns a list of available Id columns for the stream.

This populates the *Id Column* with the set of columns from the schema of type `sym` and the text array such as Character/Boolean/GUID, etc. The selected *Id Column* can be used to select a key column to manage data updates and inserts.

NOTE: Every message definition needs a text column to be defined as the *Id column*. By default, only the latest data will be loaded into memory.

Furthermore, a streaming time series window can be generated by creating a compound key with the *Id Column*, plus a separately specified *Time Id* column. This *Time Id* column can be from the source dataset, or alternatively automatically generated.

If the *Time Id* column is selected, then a scrolling time window can be specified.

The screenshot shows a configuration interface with two fields. The first field is labeled 'Time Id Column' with a dropdown menu open, showing the option 'Automatic Time Id' selected. The second field is labeled 'Time Id Column Name' with the value 'Automatic_Timestamp_Column' entered.

For *Automatic Time Id*, define the *Time Id Column Name*.

As new data arrives from the subscription, new time slices will automatically be added, and old ones will be deleted.

If a new ID is received, a new row is added to the in-memory data set representing the StreamBase topic subscription. While if an existing ID is received, an existing row is updated.

- Enter the *Predicate* expression to force emission.
- The time zone of input parameters and output data is by default unchanged. Changing the time zone is supported through the *Show in Timezone* list box, based on the assumption that data is stored in UTC time and outputs are presented in the selected time zone.
- Select whether the parameters should be automatically enclosed in quotes, by selecting the **Enclose parameters in quotes** box.
- Select the **Reset Data on Reconnect** check box to flush out the stale data and reload data after reconnection.

- Click **Save**. The new data source is added in the *Data Sources* list.

Creating StreamBase LiveView Input Data Source

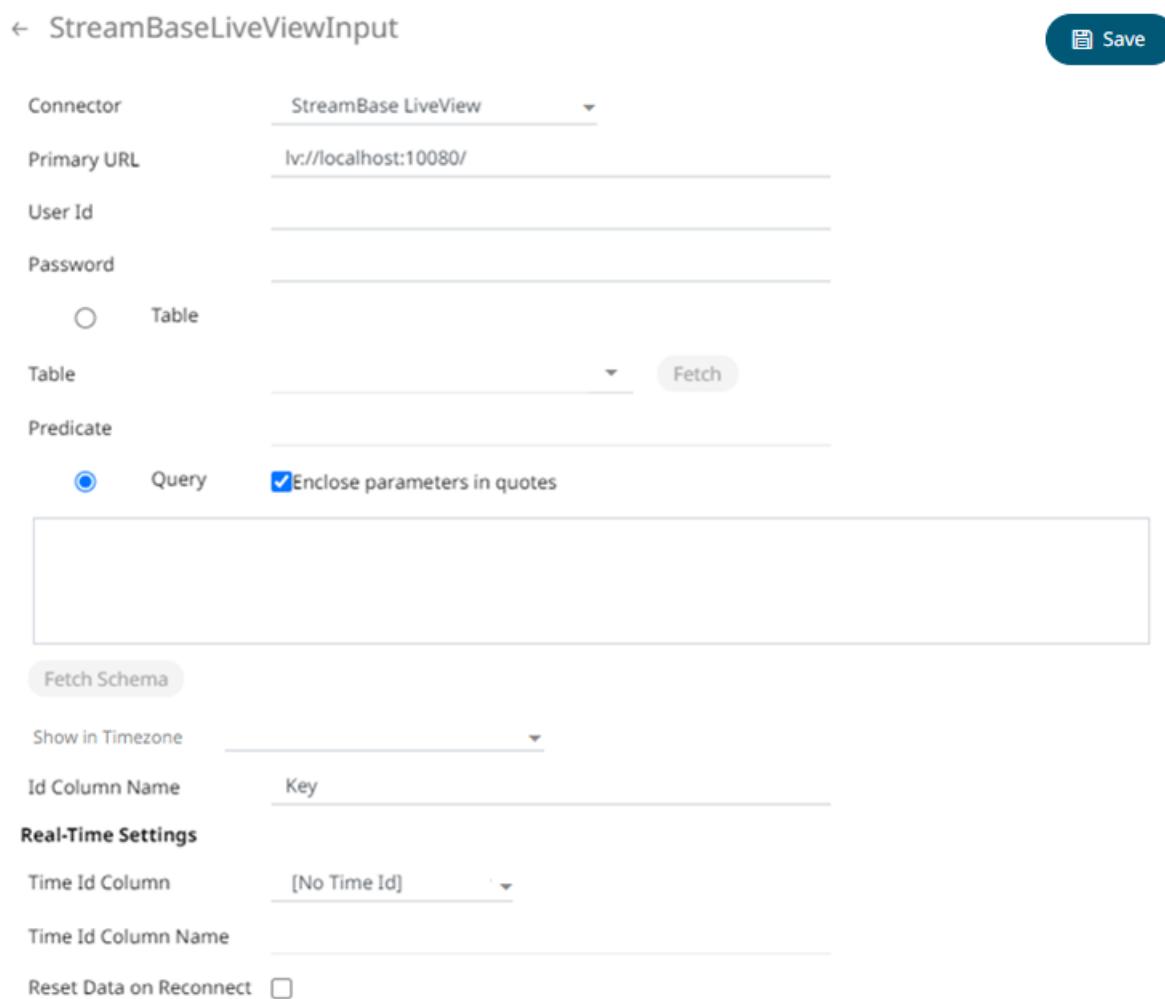
The StreamBase LiveView connector allows connection to the StreamBase LiveView instance on a real-time streaming basis.

NOTE

Supported Java versions must be earlier than 255. For example, if you are using JDK or JRE, it should be less than 1.8.0_255.

Steps:

1. In the *New Data Source* page, select **Input > StreamBase LiveView** in the *Connector* drop-down list.



← StreamBaseLiveViewInput

Connector StreamBase LiveView

Primary URL lv://localhost:10080/

User Id

Password

Table

Table

Predicate

Query Enclose parameters in quotes

Fetch Schema

Show in Timezone

Id Column Name Key

Real-Time Settings

Time Id Column [No Time Id]

Time Id Column Name

Reset Data on Reconnect

Save

2. Enter the following properties:

Property	Description
Primary URL	Primary URL of the StreamBase LiveView.
User Id	User Id that will be used to connect to StreamBase LiveView.
Password	Password that will be used to connect to StreamBase LiveView.

3. You can either:

- select the **Table** radio button then click **Fetch** to return a list of updated *Tables*.
Select the required table.
By default, the whole table will be subscribed against. To subscribe against a subset, enter a predicate.
The `IN` syntax is recommended for use of parameters to support multiple values. The square bracket notation should be used for the `IN` clause.
Example: `color IN [{color}]`

- select the **Query** radio button, enter a full query, then click **Fetch Schema**.

- The time zone of input parameters and output data is by default unchanged. Changing the time zone is supported through the *Show in Timezone* list box, based on the assumption that data is stored in UTC time and outputs are presented in the selected time zone.
- Enter the *ID Column Name*.

LiveView supplies a unique Id for each row. This Id field is by default given a title of **Key**.

Id Column Name

Key

Furthermore, a streaming time series window can be generated by creating a compound key with the *Id Column*, plus a separately specified *Time Id* column. This *Time Id* column can be from the source dataset, or alternatively automatically generated.

If the *Time Id* column is selected, then a scrolling time window can be specified.

Time Id Column	[Automatic Time Id] ▾
Time Id Column Name	Automatic_Timestamp_Column

For *Automatic Time Id*, define the *Time Id Column Name*.

As new data arrives from the subscription, new time slices will automatically be added, and old ones will be deleted.

If a new Id is received, a new row is added to the in-memory data set representing the StreamBase LiveView topic subscription. While if an existing ID is received, an existing row is updated.

- Select the **Reset Data on Reconnect** check box to flush out the stale data and reload data after reconnection.

- Click **Save**. The new data source is added in the *Data Sources* list.

Creating Text Input Data Source

The Text connector allows the retrieval and processing of delimited Text files (such as CSV, TSV, and so on), either from a disk or from a defined URL.

Steps:

- In the *New Data Source* page, select **Input > Text** in the *Connector* drop-down list.

Connector	Text
Text File Source	File
Load Type	<input type="button" value="Upload File"/> <input type="button" value="Link To File"/>
Text File Path	No file selected <input type="button" value="Browse"/>
Skip First n Rows	0
Data Type Discovery	10 Rows
Decimal Separator	Period {.}
File Encoding	
Text Qualifier	<none>
Column Delimiter	Comma {,}
First Row Headings	<input checked="" type="checkbox"/>

Column Index controls the position of a column, Must be ≥ 0 .

<input type="checkbox"/> Name	Column Index	Type	Date Format	<input checked="" type="checkbox"/> Enabled	<input type="button" value="+"/> <input type="button" value="-"/>
-------------------------------	--------------	------	-------------	---	---

2. Select the Text [File Source](#).

NOTE

Load Type

Text File Path

The **Upload File** button, when clicked, allows the user to choose files from their own computer. To choose files that resides on the Panopticon Server machine, use the **Link to File** option and fill in the *Text File Path*.

The standard settings controlling how the text file is parsed are listed.

These include:

Property	Description
Skip First N Rows	Specifies the number of rows that will be skipped.
Data Type Discovery	Specifies how many rows from the text file should be used when automatically determining the data types of the resulting columns.
Decimal Separator	Select either the dot (.) or comma (,) as the decimal separator.
File Encoding	Set the character encoding to use in text data.

	<ul style="list-style-type: none"> • UTF-8 • UTF-16 • UTF-32 • US-ASCII • Windows-1252
Text Qualifier	<p>Specifies if fields are enclosed by text qualifiers, and if present, to ignore any column delimiters within these text qualifiers.</p> <p>Can be any of the following options:</p> <ul style="list-style-type: none"> • None • Single Quote • Double Quote
Column Delimiter	Specifies the column delimiter to be used when parsing the text file.
First Row Headings	Determines if the first row should specify the retrieved column headings, and not be used in data discovery.

Generate Columns

3. Click  to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the Save button is enabled.
- By default, all the generated columns are enabled. You can uncheck the **Select All** box, then check the boxes of the columns that will be enabled.
4. You can also opt to [load or save](#) a copy of the column definition.
5. You can opt to click . A new column entry displays. Enter or select the following properties:

Property	Description
Name	The column name of the source schema.
Column Index	The column index controls the position of a column. Must be ≥ 0 .
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Enabled	Determines whether the message should be processed.

To delete a column, check its or all the column entries, check the topmost , then click .

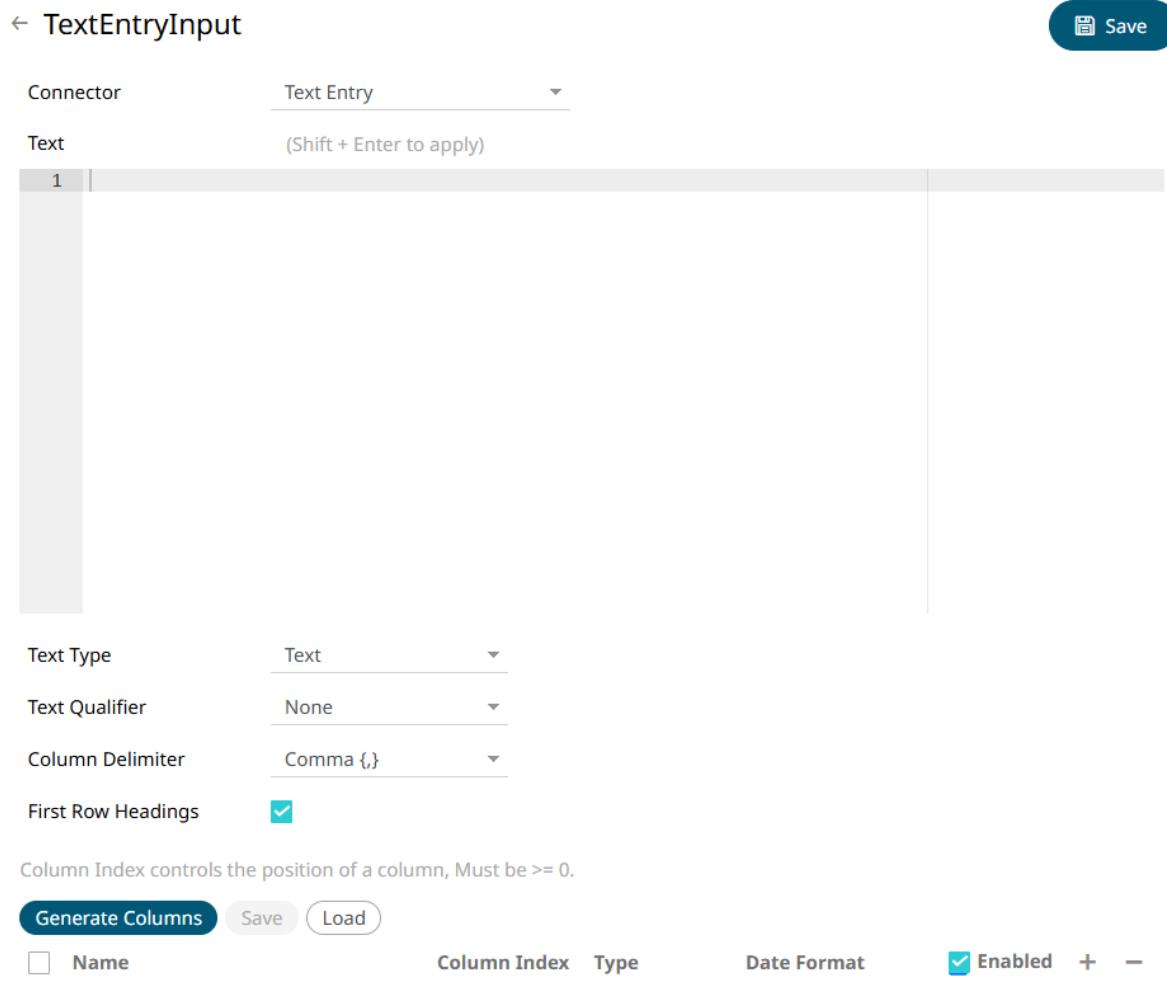
6. Click . The new data source is added to the *Data Sources* list.

Creating Text Entry Input Data Source

This is a connector that allows the creation of data tables with data typed or pasted directly into it. The supported text types include Fix, .JSON, Text, .XML, .HTML, and .SVG.

Steps:

1. In the *New Data Source* page, select **Input > Text Entry** in the **Connector** drop-down list.



Text Entry

(Shift + Enter to apply)

Text Type: Text

Text Qualifier: None

Column Delimiter: Comma {,}

First Row Headings:

Column Index controls the position of a column, Must be ≥ 0 .

Name	Column Index	Type	Date Format	Enabled	+	-
------	--------------	------	-------------	---------	---	---

2. You can either paste or type data into the *Text* box.

NOTE

- The *Paste Data* feature will only work when the page is running over SSL (HTTPS), with localhost being an exception. See [Browser Compatibility](#) for more information.
- Press **Shift + Enter** to apply the changes in the typed or pasted data.

3. Define the [Text Type](#).

4. Click **Generate Columns** to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the **Save** button is enabled.

5. You can also opt to [load or save](#) a copy of the column definition.

6. Click  to add columns to the Text Entry connection that represent sections of the message. Then enter or select:

Property	Description
Name	The column name of the source schema.
Fix Tag/JsonPath/Column Index/XPath	The Fix Tag/JsonPath/ Column Index/XPath of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Enabled	Determines whether the message field should be processed.

NOTE

To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of upper case S. There can be no additional characters following them.

For example: yyyy-MM-dd HH:mm:ss.SSSSSS

To delete a column, select its checkbox  or all the column entries, select the topmost checkbox  , then click .

Creating WebSocket Input Data Source

The WebSocket connector is very similar to the Stream Simulator connector, except that rather than looping through a file, it would either connect through web sockets, long polling, or repeatedly poll an external URL for new records to process.

Steps:

1. In the *New Data Source* page, select **Input > WebSocket** in the **Connector** drop-down list.

Connector: WebSocket

Path:

User Id:

Password: Show characters

Request Body:

Timeout: 10

Record Path:

Message Type: Json

Decimal Separator: Period (.)

Record Path: (eg. myroot.items.item)

Generate Columns **Save** **Load**

<input type="checkbox"/> Name	JsonPath	Type	Date Format	Enabled	<input type="button" value="+"/>	<input type="button" value="-"/>
<input type="checkbox"/> TradeTin	<input type="text"/>	Time <input type="button" value="▼"/>	<input type="text"/>	<input checked="" type="checkbox"/>	<input type="button" value=""/>	<input type="button" value=""/>

Real-Time Settings

Time Id Column: [No Time Id]

Time Id Column Name:

Reset Data on Reconnect:

2. Enter the connection details:

Property	Description
Path	The path to which the WebSocket server will respond to.
Proxy Server URI	The HTTP Proxy setting that will allow the WebSocket connector to reach the endpoint
User ID	The User ID that will be used to connect to the WebSocket server.
Password	The password that will be used to connect to the WebSocket server. Select the Show Characters check box to display the entered characters.
Request Body	For both the HTTP and ws:// POST requests sent to the WebSocket server.
Timeout	The length of time to wait for the server response (10 to 300). Default is 10 .

3. Select the Message Type.

4. Select either the period (.) or comma (,) as the *Decimal Separator*.

NOTE

Prepend 'default:' for the elements falling under default namespace.

5. Click  to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the Save button is enabled.
6. You can also opt to [load or save](#) a copy of the column definition.
7. You can opt to click  to add columns to the WebSocket connection that represent sections of the message. Then enter or select:

Property	Description
Name	The column name of the source schema.
JsonPath/Text Column Index/XPath	The JsonPath/Text Column Index/XPath of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Filter	Defined parameters that can be used as filter.
Enabled	Determines whether the message field should be processed.

NOTE

To parse and format times with higher than millisecond precision, the format string needs to end with a period followed by sequence of upper case S. There can be no additional characters following them.

For example: yyyy-MM-dd HH:mm:ss.SSSSSS

8. Define the [Real-time Settings](#).
9. Click . The new data source is added in the *Data Sources* list.

Creating Web Data Input Data Source

The Web Data connector allows the retrieval and processing of JSON, XML, delimited Text (such as CSV, TSV), Excel files, HTML tables and Arrow IPC files that are accessible over HTTP/HTTPS.

Steps:

1. In the *New Data Source* page, select **Input > Web Data** in the *Connector* drop-down list.

Connector Web Data

Http Method GET

URL

Test Connection and Update Settings

Test Result Authorization Headers Body Advanced

1		

Data Type Json

Record Path

Generate Columns Save Load

<input type="checkbox"/> Name	JsonPath	Type	Date Format	Enabled	+	-

2. Select the appropriate *HTTP Method* for the request from the following options:

HTTP Method	Description
GET	To retrieve data.
POST	To add new data.
PUT	To replace existing data

3. Enter the absolute URL of the web data including scheme (HTTP/HTTPS) into the *URL* field.

URL

4. Click **Test Connection and Update Settings**. A successful connection will result to the following:

- The **Connection Successful** status is displayed along with some raw data returned by the server.

```
Test Connection and Update Settings  Connection Successful

Test Result Authorization Headers Body Advanced

1 {"catalog":{ 2     "books":[{ 3         "id":"bk101", 4         "author":"Gambardella, Matthew", 5         "title":"XML Developer's Guide", 6         "genre":"Computer", 7         "price":44.95, 8         "publish_date": "2000-10-01" 9     }]} 10
```

- The connector tries auto discovery of the *Data Type* (JSON, Text, XML, Excel, HTML, Arrow IPC)

When loading JSON data, the *Record Path* can be suggested automatically by pressing the **Fetch** button.

Data Type
Json

Record Path
.catalog.books
Fetch

Array Handling
Add Rows

Generate Columns
Save
Load

<input type="checkbox"/>	Name	JsonPath	Type	Date Format	Enabled	+	-
<input type="checkbox"/>	id	.id	Text	▼	▼	<input checked="" type="checkbox"/>	-
<input type="checkbox"/>	author	.author	Text	▼	▼	<input checked="" type="checkbox"/>	-
<input type="checkbox"/>	title	.title	Text	▼	▼	<input checked="" type="checkbox"/>	-
<input type="checkbox"/>	genre	.genre	Text	▼	▼	<input checked="" type="checkbox"/>	-
<input type="checkbox"/>	price	.price	Num	▼	▼	<input checked="" type="checkbox"/>	-
<input type="checkbox"/>	publish_date	.publish_dat	Time	▼	yyyy-MM-dd	<input checked="" type="checkbox"/>	-
<input type="checkbox"/>	description	.description	Text	▼	▼	<input checked="" type="checkbox"/>	-

For the Web Data connector, you can also select the **HTML** data type.

Data Type

Table Name

Fetch
Load

Header Rows

Generate Columns
Save
Load

Name

Type

Date Format

Enabled
+
-

Property	Description
Table Name	Fetch <div style="display: flex; align-items: center; justify-content: space-between; width: 100%;"> Click to fetch HTML table elements then select the HTML table. </div>

Header Rows	Select the number of rows in the table to be retrieved as column headings. The default value is 1 , but a higher number should be set if the HTML table has column titles that use more than 1 row. If all rows of the table are data and there are no column headers, set it to 0 .
-------------	--

NOTE

If no table element is found in the HTML document, an error message displays.

You can also opt to [load or save](#) a copy of the column definition.

5. Adjust the *Authentication Type*, if required.

5.1 Select the **Authorization** tab.



Test Result **Authorization** Headers Body Advanced

Authentication Type **None**

5.2 Set the required settings:

Authentication Type	Description
None	No authentication needed.
Basic	<p>Test Result Authorization Headers Body Advanced</p> <p>Authentication Type Basic</p> <p>User Id</p> <p>Password <input type="checkbox"/> Show characters</p> <p>Enter the <i>User ID</i> and <i>Password</i> to connect to the connector's service. Select the Show Characters box to display the entered characters.</p>
OAuth	Some standard OAuth token request parameters come initialized with empty values. You can set these values and also add/remove any/all of the keys.

Test Result
Authorization
Headers
Body
Advanced

Authentication Type
OAuth

Token Url

Add Access Token To
Request Headers

Request Parameters

<input type="checkbox"/>	Key	Value	+	-
<input type="checkbox"/>	client_id			
<input type="checkbox"/>	client_secret			
<input type="checkbox"/>	grant_type			
<input type="checkbox"/>	scope			

Reset
Parameters

Enter or select the following settings:

- **Token URL** – The URL to retrieve the access token from.
- **Add Access Token To** - The Access token retrieved from the *Token URL* can be added to headers, URL or request body, depending on how the endpoint needs the token.

Request Headers

Request Url

Request Body

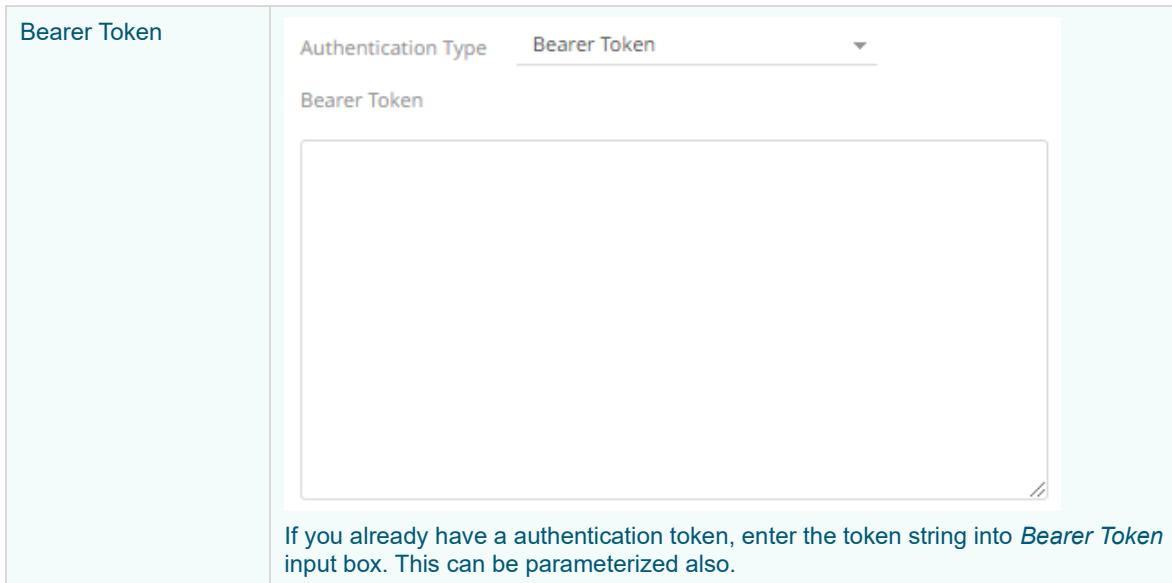
- Request Headers - A header is automatically added to the REST API request.
- Request URL - The URL needs to be manually parameterised with a {access_token} parameter, before calling the REST API, the parameter is replaced with the actual token.
- Request Body - The Request Body needs to be manually parameterised with a {access_token} parameter, before calling the REST API, the parameter is replaced with the actual token.

NOTE:

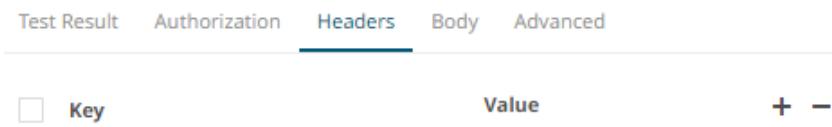
The given request parameters key/value pairs are formatted by the connector as **application/xwww-form-urlencoded**, and posted to *Token URL*.

Panopticon 2025.1 Streams Installation and Reference Guide

316

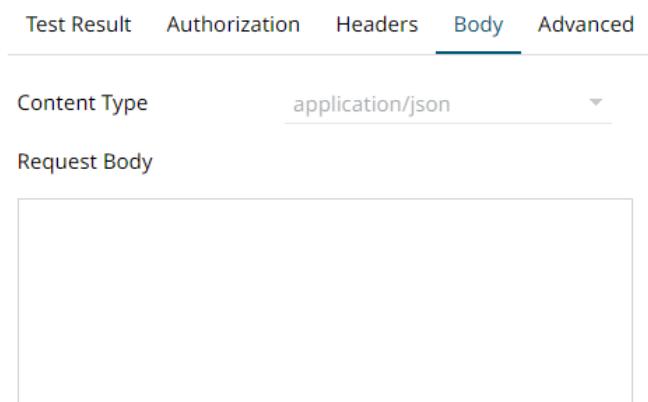


6. The **Headers** tab allows you to enter any custom headers required to be passed to the URL endpoint, typically to provide additional metadata. Enter any key/value pairs you need, and the connector will send them along with the request.



7. Set the *Body* if a POST/PUT request is required.

- 1.1. Select the **Body** tab.



- 7.2. Set the required settings:

Property	Description
Content Type	Select or enter content-type based on request body (payload) format. NOTE: This property is disabled when the HTTP Method is GET .
Request Body	The Request Body for the HTTP POST method.

8. Set the *Advanced* settings, if needed.

8.1. Select the **Advanced** tab.

Proxy Server URI	
Content Encoding	None
Timeout	10
Decimal Separator	Period (.)
File Encoding	

8.2. Set the required settings:

Property	Description
Proxy Server URI	The HTTP Proxy setting that will allow the connector to reach the endpoint.
Content Encoding	Select the <i>Content Encoding</i> with the HTTP Header: None , GZip , Deflate , or GZip and Deflate
Timeout	The length of time to wait for the server response (10 to 300). Default is 10 .
Decimal Separator	Select either the dot (.) or comma (,) as the decimal separator.
File Encoding	Set the character encoding to use in text data. <ul style="list-style-type: none">• UTF-8• UTF-16• UTF-32• US-ASCII• Windows-1252

9. Click  . The new data source is added to the *Data Sources* list.

Creating XML Input Data Source

The XML connector allows the retrieval and processing of XML files, either from a disk, a Text, or from a defined URL.

Steps:

1. In the *New Data Source* page, select **Input > Xml** in the *Connector* drop-down list.

Connector: XML

XML File Source: File

XML File Path:

Record XPath: (eg. //myroot/items/item)

Decimal Separator: Period {.}

Prepend 'default:' for the elements falling under default namespace.

Generate Columns **Save** **Load**

<input type="checkbox"/> Name	XPath	Type	Date Format	Enabled	+	-
-------------------------------	-------	------	-------------	---------	----------	----------

2. Select the XML [File Source](#).
3. Enter the Record XPath (e.g., //myroot/items/item).
4. Select either the period (.) or comma (,) as the *Decimal Separator*.
5. Click **Generate Columns** to the fetch the schema based on the connection details. Consequently, the list of columns with the data type found from inspecting the first 'n' rows of the input data source is populated and the Save button is enabled.
6. You can also opt to [load or save](#) a copy of the column definition.
7. You can opt to click **+**. A new column entry displays. Enter or select the following properties:

Property	Description
Name	The column name of the source schema.
XPath	The XPath of the source schema.
Type	The data type of the column. Can be a Text , Numeric , or Time
Date Format	The format when the data type is Time .
Enabled	Determines whether the message should be processed.

To delete a column, check its or all the column entries, check the topmost , then click **-**.

8. Click **Save**. The new data source is added in the *Data Sources* list.

MODIFYING DATA SOURCES

Steps:

1. On the **Data Sources** tab, click the link of a data source you want to modify.

The corresponding data source page is displayed.

Connector	Kdb+	▼	
Host	localhost		
Port	5001		
User Id			
Password			
Host Lookup Script	local		
Table	stocks		
<input type="checkbox"/> Source		Target	Type
<input type="checkbox"/> Ticker		Symbol	Text 
<input type="checkbox"/> Date		CloseDate	Time 
<input type="checkbox"/> Relative_Change		RelativeChange	Numeric 

All of the controls that are editable can be [modified](#).



2. Make the necessary changes then click  or the  icon. The context menu displays with two saving options:



Click to save the changes made in the data source.



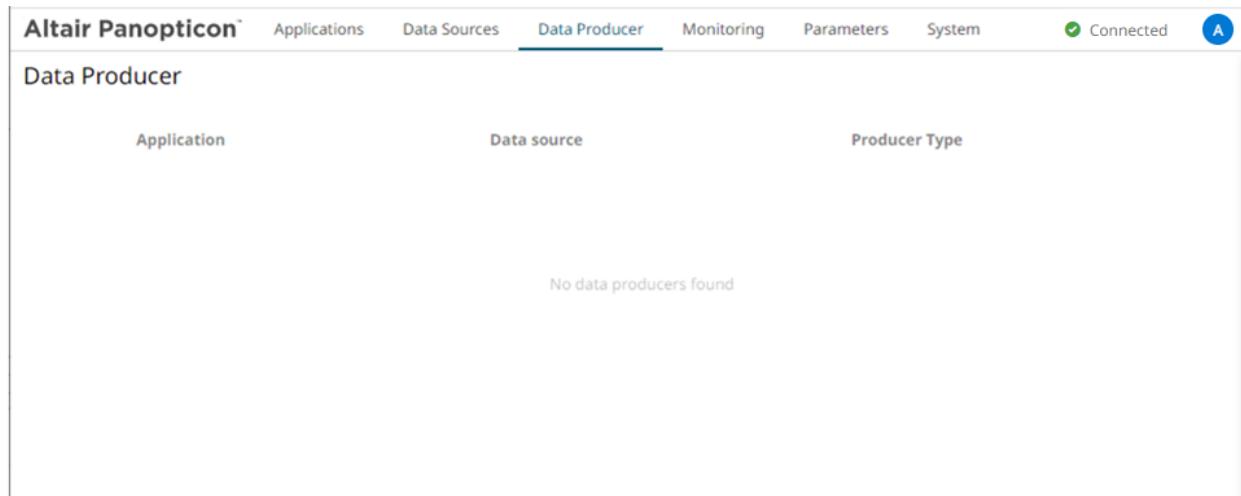
Click to make a duplicate of the data source. The original name is appended with **_Copy**.

To change the *Data Source Name*, click on it to make it editable, then enter a new one and click .

[10] MANAGING DATA PRODUCERS

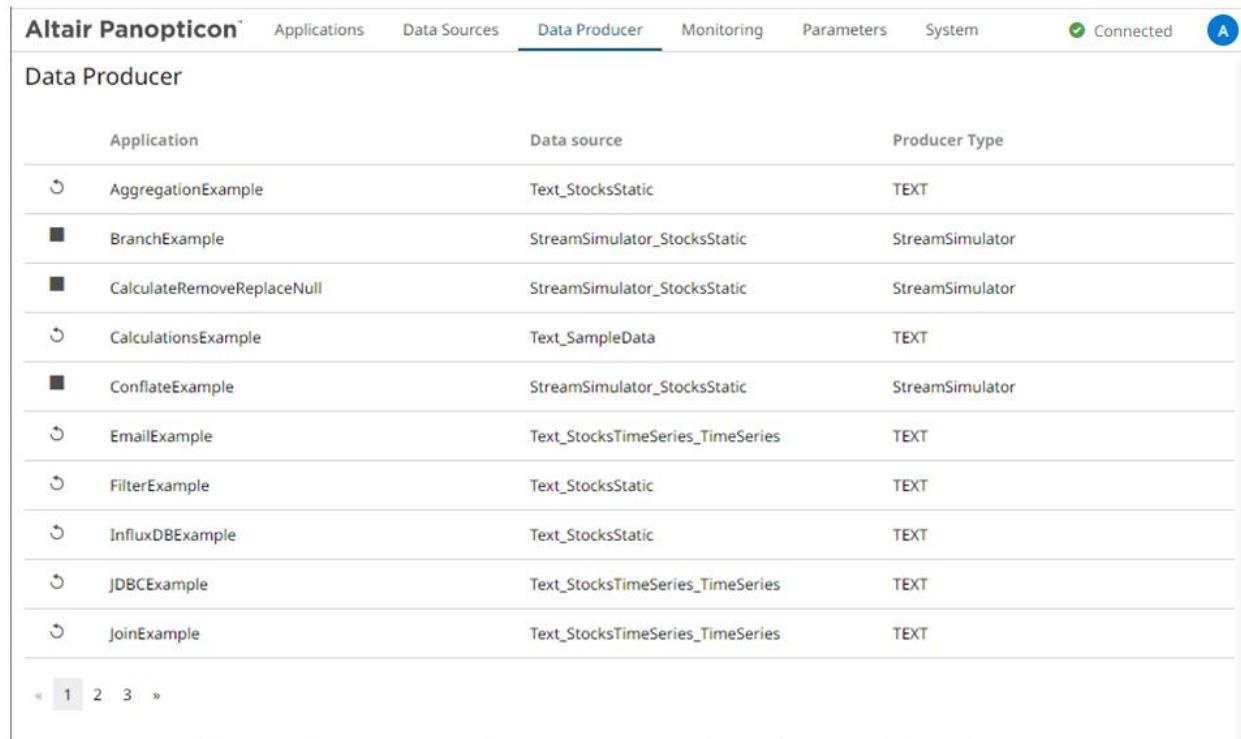
When an application has been started, the data producers used to generate data from the data sources will be displayed on the **Data Producer** tab where you can:

- Refresh data producers
- Start or Stop data producers



Application	Data source	Producer Type
No data producers found		

Data Producer tab. Initially, no records are displayed when there are no running applications or the applications that are running have no data producers



Application	Data source	Producer Type
AggregationExample	Text_StocksStatic	TEXT
BranchExample	StreamSimulator_StocksStatic	StreamSimulator
CalculateRemoveReplaceNull	StreamSimulator_StocksStatic	StreamSimulator
CalculationsExample	Text_SampleData	TEXT
ConflateExample	StreamSimulator_StocksStatic	StreamSimulator
EmailExample	Text_StocksTimeSeries_TimeSeries	TEXT
FilterExample	Text_StocksStatic	TEXT
InfluxDBExample	Text_StocksStatic	TEXT
JDBCExample	Text_StocksTimeSeries_TimeSeries	TEXT
JoinExample	Text_StocksTimeSeries_TimeSeries	TEXT

Data Producer tab with data producers currently started

Refresh Data Producers

Steps:

1. On the **Data Producers** tab, click the Refresh  icon of a data producer.

A confirmation message displays.

Yes

2. Click .

Starting or Stopping Data Producers

To start a Data Producer:

1. Click . A confirmation message displays.

2. Click **Yes**. The icon changes to .

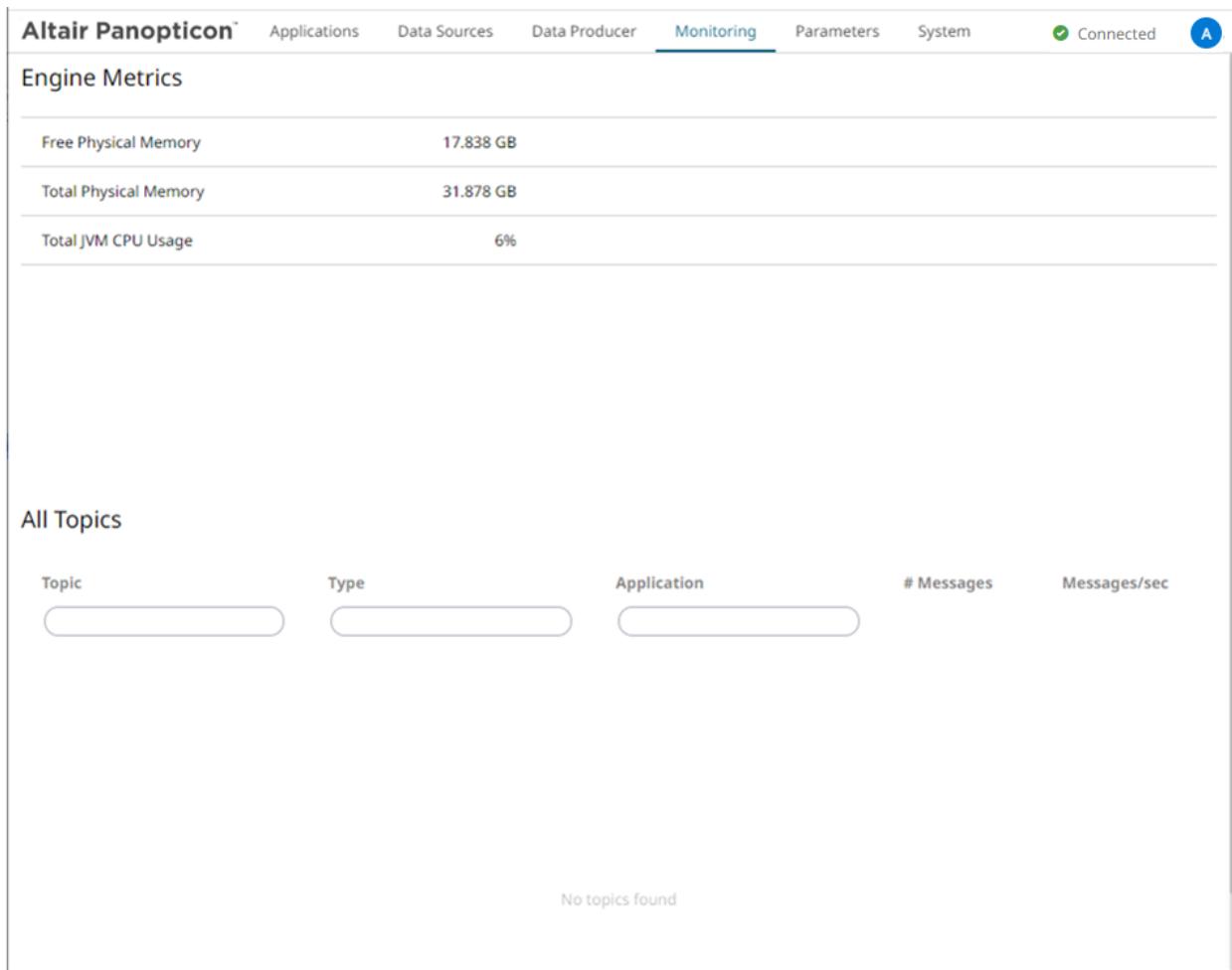
To stop the Data Producer:

1. Click . A confirmation message displays.

2. Click **Yes**. The icon changes to .

[11] MONITORING ENGINE METRICS AND APPLICATION TOPICS

The **Monitoring** tab provides the ability to monitor the engine metrics that can help determine which part of the application is causing data bottlenecks, among others.



The screenshot shows the Altair Panopticon interface with the 'Monitoring' tab selected. The 'Engine Metrics' section displays three metrics: Free Physical Memory (17.838 GB), Total Physical Memory (31.878 GB), and Total JVM CPU Usage (6%). The 'All Topics' section shows a table with columns for Topic, Type, Application, # Messages, and Messages/sec. A message at the bottom states 'No topics found'.

Engine Metric	Description
Free Physical Memory	The amount of free physical memory available to the Panopticon Streams server.
Total Physical Memory	The total amount of physical memory.
Total JVM CPU Usage	The recent CPU usage for the Java Virtual Machine process.

The screenshot shows the Altair Panopticon interface with the 'Monitoring' tab selected. The top navigation bar includes 'Applications', 'Data Sources', 'Data Producer', 'Monitoring' (underlined in blue), 'Parameters', 'System', and a 'Disconnected' status indicator with a yellow warning icon and a blue circular badge containing the letter 'A'.

Engine Metrics

Free Physical Memory	17.953 GB
Total Physical Memory	31.878 GB
Total JVM CPU Usage	2%

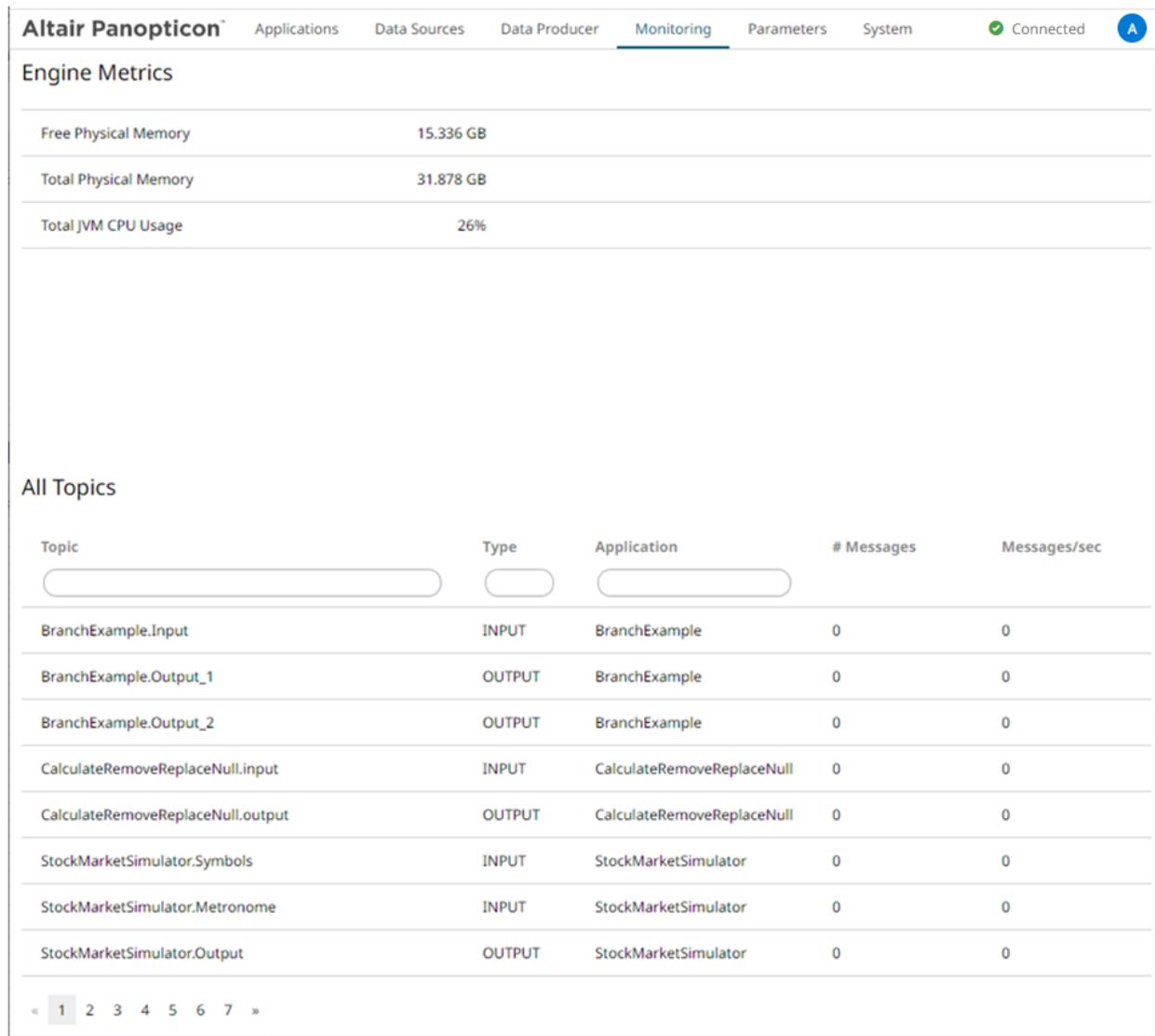
All Topics

Topic	Type	Application	# Messages	Messages/sec

No topics found

Monitoring tab when disconnected to the engine

It also displays the list of input and output topics currently running.



The screenshot shows the Altair Panopticon interface with the 'Monitoring' tab selected. At the top, there are tabs for Applications, Data Sources, Data Producer, Monitoring (which is underlined in blue), Parameters, System, and a status indicator showing 'Connected' with a green checkmark and a blue 'A' icon. Below the tabs, the 'Engine Metrics' section displays three items: 'Free Physical Memory' (15.336 GB), 'Total Physical Memory' (31.878 GB), and 'Total JVM CPU Usage' (26%). The main content area is titled 'All Topics' and contains a table with columns: Topic, Type, Application, # Messages, and Messages/sec. The table lists eight topics: 'BranchExample.Input' (INPUT, BranchExample, 0, 0), 'BranchExample.Output_1' (OUTPUT, BranchExample, 0, 0), 'BranchExample.Output_2' (OUTPUT, BranchExample, 0, 0), 'CalculateRemoveReplaceNull.input' (INPUT, CalculateRemoveReplaceNull, 0, 0), 'CalculateRemoveReplaceNull.output' (OUTPUT, CalculateRemoveReplaceNull, 0, 0), 'StockMarketSimulator.Symbols' (INPUT, StockMarketSimulator, 0, 0), 'StockMarketSimulator.Metronome' (INPUT, StockMarketSimulator, 0, 0), and 'StockMarketSimulator.Output' (OUTPUT, StockMarketSimulator, 0, 0). At the bottom of the table, there is a navigation bar with page numbers 1 through 7 and arrows for '«' and '»'.

Topic	Type	Application	# Messages	Messages/sec
BranchExample.Input	INPUT	BranchExample	0	0
BranchExample.Output_1	OUTPUT	BranchExample	0	0
BranchExample.Output_2	OUTPUT	BranchExample	0	0
CalculateRemoveReplaceNull.input	INPUT	CalculateRemoveReplaceNull	0	0
CalculateRemoveReplaceNull.output	OUTPUT	CalculateRemoveReplaceNull	0	0
StockMarketSimulator.Symbols	INPUT	StockMarketSimulator	0	0
StockMarketSimulator.Metronome	INPUT	StockMarketSimulator	0	0
StockMarketSimulator.Output	OUTPUT	StockMarketSimulator	0	0

Monitoring tab when the engine has been started along with some applications. The list of input and output topics is displayed.

MANAGING TOPICS

While [running or executing an application](#), input and output topics are retrieved and displayed on the **Monitoring** tab.

You can perform the following:

- View and monitor the number of retrieved messages and the number of retrieved messages per second
- Define a [filter](#) among the topics
- [Sort](#) the list of topics

Filter Topics

The topics can be filtered by entering letters, numbers, or underscores in the *Topic* or *Application* text box.

For the *Type* of application, enter a text (either **Output** or **Input**) into the text box above the listing.

Sorting the List of Topics

Modify the sorting of the list by clicking the  or  button of the *Topic*, *Type*, *Application*, *#Messages*, or *#Messages/sec* column. The icon beside the column that was used for the sorting will indicate if it was in an ascending or descending order.

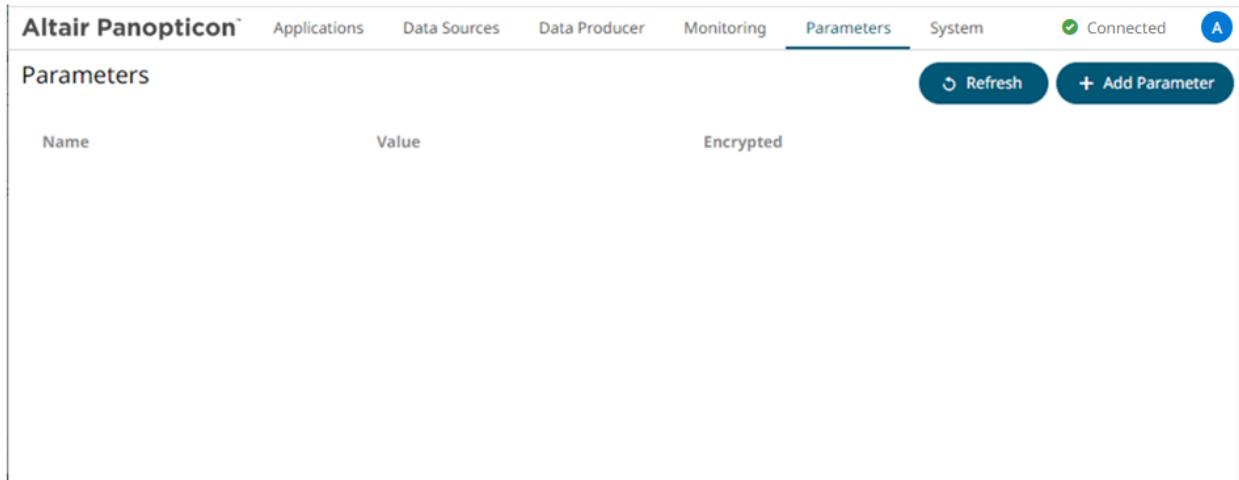
Moving to Other Topics List Pages

Go to the other topics pages by clicking:

- any link of a page number
-  . This displays the previous page
-  . This displays the next page

[12] MANAGING PARAMETERS

The **Parameters** tab supports adding, modifying, and deleting global parameters that will pull and enter specific data into the different components of an application model.



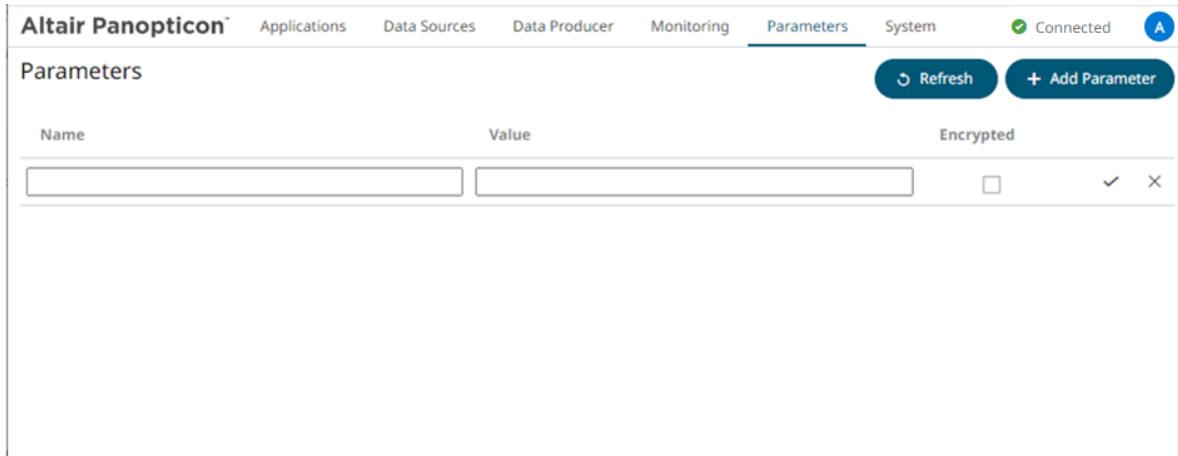
ADDING PARAMETERS

Steps:

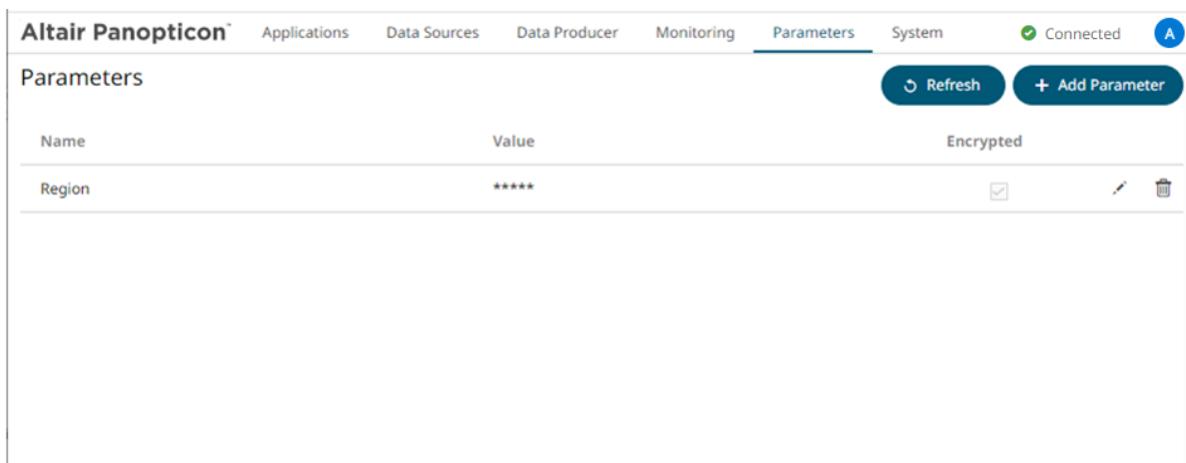
+ Add Parameter

1. On the Parameters tab, click

A new parameter entry displays.



2. Enter a *Name* for the new parameter and the *Value*.
3. Select the *Encrypted* check box to encrypt the value.
4. Click . The new parameter is added in the list.



Name	Value	Encrypted
Region	*****	<input checked="" type="checkbox"/>

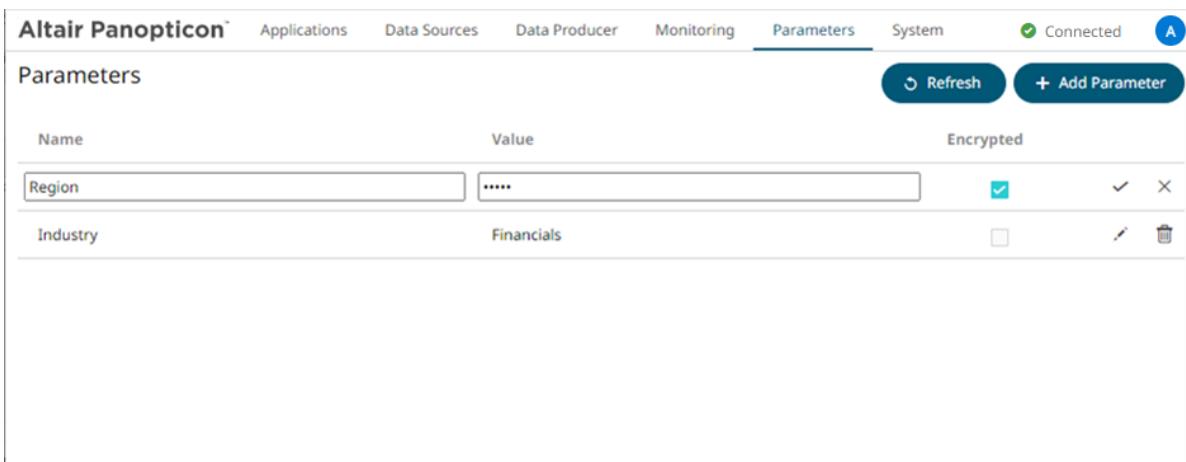
New parameters are added in the `Parameters.json` file located in the `AppData` folder (i.e., `c:\streamsseverdata`).

Modifying Parameters

Steps:

1. On the **Parameters** tab, click the **Edit** icon of a parameter you want to modify.

The *Name*, *Value*, and *Encrypted* controls are enabled.



Name	Value	Encrypted
Region	<input checked="" type="checkbox"/>
Industry	Financials	<input type="checkbox"/>

1. Make the necessary changes then click .

Deleting Parameters

Steps:

1. On the **Parameters** tab, click  of a parameter you want to delete.

A confirmation message displays.



2. Click  to delete.

Refresh Parameters

Click  to refresh the values that are being pulled by the application models.

Sorting the List of Parameters

By default, the parameters are listed based on the sequence that they were added. Modify the sorting of the list by clicking the  or  button of the *Name*, *Value*, or *Encrypted* columns. The icon beside the column that was used for the sorting will indicate if it was in an ascending or descending order.

[13] EXAMPLE APPLICATIONS

Panopticon Streams is installed with a series of example applications:

- AggregationExample – Demonstrates how to aggregate data based on a grouping key and a set of aggregated fields.
Includes simple [aggregations](#) such as avg, count, first, last, max, min, samples, sum, sdevp, sdevs, Sum, varp, and vars.
- BranchExample – Demonstrates how to split a stream into one or more branches.
- CalculateRemoveReplaceNull – Demonstrates how to:
 - remove and replace fields from output schemas
 - set a field value to null
 - set a field value to the current timestamp
- CalculationExample – Includes the SquareRoot calculation.
- CalculationsExample – Includes the following calculations:
 - Numeric calculations such as Abs, SquareRoot, Subtract, Multiply, Divide, Truncate, IF
 - Text calculations such as Upper, Lower, Proper, Left, Right, Mid, Concat, Find
 - Time Period calculations such as DateDiff

In addition, data type casting between Text, Number, and Date/Time

- ConflateExample – Demonstrates how to lower the frequency of updates by setting a fixed interval.
- EmailExample – Shows how to send an email via SMTP where the SMPT and email settings can be parameterized. Each record passed to the connector results in an email which can be primarily used as an output for alerting, having a conditional expression that would need to be fulfilled for a record to be forwarded to the output.
Requires the EmailWriter plugin.
- ExternalInputExample – Demonstrates how to directly source data from a Kafka topic (defined in the schema registry with the message format set to Avro).
- ExternalInputJsonParserExample – Demonstrates how to directly use a parsed input Json data.
- ExternalInputXMLParserExample - Demonstrates how to directly use a parsed input XML data.
- FilterExample – Demonstrates how to filter a data source based on a predicate.
- InfluxDBExample - Allows periodical dumping of records from a Kafka topic into an InfluxDB 1.x output connector. Requires the InfluxDBWriter plugin.
- JDBCExample – Allows periodical dumping of records from a Kafka topic into a JDBC database output connector. Requires the JDBCWriter plugin.
- JoinExample – Demonstrates how to join a stream to a global table.
- KdbExample - Allows periodical dumping of records from a Kafka topic into a Kx kdb+ output connector. Requires the KdbWriter plugin.
- MetronomeExample – Demonstrates how the metronome operator works in generating a timestamp field schema. A static metronome has a defined frequency while a dynamic metronome takes frequency as an input which determines the speed of the simulation.
- RetentionTimeExample – Demonstrates how to define the different retention time periods set for tables, input streams, output streams, and topics in an application.

This helps minimize memory utilization and the amount of data retrieved when subscribing from the beginning to the latest messages.

NOTE

Setting these properties in the application level overrides the defaults set in the [Streams.properties](#) file.

For example, if the following properties are defined in the `streams.properties` file:

```
cep.kafka.table.retention.ms=86400000
cep.kafka.input.retention.ms=60000
cep.kafka.output.retention.ms=900000
```

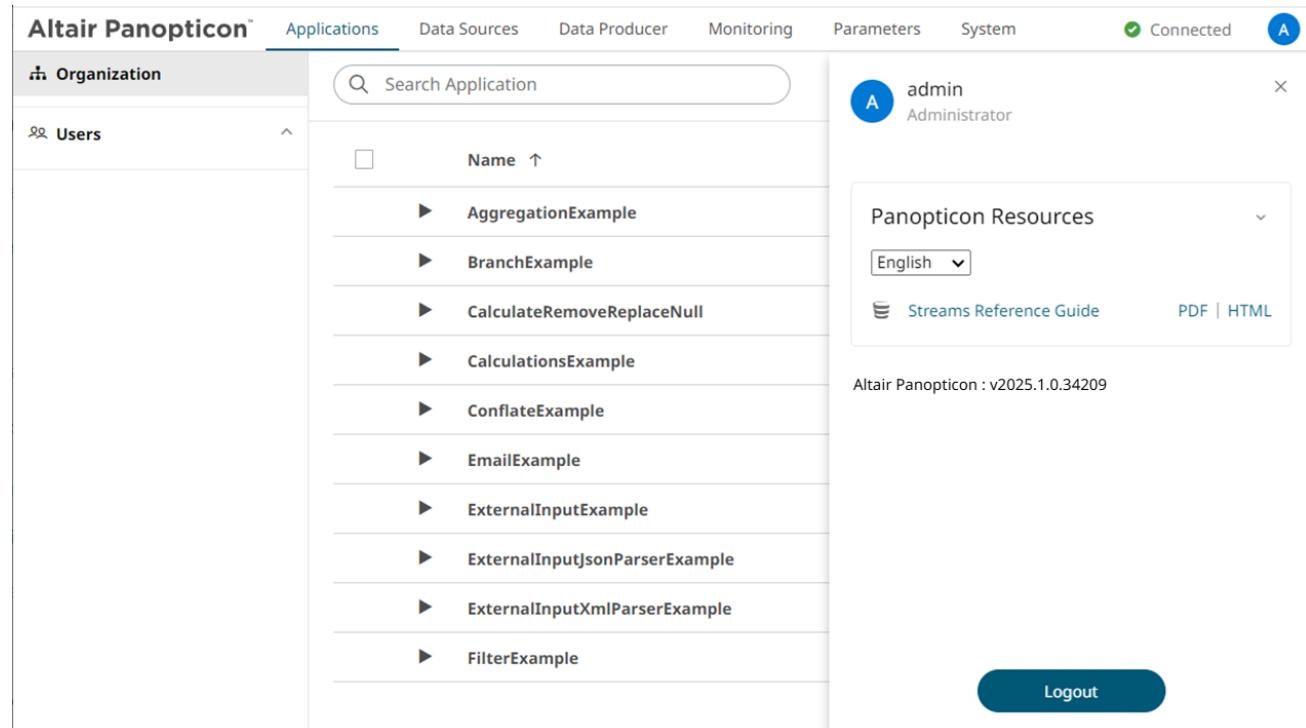
In the application level, the input retention period will be 1,000 milliseconds instead of 60,000 and the output retention period will be 1,000 milliseconds instead of 900,000. Also, a custom topic retention period has been added using the following pattern: `TopicName.retention.ms` (i.e., `TimeSeries.retention.ms`).

```
<properties>
    <!-- Keep tables alive one day -->
    <entry>
        <key>table.retention.ms</key>
        <value>86400000</value>
    </entry>
    <!-- Keep input and output streams for 1 second -->
    <entry>
        <key>input.retention.ms</key>
        <value>1000</value>
    </entry>
    <entry>
        <key>output.retention.ms</key>
        <value>1000</value>
    </entry>
    <!-- Custom retention time for InputStream topic -->
    <entry>
        <key>TimeSeries.retention.ms</key>
        <value>1111</value>
    </entry>
</properties>
```

- StockMarketSimulator – Shows a stock market simulation using a streaming data with join, calculations, and metronome operators.
- StockStaticTimeSeriesApp – Joins a static and a time series data sources using common keys. Also demonstrates adding a sum aggregation.
- StreamtoGlobalTableJoinExample – Joins stream and global table inputs using common keys.
- StreamToTableJoinExample - Joins stream and table inputs using common keys.
- TextExample - Allows periodical dumping of records from a stream Kafka topic into a Text connector. Requires the TextWriter plugin.
- UnionExample- Unioning of two streams.
- WindowedStreamExample – Demonstrates aggregation across a windowed stream.

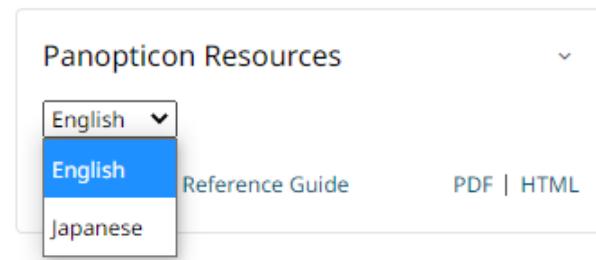
[14] PANOPTICON RESOURCES

Clicking  on the top right section of the toolbar displays the available Panopticon online resources.



The screenshot shows the Altair Panopticon application interface. The top navigation bar includes tabs for Applications, Data Sources, Data Producer, Monitoring, Parameters, System, and a Connected status indicator. A user icon with a blue 'A' is located in the top right corner. A red arrow points to the 'Panopticon Resources' dropdown menu. The main content area on the left has sections for Organization and Users. The Applications tab is selected, showing a list of examples: AggregationExample, BranchExample, CalculateRemoveReplaceNull, CalculationsExample, ConflateExample, EmailExample, ExternalInputExample, ExternalInputJsonParserExample, ExternalInputXmlParserExample, and FilterExample. The right side of the interface shows a user profile for 'admin' (Administrator) and the 'Panopticon Resources' section, which is currently set to 'English'. It also includes links to 'Streams Reference Guide' and 'PDF | HTML'.

Select the *Language* on the drop-down list: **English** or **Japanese**.



Then click *Panopticon Streams Reference Guide* either on a PDF or HTML Help format. This guide is also available upon installation.

[APPENDIX]

PROPERTIES: STREAMS

The majority of configuration options for the server are set in the `Streams.properties` file in the `<appdata>` directory (e.g., `C:\streams\serverdata\`). If this file does not exist when the server starts, it will create it with all default values. When the server starts after an upgrade, it may add new properties and remove deprecated ones.

You can optionally move sensitive properties like passwords and URLs from this file, where they are stored in clear text, into a file named `Secret.properties` in the same directory. The `Secret.properties` file stores values encrypted, and you can manage it with [PCLI](#). A property can only be defined in one of these files at a time.

The following properties can be set in the property files:

c	
Property	Access
Attribute	<code>access.administrator.groups</code>
Description	The role that is mapped to the administrator group.
Default Value	<code>admin</code>
Property	Access
Attribute	<code>access.administrator.users</code>
Description	Normally administrator access should be handled with the <code>access.administrator.groups</code> mapping, but for scenarios where the authentication cannot provide roles or you want to make exceptions for specific users, you can list individual usernames in this property. Any user listed here will get administrator access, regardless of their roles. Separate multiple users with the access.list.delimiter .
Default Value	
Property	Access
Attribute	<code>access.default.roles</code>
Description	The default roles applied to all users of the server. For example, if <code>access.default.roles=DESIGNER,ADMINISTRATOR</code> and a user with a <code>VIEWER</code> role logs on to the server, then the user will simultaneously have a <code>VIEWER</code> , <code>DESIGNER</code> , and <code>ADMINISTRATOR</code> roles. However, if no default roles are wanted, then leave the property blank. NOTE: The roles that can be assigned in this property can only be <code>ADMINISTRATOR</code> , <code>VIEWER</code> , <code>ANONYMOUS</code> , and/or <code>DESIGNER</code> . This property is case sensitive.
Default Value	<code>VIEWER</code>
Property	Access
Attribute	<code>access.designer.groups</code>
Description	The role that is mapped to the designer group.

Default Value	designer
Property	Access
Attribute	access.designer.users
Description	<p>Normally designer access should be handled with the <code>access.designer.groups</code> mapping, but for scenarios where the authentication cannot provide roles or you want to make exceptions for specific users, you can list individual usernames in this property.</p> <p>Any user listed here will get designer access, regardless of their roles.</p> <p>Separate multiple users with the <u>access.list.delimiter</u>.</p>
Default Value	
Property	Access
Attribute	access.viewer.groups
Description	<p>The role that is assigned to the viewer group.</p> <p>NOTE: Currently not in use. Development ongoing.</p>
Default Value	
Property	Access
Attribute	access.viewer.users
Description	<p>Normally viewer access should be handled with the <code>access.viewer.groups</code> mapping, but for scenarios where the authentication cannot provide roles or you want to make exceptions for specific users, you can list individual usernames in this property.</p> <p>Any user listed here will get viewer access, regardless of their roles.</p> <p>Separate multiple users with the <u>access.list.delimiter</u>.</p>
Default Value	
Property	Access
Attribute	access.list.delimiter
Description	<p>The value delimiter to use when parsing access groups.</p> <p>Examples:</p> <p><code>access.list.delimiter=,</code> <code>access.administrator.groups=group1,group2</code></p> <p>The groups are mapped to <code>{'group1', 'group2'}</code></p> <p><code>access.list.delimiter=,</code> <code>access.administrator.groups=group1;group2,group3</code></p> <p>The groups are mapped to <code>{'group1;group2', 'group3'}</code></p> <p><code>access.list.delimiter=;</code> <code>access.administrator.groups=group1;group2,group3</code></p>

	The groups are mapped to {'group1', 'group2,group3'}						
Default Value	'; (comma)						
Property	Authentication: Header						
Attribute	authentication.header.role.delimiter						
Description	The delimiter used to separate the roles. Example: role1, role2,role3						
Default Value	, (Comma)						
Property	Authentication: Header						
Attribute	authentication.header.roles						
Description	The name of the header that contains all the roles.						
Default Value							
Property	Authentication: Header						
Attribute	authentication.header.rolesdynamic						
Description	<p>Supports the ability to create dynamic roles using free form patterns or string replacement.</p> <p>To create dynamic roles, use '{header value to be used}'.</p> <p>Example: <code>authentication.header.rolesdynamic={HEADER_ROLES},financials,role_for_company_{HEADER_COMPANY}</code></p>						
	<p>Given this table:</p> <table border="1"> <thead> <tr> <th>KEY</th><th>VALUE</th></tr> </thead> <tbody> <tr> <td>HEADER_ROLES</td><td>designer, watcher</td></tr> <tr> <td>HEADER_COMPANY</td><td>industrials, consumers</td></tr> </tbody> </table>	KEY	VALUE	HEADER_ROLES	designer, watcher	HEADER_COMPANY	industrials, consumers
KEY	VALUE						
HEADER_ROLES	designer, watcher						
HEADER_COMPANY	industrials, consumers						
	<p>Then the roles to create the authentication token will be the following:</p> <ul style="list-style-type: none"> designer watcher financials role_for_company_industrials role_for_company_consumers 						
Default Value							
Property	Authentication: Header						
Attribute	authentication.header.username						
Description	The name of the header that contains the username						
Default Value							
Property	Authentication: Header						
Attribute	authentication.header.validate.token						

Description	If set to true , the authentication will validate the token. If set to false , the authentication of every request will be based on headers.
Default Value	
Property	Authentication: Logout
Attribute	authentication.logout.redirect.url
Description	Takes a URL as a parameter. Clicking the logout button redirects the user to the specified URL. If this property is not set, user will be returned to the start page of Panopticon.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	authentication.oauth2.client.ID
Description	The ID of the OAuth 2.0 client.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	authentication.oauth2.client.secret
Description	The secret used by the OAuth 2.0 client.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	authentication.oauth2.identity.attribute.roles
Description	The attribute that will be extracted from the identity response and used as the role. There can be multiple assigned roles for a user.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	authentication.oauth2.identity.attribute.roles.pattern
Description	Takes regex used to extract the roles from the OAuth 2.0 server identity response. For example, the returned string: <code>cn=admin,ou=groups,dc=openam,dc=openidentityplatform,dc=org,dc=designer,ou=groups,dc=openam,dc=openidentityplatform,dc=org</code> contains two roles, admin and designer The regex to extract the roles is cn=([^,]+) .
Default Value	
Property	Authentication: OAuth 2.0
Attribute	authentication.oauth2.identity.attribute.username
Description	The attribute that will be extracted from the identity response and used as the username.
Default Value	
Property	Authentication: OAuth 2.0

Attribute	<code>authentication.oauth2.identity.method</code>
Description	The method on how the access token is passed along in the identity request. Supported values are QUERY , BODY , and HEADER .
Default Value	QUERY
Property	Authentication: OAuth 2.0
Attribute	<code>authentication.oauth2.identity.url</code>
Description	The URL to the REST service that provides details about the authenticated user.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	<code>authentication.oauth2.login.callback.url</code>
Description	The callback URL. The URL should be the same as one of the specified callback URLs used by the client. The URL should refer to Panopticon Streams.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	<code>authentication.oauth2.login.redirect.url</code>
Description	Redirects the user to the specified URL after successfully logging in. This property can be left blank, in which case the user is redirected to the URL they requested to access.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	<code>authentication.oauth2.login.response.type</code>
Description	The response type. The only response type that is currently supported is CODE . The value can also be left blank.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	<code>authentication.oauth2.login.scope</code>
Description	The requested scope. The field can be left blank.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	<code>authentication.oauth2.login.url</code>
Description	The URL to the OAuth 2.0 login resource.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	<code>authentication.oauth2.logout.redirect.url</code>
Description	Logging out revokes the token from the authentication server if the property <code>authentication.oauth2.logout.url</code> is set to the revocation URL. If this property is not set, the server will only remove its own token.

	If none of these properties are set, the server will attempt to redirect to the start page of Panopticon when logging out.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	authentication.oauth2.logout.url
Description	The URL to the OAuth 2.0 logout resource. This field can be left blank.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	authentication.oauth2.token.method
Description	The method on how the token should be retrieved. Supported values are QUERY , BODY , and HEADER .
Default Value	
Property	Authentication: OAuth 2.0
Attribute	authentication.oauth2.token.url
Description	The URL to the OAuth 2.0 token resource.
Default Value	
Property	Service authentication level
Attribute	authentication.role
Description	The authentication role.
Default Value	
Property	Service authentication level
Attribute	authentication.required
Description	The property that will make the authentication required. It will force the user to login in order to use any of the services provided by the server.
Default Value	true
Property	Authentication: SAML
Attribute	authentication.saml.serviceprovider.id
Description	The ID of the service provider configured in the IdP.
Default Value	
Property	Authentication: SAML
Attribute	authentication.saml.assertionconsumerservice.url
Description	The URL to the Panopticon assertion consumer service. URL: [Protocol]://[Host]:[Port]/[Context]/server/rest/auth/login
Default Value	
Property	Authentication: SAML
Attribute	authentication.saml.identityprovider.url

Description	The URL to the IdP login service.
Default Value	
Property	Authentication: SAML
Attribute	authentication.saml.assertion.username
Description	User attribute for username configured in the IdP.
Default Value	
Property	Authentication: SAML
Attribute	authentication.saml.assertion.roles
Description	User attribute for roles configured in the IdP.
Default Value	
Property	Authentication: SAML
Attribute	authentication.saml.certificate.name
Description	The name of the certificate used to validate signature and/or sign outgoing SAML messages
Default Value	
Property	Authentication: SAML
Attribute	authentication.saml.certificate.password
Description	The password of the certificate used to validate signature and/or sign outgoing SAML messages.
Default Value	
Property	Authentication: SAML
Attribute	authentication.saml.challenge.required
Description	This property determines whether the IdP-first authentication with SAML is enabled or not. To enable, set this property to false .
Default Value	true
Property	Authentication: SAML
Attribute	authentication.saml.identityprovider.logout.url
Description	The URL to the IdP logout service.
Default Value	
Property	Authentication: SAML
Attribute	authentication.saml.keystore.file
Description	The location of the Keystore file that contains the certificate.
Default Value	
Property	Authentication: SAML
Attribute	authentication.saml.keystore.password

Description	The password to the Keystore file.
Default Value	
Property	Authentication: SAML
Attribute	authentication.saml.identityprovider.certificate.file
Description	Takes a file path to a certificate file that contains the IdP's public key.
Default Value	
Property	Authentication: SAML
Attribute	authentication.saml.identityprovider.signature.validation.required
Description	Specifies whether to require a valid IdP signature to be present on the SAML response. Default value is false .
Default Value	false
Property	Authentication: SAML
Attribute	authentication.saml.provider
Description	The IdP provider. Possible values are OPENSAML , OPENAM .
Default Value	OPENSAML
Property	Authentication: SAML
Attribute	authentication.saml.keystore.type
Description	The key store type. Possible values are JKS , JCEKS , PKCS12 .
Default Value	JKS
Property	Authentication: SAML
Attribute	authentication.saml.login.redirect.url
Description	Redirects the user to the specified URL after successfully logging in. This property can be left blank, in which case the user is redirected to the URL they requested to access.
Default Value	
Property	Authentication: SAML
Attribute	authentication.saml.logout.redirect.url
Description	Redirects the user back to the specified URL after logging out. This is mainly used with a proxy. In which case, Panopticon Real Time does not know the endpoint which the user is going towards to, and therefore cannot redirect the user back to the Overview page. If you are using OpenAM this is required, otherwise this property can be left blank.
Default Value	
Property	Authentication: SAML
Attribute	authentication.saml.openam.meta.alias
Description	The meta alias for the IdP if you are using OpenAM.

Default Value	
Property	Authentication: SAML
Attribute	authentication.saml.protocolbinding
Description	Protocol binding for the use of SAML authentication. Possible values are HTTP-Redirect , HTTP-POST , HTTP-Artifact , HTTP-POST-SimpleSign , or SOAP .
Default Value	HTTP-Redirect
Property	Service authentication login request
Attribute	authentication.timeout.callback
Description	The timeout (in milliseconds) for the user between initiated login and callback. The default value is five minutes.
Default Value	300000
Property	Authentication: Token
Attribute	authentication.token.cookie
Description	Used when sticky load balancer is using cookies.
Default Value	stoken
Property	Authentication: Token
Attribute	authentication.token.domain
Description	The domain in which the token cookie should be registered under.
Default Value	
Property	Authentication: Token
Attribute	authentication.token.persistence
Description	This property is used to determine if the token should persist if the browser is closed or if it should only last while the browser is open. There are two possible values: PERSISTENT and SESSION . PERSISTENT will persist the token in the browser even if the browser has been closed and reopened. SESSION will remove the token from the browser if it is shutdown. IMPORTANT: After modifying the property value to SESSION , ensure to clear the <code>AppData/Token</code> folder before starting the server.
Default Value	PERSISTENT
Property	Authentication: Token
Attribute	authentication.token.refreshable
Description	This property determines if the token can refresh itself. The web client can identify if the token is about to expire and then request a new token with the existing token. A token is refreshable if the property is set to true. The token will expire and invalidate the user session if the property is set to false.
Default Value	true
Property	Authentication: Token
Attribute	authentication.token.secret

Description	The secret is used to sign the token. The secret will be auto-generated when the server starts for the first time. NOTE: This value should be kept a secret.
Default Value	Auto-generated
Property	Authentication: Token
Attribute	authentication.token.validity.seconds
Description	The number of seconds that the token should be valid.
Default Value	604800
Property	Authentication
Attribute	authentication.type
Description	The type of the authentication mechanism that will be used on Panopticon Streams.
Default Value	BASIC
Property	Cache
Attribute	cache.plugin.ID
Description	The ID of the cache plugin that will be used. Possible value: BinaryTableFile-Cache
Default Value	BinaryTableFile-Cache
Property	Cache
Attribute	cache.purge.condition
Description	The condition for determining when the cache should be purged or cleared. Possible values: NONE, MEMORY .
Default Value	MEMORY
Property	Cache
Attribute	cache.purge.condition.memory.threshold
Description	The memory threshold used to determine if the cache should be purged or not. The values are presented in percent, 0-100. 80 means that the cache will be purged if the memory consumption reaches 80 % or more.
Default Value	80
Property	Cache
Attribute	cache.purge.enabled
Description	Enable or disable the purge functionality. Possible values: true, false
Default Value	true
Property	Cache
Attribute	cache.schedule.clear.enabled
Description	Enable the cache clearing schedule. This is scheduling the clear cache operation which will remove all the expired cache entries.
Default Value	true

Property	CEP: Application
Attribute	<code>cep.application.autostart</code>
Description	Determines whether all of the stored applications in the Streams server should auto start when the Streams server starts.
Default Value	false
Property	CEP: Application
Attribute	<code>cep.kafka.application.state.path</code>
Description	Where the tmp folder of Panopticon Streams data are created.
Default Value	C:/PanopticonStreams/Data/tmp/kafka-streams
Property	CEP: Kafka
Attribute	<code>cep.kafka.connection.timeout</code>
Description	The connection timeout towards Kafka. The value is presented in milliseconds.
Default Value	10000
Property	CEP: Kafka
Attribute	<code>cep.kafka.input.retention.ms</code>
Description	Specifies the retention period of input streams.
Default Value	60000
Property	CEP Kafka
Attribute	<code>cep.kafka.monitoring.consumer.interceptor</code>
Description	Names of classes that will be used to monitor data consumed from topics in a Streams application. In addition, these are hooks that will allow an external application to inspect this traffic. NOTE: The default value enables the Confluent Control Center to show metrics for a Streams application.
Default Value	io.confluent.monitoring.clients.interceptor.MonitoringConsumerInterceptor
Property	CEP Kafka
Attribute	<code>cep.kafka.monitoring.producer.interceptor</code>
Description	Names of classes that will be used to monitor data produced to topics in a Streams application. In addition, these are hooks that will allow an external application to inspect this traffic. NOTE: The default value enables the Confluent Control Center to show metrics for a Streams application.
Default Value	io.confluent.monitoring.clients.interceptor.MonitoringProducerInterceptor
Property	CEP: Kafka
Attribute	<code>cep.kafka.output.retention.ms</code>
Description	Specifies the retention period of output streams.
Default Value	900000
Property	CEP: Kafka

Attribute	<code>cep.kafka.properties</code>
Description	The user-defined file that contains the properties for controlling the Kafka configurations in Panopticon Streams.
Default Value	kafka.properties
Property	CEP: Kafka
Attribute	<code>cep.kafka.schemaregistry.url</code>
Description	The URL to the Schema Registry.
Default Value	http://localhost:8081
Property	CEP: Kafka
Attribute	<code>cep.kafka.servers</code>
Description	The URL to all the Kafka servers.
Default Value	localhost:9092
Property	CEP: Kafka
Attribute	<code>cep.kafka.session.timeout</code>
Description	The timeout for the Kafka session. The value is presented in milliseconds.
Default Value	15000
Property	CEP: Kafka
Attribute	<code>cep.kafka.table.retention.ms</code>
Description	Specifies the retention period for tables.
Default Value	86400000
Property	CEP: Kafka
Attribute	<code>cep.kafka.topic.partitions</code>
Description	Propagates the server-wide default for topic partitions. NOTES: <ul style="list-style-type: none"> The <i>Partition Count</i> priority is applied in the following sequence (top to bottom): <ol style="list-style-type: none"> Topic level Application level Property level The event processor create topic gets the partition count for that topic. If the topic exists, it checks for an existing partition count and deletes the topic if it has a different value, and then creates it with the provided partition count. Kafka server with auto topic creation on connect will cause issues due to preemptive metric collector component. This can be fixed with Kafka-client version 2.3 onward upgrade and adding "allow.auto.create.topics=false" in KafkaConsumer properties for TopicMetricsThread: <ul style="list-style-type: none"> PreviewSubscriptionThread TopicInputSchemaRepository
Default Value	1

Property	CEP: Kafka
Attribute	<code>cep.kafka.watcher.wait</code>
Description	The interval (in milliseconds) at which Streams will check the status of the ZooKeeper, Kafka Broker, and Schema Registry services.
Default Value	5000
Property	CEP: Kafka
Attribute	<code>cep.kafka.zookeeper.servers</code>
Description	The URL to the ZooKeeper servers.
Default Value	localhost:2181
Property	CEP: Kafka
Attribute	<code>cep.type</code>
Description	The CEP type. For now, the available value is KAFKA .
Default Value	KAFKA
Property	Server Cluster
Attribute	<code>cluster.bully.bind</code>
Description	The URL of the server in bully mode. This should be the URL to the Panopticon server web application on the server itself, by which is reachable from the other servers.
Default Value	
Property	Server Cluster
Attribute	<code>cluster.bully.boot</code>
Description	Comma-separated list of server URLs in bully mode. At least one of these servers should be running at all time for the bully mode to work correctly. The URLs should be the same as the <code>cluster.bully.bind</code> value on each boot server.
Default Value	
Property	Server Cluster
Attribute	<code>cluster.bully.id</code>
Description	The unique server ID in bully mode. Can be any string, but do not change it after the server has participated in a cluster -- the other servers will store it and expect it to identify the same server in the future. The running server with the lowest ID lexicographically will be leader.
Default Value	
Property	Server Cluster
Attribute	<code>cluster.fixed.leader</code>
Description	The leader URL in fixed mode. This should be the URL to the Panopticon server web application on the preset leader server, by which it is reachable from the follower servers. Leave blank on the leader server itself.

Default Value	
Property	Server Cluster
Attribute	<code>cluster.kubernetes.container_name</code>
Description	Optionally name of the container that runs the Panopticon server, if the pod also runs other containers. If left blank, the first container will be used.
Default Value	
Property	Server Cluster
Attribute	<code>cluster.kubernetes.id</code>
Description	Set to the name of the pod that runs the container.
Default Value	(blank)
Property	Server Cluster
Attribute	<code>cluster.kubernetes.label_selector</code>
Description	Standard Kubernetes label selector that should only match the pods that are running the server.
Default Value	
Property	Server Cluster
Attribute	<code>cluster.mode</code>
Description	NONE (default), FIXED , BULLY , or KUBERNETES Controls how multiple servers connect to each other. This needs to be the same on all connected servers.
Default Value	NONE
Property	Server Cluster
Attribute	<code>cluster.shared.secret</code>
Description	Any alphanumeric string. Secret used to encrypt a challenge in peer-to-peer communication handshake. Needs to be the same, and non-empty, on all connected servers.
Default Value	
Property	Server Cluster
Attribute	<code>cluster.shared.store.shared_directory.path</code>
Description	Shared store location in SHARED_DIRECTORY mode. This path must be reachable by all connected servers, and must point to the same physical directory on all of them.
Default Value	
Property	Server Cluster
Attribute	<code>cluster.shared.store.type</code>
Description	PRIVATE_DIRECTORY (default) or SHARED_DIRECTORY The shared store is used to store information that should be synchronized between servers but is not content, for example authentication tokens. If you have a tightly-

	coupled cluster, e.g., behind a load balancer, it is recommended that you configure this as a shared directory.
Default Value	
Property	AMPS Connector Custom Authenticator
Attribute	connector.amps.authenticators
Description	<p>This property is required when a custom authenticator is needed for AMPS connection. A custom authenticator needs be implemented as java .JAR file. The property accepts a JSON object, where key is fully qualified name of the Authenticator Java class, and values are list of constructor parameter names, e.g., <code>{"com.panopticon.examples.amps.AMPSClientAuthenticator": ["User", "Shared Key"]}</code></p>
Default Value	
Property	Host Lookup
Attribute	connector.kdb.host.lookup.script
Description	<p>Full path of the shell script file that is accessible on the server. When set, before making a new kdb+ connection, this script is executed to get the host info. This property helps in overriding connection details entered inside the kdb+ connector UI centrally, and may help when different authentications are set at kdb+ like Kerberos/Custom etc. The output of this script is expected to be a JSON object like below.</p> <pre>{ "host": "localhost", "port": 5001, "username": "", "password": "" }</pre>
Default Value	
Property	Host Lookup
Attribute	connector.kdb.host.lookup.script.arguments
Description	<p>Delimited set of arguments to be passed to the script when it is executed. '{host}, {port}, {userid}, {password}' is the default value, and these parameters are mapped to respective settings in the connector UI i.e., the value entered against these settings in the connector UI are passed as arguments to the script.</p> <p>This property can be extended or updated if you want to pass other datatable parameters as arguments. System parameter like {_user_id} or {_workbook_folder}, if added to the data table, can also be used. If the value of some parameter is null or empty at the time of execution of the script, two single quotes are passed (") against that parameter, this is to make sure that arguments count matches the arguments set at this property.</p>
Default Value	<code>{host},{port},{userid},{password}</code>
Property	Host Lookup
Attribute	connector.kdb.host.lookup.script.arguments.delimiter
Description	Used to split the arguments set at above property.
Default Value	,
Property	Host Lookup
Attribute	connector.kdb.host.lookup.script.timeout

Description	The timeout (in milliseconds) to wait for the host lookup script to run and return the host info.
Default Value	5000
Property	Amazon Kinesis – Data Streams connector
Attribute	<code>connector.kinesis.datastreams.accesskeyid</code>
Description	The Access Key ID from the AWS account.
Default Value	
Property	Amazon Kinesis – Data Streams connector
Attribute	<code>connector.kinesis.datastreams.secretaccesskey</code>
Description	The Secret Access Key ID from the AWS account.
Default Value	
Property	Python connector
Attribute	<code>connector.python.host</code>
Description	The default Python Pyro instance host address. NOTES: For <code>connector.python.host</code> , <code>connector.python.password</code> , <code>connector.python.port</code> , and <code>connector.python.serializertype</code> properties: <ul style="list-style-type: none">• If set in the <code>Streams.properties</code> file, these fields will be hidden in the Python connector and will be applied to the Python transform as well.• These default Streams Server connection properties will be applied at runtime.• These default Streams Server connection properties will override old Python connection settings.
Default Value	
Property	Python connector
Attribute	<code>connector.python.password</code>
Description	The default HMAC Key.
Default Value	
Property	Python connector
Attribute	<code>connector.python.port</code>
Description	The default Python Pyro host port.
Default Value	
Property	Python connector
Attribute	<code>connector.python.serializertype</code>
Description	The default Python serialization type. Possible values are serpent or pickle .
Default Value	
Property	Rserve connector

Attribute	connector.rserve.host
Description	<p>The default Rserve host address.</p> <p>NOTES:</p> <p>For <code>connector.rserve.host</code>, <code>connector.rserve.password</code>, <code>connector.rserve.port</code>, and <code>connector.rserve.userid</code> properties :</p> <ul style="list-style-type: none"> • If set in the <code>Streams.properties</code> file, these fields will be hidden in the Rserve connector and will be applied to the R transform as well. • These default Streams Server connection properties will be applied at runtime. • These default Streams Server connection properties will override old Rserve connection settings.
Default Value	
Property	Rserve connector
Attribute	connector.rserve.password
Description	The default password that will be used to connect to the Rserve service.
Default Value	
Property	Rserve connector
Attribute	connector.rserve.port
Description	The default Rserve host port.
Default Value	
Property	Rserve connector
Attribute	connector.rserve.userid
Description	The default user Id that will be used to connect to the Rserve service.
Default Value	
Property	REST Documentation
Attribute	documentation.enabled
Description	Enable or disable the OpenAPI Specification documentation for the REST interface.
Default Value	false
Property	REST
Attribute	error.default.message
Description	The error message that will be displayed instead of the actual error message. This is used to mask or hide error messages that may contain internal or sensitive details.
Default Value	
Property	File Upload
Attribute	file.upload.size.max.bytes
Description	Limit for files size to be uploaded through the web browser (i.e., workbooks, streams applications, streams data sources).

Default Value	30000000
Property	Log level
Attribute	<code>logger.level.file</code>
Description	Controls the level that is logged to file.
Default Value	INFO
Property	Server Metrics
Attribute	<code>metrics.authorization.level</code>
Description	Specifies the required authorization level to get server metrics. Available values are ANONYMOUS, VIEWER, DESIGNER, ADMINISTRATOR . NOTE: This property is case sensitive.
Default Value	ADMINISTRATOR
Property	Server Metrics
Attribute	<code>metrics.collection.rate</code>
Description	Specifies the rate at which metrics are collected in milliseconds.
Default Value	1000
Property	Server Metrics
Attribute	<code>metrics.file.flush.rate</code>
Description	Specifies how often metrics should be saved to disk in milliseconds. Only used if the <code>metrics.publisher.type</code> is set to FILE .
Default Value	10000
Property	Server Metrics
Attribute	<code>metrics.memory.queue.size</code>
Description	Specifies how many metric entries are stored in memory. When the number of metrics goes above the specifies value, the oldest value is removed to make room for the newest one (FIFO). Only used if the <code>metrics.publisher.type</code> is set to MEMORY .
Default Value	100
Property	Server Metrics
Attribute	<code>metrics.publisher.type</code>
Description	Specifies the current metric publisher that is used. Available values are NONE, MEMORY, FILE, EMAIL, INFLUX_DB, JDBC, KAFKA, KDB, MQTT, REST, TEXT .
Default Value	MEMORY
Property	Server Metrics
Attribute	<code>metrics.publisher.configuration</code>
Description	Specifies the id for which metric publisher configuration to use.
Default Value	
Property	Repository

Attribute	<code>repository.import.archived.applications</code>
Description	Allows to import all application backups from the <appdata>/CEP/Archive/. Refer to step 4 in the Migration to Streams Server 2025.1 from an Older Version section for more information.
Default Value	true
Property	Repository
Attribute	<code>repository.startup.filesystemcheck</code>
Description	If set to true , server runs on startup to verify the repository integrity and reports any of the following issues: <ul style="list-style-type: none"> • a deleted /HEAD file, • a modified /HEAD, • a modified /refs/heads/master file, • any file deleted inside /objects/ (e.g., /objects/94/443eec118fb8bb2021071896ff7d386a9c9518), • any file modified inside /objects/. <p>NOTE: There may be dangling files in the /objects/ directory or those that are not in use. These files are typically results of failed saves and/or sync conflicts. The check may or may not detect deleted or modified dangling files, but that is not critical.</p>
Default Value	false
Property	REST
Attribute	<code>rest.response.error.stacktrace.included</code>
Description	Include the error stacktrace in REST responses.
Default Value	false
Property	Server Downgrade
Attribute	<code>server.force_downgrade</code>
Description	The server normally refuses to start if it detects that the AppData directory has been used by a server with a newer version. This is because downgrading content and other AppData files is not supported and can cause irreversible issues. You can set this property to true to force the server to start anyway, but it is strongly recommended that you do not.
Default Value	false
Property	Server
Attribute	<code>server.id</code>
Description	Specifies an id for the current server. The value of this property will be part of each metric entry so that it can be tied to a specific server if a server cluster is used. If no value is specified, the MAC address of the localhost network will be attempted to be used to identify the server. If this is not possible, a UUID will be generated.
Default Value	
Property	Licensing
Attribute	<code>license.hwu.hosted</code>

Description	Boolean stating if you wish to use Managed or Local Altair Units licensing. Set to true if you wish to use managed licensing.
Default Value	false
Property	Licensing
Attribute	<code>license.hwu.hosted.authorization.username</code>
Description	Username to the Altair One account.
Default Value	
Property	Licensing
Attribute	<code>license.hwu.hosted.authorization.password</code>
Description	Password to the Altair One account.
Default Value	
Property	Licensing
Attribute	<code>license.hwu.hosted.authorization.token</code>
Description	An authorization token generated through the Altair One admin portal. Used to authorize a machine to the managed Altair Units system.
Default Value	
Property	Licensing
Attribute	<code>license.hwu.uri</code>
Description	The path where the License Server is running e.g., 6200@191.255.255.0 where the syntax is PORTNUMBER@HOST. If multiple servers are specified, use the ';' semicolon separator sign for Windows and the ':' colon separator sign for Linux.
	NOTE: If value is not set in the <code>Streams.properties</code> , the environment variable ALTAIR_LICENSE_PATH serves as the backup path and will be used.
Example	<p>For Windows: <code>license.hwu.uri=6200@192.168.5.51;6200@192.168.5.52</code></p> <p>For Linux: <code>license.hwu.uri=6200@192.168.5.51:6200@192.168.5.52</code></p>
Default Value	
Property	Licensing
Attribute	<code>license.hwu.use_client_timezone</code>
Description	Determines how the ALJDK should process the timezone details. If set to true , the ALJDK will process the timezone details sent by Panopticon client to the Panopticon server. If set to false , the Panopticon server timezone is used.
Default Value	true
Property	Licensing
Attribute	<code>license.mode</code>

Description	The license mode. Possible values are FILE or HWU . To use the Altair Units license, set this property to HWU .
Default Value	FILE
Property	Timeout Session
Attribute	<code>timeout.session.enabled</code>
Description	Boolean value stating if timeout functionality should be used or not.
Default Value	false
Property	Timeout Session
Attribute	<code>timeout.session.exception.delimiter</code>
Description	The delimiter to use for the usernames stated in the <code>timeout.session.exception.usernames</code> property.
Default Value	, (comma)
Property	Timeout Session
Attribute	<code>timeout.session.exception.usernames</code>
Description	Usernames that should be excluded from the timeout functionality. Separated by the delimiter stated in the <code>timeout.session.exception.delimiter</code> property.
Default Value	
Property	Timeout Session
Attribute	<code>timeout.session.minutes</code>
Description	Minutes of inactivity before a user session is terminated by logging out the user.
Default Value	480

03.2025

ABOUT PANOPTICON

For more information on Panopticon and other resources, go to <https://www.altair.com/panopticon>.