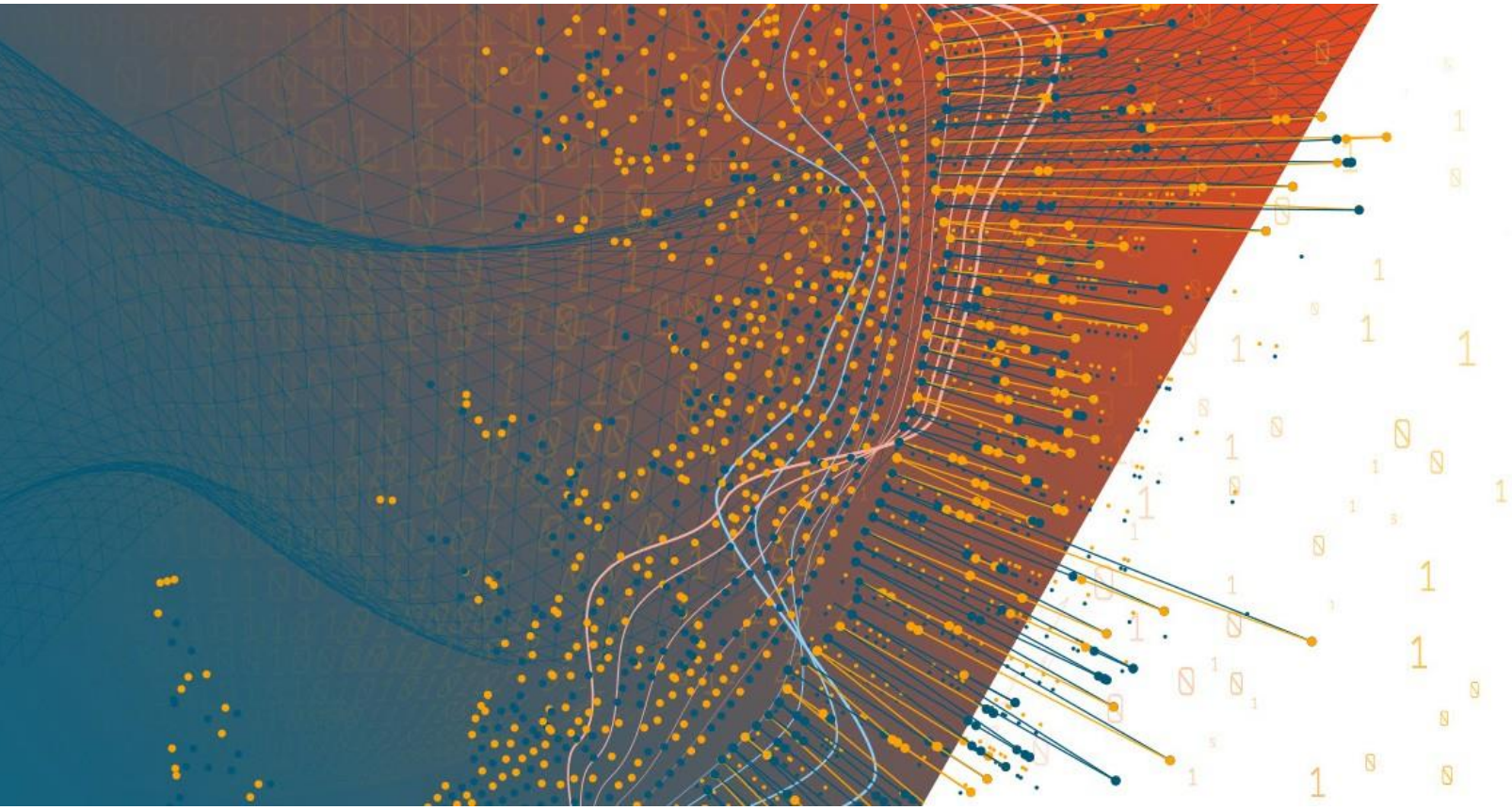


ALTAIR

Altair Panopticon™

PANOPTICON REAL TIME 2022.0 - INSTALLATION AND REFERENCE GUIDE

TABLE OF CONTENTS

[1] INTRODUCTION

OVERVIEW

Altair Panopticon™ Real Time supports the following data connectors:

- **General Connectivity**: MS Excel, Text, XML, JSON, Restful Web services, JDBC Databases
- **Big Data**: Cassandra, Elasticsearch 6.x, Elasticsearch 7.x, KsqlDB, Livy Spark, MongoDB, Splunk
- **Event Processing**: Kx kdb+Tick, OneMarketData OneTick CEP, Tibco Streambase, Tibco LiveView,Panopticon Streams
- **Messaging Streaming**: Apache/Confluent Kafka, JMS (e.g., ActiveMQ), Solace, RabbitMQ, WebSocket, JMX, Google Cloud Pub/Sub, MQTT
- **Tick Data**: OneMarketData OneTick, OneTick Cloud, Kx kdb+, InfluxDB
- **Custom code data connections, transforms, and ML model scoring**: Python, R, and REST service calls

In addition, Panopticon Real Time includes a Panopticon bundle file of example workbooks (**Examples.exz**). To use the example workbooks, the bundle file must be imported into the server after the server installation.

NOTE	
	• Beginning with version 17.1, MS Access, Valo, Apache Qpid, Valo Streaming, Ultra Messaging Streams, and OData connectors are deprecated. • Beginning with version 16.2, DataDirect based connectors, along with Vertica, are deprecated. The Database connector or JDBC Database connector should be used. • Existing workbooks will continue to operate, but connectivity will need to be migrated for subsequent releases.

SYSTEM REQUIREMENTS

Panopticon Real Time is supported on these operating systems:

- Linux which include the following distributions and versions:
  - RHEL/CentOS 7 or higher
  - Debian 8 or higher
  - Ubuntu 14 or higher
  - Fedora 21 or higher
- Windows 10 (64-bit) – For Development Environments Only
- Windows Server 2012 (64-bit)

- ❑ Windows Server 2016 (64-bit)

Panopticon Real Time also requires:

- ❑ Oracle Java SE 8, Oracle Java SE 11, Open JDK 8, and Open JDK 11 are supported after installing the dependency files that are distributed with Panopticon Real Time.

NOTE	• Unzip the contents of the dependency package file provided by Panopticon into the `TOMCAT_HOME/lib` folder to be able to run Altair Panopticon software on JRE 8 and Open JDK 8. • Please refer to Java documentation about setting up the JAVA_HOME environment variable in your system.

- ❑ Apache Tomcat 9.0.x

NOTE	When running on Windows instead of Linux, it is recommended to use the zip distribution of Apache Tomcat for Windows rather than the Windows Service Installer. This is because the zip distribution will let you run Apache Tomcat without any dependency on the Windows service manager, and management of the Apache Tomcat server will conform more with how it is done on Linux.

NOTE

Starting with Tomcat 9, Debian Linux implements a security policy which puts a harder default restriction on which folders a Tomcat 9 web application can write to.

The change is described in full detail here:

https://salsa.debian.org/java-team/tomcat9/-/commit/3ca5cbdc2f9704703419263 54f210dff032fc5f3

Quoting from the release notes:

- Tomcat is sandboxed by `systemd` and only has write access to the following directories:

Directory	Actual Directory
`/var/lib/tomcat9/conf/Catalina`	`/etc/tomcat9/Catalina`
`/var/lib/tomcat9/logs`	`/var/log/tomcat9`
`/var/lib/tomcat9/webapps`	
`/var/lib/tomcat9/work`	`/var/cache/tomcat9`

- If write access to other directories is required, override the service settings. This is done by creating an `override.conf` file in `/etc/systemd/system/tomcat9.service.d/` containing:

  [Service]
  ReadWritePaths=/path/to/the/directory/

  Ensure to restart the service afterward with:

  - systemctl daemon-reload

	○ systemctl restart tomcat9

Panopticon Real Time is supported for deployment on the following cloud providers:

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

Containerized deployment with Docker Linux containers is also supported.

Supported browsers include the latest version of:

- ❑ Google Chrome
- ❑ Safari

NOTE	• Panopticon Real Time requires administrative privileges during installation. Administrative privileges are not required after installation is complete. • Panopticon Real Time does not support Tomcat 7.x, Tomcat 8.0.x, or Tomcat 8.5.x.

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

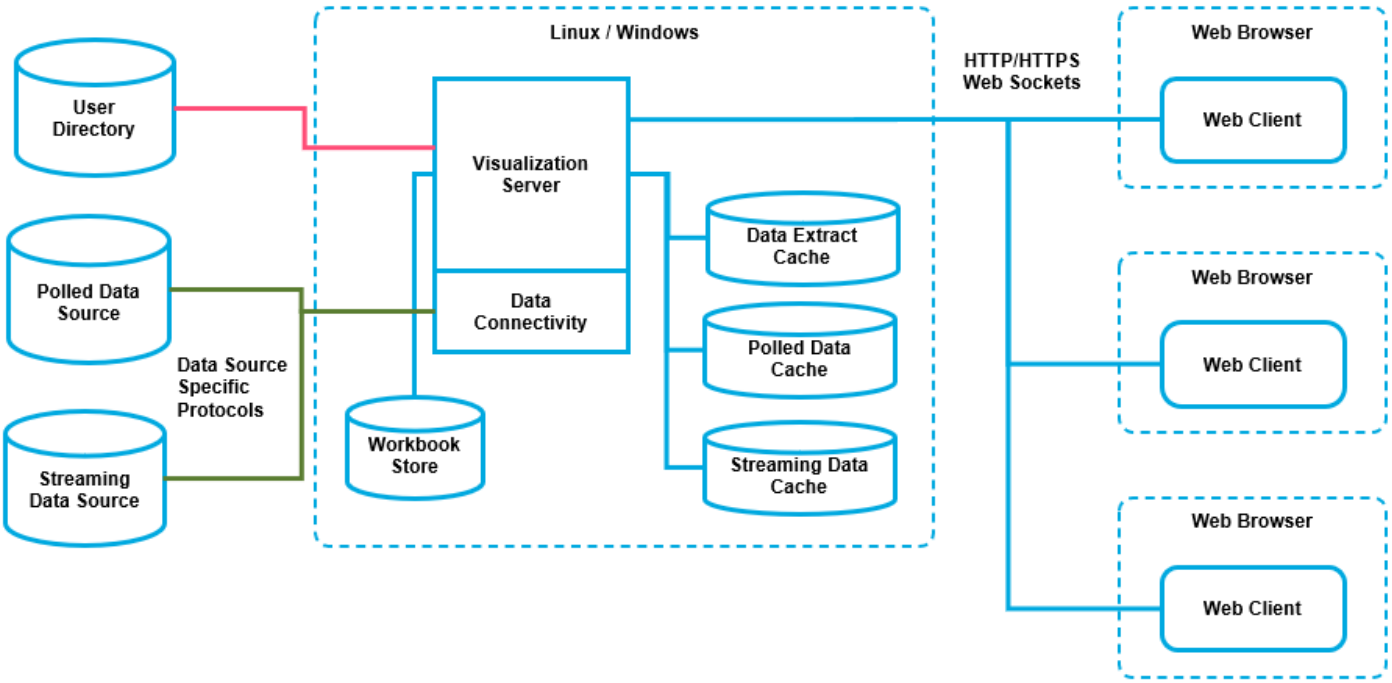
- 4 x Quad Core CPU Or Equivalent (Hyper Threaded to 32 Cores/Threads)
- 32GB RAM
- 4GB Disk (Available)
- In Memory Caching limited to available Server RAM

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] SETUP

PANOPTICON REAL TIME DEPLOYMENT MODEL



Full scale Panopticon Real Time deployment

Panopticon Real Time is deployed and hosted on an internal network. The server can be accessed from internally and/or externally from the internet. Upon allowing access to the server from the internet it is recommended to have a proxy and firewall in front of the server.

Panopticon Real Time exposes web services from both a SOAP interface and a REST interface. These interfaces are used by the Web client but can also be used to execute functionality directly on the server such as by batch jobs.

Workbook access is secured through the underlying application Panopticon Real Time security model, such as authentication and configuration of directories in Tomcat.

Furthermore, Panopticon Real Time is capable of the following features:

- Single Sign On (SSO) Support through SAML
- JDBC / JNDI Data Sources
- JMX Monitoring

Server Platforms

Panopticon Real Time consist of multiple components such as the following:

Components	Description
Panopticon Real Time	Formerly named Panopticon Visualization Server, responsible for managing all the published workbooks and all the resources that go with them. It is also responsible for authorization, data connections, transformations, scheduled tasks, report generation, alerting etc.
Panopticon Streams	Processing of data streams for real-time visualization in dashboards.
Web Client	The web client is a graphical interface for administration of the server and for creation and design of dashboard applications.

See Installation for more information.

ENVIRONMENT PROMOTION OPTIONS

You may need to set up multiple environments for Altair Panopticon. For example, you may wish to set up your system to support one of the following migration paths:

- Staging → Production
- Development → User Acceptance Testing → Production

The primary complication when promoting workbooks between environments is access to data repositories, since you may wish to use separate data repositories for each environment. If you require different data repositories in each environment, use JNDI or global parameters. These abstract the location of the data repository from the workbook.

INSTALLATION

This document provides instructions on how to install Panopticon Real Time on Linux or Windows.

NOTE	If you need to upgrade your previously installed Panopticon Real Time, proceed to the Upgrade section.

Setting Up Panopticon Real Time on Linux

Follow the steps and guidelines below to install Panopticon Real Time on Linux.

Steps:

1. Extract the contents of `AltairPanopticonVisualizationServerWAR_<version number>.zip` file to a new location.

   This zip file will contain the following folder and files:

   - pcli-java folder
   - tomcat-users_example.xml
   - start_Python_connectivity.sh
   - start_Python_connectivity.bat
   - pyro.py
   - panopticon.xml
   - panopticon.war
   - OpenJDK11Dependencies_README.txt
   - OpenJDK11Dependencies.zip
   - Examples.exz
   - CustomMessageParserExample.zip
   - Elastic_5X_Dependencies.zip
   - Elastic_6X_Dependencies.zip
   - Elastic_7X_Dependencies.zip
   - Panopticon Web Authoring Quick Start Guide
   - Panopticon Web Authoring Guide
   - Panopticon Real Time Installation and Reference Guide
   - User_License.rtf

NOTE

To support Python Transform, the following files are included in the installation zip file:

- start_Python_connectivity.sh
- start_Python_connectivity.bat
- pyro.py

Refer to Python Integration for more information.

2. Create the `AppData` folder (i.e., `/usr/share/vizserverdata`) and ensure that the user account Local Service running Tomcat has read/write and execute permissions to this folder.
3. In the Tomcat config folder (`/tomcat_home/conf/Catalina/localhost`) create the `panopticon.xml` file with the following information:

```
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/panopticon">
      <Environment name="PanopticonAppData" override="false"
type="java.lang.String" value="/usr/share/vizserverdata" />
</Context>
```

NOTE

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

Variable	New Value
PanopticonAppData	`/usr/share/panopticondata`

- If the directory path is set in both an environment variable and in the `panopticon.xml` file, the value set in the XML file will take precedence.
- Starting with 21.2, the `DatawatchVDDAppData` is replaced with `PanopticonAppData` as the specifier for the Panopticon application data directory. You can still use `DatawatchVDDAppData` as a fallback, but going forward, `PanopticonAppData` should be used**.**

4. 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 Panopticon Real Time for more information.
   - Hosted Altair Units license. Refer to Using the Hosted Altair Units License in Panopticon Real Time for more information.
5. Copy the `panopticon.war` file into the Tomcat `webapps` folder (`/tomcat_home/webapps`).
6. For a basic installation using the Tomcat inbuilt XML file user directory, copy the provided tomcat-users_example.xml and overwrite the existing `tomcat-users.xml` file which is available in the Tomcat config folder `(/tomcat_home/conf)`.

The provided `tomcat-users_example.xml` contains the following roles and users:

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

NOTE

In Panopticon 2020.0 and onwards, the `Administrators.txt` and `AdministratorGroup.txt` files are no longer used to authorize administrator users. The function provided by these files has been replaced by a set of properties in `Panopticon.properties`:

```
access.default.roles=VIEWER
access.administrator.groups=admin
access.designer.groups=designer
access.viewer.groups=
```

The `access.default.roles` property defines the default roles assigned to any user accessing the server, defaulting to VIEWER. The administration (`access.administrator.groups` property) and content creation (`access.designer.groups` property) on the server are mapped by default to groups named "admin" and "designer".

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., `(/tomcat_home/temp)` is available.
- You can opt to choose a different `temp` folder with the CATALINA_TMPDIR environment variable. For example:

Variable	Value
CATALINA_TMPDIR	`/tomcat_home/dev/temp`

7. Start Tomcat to deploy the `panopticon.war` file.

   The server initializes the `AppData` directory with an empty content repository and empty subdirectories for other types of data. The `Panopticon.properties` file is created with the default server properties.

8. Increase the Java heap size of Tomcat.
9. You can also opt to install Java data connector's dependencies.
10. You should now be able to log on to Panopticon Real Time using the following:

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

For example:

`http://localhost:8080/panopticon`

The more advanced configuration options are also discussed in this document.

Setting Up Panopticon Real Time on Windows

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

Steps:

1. Extract the contents of `AltairPanopticonVisualizationServerWAR_<version number>.zip` file to a new location.

   This zip file will contain the following folder and files:

   - `pcli-java` folder
   - tomcat-users_example.xml
   - start_Python_connectivity.sh
   - start_Python_connectivity.bat
   - pyro.py
   - panopticon.xml
   - panopticon.war
   - OpenJDK11Dependencies_README.txt
   - OpenJDK11Dependencies.zip
   - Examples.exz
   - CustomMessageParserExample.zip
   - Elastic_5X_Dependencies.zip
   - Elastic_6X_Dependencies.zip
   - Elastic_7X_Dependencies.zip
   - Panopticon Web Authoring Quick Start Guide
   - Panopticon Web Authoring Guide
   - Panopticon Real Time Installation and Reference Guide
   - User_License.rtf

NOTE

To support Python Transform, the following files are included in the installation zip file:

- start_Python_connectivity.sh
- start_Python_connectivity.bat
- pyro.py

Refer to Python Integration for more information.

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

   Example: `c:\vizserverdata`

3. 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, or
   - Altair Units license. Refer to Using Altair Units License in Panopticon Real Time for more information.
   - Hosted Altair Units license. Refer to Using the Hosted Altair Units License in Panopticon Real Time for more information.

4. Copy the extracted `panopticon.xml` file into the Tomcat config folder (`\Apache Software Foundation\Tomcat 9.0\conf\Catalina\localhost`). This file contains the following information:

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

NOTE

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

Variable	New Value
PanopticonAppData	`c:\panopticondata`

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

Starting with 21.2, the `DatawatchVDDAppData` is replaced with `PanopticonAppData` as the specifier for the Panopticon application data directory. You can still use `DatawatchVDDAppData` as a fallback, but going forward, `PanopticonAppData` should be used.

5. Copy the `panopticon.war` file into the Tomcat `webapps` folder (`\Apache Software Foundation\Tomcat 9.0\webapps`).

6. For a basic install using the Tomcat inbuilt XML file user directory, copy the provided `tomcat-users_example.xml` and overwrite the existing `tomcat-users.xml` file which is available in the Tomcat config folder (`\Apache Software Foundation\Tomcat 9.0\conf`).

   The provided `tomcat-users_example.xml` contains the following roles and users:

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

NOTE

In Panopticon 2020.0 and onwards, the `Administrators.txt` and `AdministratorGroup.txt` files are no longer used to authorize administrator users. The function provided by these files has been replaced by a set of properties in `Panopticon.properties`:

```
access.administrator.groups=admin
access.default.roles=VIEWER
access.designer.groups=designer
access.viewer.groups=user
```

The `access.default.roles` property defines the default roles assigned to any user accessing the server, defaulting to VIEWER. The administration (`access.administrator.groups` property) and content creation (`access.designer.groups` property) on the server are mapped by default to groups named "admin" and "designer".

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

7. You can also opt to install Java data connector's dependencies, and JDBC driver JAR files as required.

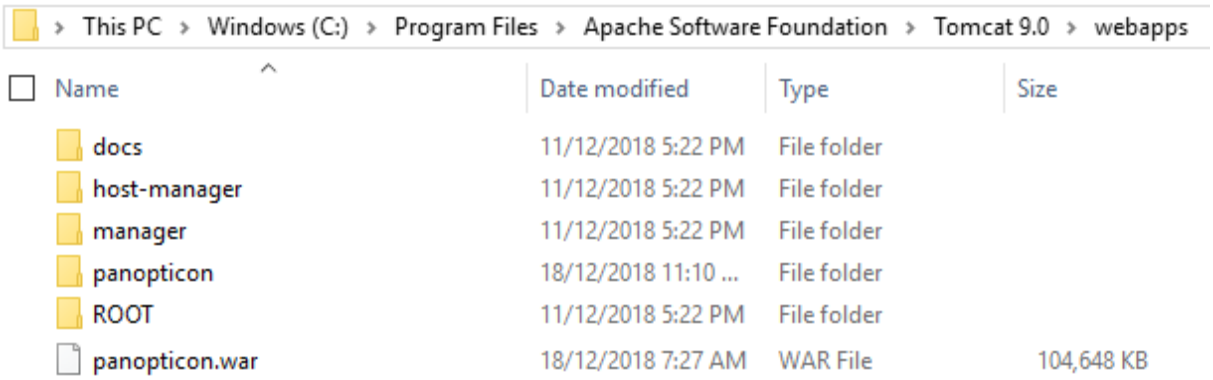
IMPORTANT

- Before proceeding to step 8, ensure the Tomcat `temp` folder (e.g., `(\Apache Software Foundation\Tomcat 9.0\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`

8. Start Tomcat to deploy the `.war` file.

   The `panopticon` folder is extracted in the Tomcat `webapps` folder:



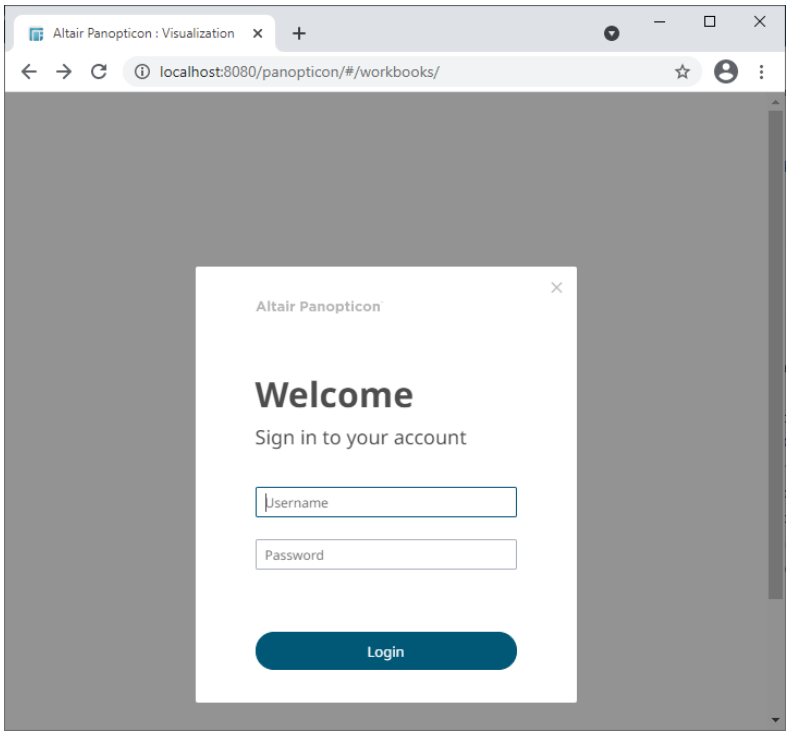
The server initializes the `AppData` directory with an empty content repository and empty subdirectories for other types of data. The `Panopticon.properties` file is created with the default server properties.

9. Increase the Java heap size of Tomcat.
10. You should now be able to log on to Panopticon Real Time using the following:

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

    For example:

    `http://localhost:8080/panopticon`

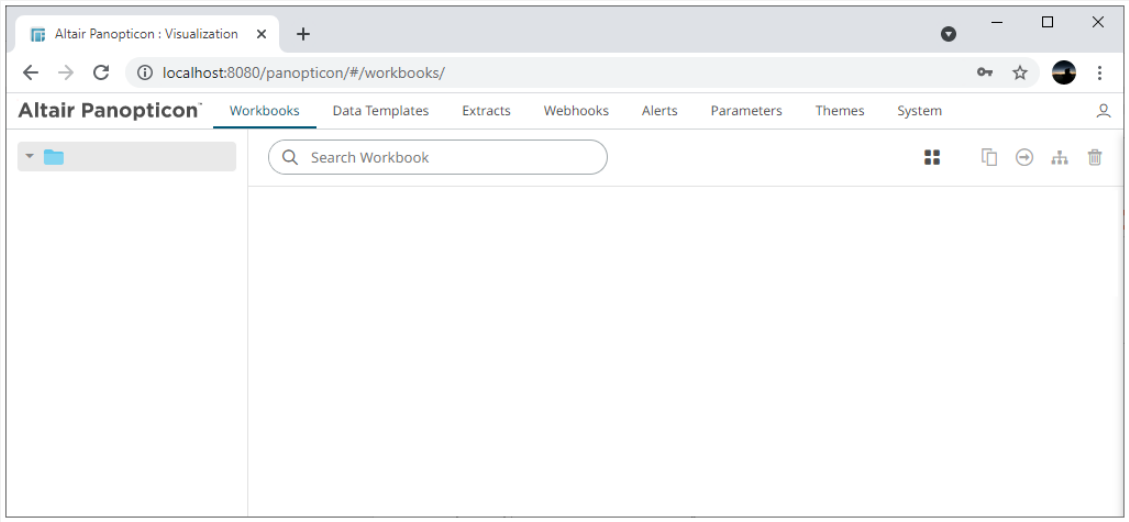


The more advanced configuration options are also discussed in this document.

NOTE

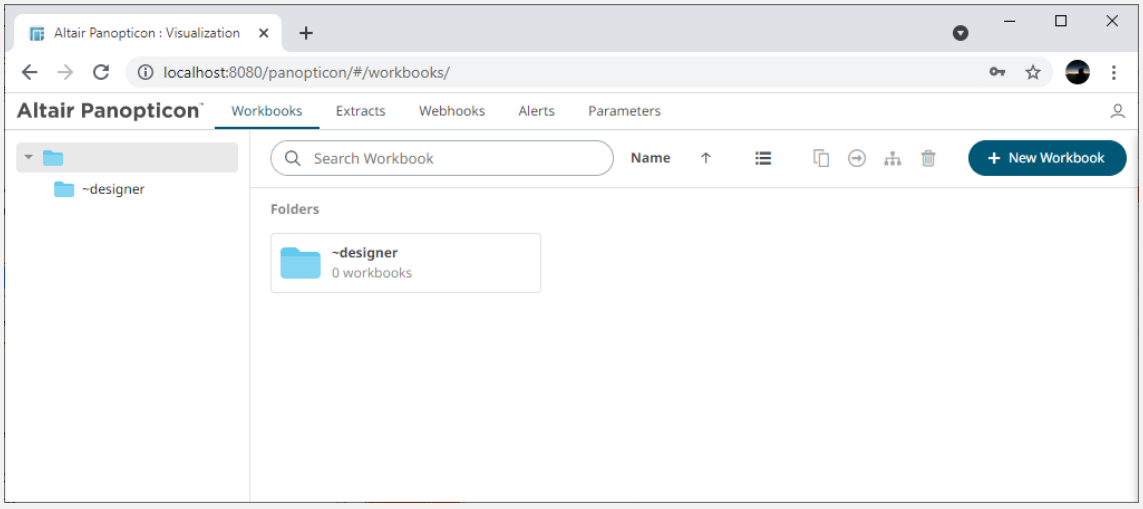
Panopticon Real Time supports different user roles. To have full access to all the services, the user is required to have an ADMINISTRATOR and DESIGNER roles.

For example, logging on using the ADMINISTRATOR role added in step 6 (i.e., admin/admin), will display:



All of the available user specific folders in the authentication method used are displayed.

However, logging on with a DESIGNER role (i.e., designer/designer) will only display:



The allowed features available for the DESIGNER role is extensively discussed in the Web Authoring Guide.

For more information on how to set up the user groups and map them to the user roles, refer to Supported Roles in Panopticon Real Time.

Open JDK 11+ Dependencies

`The AltairPanopticonVisualizationServerWAR_<version number>.zip` file includes `OpenJDK11Dependencies.zip` which contains necessary dependencies for running Altair Panopticon software on Open JDK 11 and up.

The overview, installation, and list of the contents of `OpenJDK11Dependencies.zip` are provided and discussed in the `OpenJDK11Dependencies_README.txt` file.

Background

In Java 9, a number of Java EE modules were marked for deprecation, and subsequently removed completely from Java 11.

With missing Java EE dependencies, the typical exceptions would include `NoClassDefFoundError` exceptions being thrown for `javax/xml/bind` classes.

```
Exception in thread "main" java.lang.NoClassDefFoundError:
javax/xml/bind/JAXBException
    at monitor.Main.main(Main.java:27)
Caused by: java.lang.ClassNotFoundException: javax.xml.bind.JAXBException
    at
java.base/jdk.internal.loader.BuiltinClassLoader.loadClass(BuiltinClassLoader.java:582
)
    at
java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(ClassLoaders.java:
185)
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:496)
    ... 1 more
```

In order to support deployment on either Java 1.8 or Open JDK 11+, we have packaged the necessary Java EE dependencies separately for simple installation in Tomcat.

Installation

Do the following to make the dependencies available to the JVM and the Altair Panopticon server:

1. Stop Tomcat.
2. Unzip the contents of `OpenJDK11Dependencies.zip` into the `TOMCAT_HOME/lib` folder.
3. Start Tomcat.

Zip File Content

- Jakarta XML Binding API (jakarta.xml.bind-api), version 2.3.2
  - jakarta.xml.bind-api-2.3.2.jar
  - jakarta.activation-api-1.2.1.jar
- JAXB Runtime (jaxb-runtime), version 2.3.2
  - jakarta.xml.bind-api-2.3.2.jar
  - txw2-2.3.2.jar
  - istack-commons-runtime-3.0.8.jar
  - jakarta.activation-api-1.2.1.jar

- stax-ex-1.8.1.jar
- jakarta.activation-api-1.2.1.jar
- jakarta.xml.bind-api-2.3.2.jar
- FastInfoset-1.2.16.jar
- jakarta.activation-api-1.2.1.jar

- Jakarta SOAP Implementation (saaj-impl), version 1.5.1
  - saaj-impl-1.5.1.jar
  - jakarta.xml.bind-api-2.3.2.jar
  - jakarta.activation-api-1.2.1.jar
  - jakarta.xml.soap-api-1.4.1.jar
  - mimepull-1.9.11.jar
  - stax-ex-1.8.1.jar
- Java API for XML Web Services (jaxws-api), version 2.3.1
  - jaxws-api-2.3.1.jar
  - jaxb-api-2.3.1.jar
  - javax.activation-api-1.2.0.jar
  - javax.xml.soap-api-1.4.0.jar
  - javax.annotation-api-1.3.2.jar

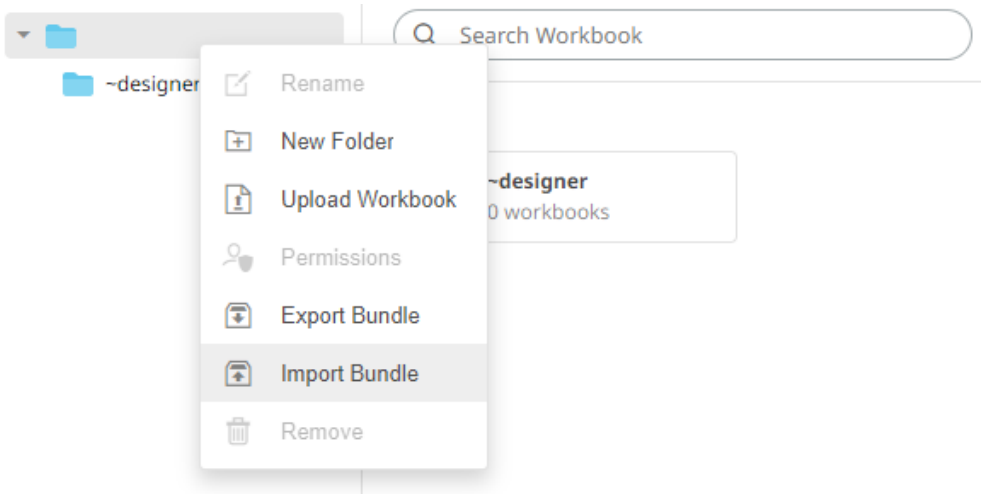
Importing the Bundle of Example Workbooks

`The AltairPanopticonVisualizationServerWAR_<version number>.zip` file includes the bundle file of the example workbooks and their associated data files (`Examples.exz`).

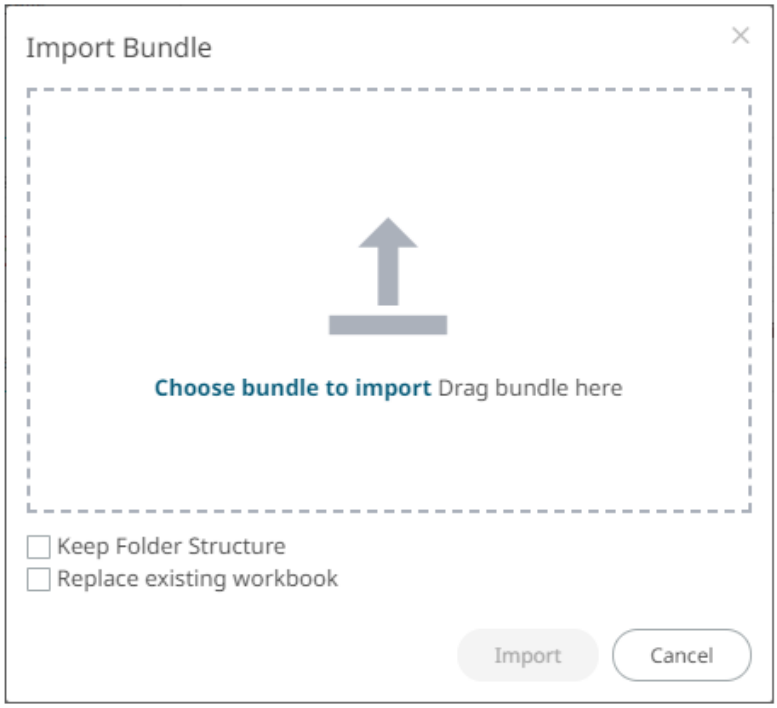
Follow the instructions below to import this bundle to Panopticon Real Time.

Steps:

1. On the *Workbooks and Folders Summary* page, right-click on a folder and select **Import Bundle** on the context menu.



The *Import Bundle* dialog displays.



2. To import the bundle, you can either:
   - drag it from your desktop and drop on 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 on the dialog box.

3. Check the **Keep Folder Structure** box.

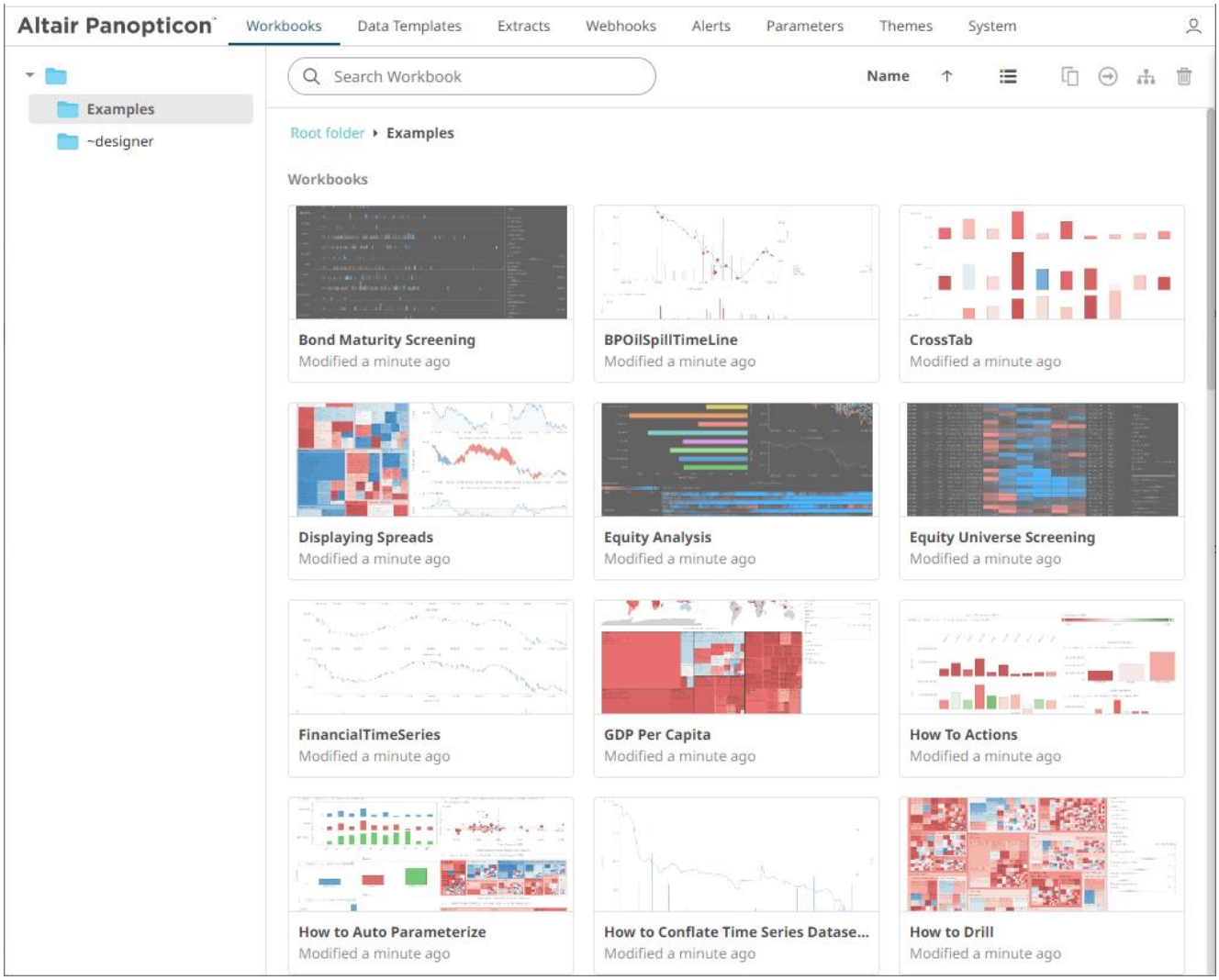
   This means the exported folder structure is maintained when uploading the bundle. If the folders do not exist on the server, they will be created.

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

5. Click 

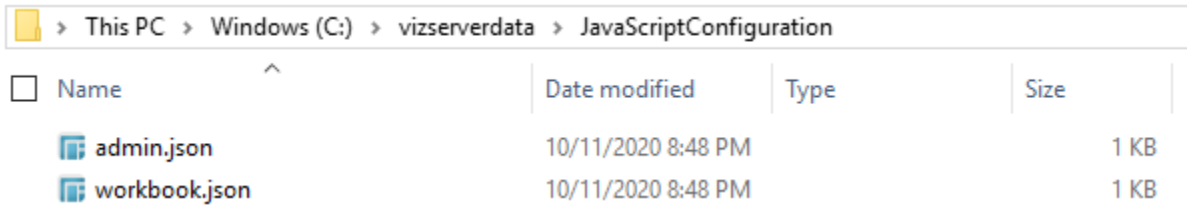
.

   The example workbooks that you can view and explore are imported.


Altair Panopticon
Workbooks
Data Templates
Extracts
Webhooks
Alerts
Parameters
Themes
System
Search Workbook
Name
Examples
~designer
Root folder ▸ Examples
Workbooks
Bond Maturity Screening
Modified a minute ago
BPOilSpillTimeLine
Modified a minute ago
CrossTab
Modified a minute ago
Displaying Spreads
Modified a minute ago
Equity Analysis
Modified a minute ago
Equity Universe Screening
Modified a minute ago
FinancialTimeSeries
Modified a minute ago
GDP Per Capita
Modified a minute ago
How To Actions
Modified a minute ago
How to Auto Parameterize
Modified a minute ago
How to Conflate Time Series Datase...
Modified a minute ago
How to Drill
Modified a minute ago

CONFIGURATION OF THE CLIENT PROPERTIES

Starting with version 2020.1, Panopticon Real Time generates JSON configuration files in the `JavaScriptConfiguration` directory of the `AppData` folder (i.e., **c:\vizserverdata**).



NOTE

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

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

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

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

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

Object/Name	`automaticReconnectOnServerDisconnect`
Description	If set to **true**, 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	`baseUrl`
Description	Location of Panopticon Real Time.
Default Value	**"."**
Required	Yes
Object/Name	`dataLoading.transport`

Description	Controls 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	`hideAuthenticationButton`
Description	Boolean. Hides the login and logout buttons.
Default Value	**false**
Required	No
Object/Name	`localization.defaultLocale`
Description	Locale used if the browser locale is not supported, or if the `useBrowserLocale` is set to **false**.
Default Value	**"en-US"**
Required	No
Object/Name	`localization.fallbackLocale`
Description	Locale used if a resource string is missing from the locale in use. Should be specified if `localization.defaultLocale` is specified.
Default Value	value of `localization.defaultLocale`
Required	No
Object/Name	`localization.useBrowserLocale`
Description	Boolean. If set to **true**, the browser navigator.language, navigator.userLanguage on IE11, controls the localization of the UI. Not all languages are supported.
Default Value	**true**
Required	No
Object/Name	`localizationOverride`
Description	Nested object with resource strings per language. Used to customize resource strings.
Default Value	
Required	No
Object/Name	`logLevel`
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	`workbookUrl`
Description	Location of the workbook application. **NOTE:** The workbookUrl property should reflect the actual location of the workbook application, but it doesn't rename or change the location of said application.

Default Value	**"workbook"**
Required	Yes

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

```
{
  "baseUrl" : "..",
  "forceClientSelectionHandling" : true,
  "startUrl" : "../",
  "subscriptionCompression" : true,
  "dataLoading" : {
    "transport" : "websocket"
  },
  "webGlEnabled" : true,
  "pdfMultiplePagesEnabled" : true
}
```

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

Object/Name	`baseUrl`
Description	Location of Panopticon Real Time.
Default Value	**".."**
Required	Yes
Object/Name	`localization.useBrowserLocale`
Description	Boolean. If set to **true**, the browser `navigator.language`, `navigator.userLanguage` on IE11, controls the localization of the UI. Not all languages are supported.
Default Value	**true**
Required	No
Object/Name	`localization.defaultLocale`
Description	Locale used if the browser locale is not supported, or if `useBrowserLocale` is set to **false**.
Default Value	**"en-US"**
Required	No
Object/Name	`localization.fallbackLocale`
Description	Locale used if a resource string is missing from the locale in use. Should be specified if `localization.defaultLocale` is specified.
Default Value	value of `localization.defaultLocale`
Required	No
Object/Name	`localizationOverride`
Description	Nested object with resource strings per language. Used to customize resource strings.

Default Value	
Required	No
Object/Name	`logLevel`
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	`disableExternalHelpText`
Description	Disables the browser dev console splash screen.
Default Value	**false**
Required	No
Object/Name	`theme`
Description	Name of the default theme for all workbooks. Per default, the first theme available is picked as the default theme.
Default Value	
Required	No
Object/Name	`allowOrigin`
Description	A comma separated list of allow origins, used by the Panopticon POST message API.
Default Value	
Required	No
Object/Name	`automaticReconnectOnServerDisconnect`
Description	If set to **true**, 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	`alwaysHideNonInteractiveParametersInDialog`
Description	Hides parameters that are not interactive when displaying the interactive parameter dialog.
Default Value	**false**
Required	No
Object/Name	`enableDevicePixelRatioCanvasScaling`
Description	Enable or disable handling of device pixel ratio for 2D visualizations.
Default Value	**true**
Required	No
Object/Name	`staleStateTimeout`

Description	Time (in milliseconds) that controls how fast the "stale data" overlay should be rendered on top of visualizations. Only applies to those interactions that doesn't show "data loading" animation. A value equal to or less than zero will disable the stale data overlay.
Default Value	**150**
Required	No
Object/Name	`preloadDetailsPopup`
Description	Data to be shown in the *Details* pop-up is preloaded as part of the visualization data request. If set to **false**, then the details data will be loaded on demand.
Default Value	**true**
Required	No
Object/Name	`forceClientSelectionHandling`
Description	If set to **true**, then selection handling in the visualizations will be performed by the client. If set to **false**, then the server will calculate the selection.
Default Value	**false**
Required	No
Object/Name	`subscriptionCompression`
Description	Controls if data query strings longer than 2048 chars should be compressed by the web client before sending them to the server.
Default Value	**false**
Required	No
Object/Name	`pdfMultiplePagesEnabled`
Description	Controls the visibility of the "Create multiple pages..." checkbox in the ad hoc PDF report dialog.
Default Value	**true**
Required	No
Object/Name	`startUrl`
Description	URL used by the **Back** button, and by the dialog for unexpected errors. If this property is removed and `useBrowserHistoryToNavigateBack` is **false**, then the **Back** button will not be displayed.
Default Value	
Required	No
Object/Name	`useBrowserHistoryToNavigateBack`
Description	If set to **true,** then the **Back** button will be visible and the button will execute `window.history.back()` when pressed. This setting takes precedence over having a configured `startUrl`.
Default Value	**true**
Required	No
Object/Name	`hideThemeSelection`

Description	Controls the visibility of the theme drop down.
Default Value	**false**
Required	No
Object/Name	`dataLoading.transport`
Description	Controls the which transport should be used when loading data from the server and receiving notifications. Valid values are "**websocket**" and "**long-polling**". If configured to "**websocket**", but the WebSocket connection fails, then the web client will automatically fall back on "**long-polling**".
Default Value	**"websocket"**
Required	No
Object/Name	`dataLoading.connectTimeout`
Description	Controls the timeout used, in milliseconds, when opening the data loading connection to the server.
Default Value	**10000**
Required	No
Object/Name	`preventVisualizationMouseWheelDefaultEvents`
Description	Prevents the browser default action when using the mouse wheel over a visualization. Useful in an embed scenario if the hosting web page is scrolled when the user tries to zoom in a visualization using the mouse wheel.
Default Value	**false**
Required	No
Object/Name	`webGlEnabled`
Description	Enables the use of WebGL in visualizations that supports it.
Default Value	**true**
Required	No
Object/Name	`maxClipboardLength`
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	`selectionInDetailsPopup`
Description	Enables/disables selection data in the visualization details popup. Primary use case for this setting is to disable it on a server level.
Default Value	**true**
Required	**No**
Object/Name	`showAlertsButton`
Description	Controls the visibility of the **Alerts** workbook button.
Default Value	**true**

Required	**No**
Object/Name	`showBookmarksButton`
Description	Controls the visibility of the **Bookmarks** workbook button.
Default Value	**true**
Required	**No**
Object/Name	`showCopyDashboardImageButton`
Description	Controls the visibility of the **Copy Image** workbook button.
Default Value	**true**
Required	**No**
Object/Name	`showPdfExportButton`
Description	Controls the visibility of the **Create PDF Report** workbook button.
Default Value	**true**
Required	**No**
Object/Name	`showRefreshDataButton`
Description	Controls the visibility of the **Refresh** workbook button.
Default Value	**true**
Required	**No**
Object/Name	`showPauseRealtimeButton`
Description	Controls the visibility of the **Pause Real-Time** workbook button.
Default Value	**true**
Required	**No**
Object/Name	`pluginDenyList`
Description	Array of plugin IDs, used to block the specified dashboard parts and visualizations.
Default Value	**empty array**
Required	No
Object/Name	`pluginAllowList`
Description	Array of plugin IDs, used to allow only the specified dashboard parts and visualizations. The default value, an empty array, allows all plugins**.**
Default Value	**empty array**
Required	No

NOTE

- With the new application configuration files, the `workbook/config.js` inside the extracted war file is no longer valid.
- 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.

LICENSING

Licensing within Panopticon Real Time supports the following license types:

- a volume-based XML file (named **PanopticonLicense.xml**) which is used to store all license information for a specific customer, must be copied to the designated `AppData` folder (i.e., **c:\vizserverdata**)

NOTE

Starting with 21.2, the newly issued volume-based license file is named `PanopticonLicense.xml`. For customers with the `DatawatchLicense.xml` file, it can still be used but it is strongly recommended to rename it to `PanopticonLicense.xml`.

- Altair Units license which is available in Altair License Server you are connected to (local or over the network)
- Hosted Altair Units license

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

NOTE

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

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/.

Using Altair Units License in Panopticon Real Time

Before using the Altair Units license type in Panopticon Real Time, it is required to configure certain properties in the `Panopticon.properties` file located in the `AppData` folder or `c:\vizserverdata`:

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.operating.system`
Description	The operating system where Panopticon Real Time is installed. Possible values are: **WIN_X86**, **WIN_X64**, **MAC**, **LINUX_X64**, or **LINUX_ARM64**. **NOTE:** If the Java bitness (e.g., 32-bit) is different from the operating system (e.g., 64-bit), it is recommended to add the Java bitness in this property (e.g., WIN_X86).
Default Value	
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. **NOTES:** If value is not set in the `Panopticon.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.hwu.version`
Description	Value must match the license version found in the Altair Units license file.
Default Value	**19.0**
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.operating.system=WIN_X64
license.hwu.uri=6200@192.168.5.51;6200@192.168.5.52
license.hwu.version=19.0
license.mode=HWU
```

NOTE

- Panopticon Real Time supports different user roles which check out different numbers of Altair Units.

Role	Altair Units License Draw
Viewer	2
Designer	2 21 when designing a workbook
Administrator	2

- Alerts and scheduled tasks are leveled towards each other. Regardless of the number of alerts or scheduled tasks a user creates, only two Altair Units licenses will be checked out.
- These units are separate from the units that are checked out for a user of the server. For example, if a user is logged on to the server (two units) and starts an alert (two units), a total of four units are checked out. If the user then starts two more alerts and a scheduled task, the total number of checked out units will still be four. If the user logs out without shutting off any alerts, two units will remain checked out.
- Two products (e.g., Panopticon Real Time and Panopticon Streams) or two instances of one product, must not be configured to use Altair unit licensing if they run on the same Tomcat.

Using the Hosted Altair Units License in Panopticon Real Time

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

Before using the Altair Units license type in Panopticon Real Time, it is required to configure certain properties in the `Panopticon.properties` file located in the `AppData` folder or `c:\vizserverdata`:

Property	Licensing
Attribute	`license.hwu.hosted`
Description	Boolean stating if you wish to use Hosted or Local Altair Units licensing. Set to **true** if you wish to use hosted 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 Hosted Altair Units system.
Default Value	

NOTE

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

```
license.hwu.hosted=true
license.mode=HWU
license.hwu.operating.system= WIN_X64
authentication.required=true
license.hwu.uri=6200@localhost
license.hwu.version=20.0
```

- Add the Panopticon application to your Altair One account.

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

Option 1

If you wish to generate the authorization token through Altair One:

1. Log on to Altair One (https://admin.altairone.com) then navigate to **User Profile -> View My Authorized Machines -> Generate Auth Code** (up in the right corner).
2. Paste the generated code into the `license.hwu.hosted.authorization.token` property in the `Panopticon.properties` file.
3. Start the server.

Option 2

To eliminate token generation on your own:

1. Enter your Altair One credentials into the `license.hwu.hosted.authorization.username` and `license.hwu.hosted.authorization.password` properties in the `Panopticon.properties` file.
2. 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 Hosted Altair Units system. • A working Internet connection is required to use Hosted Altair Units licensing. • 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.

Configuration Properties

Encoding

The default encoding of the JVM is the same as the system it is running on. It is recommended to configure your Java and Apache Tomcat to use the UTF-encoding. This is achieved by setting the property `file.encoding` to **UTF-8**.

There are several ways to configure the property and one method is to create a `setenv` file in your Apache Tomcat `bin` folder:

- `setenv.bat` for Windows
- `setenv.sh` for Linux

  The following operating systems should contain the following information in order to use the UTF-8 encoding:

  For Windows:

  ```
  set JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=UTF-8
  ```

  For Linux:

  ```
  JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=UTF-8"
  ```

- Restart the Apache Tomcat to save the changes.

PROXY

A proxy is a server or software running on a server that acts as an intermediary for requests from clients seeking resources from other servers. Instead of using a proxy, you can use a load balancer.

It is recommended to use a proxy when setting up Panopticon Real Time. There are a variety of proxies available. One of the most commonly used proxies is Apache HTTP Server with the proxy module. Refer to the section below on how to setup an Apache HTTP Server with Proxy functionality.

Apache HTTP Server

This section describes the steps on how to install and configure an Apache Proxy. The guide expects that the Apache HTTP Server is being setup for the first time. Please note that the installation steps might vary depending on your environment. These steps cover how to install and configure an Apache HTTP Server with proxy support for Microsoft Windows.

1. Download the Apache HTTP Server from the official webpage:
   https://httpd.apache.org/download.cgi
2. Unzip and copy the files to a folder.
3. Configure the proxy by opening the `httpd.conf` file in the `conf` folder.
4. Update the `SRVROOT` variable. The value must be updated to the file location of the Apache HTTP server.

```
Define SRVROOT "/Path/To/Apache"
ServerRoot "${SRVROOT}"
```

5. Modules are required to be loaded to make the Apache HTTP Server into a proxy. Add the following lines in the `httpd.conf` file.

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
LoadModule rewrite_module modules/mod_rewrite.so
```

NOTE	In the `httpd.conf` file: • Ensure that the line `Include conf/extra/httpd-vhosts.conf` is uncommented. • Replace Listen 80 with Listen 10088.

6. Configure the logic for the proxy and how requests should be passed. The Virtual host config should look like this and be added to the `httpd-vhosts.conf` file in the `conf/extra` folder:

```
<VirtualHost :10088>
  ServerAdmin webmaster@localhost
  ProxyPreserveHost On
  ProxyPass /panopticon http://localhost:8080/panopticon
  ProxyPassReverse /panopticon http://localhost:8080/panopticon
  ProxyPass /panopticon/server/ws http://localhost:8080/panopticon/server/ws
  ServerName localhost:8080
  RewriteEngine on
  RewriteCond %{HTTP:UPGRADE} ^websocket$ [NC]
  RewriteCond %{HTTP:CONNECTION} Upgrade [NC]
  RewriteRule . ws://localhost:8080%{REQUEST_URI} [P]
</VirtualHost>
```

7. The Apache HTTP Server can be started when all the configurations are in place. This is done by running the `httpd` script or application in the Apache `bin` folder.

LOAD BALANCER

A load balancer is a server used to distribute the workload across multiple computer resources. A load balancer allows you to scale the system to max and optimize the resource use and throughput, and at the same time minimize the response time. A load balancer can also be used to ensure that the system will still be available, even during downtime on a computer resource.

Very much like proxies, there are a variety of load balancers available. The only requirement Panopticon Real Time has on the load balancer is that it supports persistence or stickiness. This means that the proxy will establish a user session and ensure that the user continues to use the same computer resource.

Stickiness are mainly implemented in two means: **Cookies** or **URL encoding**. These two alternatives will be used to determine which route the user will continue to take in the load balancer. The rest of this section will cover how to implement stickiness with cookies.

Sticky load balancer that are using cookies are normally using session tokens. Due to this, it is required to configure Panopticon to use session tokens. This is done by updating the following property to **SESSION** in the `Panopticon.properties` file: `authentication.token.persistence`.

```
authentication.token.persistence=SESSION
```

> **IMPORTANT** After modifying the property value to SESSION, ensure to clear the `AppData/Token` folder before starting the server.

For details on how to configure multiple servers to run in a cluster and synchronize content between them, see Server Cluster Configuration.

Apache HTTP Server

The following section describes the steps on how to install and configure an Apache Load Balancer. The guide expects that the Apache HTTP Server is being setup for the first time. Please note that the installation steps might vary depending on your environment. These steps cover how to install and configure an Apache HTTP Server with proxy support for Microsoft Windows.

1. Download the Apache HTTP Server from the official webpage: https://httpd.apache.org/download.cgi
2. Unzip and copy the files to a folder.
3. Configure the proxy by opening the `httpd.conf` file in the `conf` folder.
4. Update the `SRVROOT` variable. The value must be updated to the file location of the Apache HTTP server.

```
Define SRVROOT "/Path/To/Apache"
ServerRoot "${SRVROOT}"
```

5. Modules are required to be loaded to make the Apache HTTP Server into a load balancer. Add or uncomment the following lines in the `httpd.conf` file.

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule headers_module modules/mod_headers.so
LoadModule lbmethod_byrequests_module
modules/mod_lbmethod_byrequests.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule slotmem_shm_module modules/mod_slotmem_shm.so
```

6. Configure the logic for the load balancer and how requests should be passed.

   In the following example, we have configured the load balancer to listen to port **10080** and to use two balancer members (**Route 1** and **Route 2**). The example will also set a session cookie named **ROUTEID**. The cookie contains the route that the user took and will continue to use throughout the active session.

```
<VirtualHost *:10080>
  ServerAdmin webmaster@localhost
  ProxyPreserveHost On

  Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/"
env=BALANCER_ROUTE_CHANGED

  <Proxy "balancer://panopticoncluster">
    BalancerMember "http://localhost:8080/panopticon" route=1
    BalancerMember "http://localhost:8081/panopticon" route=2
    ProxySet stickysession=ROUTEID
  </Proxy>

  ProxyPass         /panopticon balancer://panopticoncluster
  ProxyPassReverse  /panopticon balancer://panopticoncluster

  ServerName localhost:8080
</VirtualHost>
```

7. The Apache HTTP Server can be started when all the configurations are in place. This is done by running the `httpd` script or application in the Apache `bin` folder.

MULTIPLE INSTANCES

Multiple instances of Panopticon Real Time can be deployed onto a single machine.

The common usage models for multiple instances are:

- Multi-tenant deployments, providing separate **Sand boxes** for each tenant
- Multi environments (Development, Test, Production)
- Regression Testing
- To deploy multiple servers, the WAR and corresponding configuration file must be updated to have a unique name.

BACKUP

Panopticon Real Time consists of:

- Software Installation & Server Configuration
- License
- Usage Configuration
- Published Workbooks
- Data
- Caches

Backup is typically divided into the above sections, with published workbook backup occurring on a regular basis from the configured `AppData` (i.e., `c:\vizserverdata`) folder.

DATA ACCESS AND CACHING

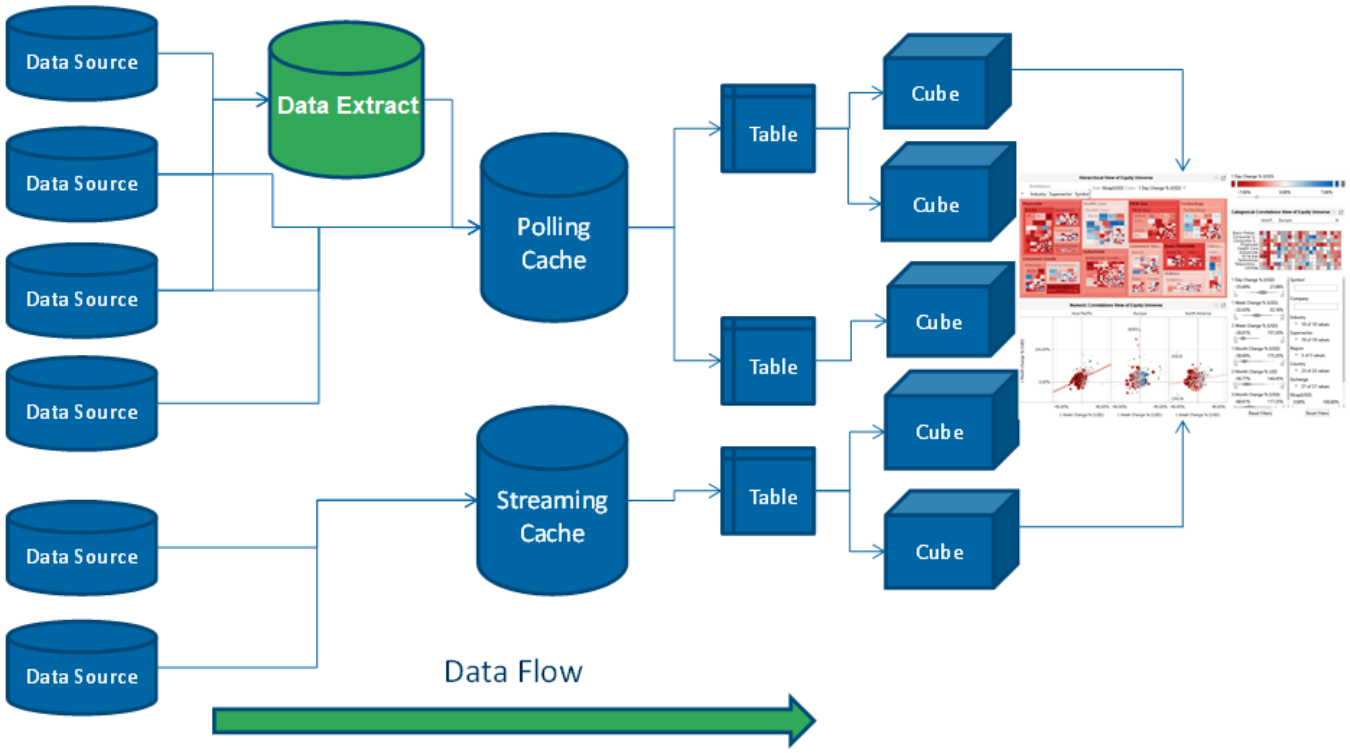
Panopticon assumes in general that data is never at rest and too big to be simply loaded into memory. The data can either be subscribed against or polled (automatically refreshed on a defined period).

This means either:

- Load Subset of Data in Memory
- Load Summary and Parameterized Detail Views
- ROLAP (Dynamically explore datasets)

Consequently, for direct access, Panopticon is only as fast as the underlying data platform, or the refreshing of result set caches.

When data is not changing on a timely basis, such as a daily updated data warehouse, there is the additional option of retrieving data into a data extract.



Consequently:

- Only required data is retrieved. Majority of the data stays in the underlying data sources
- Typically aggregated, conflated, filtered data is retrieved
- Behind each dashboard part (visualization) is a micro-cube
- Each cube is designed for streaming real time updates
- Behind each cube is a real-time data table (also powering filters)
- Behind each data table is a resultset cache
- Behind the cache is the underlying data repository
- Caches can be loaded on the fly, or pre-loaded on a periodic basis
- All caching is optional
- Consequently, data access is either:
    - **Work Directly** against underlying sources (either Exploratory Analysis (ROLAP), Or Pre-Defined Parameterised Views)
    - **Extract & Cache Data** from slower underlying sources. And query this data extract locally. (Similar to competitor products).

In reality, usage is typically **Hybrid**. Based on the characteristics of the underlying data, you choose whether to extract and load, or query directly.

This is to cater for real world data landscapes, where different data has different data retrieval latency characteristics, and different timeliness; and where there is too much data to simply load all into memory.

CONTENT REPOSITORY

Previous versions of the server stored workbook files in the `Workbooks` subdirectory of the application data directory (i.e., `c:\vizserverdata\Workbooks`), and older versions of those workbooks in the `Archive` subdirectory (i.e., `c:\vizserverdata\Archive`). Starting with the 2020.0 release, workbooks are now stored in a version tracking repository, located in the `.repository` subdirectory (i.e., `c:\vizserverdata\.repository`).

The repository also stores other types of content that was previously stored in separate subdirectories of the application data directory, such as data files and bookmarks.

The new workbook repository tracks changes to content, folders, and permissions. It also makes it possible for multiple servers to synchronize their repositories, so you only have to make changes to a workbook on one node in a server cluster, and it will automatically propagate to the other servers [see section on cluster configuration].

The first time you start the 2020.0 server, it will initialize an empty repository in the application data directory. If you are upgrading an existing install you have the option of migrating content from the old application data directory into the new repository [see section on content migration].

For the list of properties that control the repository behavior, refer to those starting with **'repository.'** in the Properties: Panopticon section.

SERVER CLUSTER CONFIGURATION

When you have muliple servers running, you can set them up so they synchronize content between them. The servers will use an internal protocol over http(s) to propagate changes and make sure their content is the same.

The cluster component discovers the other servers and the topology that connects them, and can use various methods to do so. The cluster component also identifies one of the running servers as the **leader**, the others are **followers**.

The leader-follower relationship determines how content is synchronized. A follower will immediately push any local change to the leader, for example, when you save a workbook after editing it. On the other hand, a follower periodically polls the leader for changes. This means the leader has the "latest" version of the content, whereas a follower may lag behind by a few seconds. The leader is also special if there are conflicting changes, for example, if two users edits and saves the same dashboard. In this case, the leader's changes always wins.

The REST services, that the servers call to synchronize content, expose potentially sensitive information such as data tables and data source settings. They are protected by token validation just as other services on the server, and only accepts special "server" tokens that are never issued to users. A server can only get a token from another server if they have both been configured with the same shared secret. That said, the calls are not encrypted, so if you connect two servers over the internet, you will want to use https.

Even though the content synchronization makes it easier to run a set of servers as a cluster behind a load balancer, you still need to use sticky sessions (session affinity). The server requires that a single user stays with the same server instance for the duration of a session.

There are four different cluster modes:

- **None** - Each server is completely stand-alone, and nothing will be synchronized. This is the default, and no further configuration is needed.
- **Fixed** - One server is the permanent leader. The other servers will synchronize with it if it is up. If the leader goes down, the followers will log the problem, but will continue to run basically as stand-alone servers. When the leader comes back up, they will start synchronizing again.

In practice, the fixed mode has a single point of failure. Because the followers connect through the leader, even if they keep running, their content will not be synchronized, and conflicts become more likely the more their content diverge.

To configure fixed mode, set `cluster.shared.secret` to the same non-empty string on all, set `cluster.mode` to **FIXED** on all, and then set `cluster.fixed.leader` to the URL of the leader on the followers only (leave it blank on the leader).

The leader URL should be the path to the web application, for example `http://panoserver:8080/panopticon/`. It needs to identify the leader server, and be resolvable on the network that the followers run on. If you use a load balancer, you cannot use the externally exposed URL, as it always needs to resolve to the leader server. If the leader server is dynamically assigned an IP, you need to take extra steps to assign it with a URL that does not change.

- **Bully** - The server with the lowest ID (lexicographically) of the running servers is chosen as leader, and if it goes down a new leader is automatically appointed.

When a new server joins a bully cluster, it needs to discover the current list of members and their IDs. To do this, it tries to contact any running server from a list of known servers, called the boot servers. If any one of them answers, it replies with the current members and leader. If none of them answers, it starts as the single member of the cluster if it is one of the boot servers, or refuses to start if not.

In a sense, the bully mode is more flexible than the fixed mode, since it eliminates the single point of failure. As long as one server is still running, there will be a leader, so synchronization will happen. In another sense, it's less flexible as you need to provide more non-changing URLs, one for each server.

To configure the bully mode, set `cluster.shared.secret` (see above), set `cluster.mode` to **BULLY** on all, set `cluster.bully.id` to a unique ID string for each server (lower ID has higher leader priority), set `cluster.bully.bind` on each to the URL on which the other servers can reach it, and `cluster.bully.boot` to a comma-separated list of known server URLs.

An example bully configuration with three servers:

On server #1:

```
cluster.shared.secret=supersecretpassword
cluster.mode=BULLY
cluster.bully.id=panopticon-1
cluster.bully.bind=http://192.168.0.10/panopticon
cluster.bully.boot=\
http://192.168.0.10/panopticon,\
http://192.168.0.11/panopticon
```

On server #2:

```
cluster.shared.secret=supersecretpassword
cluster.mode=BULLY
cluster.bully.id=panopticon-2
cluster.bully.bind=http://192.168.0.11/panopticon
cluster.bully.boot=\
http://192.168.0.10/panopticon,\
http://192.168.0.11/panopticon
```

On server #3:

```
cluster.shared.secret=supersecretpassword
cluster.mode=BULLY
cluster.bully.id=panopticon-3
cluster.bully.bind=http://192.168.0.12/panopticon
cluster.bully.boot=\
http://192.168.0.10/panopticon,\
http://192.168.0.11/panopticon
```

Note that only servers #1 and #2 are boot servers, and that only id and bind differ between servers. With this configuration, servers #1 and #2 can be started in any order, but at least one of them must be up before #3 starts. On the other hand, you can add server #3 without #1 and #2 knowing about it up front, so non-boot servers can be useful in auto-scaling scenarios.

One caveat with non-boot servers is that if all the boot servers go down, a non-boot server will become the leader. If a new server joins, or a boot server rejoins, there is now way for them to see this, and you will end up with two separate clusters.

- **Kubernetes** - The servers discover each other through the Kubernetes API Server, and the one whose pod has the lowest name (lexicographically) is chosen as leader. Each server periodically refreshes this information, so if the list of available pods change, they adapt.

  To call the Kubernetes API, the server needs to know the address of the API Server and also have valid credentials. By default, the address is passed into the pod via Kubernetes downward API as environment variables `KUBERNETES_SERVICE_HOST/PORT`, and the credentials are mounted to `/var/run/secrets/kubernetes.io/serviceaccount/`, and the server will use these, so no extra configuration is needed.

  The server discovers the other servers (pods) with a Kubernetes label selector. You can use any label and any selector for this, e.g., give each pod the metadata label "app" with value "panopticon" and use the selector "app=panopticon". The server will assume that all pods returned by the query are standard Panopticon servers.

  You also need to tell each server what its own pod name is, so it can tell if it's supposed to be a leader or follower, and avoid calling itself. You can use the Kubernetes downward API to pass this in: use valueFrom, fieldRef and fieldPath "metadata.name" (see example below).

  To configure the Kubernetes mode, set `cluster.shared.secret` (see above), set `cluster.mode` to **KUBERNETES**, set `cluster.kubernetes.id` to the pod's name, set `cluster.kubernetes.label_selector` to the pod selector, and `cluster.kubernetes.peer_path` to the web application path.

  If the pod that runs the Panopticon server container also runs other containers, the first container will be used. If this is not the case, you can set `cluster.kubernetes.container_name` to the name of the container that runs the Panopticon server.

  Example yaml snippet:

```
template:
  metadata:
    labels:
      app: panopticon
  spec:
    containers:
      ...
      env:
        - name: CLUSTER_SHARED_SECRET
          value: supersecretpassword
        - name: CLUSTER_MODE
          value: KUBERNETES
        - name: CLUSTER_KUBERNETES_ID
          valueFrom:
            fieldRef:
              fieldPath metadata.name
        - name: CLUSTER_KUBERNETES_LABEL_SELECTOR
          value: app=panopticon
        - name: CLUSTER_KUBERNETES_PEER_PATH
          value: panopticon/
```

SYNCHRONIZING TOKENS

When a user has authenticated successfully with a server, a token is issued that is passed back and forth in a cookie. These tokens may be long-lived with a default expiration time of seven days and normally automatically reissued. So when the server is regularly used, the user will rarely need to log in again. Similarly, API tokens never expire.

Normally, a token issued by one server is only valid on that server. The server keeps track of issued tokens and validates each incoming token against its stored tokens. Furthermore, tokens are revoked when an administrator logs out a user, and the token is removed from the server's list.

If there are multiple servers that is being used as a cluster with a load balancer in front, they should be configured to have synchronized tokens. Even if the load balancer usess sticky sessions as it should, a token is typically valid for a longer time than a sessions lasts, and the user should not have to log in again just because a new session is directed to a different server than last time.

Token synchronization uses a different mechanism from repository synchronization. The repository stores content with its change history, and there are scenarios where the user may want to synchronize one and not the other. For example, there may be one QA server and one production server then use a common login.

Tokens are synchronized through the cluster shared store. By default, this is just a subdirectory `<appdata>/shared/` which is not synchronized. The store also keeps other types of non-content data that the user may want to synchronize between servers.

To enable token synchronization:

- Change the property `cluster.shared.store.type` from its default **PRIVATE_DIRECTORY** to **SHARED_DIRECTORY**.
- Set `cluster.shared.store.shared_directory.path` to a location that is accessible from all servers. On Kubernetes, this would typically be a volume that you mount on a path in the container.
- Ensure all servers use the same cookie name in `authentication.token.cookie`, or a cookie issued by one server would not be visible to another.
- Ensure all servers (i.e., validating server and the one that issued the token) have the same `authentication.token.secret`. This secret is used to sign and validate tokens.

EXPORTING LEGACY FILES

Starting with Panopticon 2020.0, new server installation will no longer include `Workbooks` and `Data` folders in the `AppData` folder. To recreate these folders and export workbooks and their associated data files, `GroupAccessRestrictions`, and parameters that are stored inside the repository, you can either:

- do a POST call to http://[host]:[port]/[serverappname]/server/rest/server/export/legacy/files?replaceExistingDataFiles=true&replaceExistingWorkbooks=true

  Example:

  `http://localhost:8080/panopticon/server/rest/server/export/legacy/files?replaceExistingDataFiles=true&replaceExistingWorkbooks=true`

- or perform the following steps

1. Set the `documentation.enabled` property in `Panopticon.properties` to **true**.

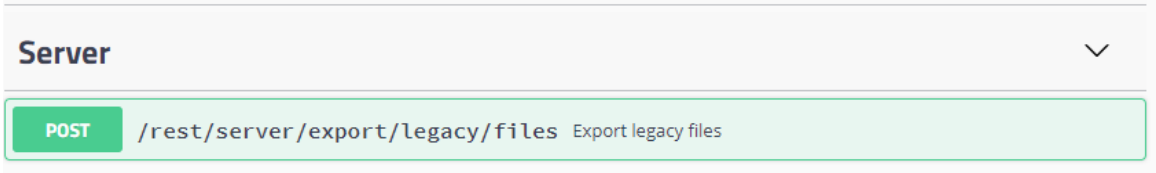
```
documentation.enabled=true
```

2. Restart the server.
3. Access the REST API to call the `/rest/server/export/legacy/files` service using this URL:

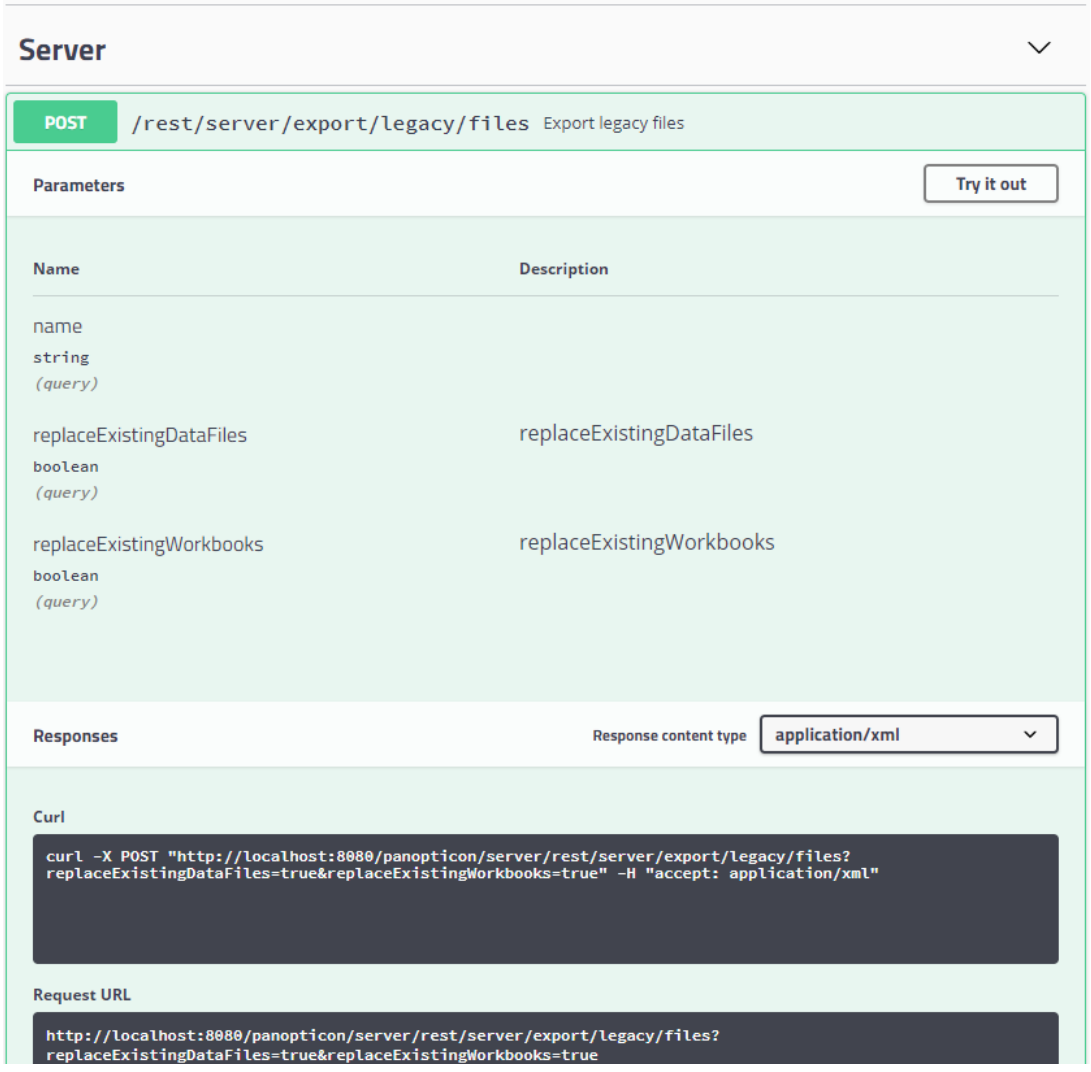
   `http://[host]:[port]/[serverappname]/server/swagger-ui.html`

   Example: `http://localhost:8080/panopticon/server/swagger-ui.html`

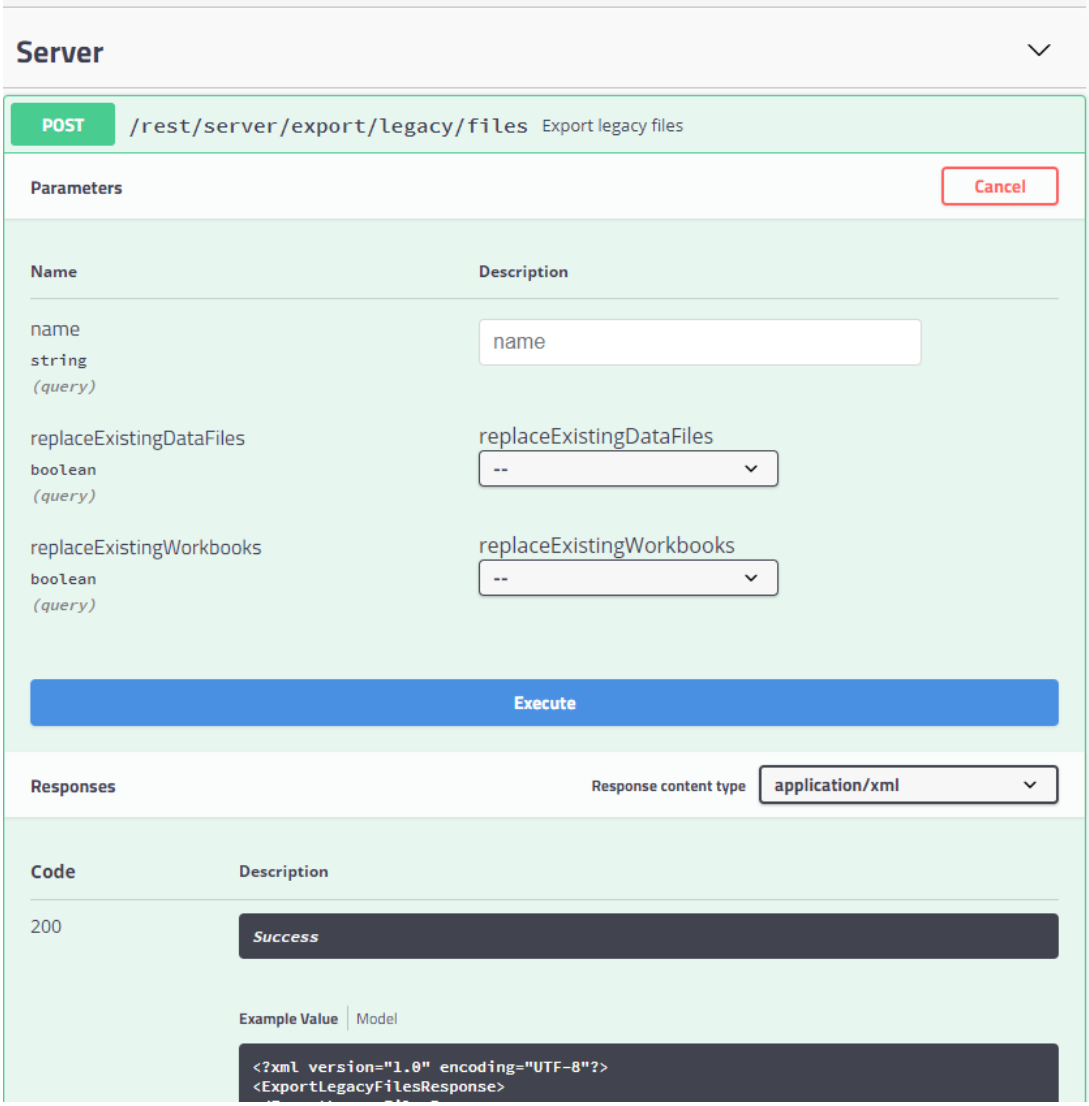
   The Panopticon REST API page displays.
4. Expand the *Server* section. The *Export Legacy Files* service displays.



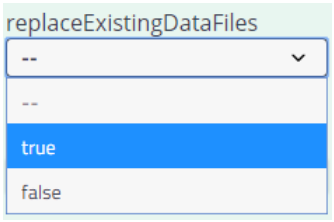
5. Again, expand this section to display the service settings.



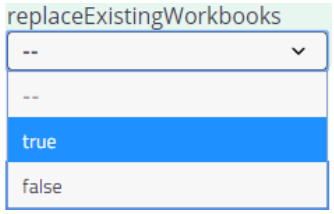
6. Click **Try it out.** The parameters and variables that you can set are displayed.



7. Set the `replaceExistingDataFiles` to **true** to replace existing files in the `Data` folder during the export.



8. Set the `replaceExistingWorkbooks` to **true** to replace existing files in the `Workbooks` folder during the export.



9. Click **Execute**.

   If successful, the `Data` and `Workbooks` folders are recreated in the `AppData` folder with the exported workbooks, data files, parameters, and GroupAccessRestrictions.

UPGRADING PANOPTICON REAL TIME

The server stores all its content in the `AppData` (e.g., `c:\vizserverdata`) directory. Ensure that you back up this directory frequently. You can always revert the server to an earlier state by restoring the entire `AppData` directory from a backup.

In general, a newer server will use an `AppData` from an older server, with exceptions for some types of content that you may need to migrate manually. In contrast, the server will refuse to start if you point an older server to an `AppData` that has been used with a newer server.

> **NOTE** Two servers should never share the same `AppData` directory.

It is recommended that you try out the new server version with your existing content before you decide to upgrade. The best way to do this is on a dedicated server machine, or at a minimum on a separate Tomcat instance. You should use a separate `AppData` directory for the new server while you are evaluating it --- if for some reason you decide to wait with the upgrade, you will not be able to use the new version's `AppData` on the old server, even if you have not made any changes.

Summary of steps:

1. Make a full backup of the old server's `AppData` directory.
2. Configure a new Tomcat, ideally on a separate machine.
3. Put the new server WAR file in the new Tomcat's `webapps` directory.
4. Copy the entire `AppData` from the old server to the new server.
5. Make sure the new server is pointed at the new `AppData` copy.
6. Read below for how content is migrated between versions.
7. Start the new server and then review its log file carefully to see if there were any warnings related to migration.
8. Try out the new server. Keep in mind that content created or changed on the new server cannot be moved back to the old server if you decide to roll back.
9. If you have continued to use the old server in parallel with testing, you should copy over the `AppData` again to make sure you have the latest content.

Content Migration

The format of the `AppData` changes between versions. Certain types of content may be moved to a different location with the `AppData` e.g., bookmarks from loose files into the repository for version tracking and cluster replication. Frequently, new features may be added to the content which may include changes to the content models, e.g., workbooks with new property. Typically a newer server version will do this migration behind the scenes when it first starts up, and any issues found will be logged with at least a warning level.

NOTE After starting a new server version for the first time, check the log file for warnings. It is much easier to address these issues immediately than later on, for example, if you wish to redo a one-time migration step.

By default, the server will check for content in an old location in `AppData` and do a one-time migration of anything it finds when it starts. Typically, the server will copy old content from loose files into the repository. This type of migration is controlled through the `repository.migrate.<type>.path` properties in `Panopticon.properties`.

For example, in versions 21.1 and older, the server stored workbook bookmarks as loose files in the `AppData/Bookmarks/` directory. Version 21.2 stores them inside the repository in `AppData /.repository/` instead. The property `repository.migrate.bookmarks.path` defaults to **Bookmarks** which is relative to `AppData` and therefore points to the 21.1 bookmark files. When the 21.2 server starts, this can happen:

- There are no bookmarks in the repository, but are available in `AppData/Bookmarks/` or to some other location you have set the property to. The server will do a one-time migration and move them into the repository. The result of this will be logged. The old bookmark files are left in the old location but will no longer be used.
- There are bookmarks in the repository. You may have created them on the new server, or the migration has run already. In this case, it does not matter if the old location has bookmarks or not, and the server will log a warning that it will not run a second migration. To get rid of the warning, simply blank out the property value.
- There are no bookmarks in either location, but the property is still set. This would be the default on a new server. In this case, you will get a false migration warning because the server cannot find `AppData/Bookmarks/`. Again, just blank out the property value to get rid of the warning.

NOTE

- Because the server will not migrate a type of content (e.g., bookmarks) if that type of content is already in the repository, you will need to delete the repository to run the migration again. The easiest way is to start over with the `AppData` from the old server.
- After the content has been migrated, the original files are left in the old location in `AppData`. They are no longer used, to clean up, you may want to delete them after you have checked the logs for any migration issues.

There are some types of content that have changed so much between releases that they cannot be automatically migrated like this.

Workbooks and their history in version 17 and older were stored separately in `AppData/Workbooks/` and `AppData/Archive/`. You can use the `repository.migrate.workbooks.path` and `repository.migrate.archive.path` properties to migrate them, but we do not default these to the old locations (e.g., you may not want to migrate the entire history), and for clarity you need to use absolute paths if you set them.

Data files used with text data sources can now optionally be stored in the repository so they can be bundled with the workbook, and also replicated to other servers in a cluster. You can still have data files in `AppData/Data/`, so old workbooks will continue to work on new servers, but old files are not automatically migrated into the repository.

Permissions on workbook folders were in version 17 and older stored in XML files in the `AppData/Workbooks/` subtree. The permissions model has changed completely since then, so they are not automatically migrated. To migrate permissions from version 17, you need to:

1. Use PCLI `convertpermissions` to create a permissions template which, as closely as possible, reflects the old permissions. This is a single JSON file which the new server can apply to its repository.

2. Review the generated permissions template in a text editor to make sure it is correct.
3. Point the `repository.startup.apply.permissions.path` to the template file and start the server. You can control how the template is applied with the properties `repository.startup.apply.permissions.clean` and `repository.startup.apply.permissions.create`.
4. Clear the properties after the server has started, or they will be applied on each startup overwriting changes you make.

> **NOTE** In version 17 and earlier, the files `AppData/Administrators.txt` and `AppData/AdministratorsGroup.txt` were used to give users administrative permissions. Currently, with the normal permissions model, these files are no longer used.

Authentication tokens are server-specific. They will only work on a new server if it has the same `authentication.token.secret` value as the old server that created them. In addition, a normal user token is also stored as cookie in the user's browser, and will only get sent to a new server if it has the same URL as the old server. For these reasons, tokens are not automatically migrated and users will have to log in again.

The exception to token migration is API tokens. In version 21.1 and older, these were stored in `AppData/APIToken/`. In 21.2, all tokens, including the API tokens, are stored in the shared cluster storage (even if you only have one single server), by default in `AppData/shared/tokens/`. See also the section on Synchronizing Tokens. If the server finds API tokens in the old location, they will be migrated on startup.

[3] AUTHENTICATION

INTRODUCTION

Panopticon Real Time provides multiple approaches on 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**, **WINDOWS**.	**BASIC**
authentication.domain	The default domain information for user authentication.	

Supported Roles in Panopticon Real Time

There are four roles supported in Panopticon Real Time:

Role	Description

Administrator	Allowed to perform the following:
	• View, rename, move, copy, upload, download, interact, and publish/republish workbooks into folders to which the user has permissions to.
	• Add, rename, or remove folders and subfolders and manage users and groups that should be granted or denied access.
	• Import or export workbooks bundle
	• Administer the server which includes:
	  ○ manage data templates
	  ○ create data extracts from connectors
	  ○ create and manage alerts and specify the email address of the user or group who will receive the alert

	○ view and manage plugin subscriptions ○ manage caches ○ specify logging level and view and copy logs ○ create and manage scheduled tasks ○ create and manage global parameters ○ manage workbook themes ○ view logged in users to the server and log them out ○ view server settings, set the file logging level, and clear cache
Viewer	Allowed to view and analyze fully interactive dashboards. This is the default role assigned to users that cannot be mapped to other roles.
Designer	Allowed to perform the following: • View, create, upload, rename, move, copy, merge, download, remove, workbooks, and publish/republish them into folders to which the user has permissions • Import or export workbooks bundle • Add, rename, or remove folders and subfolders and manage users and groups that should be granted or denied access • Create and manage global parameters • Create and manage alerts and specify the email address of the user or group who will receive the alert These topics are discussed in Altair Panopticon Web Authoring Guide.
Anonymous	Allowed to view workbooks if `authentication.required` is set to **false**.

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 `Panopticon.properties`.

Property	Description	Default Value
access.administrator.groups	The role that is mapped to the administrator group.	**admin**
access.default.roles	The default roles applied to all users of the server. For example, if `access.default.roles`=DESIGNER,ADMINISTRATOR and a user with a VIEWER role logs on to the server, then the user will simultaneously have a VIEWER, DESIGNER, and ADMINISTRATOR 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 ADMINISTRATOR, VIEWER, ANONYMOUS, and/or DESIGNER. This property is case sensitive.	**VIEWER**
access.designer.groups	The role that is mapped to the designer group.	**designer**
access.viewer.groups	The role that is assigned to the viewer group.	

NOTE

- Group sets can be added for a role, separated by a comma.
- To be able to use all of the features of Panopticon Real Time, a user is required to have Designer and Administrator roles.
- When using Altair Units licensing, different user roles will check out different numbers of Altair Units.

Role	Altair Unit Draw
Viewer	2
Designer	2 21 when designing a workbook
Administrator	2

Token

A web token is used when the user has successfully logged into Panopticon Real Time when using one of the following authentication types: **BASIC**, **SAML**, or **WINDOWS**. 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 Real Time in the form of a cookie when the user has been authenticated. By default (false), the cookie will be stored in the browser as https cookie and is accessible to the JavaScript.

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, if it should be persistent or if it should only last for a user session (while the browser is still open), and/or it can be stored as a HttpOnly cookie. All this can be configured in the `Panopticon.properties`. The table below lists all available token properties.

Property	Description	Default Value
authentication.token.persistence	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 `AppData/Token` folder before starting the server.	**PERSISTENT**
authentication.token.refreshable	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	**true**

	will expire and invalidate the user session if the property is set to **false**.	
authentication.token.secret	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.*	**Auto-generated**
authentication.token.validity.seconds	The number of seconds that the token should be valid.	**604800**
authentication.token.cookie	The name of the cookie used to store the authentication cookie. Must be unique for each server instance on the host.	**ptoken**
authentication.token.cookie.httponly	This property determines how the browser will treat the cookie. If set to **true**, the cookie will be stored in the browser as a HttpOnly cookie and will not be available to the JavaScript. If set to **false** (default), the cookie will be stored in the browser as https and will be accessible to the JavaScript.	**false**
authentication.token.cookie.secure	This property determines how the browser will treat the cookie depending on the security of the connection. If set to **true**, when the browser receives a secure cookie (HttpOnly cookie), you will not be able to transmit it unless the connection is secure.	**false**
authentication.token.in.login.response.body	This property determines if the REST login response body should contain a token info. **NOTE:** Does not affect the SOAP login response body.	**false**

TOMCAT REALM

Panopticon Real Time 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 Real Time 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 `Panopticon.properties` file:

`authentication.type`=**BASIC**

NOTE	

	• Reading the Apache Tomcat documentation is recommended: https://tomcat.apache.org/tomcat-9.0-doc/realm-howto.html. Abbreviations used: CN = Common Name, OU = Organizational Unit, DC = Domain Component.
	
	• 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 should 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.

Example 1

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

```
<role rolename="administrator"/>
<user username="James" password="james" roles="administrator"/>
```

By adding these two lines you have achieved the following:

- ❑ Created a new role named **administrator**
- ❑ Created a new user with username **James** and password **james**
- ❑ Assigned the newly created user the role **administrator**

Example 2

```
<role rolename="admin"/>
<role rolename="designer"/>
<role rolename="user"/>
<user username="viewer" password="viewer" roles="user"/>
<user username="John" password="john" roles="user,admin"/>
<user username="Paul" password="paul" roles="user,designer"/>
<user username="Austin" password="austin" roles="user,designer,admin"/>
```

By adding these seven lines, you have achieved the following:

- Created three new roles named **admin, designer, user**
- For the role **user**, created four users:
  - with username **viewer** and password **viewer**
  - with username **John** and password **john**
  - with username **Paul** and password **paul**
  - with username **Austin** and password **austin**
- For the role **admin**, created two users:
  - with username **John** and password **john**
  - with username **Austin** and password **austin**
- For the role **designer**, created two users:
  - a user with username **Paul** and password **paul**
  - with username **Austin** and password **austin**

NOTE

User Austin has both administrator and designer roles and is considered a super user.

A sample `tomcat-users_example.xml` is provided in the `AltairPanopticonVisualizationServerWAR_<version number>.zip` file. You can modify or add new users and roles in this file.

In Panopticon 2020 and onwards, the `Administrators.txt` and `AdministratorGroup.txt` files are no longer used to authorize administrator users. The function provided by these files has been replaced by a set of properties in `Panopticon.properties`:

```
access.default.roles=VIEWER
access.administrator.groups=admin
access.designer.groups=designer
access.list.delimiter=,
access.viewer.groups=
```

The `access.default.roles` property defines the default roles assigned to any user accessing the server, defaulting to VIEWER. The administration (`access.administrator.groups` property) and content creation (`access.designer.groups` property) on the server are mapped by default to the admin and designer user groups.

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

LDAP

Panopticon Real Time 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, in `conf/server.xml`. 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 th ese details.

You can specify more than one LDAP domain by defining a **Combined Realm**. This is done by putting more than one Realm configuration within a parent CombinedRealm:

```
<Realm className="org.apache.catalina.realm.CombinedRealm" >
  <Realm className="org.apache.catalina.realm.JNDIRealm"
    (realm details...) />
  <Realm className="org.apache.catalina.realm.JNDIRealm"
    (realm details...) />
</Realm>
```

NOTE	LockOutRealm (mentioned at the start of this chapter) is an implementation of the Tomcat Realm interface that extends the CombinedRealm. For futher information, please see Apache Tomcat 9 documentation on https://tomcat.apache.org/tomcat-9.0-doc/realm-howto.html

Active Directory

Panopticon Real Time can be configured to authenticate towards an Active Directory server. Panopticon Real Time 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 extra[illegible] these details.

A useful tool when configuring your Active Directory realm is Active Directory Explorer from Microsoft Sysinternals: https://docs.microsoft.com/en-us/sysinternals/downloads/adexplorer.

USEFUL TIP

Depending on how your Active Directory is set up, you may need to specify different attribute values for your userBase and your roleBase. For further info, see Apache Tomcat 9 documentation about realms: https://tomcat.apache.org/tomcat-9.0-doc/realm-howto.html.

Abbreviations used: CN = Common Name, OU = Organizational Unit, DC = Domain Component

WINDOWS AUTHENTICATION

Panopticon Real Time supports Windows authentication. Panopticon Real Time will authenticate a user towards the local machine and verify its credentials with the existing and configured users on the Windows machine. The Windows authentication operates similarly to the Basic authentication function. Both the username and the password are sent to Panopticon Real Time which they are then verified.

To enable Windows authentication, set this property in the `Panopticon.properties` file:

`authentication.type`=**WINDOWS**

NOTE	Single Sign On is currently not supported with the Windows authentication. In addition, Windows authentication only supports authentication towards the local machine. This means that the machine where Panopticon Real Time is deployed on also has to manage all of the users.

SAML

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

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

Property	Description
authentication.saml.assertion.roles	User attribute for roles configured in the IdP.
authentication.saml.assertion.username	User attribute for username configured in the IdP.
authentication.saml.assertionconsumerservice.url	The URL to the Panopticon assertion consumer service. URL: [Protocol]://[Host]:[Port]/[Context]/server/rest/auth/login Example: `http://localhost:8080/panopticon/server/rest/auth/login`
authentication.saml.certificate.name	The name of the certificate used to validate signature and/or sign outgoing SAML messages
authentication.saml.certificate.password	The password of the certificate used to validate signature and/or sign outgoing SAML messages..
authentication.saml.challenge.required	Determines whether the IdP-first authentication with SAML is enabled or not. To enable, set this property to **false**.
authentication.saml.identityprovider.logout.url	The URL to the IdP logout service.
authentication.saml.identityprovider.url	The URL to the IdP login service.
authentication.saml.keystore.file	The location of the Keystore file that contains the certificate.
authentication.saml.keystore.password	The password to the Keystore file.
authentication.saml.serviceprovider.id	The ID of the service provider configured in the IdP.
authentication.saml.identityprovider.certificate.file	Takes a file path to a certificate file that contains the IdP's public key.
authentication.saml.identityprovider.signature.validation.required	Specifies whether to require a valid IdP signature to be present on the SAML response. Default value is **false**.

authentication.saml.provider	The IdP provider. Possible values are **OPENSAML**, **OPENAM**. Default value is **OPENSAML**.
authentication.saml.keystore.type	The key store type. Possible values are **JKS, JCEKS**, **PKCS12**. Default value is **JKS**.
authentication.saml.login.redirect.url	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.
authentication.saml.logout.redirect.url	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.
authentication.saml.openam.meta.alias	The meta alias for the IdP if you are using OpenAM.

Using SAML Through OpenAM

To enable roles mapping with OpenAM, perform the following configuration in the IdP:

1. Add value **isMemberOf** to the LDAP User Attributes list in your Data Store.
2. Add value **isMemberOf** to the attributes mapping in your IdP configuration.
3. Set `authentication.saml.assertion.roles=isMemberOf` in the `Panopticon.properties`.

OAUTH 2.0

This section discusses how to configure Panopticon Real Time to use the OAuth 2.0 for authorization. Upon a logon request, Panopticon Real Time 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 Real Time will only know if the user is authorized or not. To authenticate the user, Panopticon Real Time 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.

If user roles are configured, the user identity attribute (i.e., `authentication.oauth2.identity.attribute.roles=<Name of role attribute in OAuth2 server>`) will expect a list of roles that enable role based access restrictions on workbook folders.

For example: `"roles": ["VIEWER","DESIGNER"]`

If this attribute is not set, the default role VIEWER is used.

To use OAuth2.0, change the authentication type:

`authentication.type`=**OAUTH2**

Afterwards, go through the following properties to be configured with respect to your OAuth server configuration.

Property	Description
authentication.oauth2.client.id	The ID of the OAuth 2.0 client.
authentication.oauth2.client.secret	The secret used by the OAuth 2.0 client.
authentication.oauth2.identity.attribute.roles	The attribute that will be extracted from the identity response and used as the role.
authentication.oauth2.identity.attribute.roles.pattern	Takes regex used to extract the roles from the OAuth 2.0 server identity response. For example, the returned string: `cn=admin,ou=groups,dc=openam,dc=openidentityplatform,dc=org,cn=designer,ou=groups,dc=openam,dc=openidentityplatform,dc=org` contains two roles, **admin** and **designer** The regex to extract the roles is **cn=([^,]+)**.
authentication.oauth2.identity.attribute.username	The attribute that will be extracted from the identity response and used as the username.
authentication.oauth2.identity.url	The URL to the REST service that provides details about the authenticated user.
authentication.oauth2.login.callback.url	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 Real Time.
authentication.oauth2.login.response.type	The response type. The only response type that is currently supported is **code**. The value can also be left blank.
authentication.oauth2.login.redirect.url	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.
authentication.oauth2.login.scope	The requested scope. The property can be left blank.
authentication.oauth2.login.state	The requested state. The property can be left blank.
authentication.oauth2.login.url	The URL to the OAuth 2.0 login resource. This field can be left blank.
authentication.oauth2.logout.redirect.url	Logging out revokes the token from the authentication server if the property `authentication.oauth2.logout.url` 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 the Panopticon when logging out.
authentication.oauth2.logout.url	The URL to the OAuth 2.0 logout resource. This property can be left blank.
authentication.oauth2.token.method	The method on how the token should be retrieved. Supported values are **QUERY, BODY**, and **HEADER**.
authentication.oauth2.token.url	The URL to the OAuth 2.0 token resource.
access.default.roles	The default role that will be assigned to everyone to execute a specific service. The field can be left blank. If left blank, **VIEWER** role is used.
access.administrator.groups	The list of administrator roles authenticated by the OAuth2.0 server.
access.designer.groups	The list of designer roles authenticated by the OAuth2.0 server.

access.viewer.groups	The list of viewer roles authenticated by the OAuth2.0 server.

Example

For an example of OAuth2.0 server configuration, here are some roles and users:

Roles/Groups available:

- admin-backend
- admin-all
- server-designers
- server-viewers
- server-viewer-groupX

Users:

1. `username:testuser, roles:["server-viewers"]`
2. `username:testuser2, roles:["server-viewers","server-designers","admin-all"]`

For the OAuth2.0 server authentication, you would have to generate **ClientID** and **ClientSecret**.

To be authenticated by the OAuth2.0 server, the following Panopticon properties will be set:

```
authentication.oauth2.client.id=ClientId
authentication.oauth2.client.secret=ClientSecret
authentication.oauth2.identity.attribute.roles=roles
authentication.oauth2.identity.attribute.username=username
authentication.oauth2.identity.url=https://oauth2/me
authentication.oauth2.login.callback.url=http://localhost:8080/panopticon/server/rest/auth/login
authentication.oauth2.login.redirect.url=
authentication.oauth2.login.response.type=code
authentication.oauth2.login.scope=
authentication.oauth2.login.state=
authentication.oauth2.login.url=https://oauth2/authorize
authentication.oauth2.logout.redirect.url=
authentication.oauth2.logout.url=
authentication.oauth2.token.method=query
authentication.oauth2.token.url=https://oauth2/access_token
authentication.type=OAUTH2
access.default.roles=
access.administrator.groups=admin-backend,admin-all
access.designer.groups=server-designers
access.viewer.groups=server-viewers,server-viewer-groupX
```

Using OAuth 2.0 Through OpenAM

To enable roles mapping and username retrieval with OpenAM, perform the following configuration in the IdP:

1. Add value **isMemberOf** to the LDAP User Attributes list in your Data Store.
2. Add values **uid|Username** and **isMemberOf|Groups** to the scopes mapping in your OAuth agent configuration.

Example configuration in Panopticon.properties:

```
authentication.oauth2.client.id=panopticon
authentication.oauth2.client.secret=password123
authentication.oauth2.identity.attribute.roles=isMemberOf
authentication.oauth2.identity.attribute.roles.pattern=cn=([^,]+)
authentication.oauth2.identity.attribute.username=uid
authentication.oauth2.identity.url=http://localhost:9080/openam/oauth2/tokeninfo
authentication.oauth2.login.callback.url=http://localhost:8080/panopticon/server
/rest/auth/login
authentication.oauth2.login.redirect.url=
authentication.oauth2.login.response.type=code
authentication.oauth2.login.scope=uid isMemberOf
authentication.oauth2.login.state=
authentication.oauth2.login.url=http://localhost:9080/openam/oauth2/authorize
authentication.oauth2.logout.redirect.url=http://localhost:8080/panopticon
authentication.oauth2.logout.url=http://localhost:9080/openam/oauth2/token/revok
e
authentication.oauth2.token.method=query
authentication.oauth2.token.url=http://localhost:9080/openam/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 Real Time 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 Panopticon.properties file in the AppData folder (**c:\vizserverdata**).
2. Enable authentication.type=FILTER in Panopticon.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 Real Time 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 Real Time 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 Real Time.

```
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.datawatch.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 Real Time behind a proxy server that will handle the authentication of users. The proxy server forwards the name of the user and roles to Panopticon Real Time 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 are 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 Real Time with regard to 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 `Panopticon.properties` file:

```
authentication.type=HEADER
authentication.header.role.delimiter=,
authentication.header.roles={roles header}
authentication.header.rolesdynamic={dynamic roles header}
authentication.header.username={userid header}
```

For example:

```
authentication.type=HEADER
authentication.header.role.delimiter=,
authentication.header.roles=X-Roles,X-Company
```

```
authentication.header.rolesdynamic=watcher,role_for_{X-Company}
authentication.header.username=X-User
```

[4] ADDITIONAL OR OPTIONAL STEPS

REPLACING PARAMETER VALUES WITH HTTP HEADERS AND COOKIES

Panopticon Real Time can be configured to replace both the incoming and outgoing parameters with HTTP headers and cookies:

- **Incoming Parameters** are parameters sent to Panopticon Real Time when requesting data. These types of parameters are also referred to as **request parameters**.
- **Outgoing Parameters** are parameters which are returned to the Client when retrieving a workbook. These types of parameters are also referred to as **response parameters**.

This feature is used for employing the user identifier as a parameter and sending the user identifier as a *Header* and *Cookie*. The Server Administrator can configure these properties so that the incoming parameters employ the user identifier value when requesting data. Consequently, the requested HTTP *Header* and *Cookie* values will be tailored for each user. The Server Administrator can also update these properties so that the outgoing parameters get updated when loading a workbook. For example, if you want the user's identifier to be shown in the workbook as a Title.

Replacing the parameter values with Header and Cookie values is achieved by configuring certain properties in the `Panopticon.properties` file located in the `Appdata` folder or `c:\vizserverdata`).

Updating incoming parameters can be achieved by configuring the following properties:

Property	Request parameter mapping
Attribute	`request.cookie.parameters.mapping.required`
Description	The parameters that are required to be updated with certain cookie values. This property will only affect incoming parameters. The operation will fail if configured cookie values are not present in the request. The property should be formatted as follows: Parameter name (Value delimiter) Cookie name.
Default Value	
Property	Request parameter mapping
Attribute	`request.cookie.parameters.mapping.optional`
Description	The parameters that could be updated with certain cookie values. This property will only affect incoming parameters. The operation will not fail if the cookie values are not present in the request. The parameters will keep their default value instead of the configured cookie value if the cookie is not present. The property should be formatted as follows: Parameter name (Value delimiter) Cookie name.
Default Value	
Property	Request parameter mapping

Attribute	`request.cookie.parameters.mapping.entry.delimiter`
Description	The delimiter that separates the configuration entries. This property will only affect incoming parameters.
Default Value	**, (Comma)**
Property	Request parameter mapping
Attribute	`request.cookie.parameters.mapping.value.delimiter`
Description	The delimiter that separates the parameter name and the cookie name. This property will only affect incoming parameters.
Default Value	**: (Colon)**
Property	Request parameter mapping
Attribute	`request.header.parameters.mapping.required`
Description	The parameters that are required to be updated with certain header values. This property will only affect incoming parameters. The operation will fail if a configured header values are not present in the request. The property should be formatted as follows: Parameter name (Value delimiter) Header name.
Default Value	
Property	Request parameter mapping
Attribute	`request.header.parameters.mapping.optional`
Description	The parameters that could be updated with certain header values. This property will only affect incoming parameters. The operation will not fail if the header values are not present in the request. The parameters will keep their default value instead of the configured header value if the header is not present. The property should be formatted as follows: Parameter name (Value delimiter) Header name.
Default Value	
Property	Request parameter mapping
Attribute	`request.header.parameters.mapping.entry.delimiter`
Description	The delimiter that separates the configuration entries. This property will only affect incoming parameters.
Default Value	**, (Comma)**
Property	Request parameter mapping
Attribute	`request.header.parameters.mapping.value.delimiter`
Description	The delimiter that separates the parameter name and the header name. This property will only affect incoming parameters.
Default Value	**: (Colon)**

The following properties can be configured to update outgoing parameters:

Property	Response parameter mapping
Attribute	`response.operation.parameters.mapping.required`
Description	The parameters that are required to be updated with certain Header values. This property will only affect outgoing parameters. The operation will fail if configured Header values are not present in the request. The property should be formatted as follows: Parameter name (Value delimiter) Header name.
Default Value	
Property	Response parameter mapping
Attribute	`response.operation.parameters.mapping.optional`
Description	The parameters that could be updated with certain Header values. This property will only affect outgoing parameters. The operation will not fail if the Header values are not present in the request. The parameters will keep their default value instead of the configured Header value if the Header is not present. The property should be formatted as follows: Parameter name (Value delimiter) Header name.
Default Value	
Property	Response parameter mapping
Attribute	`response.operation.parameters.mapping.entry.delimiter`
Description	The delimiter that separates the configuration entries. This property will only affect outgoing parameters.
Default Value	**, (Comma)**
Property	Response parameter mapping
Attribute	`response.operation.parameters.mapping.value.delimiter`
Description	The delimiter that separates the parameter name and the Header name. This property will only affect incoming parameters.
Default Value	**: (Colon)**

Example

This section describes how incoming parameters are replaced with Header values. For example, Panopticon Real Time is required to update parameters **uid** and **uname**.

Parameter Name	Update With Header
uid	userIDHeader
uname	userNameHeader

The request will fail if the required *Headers* are not present in the incoming request.

For the next example, Panopticon Real Time will try to update the parameter **ulocation** with **userLocationHeader** header. The parameter value will only be updated if the Header is available.

In both of these configurations, comma was used as an entry delimiter and colon as a delimiter between the parameter name and the Header name.

However, for outgoing parameters, the property prefix (request) has to be changed to **response** instead.

Configurations:

```
request.header.parameters.mapping.required=uid:userIdHeader,uname:userNameHea
der
request.header.parameters.mapping.optional=ulocation:userLocationHeader
request.header.parameters.mapping.entry.delimiter=,
request.header.parameters.mapping.value.delimiter=:
```

NOTE	Mapping the same parameter in both the header and cookie will thrown an exception on initialize.

FILE UPLOAD SIZE LIMITS SETTINGS IN TOMCAT AND PANOPTICON

Starting with version 2020.0 and the introduction of web authoring, any connection to a file data source involves uploading the file first to the server then loading its data into Panopticon. The upload happens as part of using the data connector for the file.

Setting the limit of the file upload sizes are done in the following properties:

- maxSwallowSize

  This setting is part of the overall Tomcat configuration, particularly for the HTTP connector, and is found in the `<tomcat>/conf/server.xml` file.

  `maxSwallowSize` controls how much data Tomcat will accept for upload before it is cancelled or terminated. If the file size is larger than the `file.upload.size.max.bytes`, and the limit of `maxSwallowSize` is hit, then Panopticon will never get a chance to send a proper error message about the file being too large. The upload will simply be terminated with a message about an unknown error. It is therefore recommended to set the `maxSwallowSize` value high enough to the file size that Panopticon users are expected to load.

  Any minus value (e.g., -1), means unlimited. Setting a minus value for `maxSwallowSize` creates a risk of getting the Tomcat connection saturated by a very large file upload or being stuck in an infinite file upload. A reasonable setting would be something between one to two times of the Panopticon `file.upload.size.max.bytes` property value.

  Example:

  ```
  <Connector port="8080" protocol="HTTP/1.1"
             connectionTimeout="20000"
             redirectPort="8443"
             maxSwallowSize="100000000"/>
  ```

- file.upload.size.max.bytes

  This property is part of the Panopticon specific settings found in the `Panopticon.properties` file in the `PanopticonAppdata` folder (i.e., `c:\vizserverdata`).

This size limit property (in bytes) controls how large are the files Panopticon will accept to connect to for loading data.

If the file exceeds the size limit, there will be an informative error message that indicates the current size limit. The size limit check can only take place on the condition that the file has already been successfully uploaded to the server. The upload success depends on the limit set in `maxSwallowSize`.

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.

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.

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
   ```

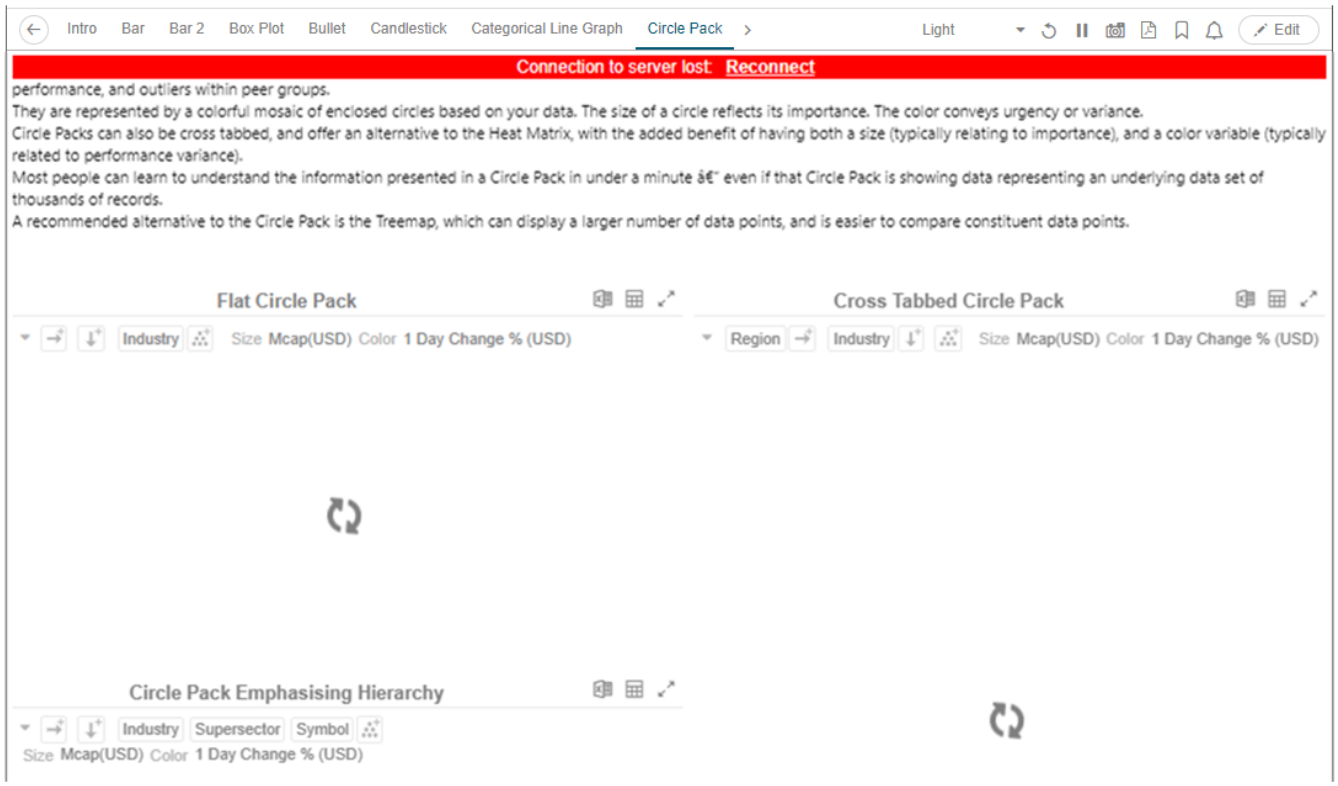
NOTE	Setting the maximum value should be dependent on your system. Ensure that the heap size is not larger than the available free RAM on your system. It is recommended to use 80% of the available RAM not taken by the operating system or other processes of your JVM.

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

SET CLIENT AUTOMATIC RECONNECTION TO THE SERVER WHEN DISCONNECTED

When the client loses connection to the server, by default, it will no longer attempt to automatically reconnect to the server.

A notification message displays such as below. Clicking the **Reconnect** link will attempt to reconnect to the server.



The `AppData` folder of Panopticon Real Time has a subfolder named `JavaScriptConfiguration` which contains the file named `workbook.json`.

Below is an example default content of a `\JavaScriptConfiguration\workbook.json` file:

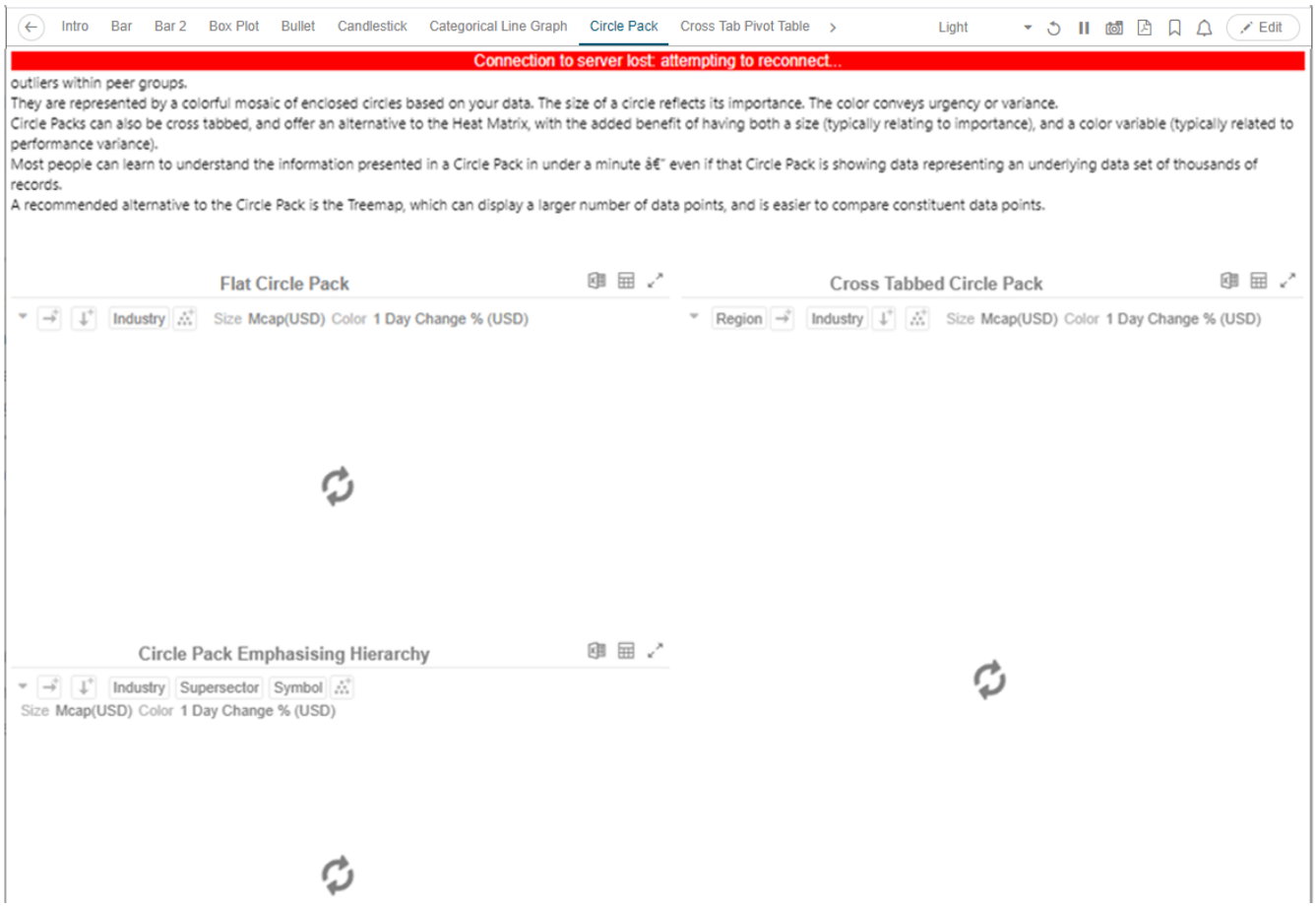
```
{
  "baseUrl" : "..",
  "forceClientSelectionHandling" : true,
  "startUrl" : "../",
  "subscriptionCompression" : true,
  "dataLoading" : {
    "transport" : "websocket"
  },
  "webGlEnabled" : true,
  "pdfMultiplePagesEnabled" : true
}
```

To set automatic reconnection to the server, add the following parameter in the `JavaScriptConfiguration\workbook.json` file:

```
"automaticReconnectOnServerDisconnect" : true,
```

NOTE	After each change in the `workbook.json` file, the Panopticon application must be restarted.

A notification message displays such as below. **Connection to server lost: attempting to reconnect…**



AUTOMATIC LOGOUT OF USERS ON TIMEOUT TO SAVE UNUSED LICENSES

Users who have no activity and leave their browsers open may be logged out and their license units are checked out by setting the following properties in the `Panopticon.properties` file:

Property	Timeout Session
Attribute	`timeout.session.enabled`
Description	Boolean value stating if timeout functionality should be used or not.
Default Value	**false**
Property	Timeout Session
Attribute	`timeout.session.exception.delimiter`
Description	The delimiter to use for the usernames stated in the timeout.session.exception.usernames property.

Default Value	**, (comma)**
Property	Timeout Session
Attribute	`timeout.session.exception.usernames`
Description	Usernames that should be excluded from the timeout functionality. Separated by the delimiter stated in the `timeout.session.exception.delimiter` property.
Default Value	
Property	Timeout Session
Attribute	`timeout.session.minutes`
Description	Minutes of inactivity before a user session is terminated by logging out the user.
Default Value	**480**
Property	Timeout Session
Attribute	`timeout.session.notification.minutes`
Description	Minutes before a timeout that a notification about session timeout is sent to the user.
Default Value	**1**

NOTE

- The timeout functionality is only enabled if property `timeout.session.enabled` is set to **true**.
- Each time a user actively interacts with the server, the timeout timer for that user is reset. Just sitting idle on a tab in the UI or having a dashboard open will not reset the timer.
- If `timeout.session.notification.minutes` has been set to a value > 0, a notification will be sent to the user on an established notification subscription on the websocket, X minutes before the timeout happens. This is the format of the timeout notification:

  ```
  {"TimeoutNotification":{"minutesUntilTimeout":1}}
  ```

- When a session times out, a logout notification will be sent on an established notification subscription on the WebSocket. This is the format of the logout notification:

  ```
  {"LogoutNotification":{"reason":"Logged out due to session timeout"}}
  ```

- To abort a session timeout, all that is required is that the user interacts with the server. To facilitate the process there is a new service that can be called using GET on URL `/rest/user/timeout/reset` that will reset the timeout for the calling user. The service itself does not do anything, but the layers the message interacts with before reaching the service will count it as a user activity and resets the timeout.

SETTING THE TRANSPORTATION PROTOCOL

In previous versions, you can control which transportation protocol the browser would use for subscriptions to the server by setting the value (**WEBSOCKET** or **LONG_POLLING**) in the `client.data.load.transport` property in the `Panopticon.properties` file.

Starting with version 2020.2, you can instead edit the `workbook.json` and `admin.json` in `<appdata>/JavaScriptConfiguration/`, where you add the section:

```
"dataLoading" : {
    "transport" : "websocket" or "long-polling"
  }
```

> **NOTE** After each change in the `workbook.json` and `admin.json` files, the Panopticon application must be restarted.

PANOPTICON REAL TIME CONFIGURATIONS FOR EMAIL SEND OUTS AND ALERTS

> **NOTE** When triggering email send out via the REST API, scheduling email send outs, or sending email alerts, Panopticon Real Time needs to be configured with valid email server information.

The following values need to be configured in the `Panopticon.properties` file located in the `AppData` folder (e.g., `c:\vizserverdata`):

Attribute	Description
`email.address`	Email address where the alert will be sent from.
`email.host`	Host name used by the email server.
`email.password`	Email password, if available.
`email.port`	Port number used by the email server.
`email.security.mode`	Security mode used when sending emails. Possible values: **NONE**, **SSL**, **TLS**. The value **NONE** will be used if there was no value configured for the property.
`email.username`	Email account username.

FONT INSTALLATION REQUIREMENT FOR PDFS AND IMAGE EXPORT WITH CJK CHARACTERS

When creating PDF reports or exporting images from workbooks that contain text in Chinese, Japanese or Korean (CJK), a font with CJK support is required. The font must be installed on the server operating system. Refer to your operating system documentation on how to install a new font.

The PDF and image export functionalities in Panopticon will use the font specified in the workbook Theme. While used in the browser, the workbooks and dashboards will get the suitable font by the browser if a font with CJK support is needed.

When creating a PDF or an image, the browser is not involved since it happens on the server-side, and correct characters depend on the availability of a font with CJK support on the local system of the server, plus the specification of that font (e.g., **Yu Gothic**), in the workbook Theme. For the list of CJK-supporting fonts, you may refer to https://en.wikipedia.org/wiki/List_of_CJK_fonts.

SETTING SERVER PROPERTIES THROUGH THE ENVIRONMENT VARIABLES

Server properties set in the Panopticon.properties file in the `AppData` folder (i.e., `c:\vizserverdata`) are overridden by environment variables.

For example, you can supply a JSON object through the environment variable SPRING_APPLICATION_JSON that will be parsed during server start up:

```
{
      "server.id": "Test_Server",
      "subscription": {
            "data.loading.pool.max.size": "5",
            "broadcasting.pool.max.size": "6"
      }
}
```

NOTE	Ensure that you minify the JSON object before setting the environment variable.

This will override and set the following property values:

```
server.id=Test_Server
subscription.data.loading.pool.max.size=5
subscription.broadcasting.pool.max.size=6
```

As seen from the example above, you can use inline JSON annotations for properties that share the same prefix, which in this case is **subscription**.

To override a single property, create an environment variable with the same name, but replace each '.' with an '_' and use upper case.

Example: Override the property `server.id`
Name: `SERVER_ID`
Value: <some value>

You can also override individual properties with environment variables. Just set a variable with the same name as the property but with all letters in upper case and periods replaced with underscores. For example, **REPOSITORY_STARTUP_IMPORT_PATHS** will override the `repository.startup.import.paths` in `Panopticon.properties`.

[5] ADVANCED SERVER DEPLOYMENTS

USAGE IN SSL ENABLED ENVIRONMENTS

Enabling SSL for Panopticon Real Time

The steps shown in this guide use the keytool command for managing keyStores and certificates. The keytool command is part of the Java distribution and can be found in the `JAVA_HOME\bin`. Make sure you have the `JAVA_HOME\bin` folder in your `PATH` environment variable, in order to run the command. Details on the keytool command can be found here: https://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html

Follow the steps below to configure SSL for Panopticon Real Time.

Steps:

1. Change directory to the `CATALINA_HOME\conf` folder, which is where we want to generate the Tomcat keystore.
2. Create a keyStore file to store the private key and self-signed certificate used to identify the server:

   ```
   keytool -genkey -alias myalias -keyalg RSA -keystore keystore.jks
   ```

NOTE	Java is strict when validating the certificate of a host. If the domain name store in the certificate does not match the domain of the server, the connection will be rejected. Enter the target domain name (`www.mydomain.com`) when keytool asks for “your first and last name”, when running the command above.

3. Add an **SSL HTTP/1.1 Connector** entry in `$CATALINA_BASE/conf/server.xml`

```
<!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
    <Connector

protocol="org.apache.coyote.http11.Http11NioProtocol"
           port="8443" maxThreads="200"
           scheme="https" secure="true" SSLEnabled="true"
           keystoreFile="conf/keystore.jks"
           keystorePass="keystorepassword"
     clientAuth="false" sslProtocol="TLS"/>
```

4. Disable unencrypted server access by commenting out the default HTTP connector for port 8080.

```
<!--
    <Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
-->
```

5. After completing the configuration changes, you must restart Tomcat. When the process is back up you should be able to connect over SSL using the URL below:

   `https://localhost:8443/panopticon`

Details on how to configure Apache Tomcat SSL can be found at:

`https://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html`

Defining a TrustStore

In scenarios that require TLS-enabled intra-service communication, we need to configure a trustStore. These scenarios include, for instance, LDAP, SAML or OAuth integration.

A trustStore is essentially a keyStore, but where the keyStore is used to store private keys used to identify the server, the trustStore is used to store public keys of trusted *Certificate Authorities* (CA). The trustStore is used to verify certificates presented to the server when establishing an SSL connection.

Follow the steps below to create a new trustStore, import a certificate and configure Java to use the new trustStore:

Steps:

1. Create a new keyStore called **truststore**:

   ```
   keytool -genkey -alias truststore -keyalg RSA -keystore
   truststore.jks
   ```

2. Export a certificate from a keyStore:

   ```
   keytool -export -keystore keystore.jks -alias myalias -file cert.cer
   ```

3. Import the certificate into the trustStore:

   ```
   keytool -import -trustcacerts -alias myalias -file cert.cer -keystore
   truststore.jks
   ```

You can also re-use a keyStore as a trustStore in which case the certificate does not need to be exported and imported.

To configure a trustStore for Apache Tomcat you need to edit the JAVA_OPTS environment variable in the `setenv` script, located in the Tomcat `conf` folder.

- On Windows, setenv.bat:

  ```
  set JAVA_OPTS=-Djavax.net.ssl.trustStore="C:/location/to/truststore
  /truststore.jks"
  ```

- On Linux, setenv.sh:

  ```
  export JAVA_OPTS="$JAVA_OPTS -
  Djavax.net.ssl.trustStore='/location/to/truststore/truststore.jks'";
  ```

[6] 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 `Panopticon.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 workbooks published to the folders or subfolders in Panopticon Real Time can be secured by granting allowed or denied permissions.

NOTE	Beginning with version 16.1.0, new workbooks must be published to a folder or subfolder to use their access restrictions. However, workbook access restriction is still available and supported on older workbooks that will be accessed in the current and later Panopticon Real Time versions.

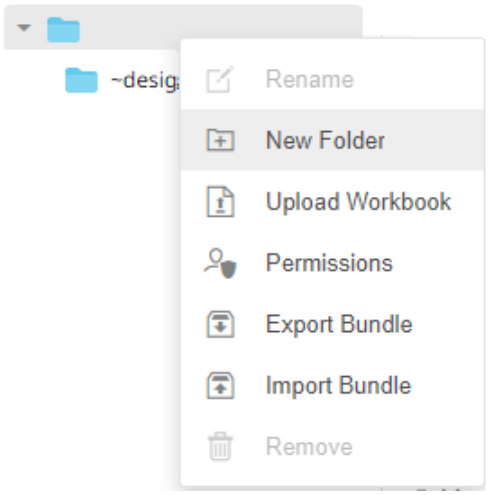
Creating Folders

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 Workbooks page (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 workbooks and build dashboards. For more information, refer to *Altair Panopticon Web Authoring Guide* on how to create workbooks on the Web client.

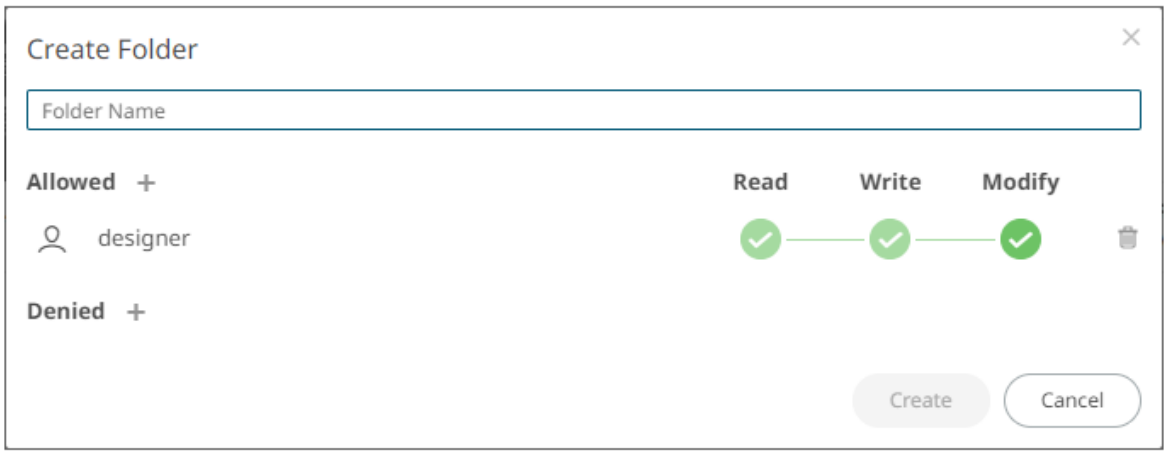
Steps:

1. On the **Workbooks** tab, right-click on the topmost folder, and select **New Folder**.



NOTE	Only Administrators are allowed to change the permissions on the root folder.

The *Create Folder* dialog displays.

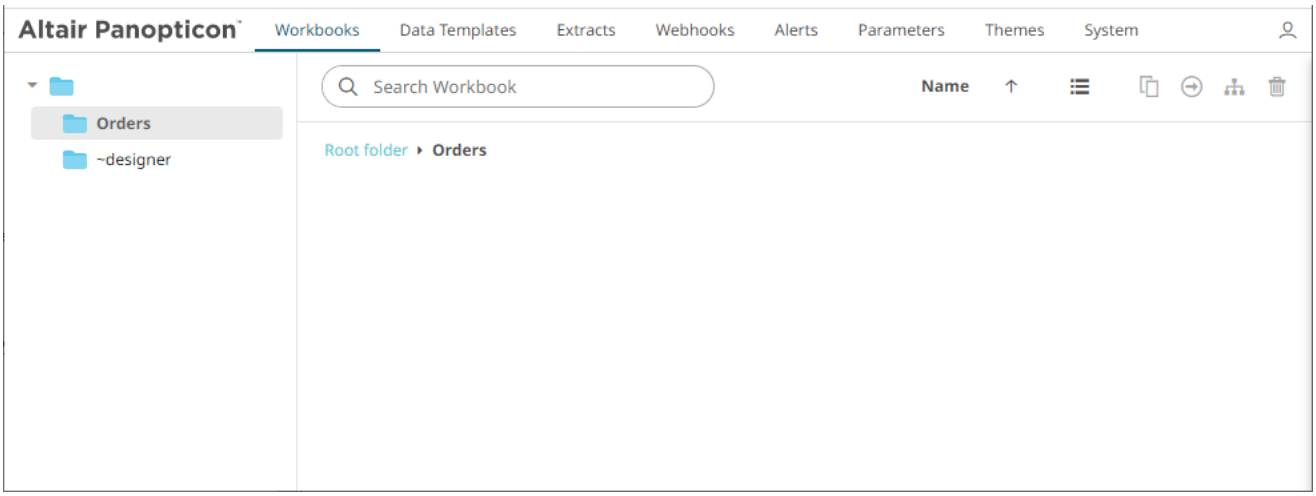


NOTE	• Everyone is available in the *Allowed* section by default. • Removing the Everyone group will mean that the folder and its subfolders will not be available for public access. • The default group permissions on the root folder are WRITE + READ.

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

4. Click Create .

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



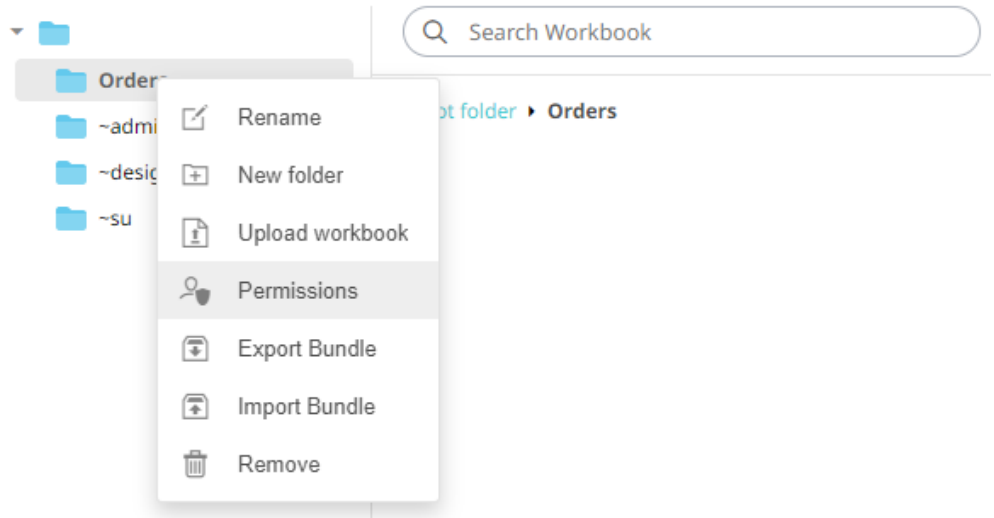
NOTE	• Folders and subfolders can be deleted as long as they do not contain published workbooks. • The folders and subfolders on the Workbooks tab will also be available on the Extracts and Webhooks tabs.

Adding Groups and Users with Allowed Authorization

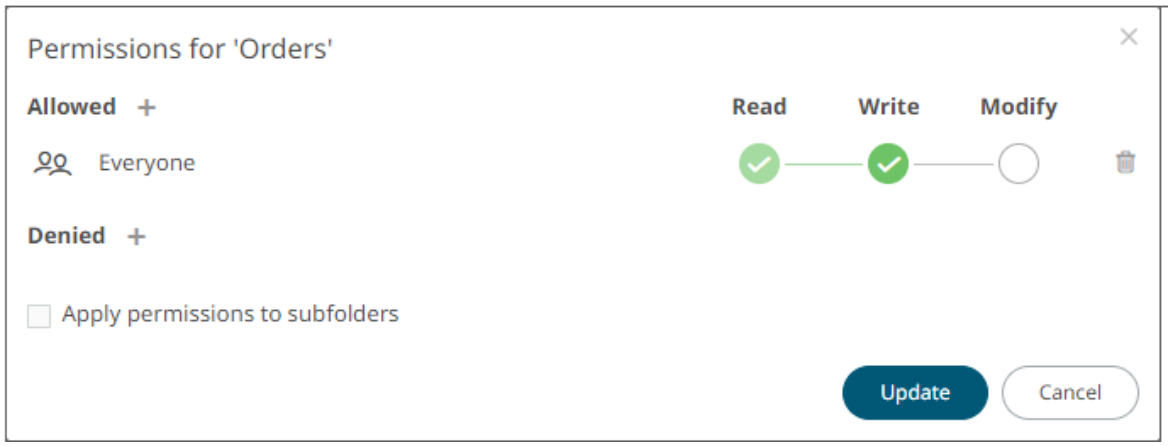
A user with an Administrator or Designer role can grant permissions for users or groups to a workbook folder or subfolder.

Steps:

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

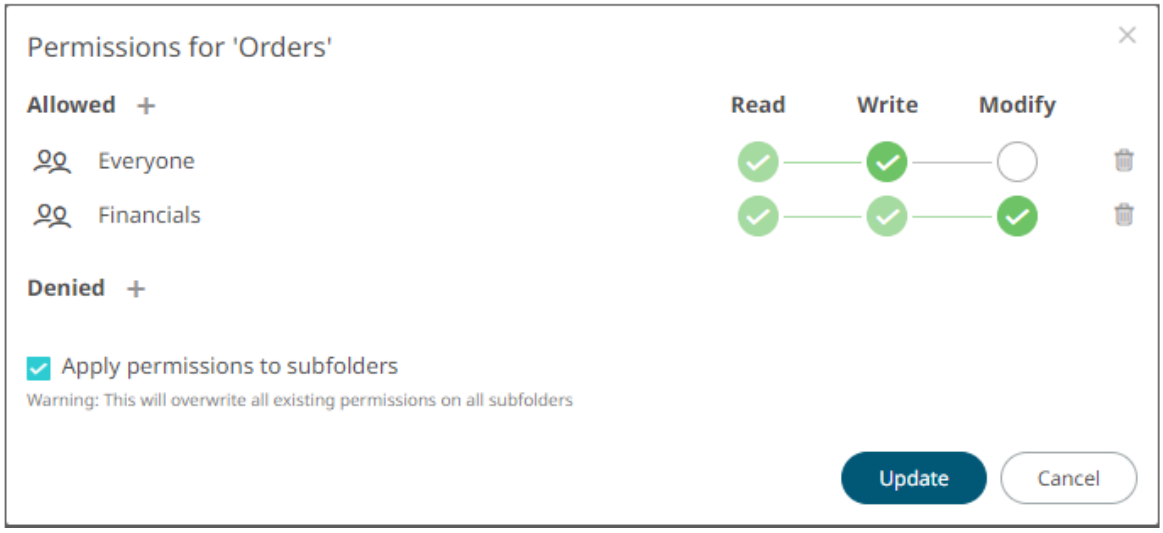
- MODIFY + WRITE + READ

  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:

   - check the **Apply Permissions to Subfolders** 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: • is only enabled when there is an existing subfolder. • does not affect the private folders.

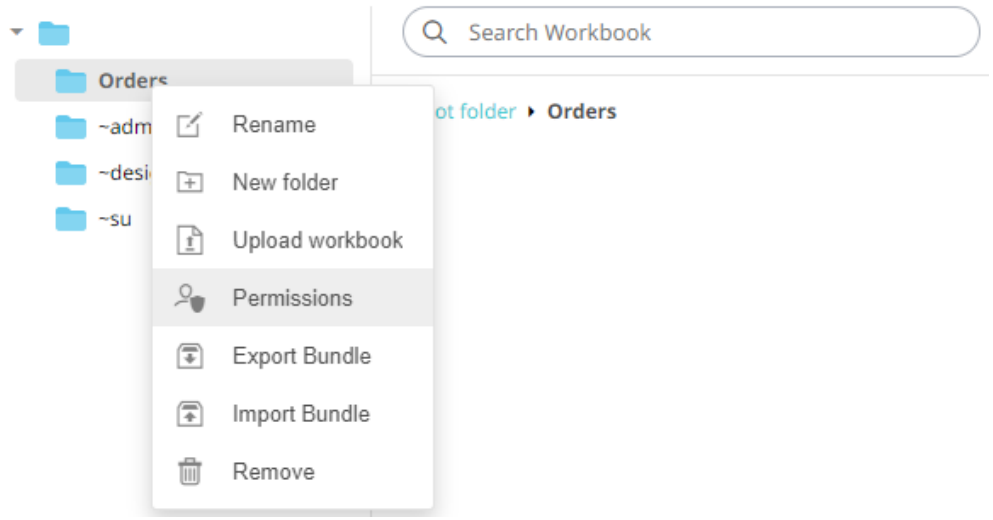
   - leave the **Apply Permissions to Subfolders** box unchecked and modify the permission properties of the subfolders

8. Click Update to save the changes.

Adding Groups and Users with Denied Access

Steps:

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



The *Permissions* dialog displays.

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

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

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.

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

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

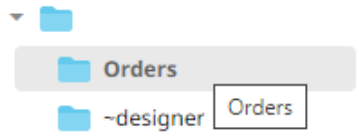
7. Click 

to save the changes.

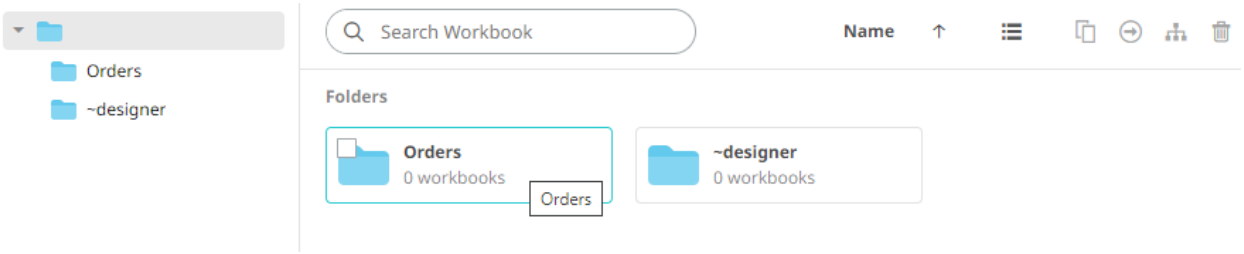
Creating Subfolders

Steps:

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

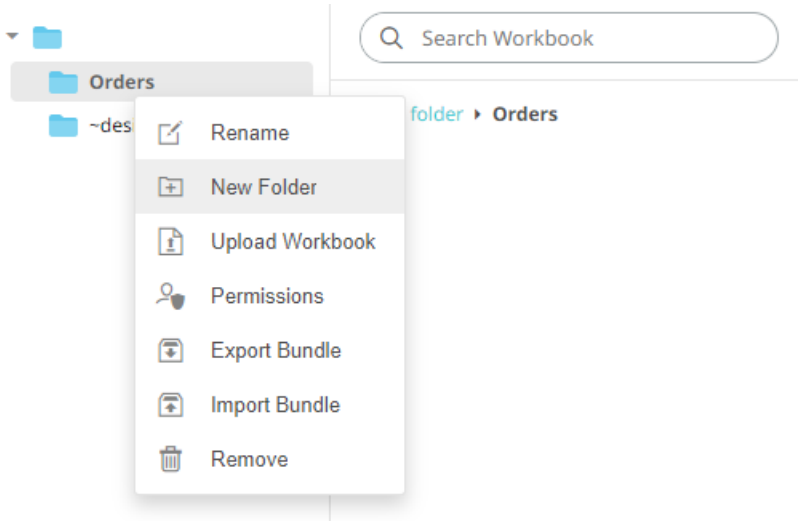


   - on the Root workbooks/folders list



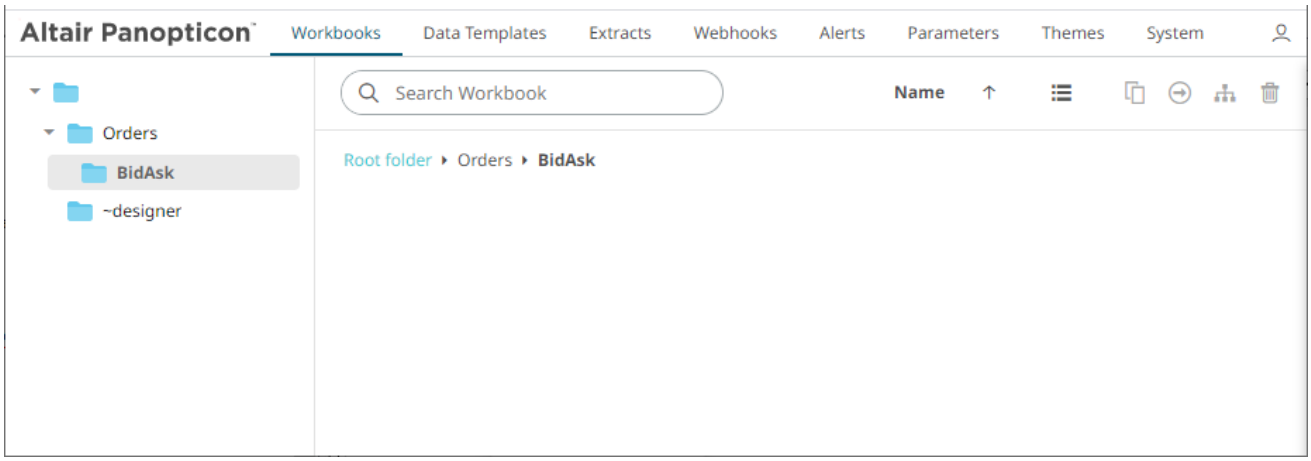
   The *Folders* page is displayed.

2. Right-click on the folder and select **New Folder**.

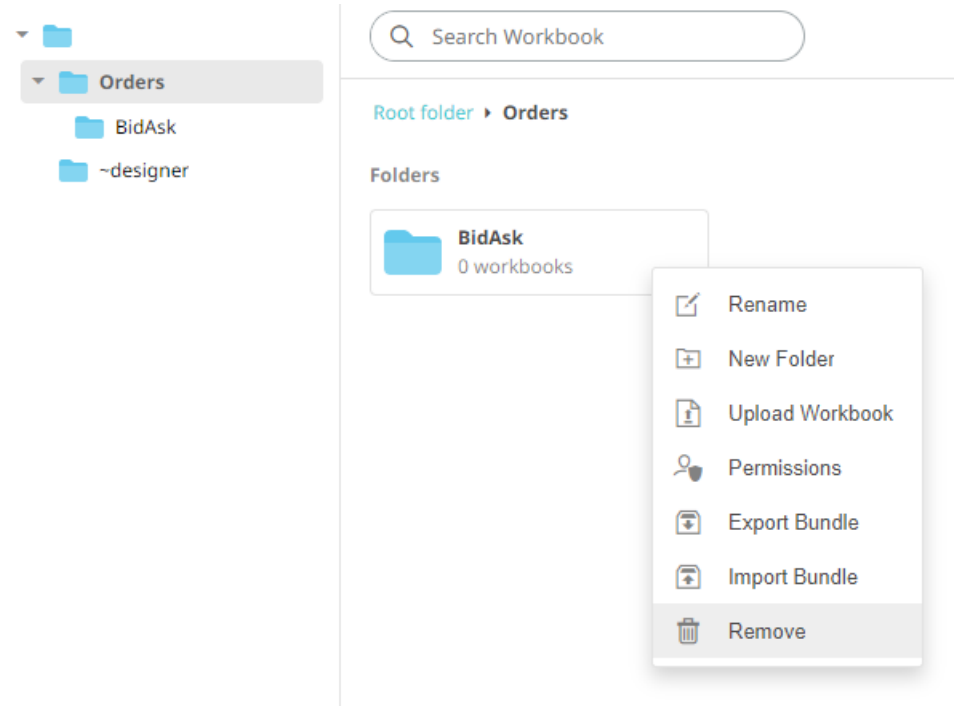


   Refer to Creating Folders for the steps in creating the subfolders. Also, Adding Groups and Users with Allowed Authorization and Adding Groups and Users with Denied Access for more information on adding Users and Groups with allowed or denied authorization.

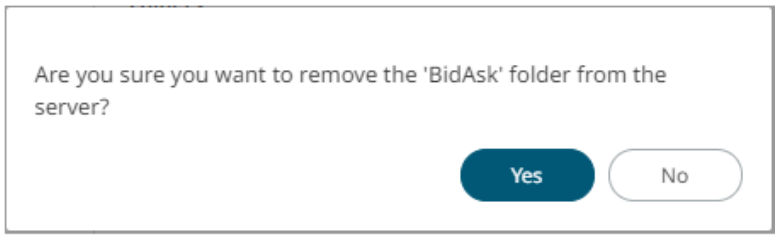
   The subfolder is added.



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



A confirmation message displays.



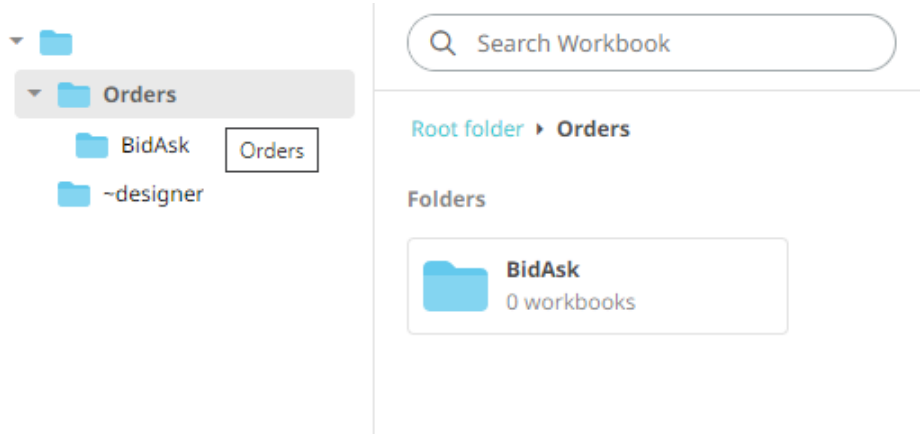
Click 

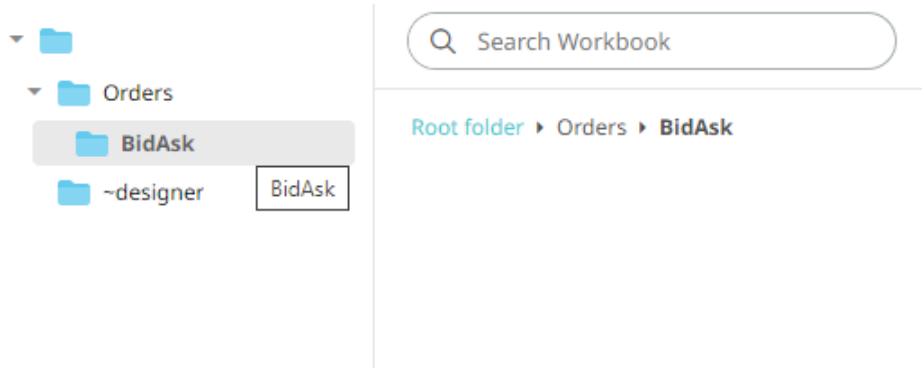
.

Updating Folder or Subfolder Properties

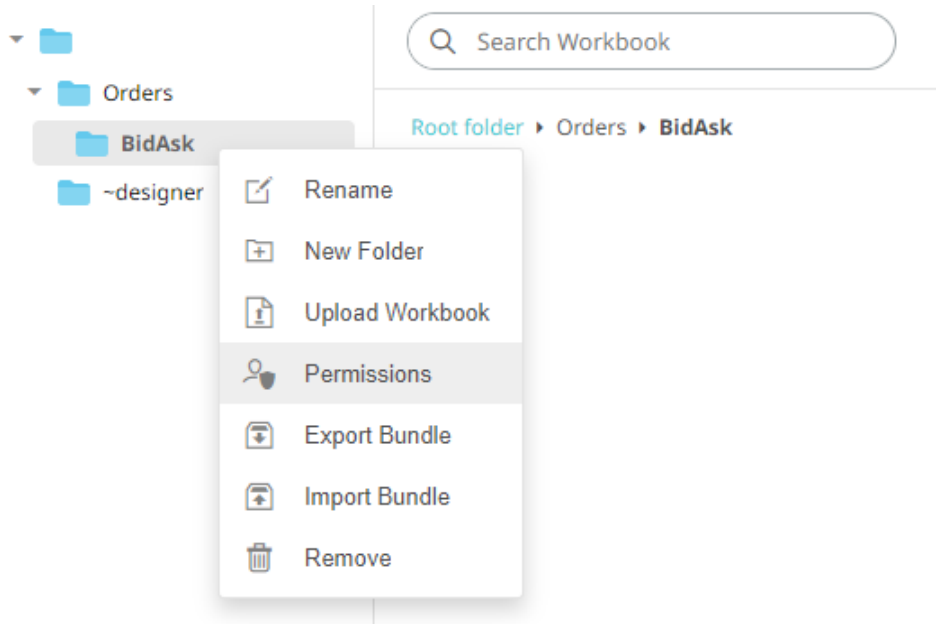
Steps:

1. To update folder properties, click a folder or a subfolder.

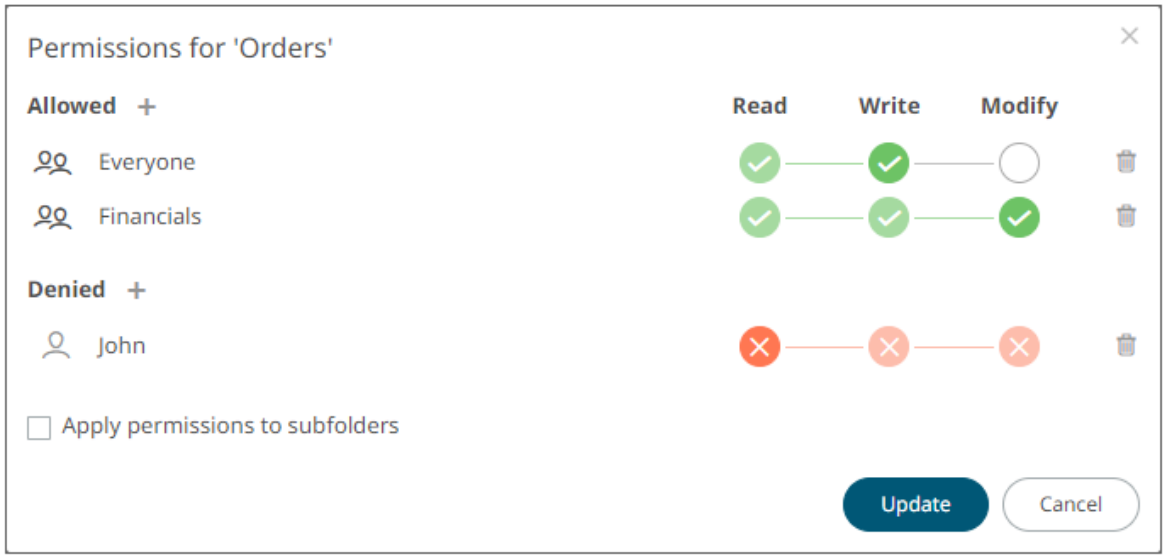




2. Right-click on the folder or subfolder and select **Permissions**.



The corresponding *Permissions* dialog displays.



3. Make the necessary changes such as new folder name, add or delete users and groups.
4. You can either:
   - check the **Apply Permissions to Subfolders** 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.

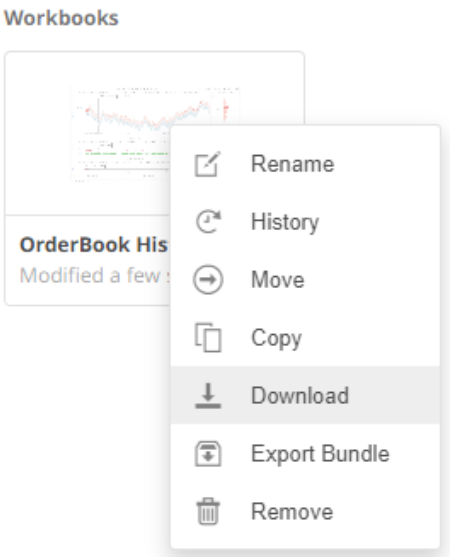
5. Click 

to save the changes.

Downloading a Workbook

A user with an Administrator or Designer role with READ + WRITE permission to the folder is allowed to download a copy of a workbook available in it.

Right-click on a workbook and select **Download** on the context menu.



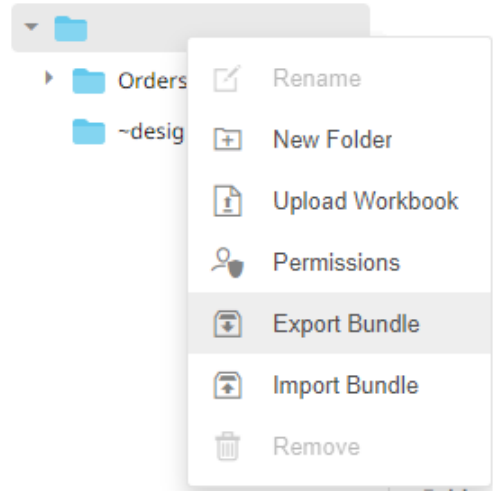
A copy of the workbook is downloaded.

Exporting Workbook or Folder Bundle

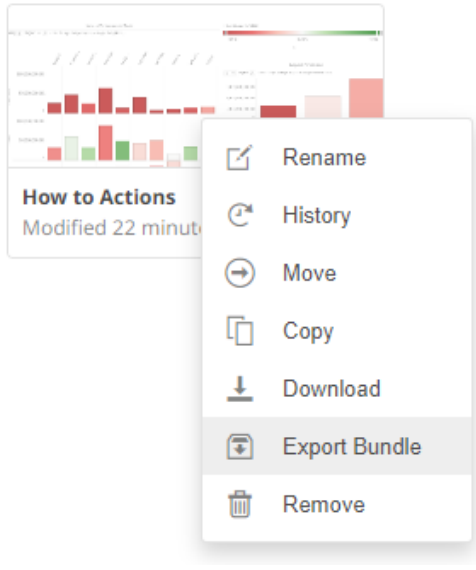
NOTE	• Data files associated with workbooks will only be included in the download if they are available inside the repository. • Users will only be able to download workbooks from folders where they have WRITE permission.

Steps:

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

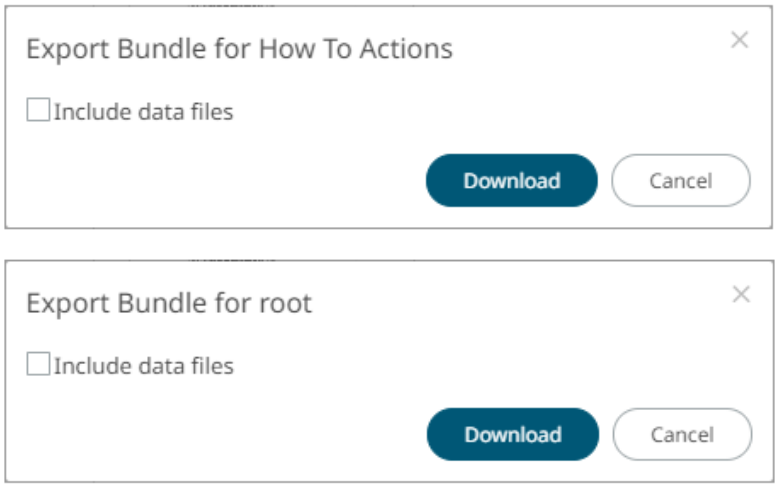


Workbook Folder or Subfolder Context Menu



Workbook Context Menu

A notification message displays.



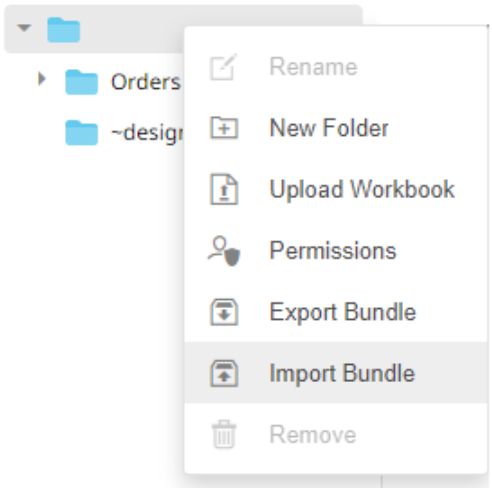
2. Check the **Include Data Files** box to include the associated workbook data files in the download.
3. Click Download. A copy of the workbook or folder bundle is downloaded.

Importing Workbooks Bundle

NOTE	• Users will only be able to import a bundle to folders where they have WRITE permission. • Existing workbooks with the same name as the uploaded workbooks will be archived, only if the new workbook differs from the current one. Consequently, the uploaded version will be the current one. • The bundle must not exceed the value set in the property `file.upload.size.max.bytes` in the `Panopticon.properties`.

Steps:

1. Right-click on a folder and select **Import Bundle** on the context menu.



The *Import Bundle* dialog displays.

2. To import a bundle, you can either:
   - drag it from your desktop and drop on 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 on the dialog box.

3. Check the **Keep Folder Structure** box.

   This means the exported folder structure is maintained when uploading the bundle. If the folders do not exist on the server, they will be created.

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

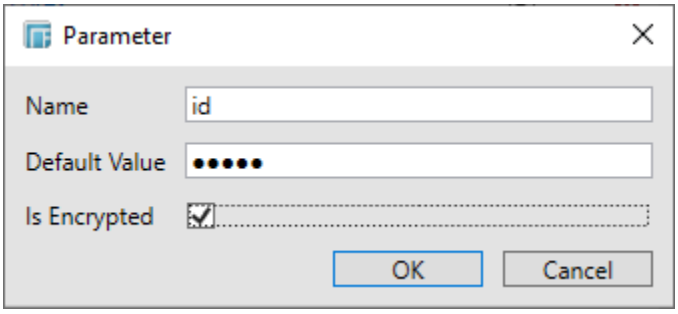
5. Click Import .

Data Level Secure Access

In this case the data being displayed is filtered to a particular authenticated user.

Data is filtered through the use of the special parameter **_user_id.**

This **_user_id** parameter is replaced at run time by the authenticated user id in lower case.

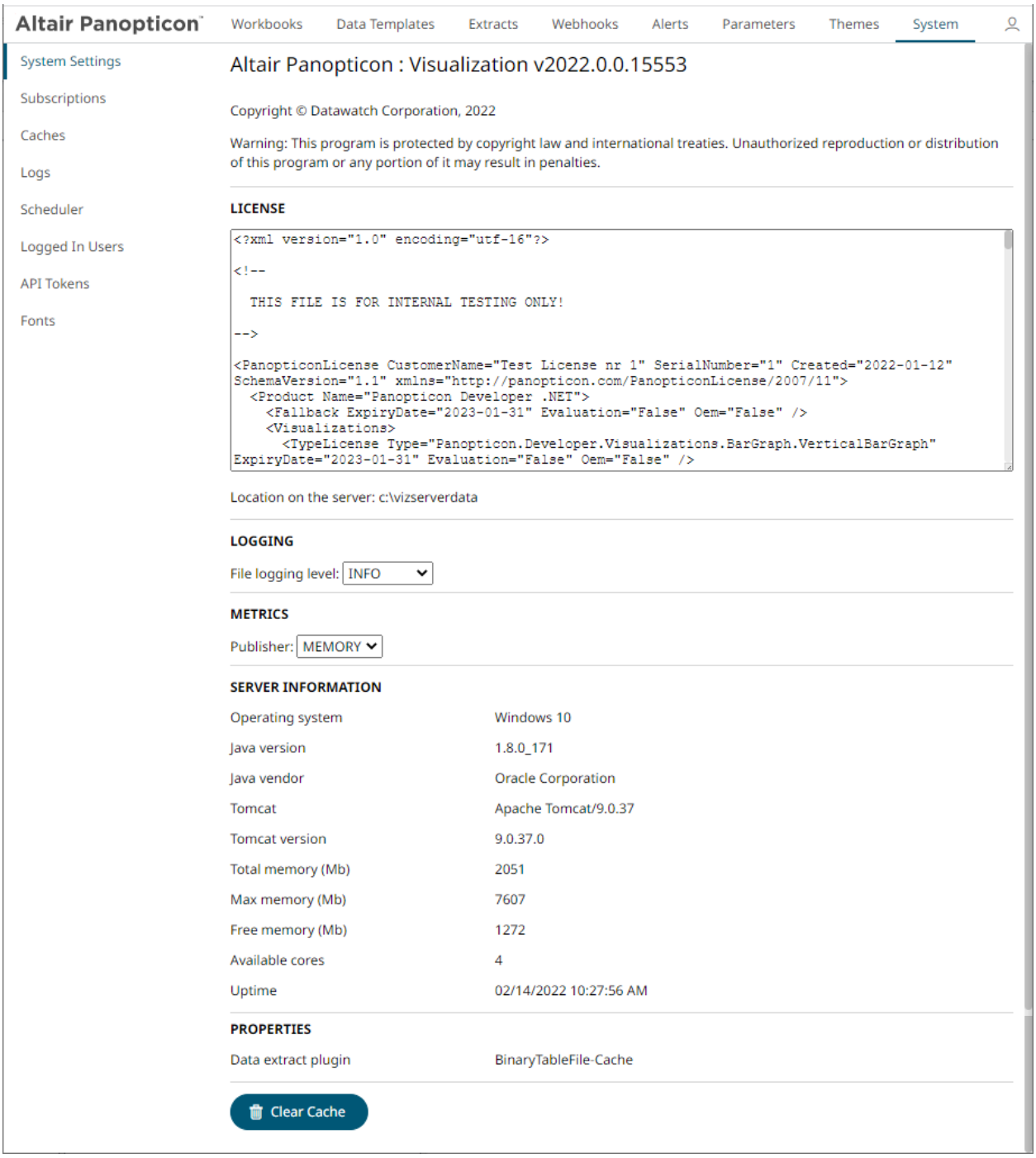


This parameter can then be used to restrict the data being retrieved, though use in either:

- Connection Details to Data Sources
- Filter constraints on data queries (e.g., SQL WHERE Clauses)

[7] SYSTEM ADMINISTRATION

Panopticon Real Time system administration is done on the following pages of the **System** tab:



Page	Description
System Settings	Allows to set file logging level and view the license and server information.
Subscriptions	Allows to view and manage real-time plugin subscriptions.
Caches	Allows to view, refresh, clear, or delete caches that are currently running on the server.
Logs	Allows to set the logging level and view logs. Also, pause or resume logging, and copy or clear logs.
Sheduler	Allows scheduling of email send outs and extracting of data.
Logged In User	Allows to view and manage logged in users.
API Tokens	Alows to add, delete, and view API Tokens.
Fonts	Allows to add custom fonts that can be used in a part or workbook.

SYSTEM SETTINGS

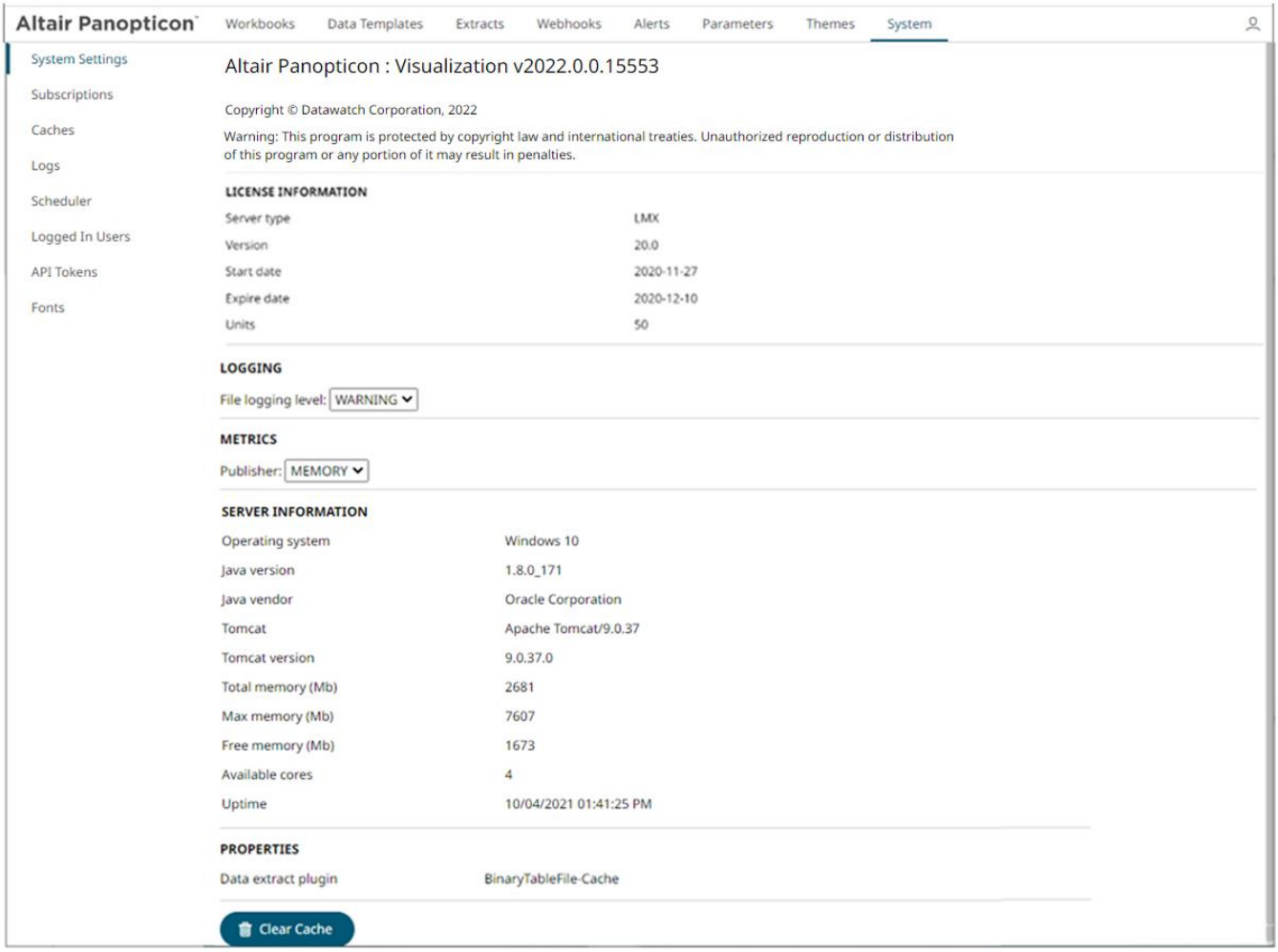
The *System Settings* page include the following panes or sections:

- License Information
- Logging Level
- Metrics Publisher
- Server Information

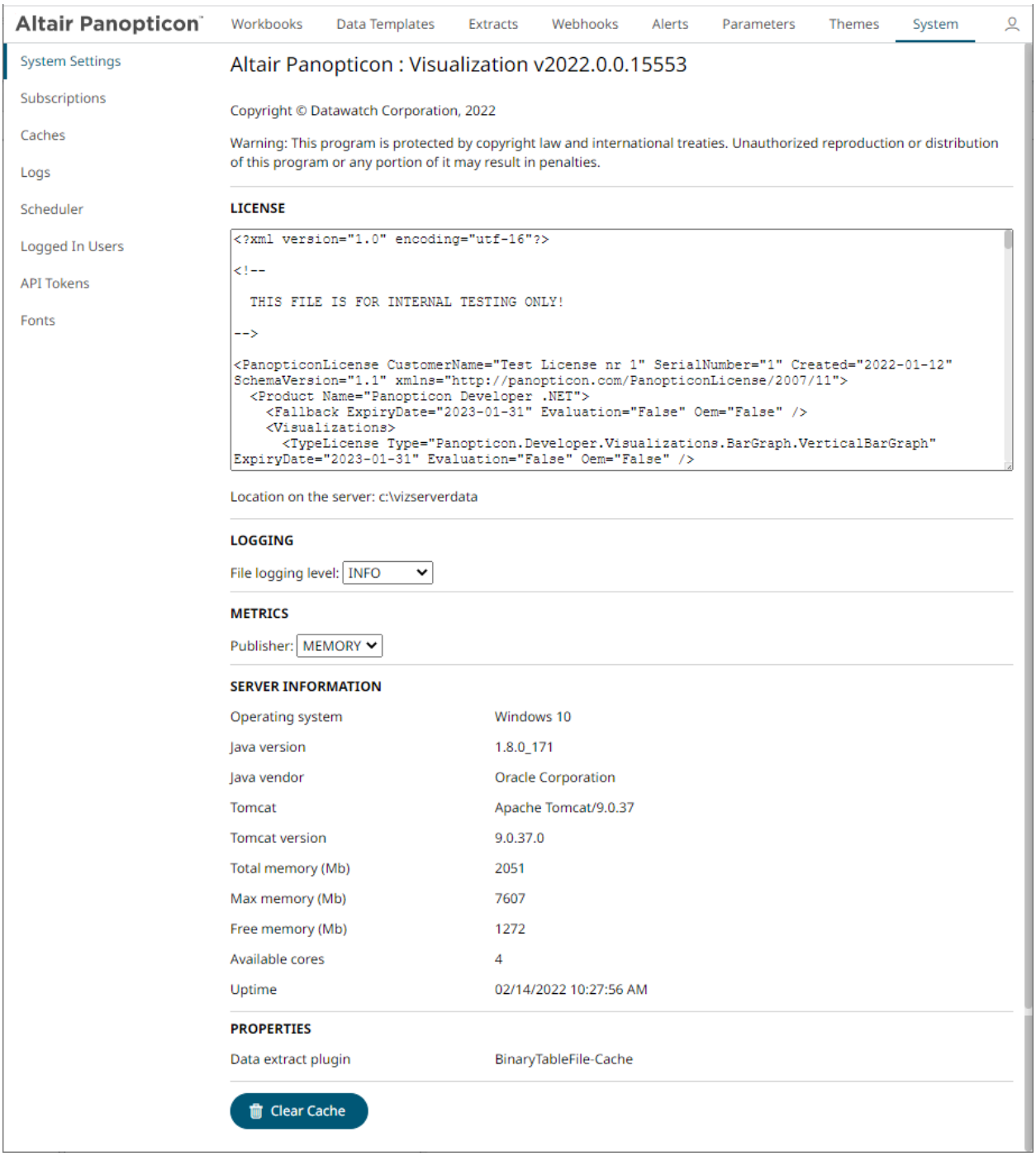
View License Information

If the licensing used is Altair Units license, the following license information are displayed:

- License server type
- License version
- Start Date and End Date of the license
- Total number of units available in the license

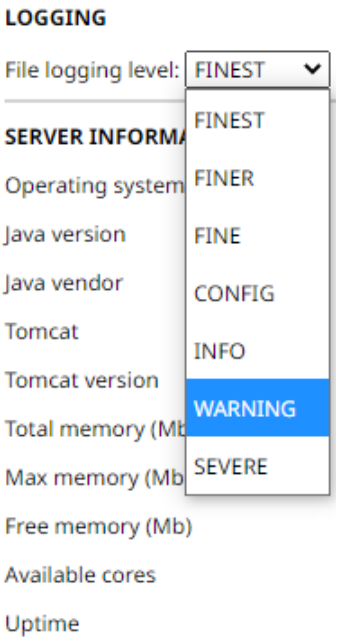


If the licensing used is the volume-based XML file (named **PanopticonLicense.xml**), the content and location (i.e., `c:\vizserverdata`) of the license are displayed.


Altair Panopticon
Workbooks
Data Templates
Extracts
Webhooks
Alerts
Parameters
Themes
System
System Settings
Subscriptions
Caches
Logs
Scheduler
Logged In Users
API Tokens
Fonts
Altair Panopticon : Visualization v2022.0.0.15553
Copyright © Datawatch Corporation, 2022
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
<?xml version="1.0" encoding="utf-16"?>
<!--
THIS FILE IS FOR INTERNAL TESTING ONLY!
-->
<PanopticonLicense CustomerName="Test License nr 1" SerialNumber="1" Created="2022-01-12" SchemaVersion="1.1" xmlns="http://panopticon.com/PanopticonLicense/2007/11">
<Product Name="Panopticon Developer .NET">
<Fallback ExpiryDate="2023-01-31" Evaluation="False" Oem="False" />
<Visualizations>
<TypeLicense Type="Panopticon.Developer.Visualizations.BarGraph.VerticalBarGraph" ExpiryDate="2023-01-31" Evaluation="False" Oem="False" />
Location on the server: c:\vizserverdata
LOGGING
File logging level: INFO
METRICS
Publisher: MEMORY
SERVER INFORMATION
Operating system Windows 10
Java version 1.8.0_171
Java vendor Oracle Corporation
Tomcat Apache Tomcat/9.0.37
Tomcat version 9.0.37.0
Total memory (Mb) 2051
Max memory (Mb) 7607
Free memory (Mb) 1272
Available cores 4
Uptime 02/14/2022 10:27:56 AM
PROPERTIES
Data extract plugin BinaryTableFile-Cache
Clear Cache

Setting the File Logging Level

The current set level (e.g., **FINEST**) is displayed. To change, click the drop-down list and select another log level.



The new log level is written in the **`Panopticon.properties`** file:

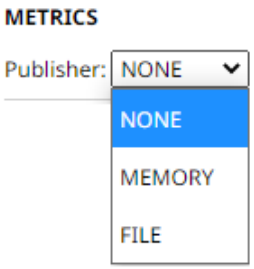
```
logger.level.file=WARNING
```

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

On the *Metrics* section of the *System Settings* page, select the *Publisher* of the server performance metrics.



Metrics Publisher	Description
None	No metrics are published.
Memory	Metrics are published to a queue in memory.
File	Metrics are published to a file on disk located in the `AppData/Metrics/` folder (i.e., `c:\vizserverdata\Metrics`).

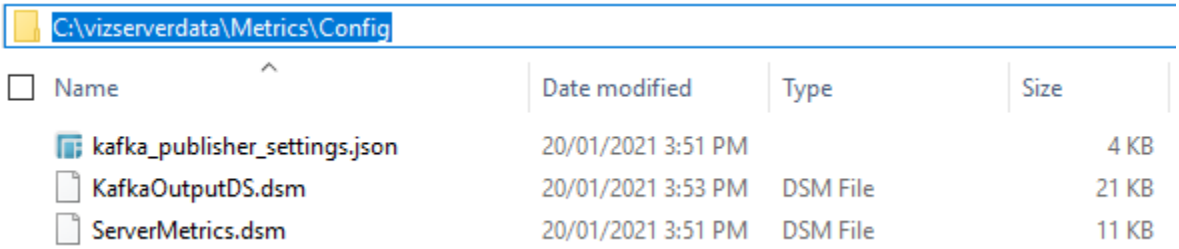
To add other Kafka publishers in the drop-down list, ensure their configuration file are available in the `AppData/Metrics/Config` folder.

A configuration file can be generated by creating a new data source in the Panopticon Streams Server and selecting any of the *Output* connectors. You can either:

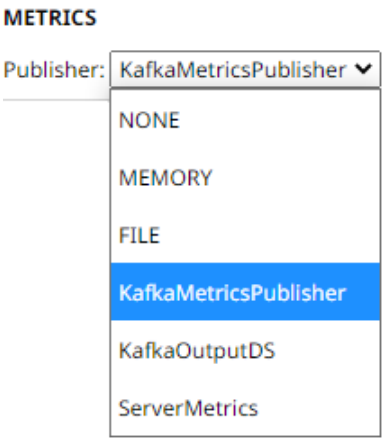
- export the JSON file from the repository, or
- download the DSM file by right-clicking the **Data Source** and clicking **Download** on the context menu

Refer to the Panopticon Streams Server Installation and Reference Guide for more information.

For example, when the generated Kafla data sources are added in the `AppData/Metrics/Config` folder:



The ID of the new configuration files are displayed in the *Publisher* drop-down list.



Selecting any of these specific Kafka data sources means that this is only place where metrics will be published to.

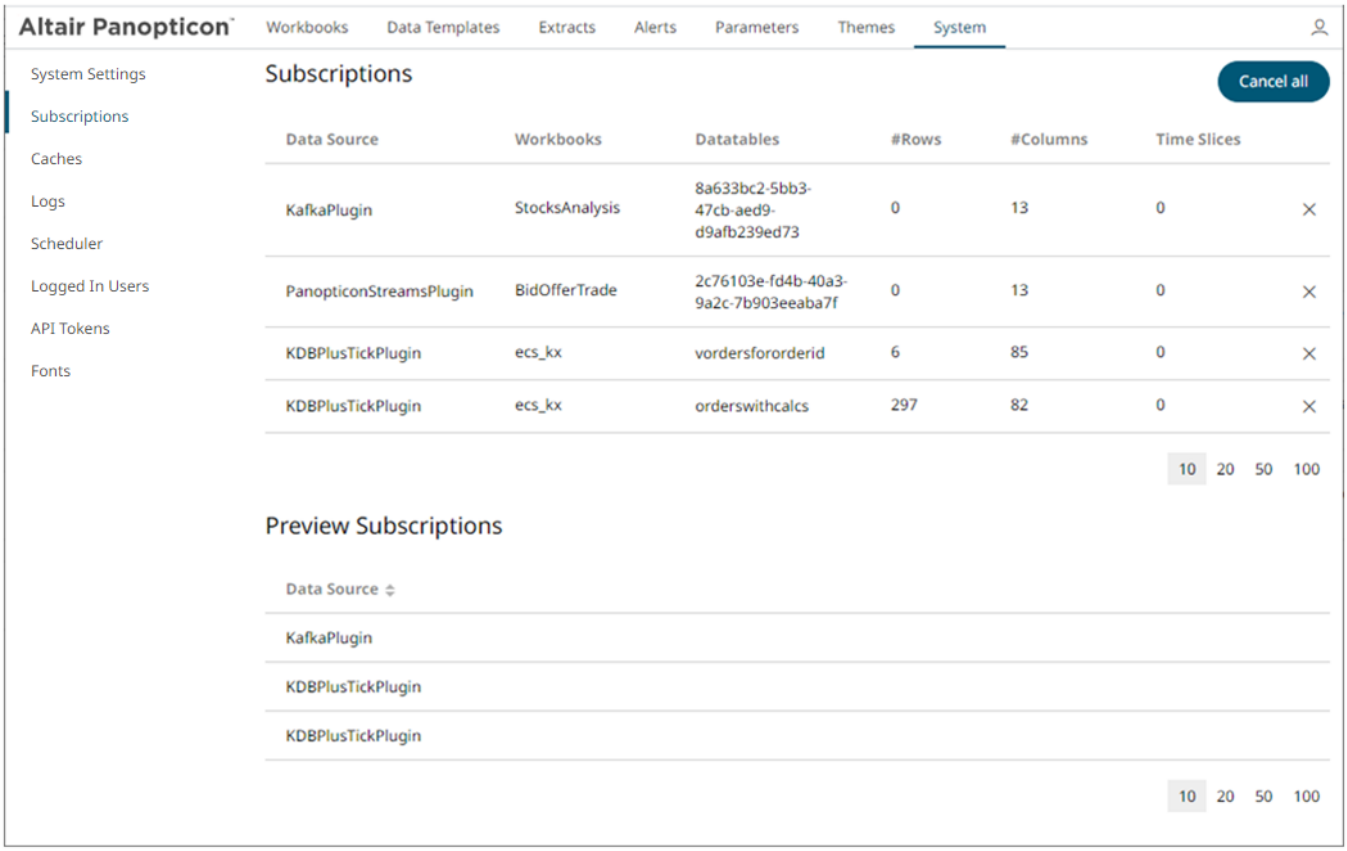
View Panopticon Real Time Information

On the *System Settings* page, the following server information are displayed:

Server Property	Description
Operating System	The server host operating system.
Java Version	The version of the Java Runtime Environment.
Java Vendor	The vendor of the Java Runtime Environment.
Tomcat	Identifies the Tomcat hosting the server
Tomcat Version	The Tomcat version.
Tomcat Memory (Mb)	The total amount of memory available to the Java Virtual Machine.
Max Memory (Mb)	The maximum amount of memory that the Java Virtual Machine will attempt to use.
Free Memory (mb)	The amount of free memory in the Java Virtual Machine.
Available Cores	The number of cores available to the Java Virtual Machine.
Uptime	The time when Panopticon Real Time was last started.

VIEW PLUGIN SUBSCRIPTIONS

View all of the currently running real-time plugin subscriptions.

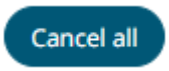


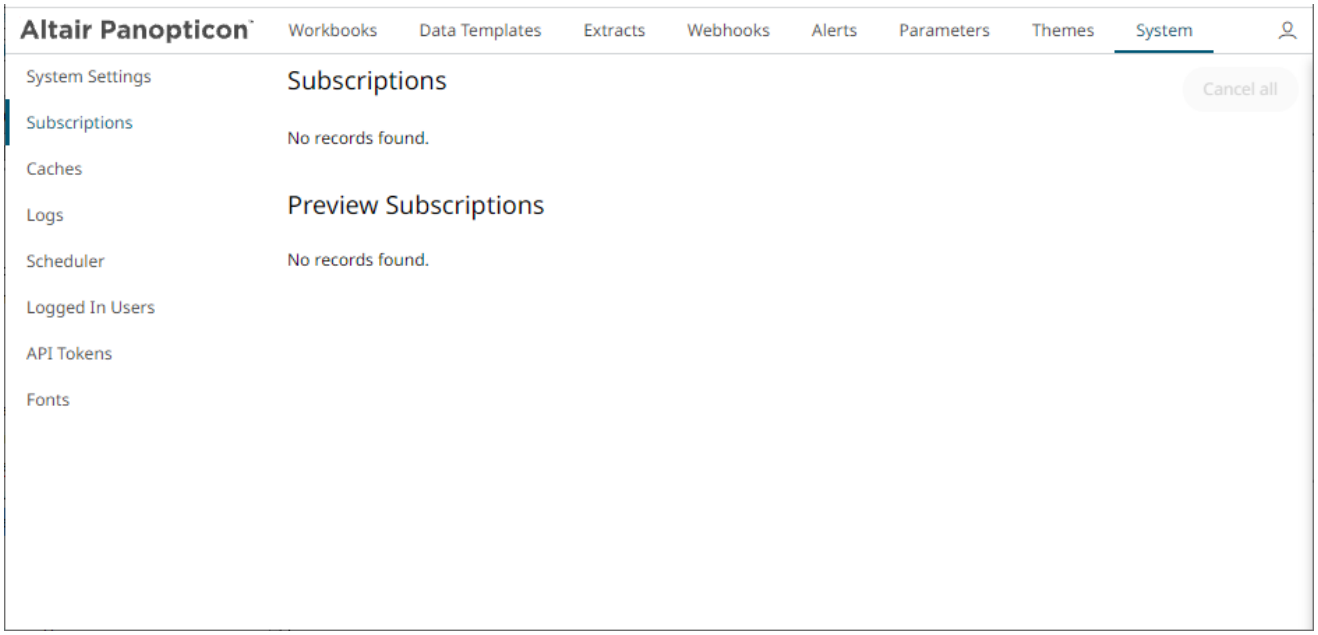
Including the following information:

- ❑ Data source with an installed plugin
- ❑ Workbook name
- ❑ Data table name
- ❑ Current size of the real-time table held by the plugin such as number of rows, columns, and time slices

For subscriptions created by ad hoc services, or those with no owner (workbook reference), they can be viewed on the *Preview Subscriptions* section.

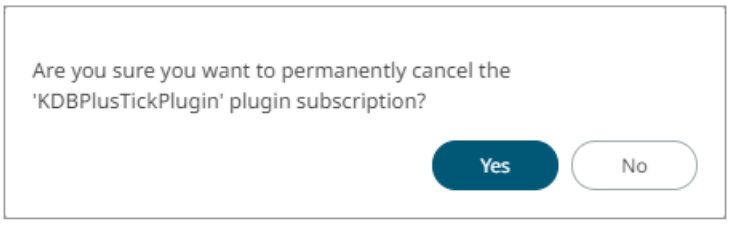
You can also opt to do any of the following:

- ❑ Click **Cancel All**  to cancel all of the subscriptions on the *Subscriptions* section.



- Cancel a plugin subscription by clicking ×

A notification message displays.

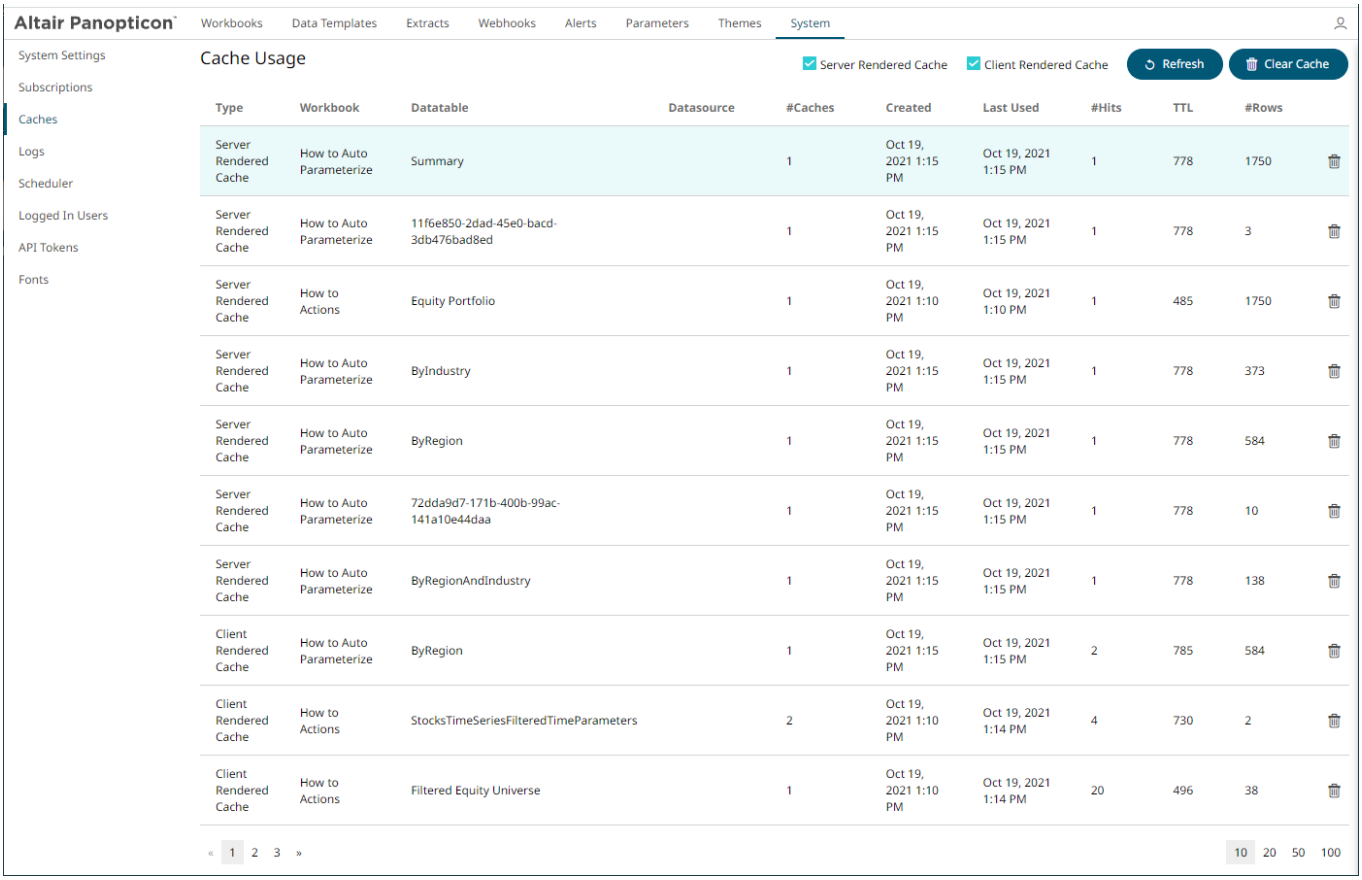


Click Yes to cancel.

- Move to other pages

VIEW CACHE USAGE

View the caches currently in use on the server.



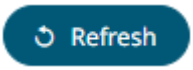
Including the following information:

- ❑ Cache Rendering Type
- ❑ Workbook Name
- ❑ Data Table Name
- ❑ Cache Usage Count
- ❑ Date Created
- ❑ Date Last Used
- ❑ Cache Usage Hits
- ❑ The Time to Live
- ❑ Row Count of the cached dataset

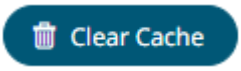
You can also opt to do the following:

- ❑ Clear Cache and refresh page
- ❑ Display Server-rendered Cache

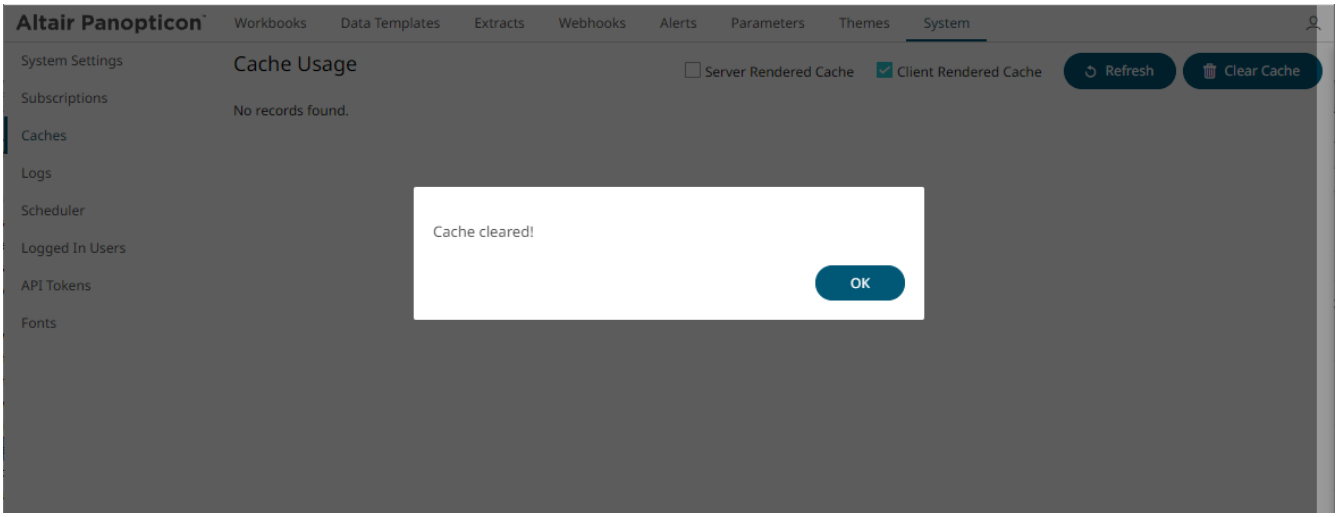
- ❑ Display Client-rendered Cache
- ❑ Refresh the Cache Usage list
- ❑ Move to other pages
- ❑ Delete a Cache Usage instance in the list

Click the **Refresh** 

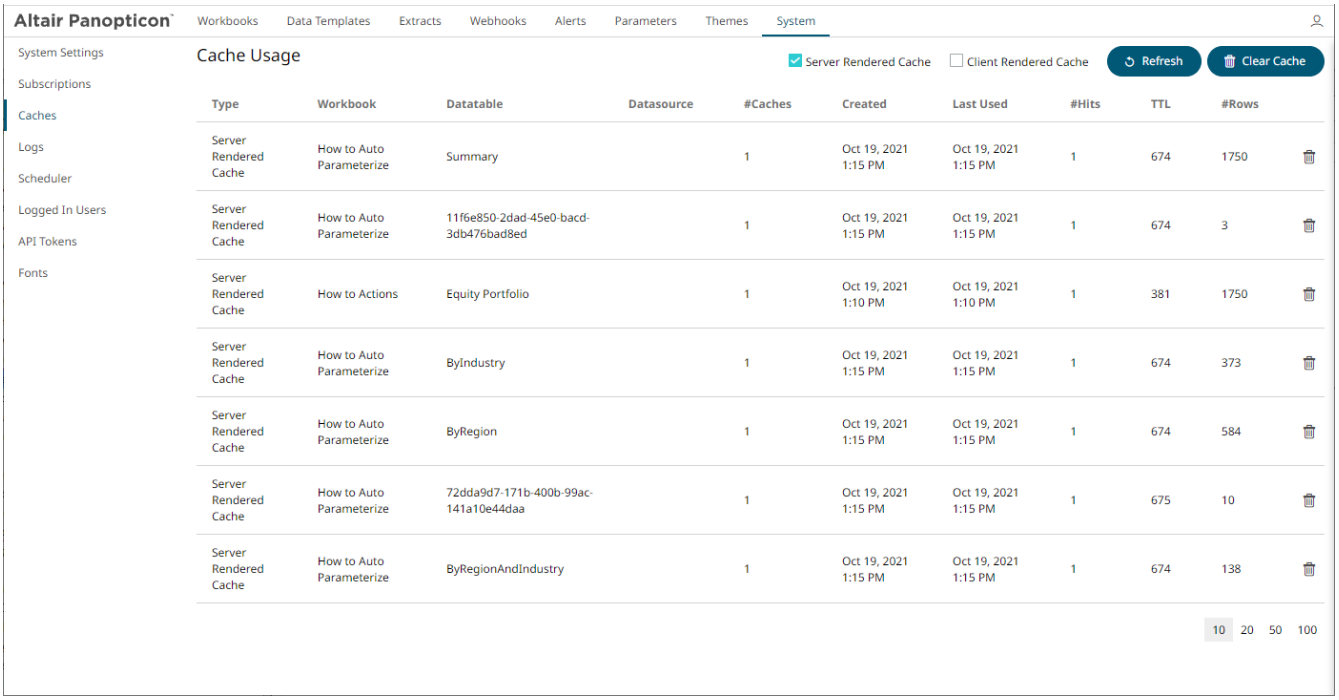
button to refresh the list.

Clicking **Clear Cache** 

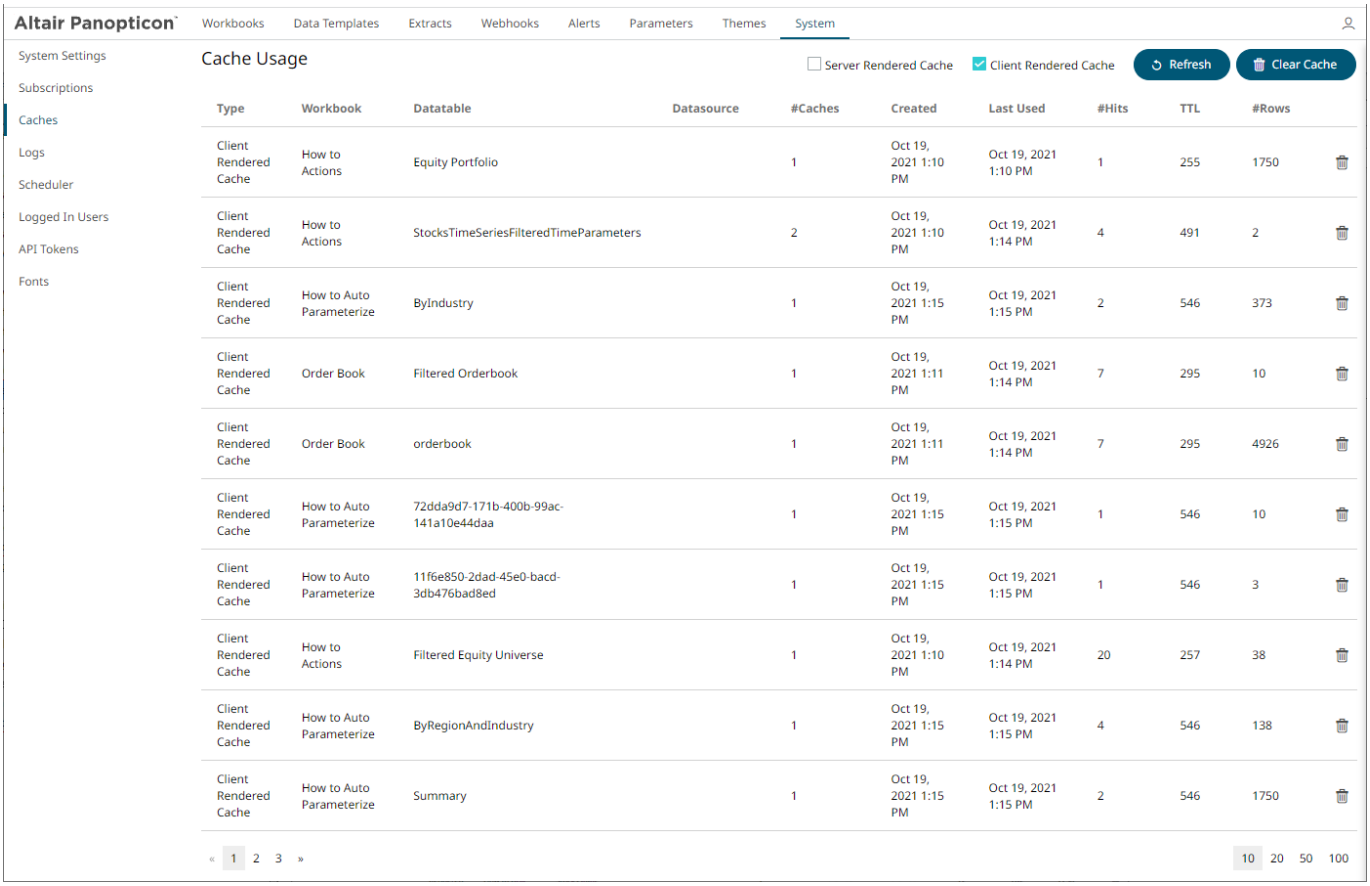
will clear all caches of data, ensuring that any subsequent workbook access that utilizes a cache, will cause a cache reload.



Checking the *Server Rendered Cache* box displays the list of server-rendered caches.



Checking the *Client Rendered Cache* box displays the list of client-rendered caches.



Click the **Delete Cache Usage** icon of a cache usage in the list.

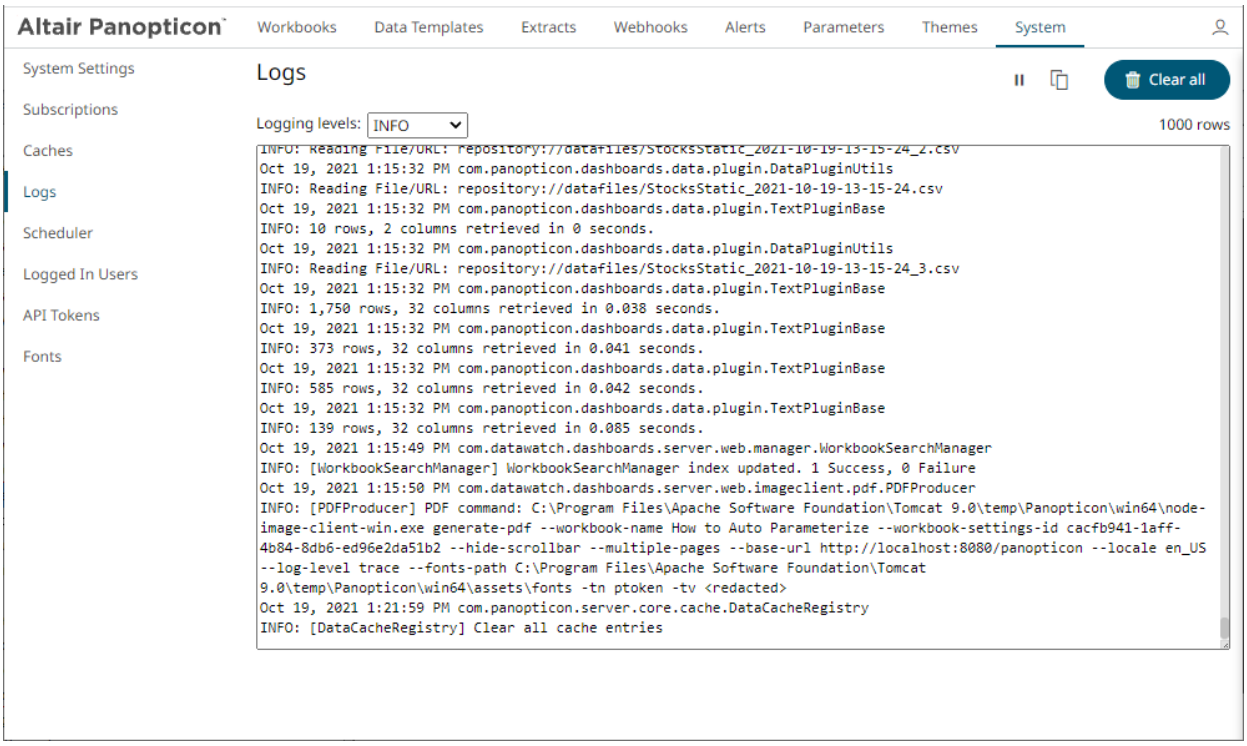
VIEW LOGS

View the latest 300 rows of a *Logging Level* in the **Logs** tab:

- ❑ FINEST (lowest level)
- ❑ FINER
- ❑ FINE
- ❑ CONFIG
- ❑ INFO (default level)
- ❑ WARNING
- ❑ SEVERE (highest level)

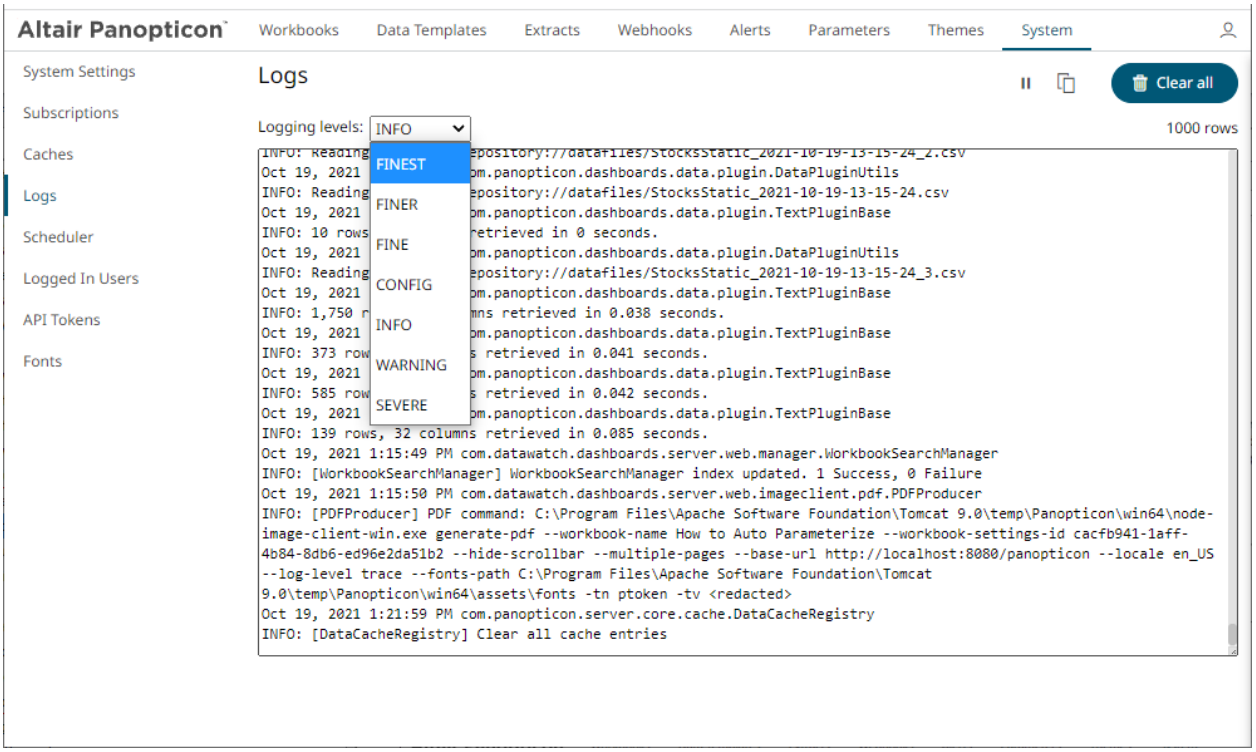
Steps:

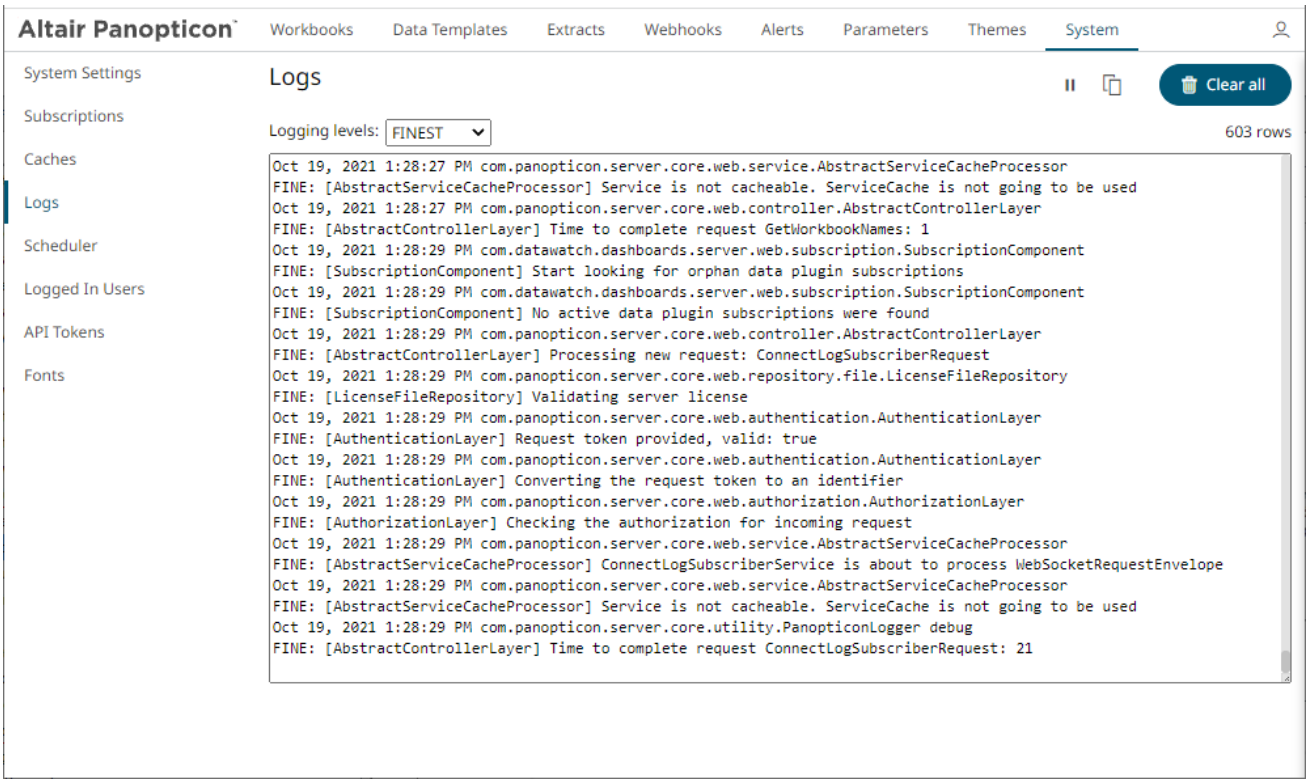
1. On the *System* page, click the **Logs** tab. Initially, the default level (**INFO**) logs are displayed.



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

   For example: FINEST





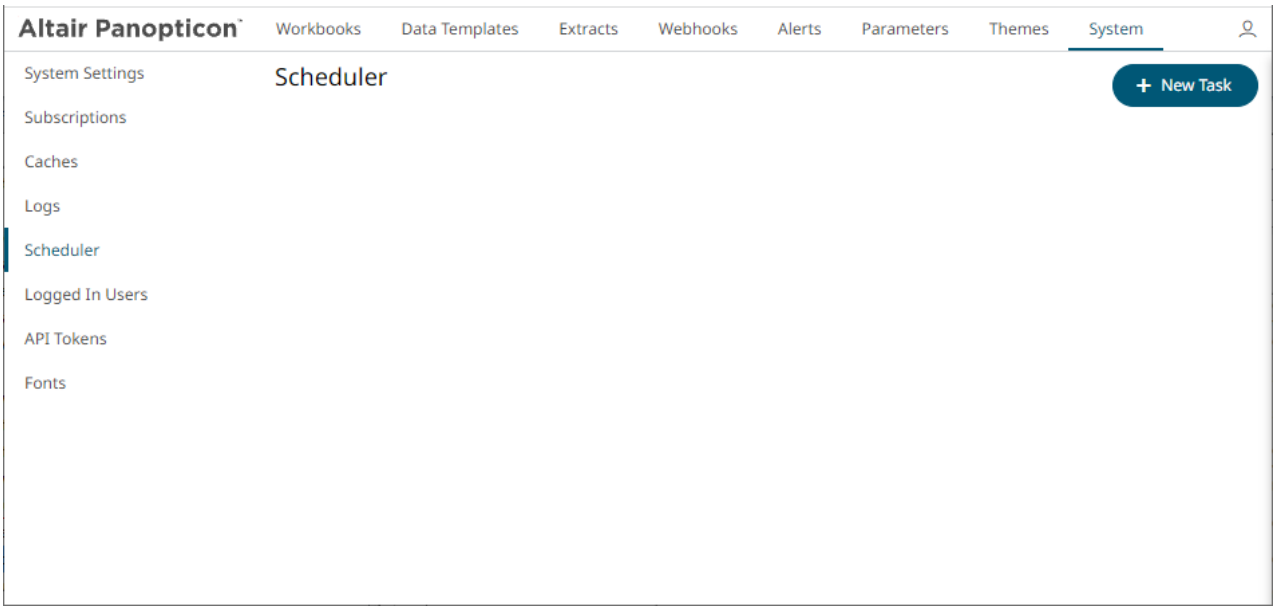
The latest 1000 rows of the selected log level or higher are fetched.

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

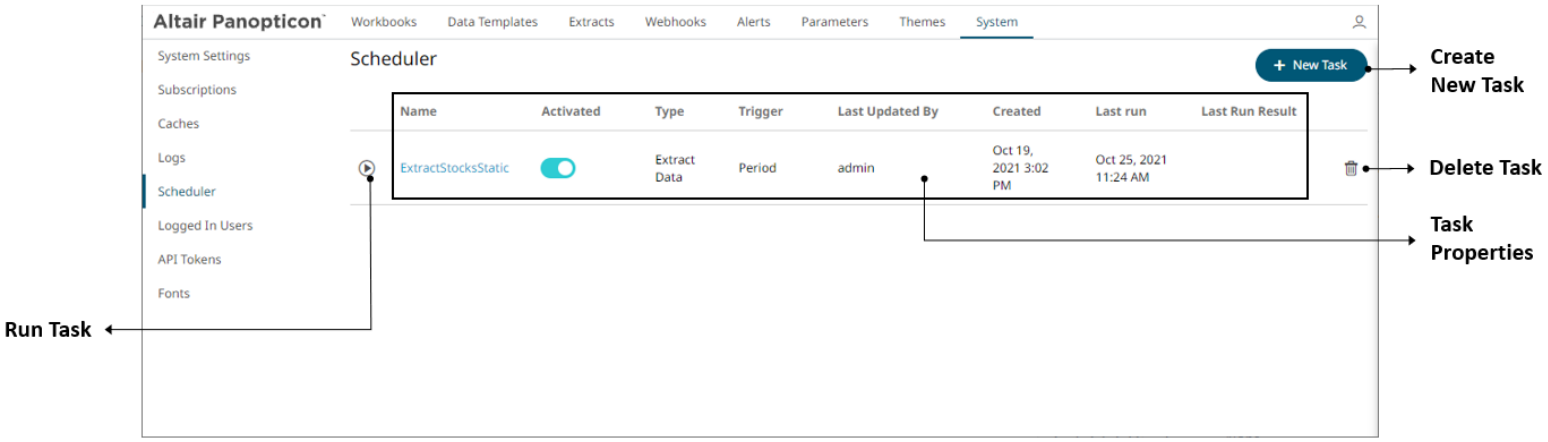
- to pause the logging, it changes to
- to resume the logging
- to copy log to clipboard
- Clear all to clear the logs

SCHEDULING TASKS

On the **Scheduler** tab of the *System Settings* page, Panopticon Real Time allows scheduling of tasks.



A new scheduled task is added in the list with the properties.

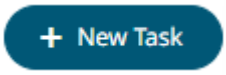


Scheduler Property	Description
Run Task	Manually run scheduled task.
Create New Task	Create a new scheduled task.
Delete Task	Delete a scheduled task.
Task Properties	The task properties include: • Name of the task • Activated status

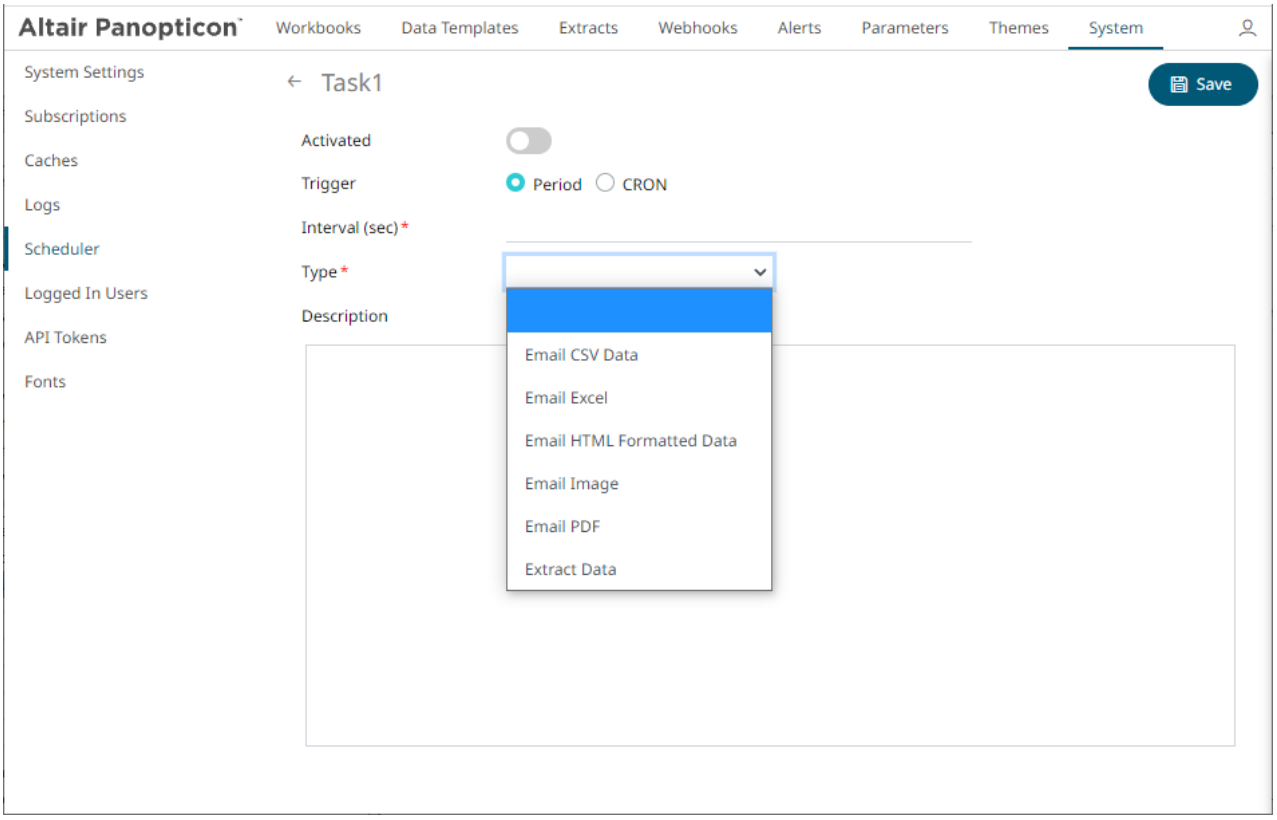
- Type of the scheduled task
- Trigger type: **Period** or **CRON**
- Last user who made an update
- Date/Time when the task was created
- Date/Time when the task was last ran
- Last run result: **Success** or **Failed**

  For failed results, you can hover on the tooltip to view the error.

  NOTE: The result is not displayed for **Extract Data** scheduler type.

To create a new task, click **New Task** 

. The *New Task* pane displays that allows you to define the following tasks:



- ❑ Sending of a CSV Data via Email
- ❑ Sending of an MS Excel file via Email
- ❑ Sending of an HTML Formatted Data via Email
- ❑ Sending of an Image file via Email
- ❑ Sending of a PDF file via Email
- ❑ Extracting Data

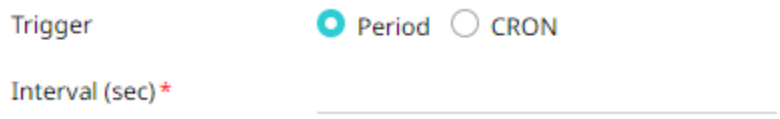
NOTE	
	• To allow scheduling of email send outs, Panopticon Real Time must be configured with valid email server information in the `Panopticon.properties` file located in the `AppData` folder (e.g., `c:\vizserverdata`). See Panopticon Real Time Configurations for Email Send Outs and Alerts for instructions. • If any data load fails, such as when the data source is offline, the PDF/Image generation fails as well and an email will not be sent.

Create Task to Send CSV Data via Email

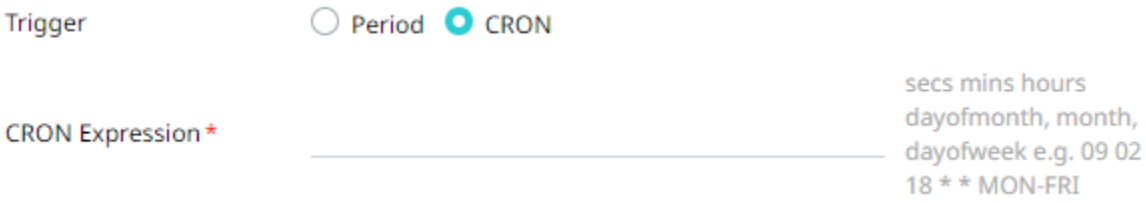
Panopticon Real Time provides the functionality to create tasks to generate and email CSV data from a workbook, dashboard, or visualization.

Steps:

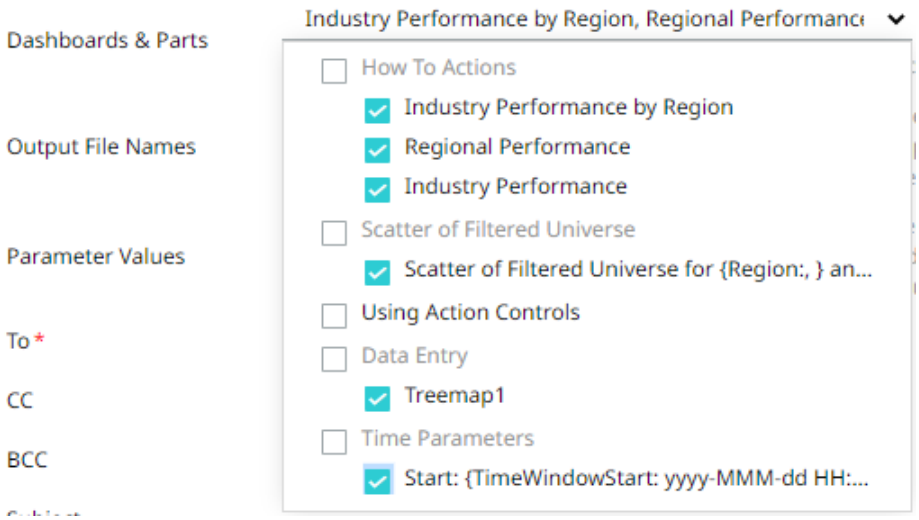
1. On the *New Task pane*, enter the *Name* of the task and click ✓. Ensure the name is unique.
2. Tap the *Activated* slider to turn it on.
3. Select the *Trigger.* You can either select:
   - **Period** then enter the *Interval* (in seconds), or



   - **CRON** then enter the CRON Expression.



4. Select the task *Type*: **Email CSV Data**.
5. Enter the *Description* of the task.
6. Upon selecting **Email CSV Data**, the *Scheduler* page changes to allow specification of the following:
   - Select the *Workbook Name* in the drop-down list. These are the published workbooks available in the *Workbooks* page.
   - Select dashboards or parts where to source the CSV data from, by checking their corresponding boxes in the *Dashboards & Parts* drop-down list.



- enter comma-separated list of *Output File Names*.

NOTE	
	• The items in the list must be either unique or empty. • Empty string items indicate that the default title should be used. • By default, the text box is blank causing the implicit naming to be used. • If the supplied names are fewer than the selected data sets, the default naming comes into effect for non-specified names

- You can also opt to enter the *Parameter Values* that will be added as parameters to the subject line of emails or as dashboard values in the CSV Data.

  Such as `Parameter=Value`, and are comma separated. For example:

  ```
  Region=Europe,Industry=Financials
  ```

NOTE	
	The following Date/Time range querying parameters are also supported in the Email CSV Data task: • CurrentTime • LastWorkDay • WeekStart • QuarterStart For example: `{CurrentTime:dd-MMM-yyyy}` However, when there is no Date/Time format supplied, the default format `yyyy-MM-dd` will be used instead.

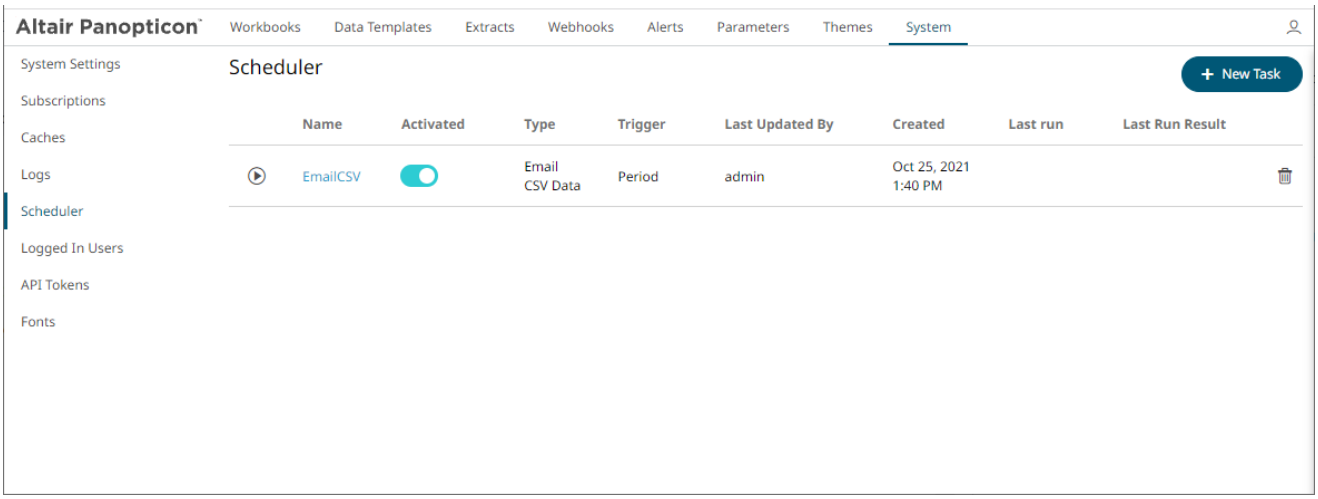
7. Enter the email address of the recipient in the *To* field.

8. You can opt to enter the following:
   - the *CC* and/or *BCC* recipients of the email separated by a comma.
   - the mail message subject to be used in the email notifications in the *Subject* field.

NOTE	Supports dashboard parameters.

   - the content of the email in the *Body* box.
9. Click Save.

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

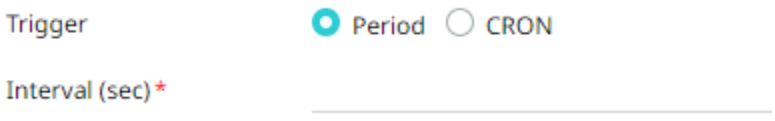


Create Task to Send an MS Excel File via Email

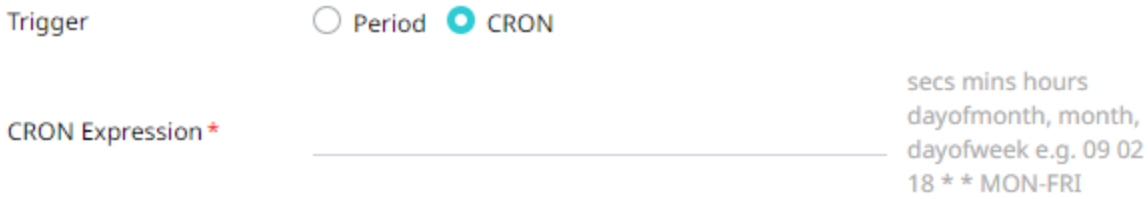
Panopticon Real Time provides the functionality to create tasks to generate and email MS Excel files.

Steps:

1. On the *New Task pane*, enter the *Name* of the task and click ✓. Ensure the name is unique.
2. Tap the *Activated* slider to turn it on.
3. Select the *Trigger.* You can either select:
   - **Period** then enter the *Interval* (in seconds), or



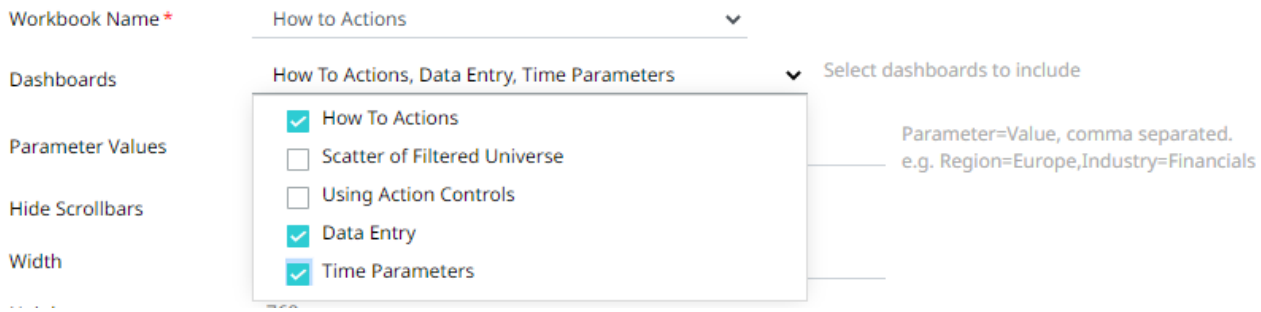
   - **CRON** then enter the CRON Expression.



4. Select the task *Type*: **Email Excel**.

   Upon selecting **Email Excel**, the *Scheduler* page changes to allow specification of the following:

   - Enter the *Description* of the task.
   - Select the *Workbook Name* in the drop-down list. These are the published workbooks available in the *Workbooks* page.
   - Select the dashboards to include in the MS Excel file by checking their corresponding boxes in the *Dashboards* drop-down list.



NOTE	Multiple tables per dashboard are inserted in a sheet of the MS Excel file.

5. You can also opt to:

   - enter the *Parameter Values* that will be added as parameters to the subject line of emails or as dashboard values in the MS Excel file

     Such as `Parameter=Value`, and are comma separated. For example:

     ```
     Region=Europe,Industry=Financials
     ```

NOTE	The following Date/Time range querying parameters are also supported in the Email Excel task: • CurrentTime • LastWorkDay • WeekStart • QuarterStart For example: `{CurrentTime:dd-MMM-yyyy}` However, when there is no Date/Time format supplied, the default format `yyyy-MM-dd` will be used instead.

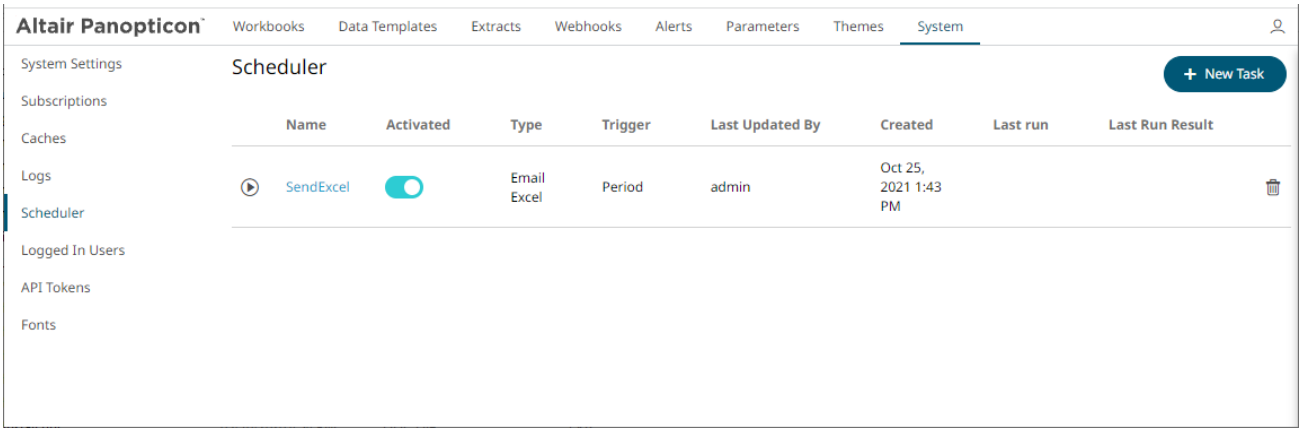
- check the **Hide Scrollbars** box.

6. Enter the *Width* and *Height* of the MS Excel file. Default values are **1024px** and **768px,** respectively.
7. Enter the *Table Style*. Default is **TableStyleMedium4**.
8. Enter the email address of the recipient in the *To* field.
9. You can opt to enter the following:
   - the *CC* and/or *BCC* recipients of the email separated by a comma.
   - the mail message subject to be used in the email notifications in the *Subject* field.

NOTE	Supports dashboard parameters.

   - the content of the email in the *Body* box.
10. Click Save.

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



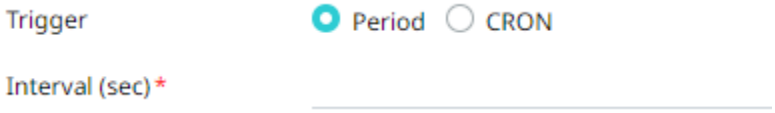
Create Task to Send an HTML Formatted Data via Email

Panopticon Real Time provides the functionality to create tasks to generate and email HTML-formatted table exported from a selected workbook and dashboards.

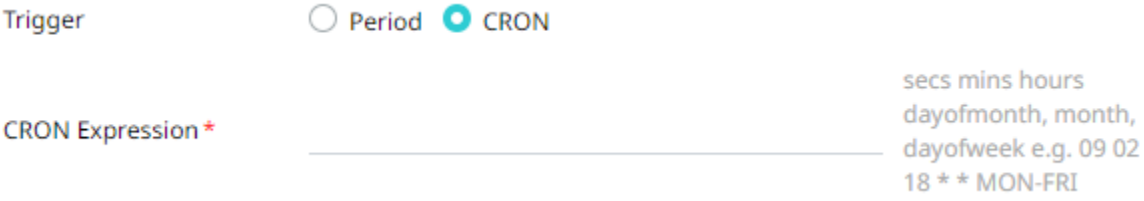
IMPORTANT	Use with caution! When emailing HTML formatted data, the email message size runs the risk of becoming very large if the data used in the visualization is too large and/or complex. The data volume will not stop Panopticon Real Time from creating the message and the HTML-formatted data, but email servers may struggle to send and/or receive the message.

Steps:

1. On the *New Task pane*, enter the *Name* of the task and click ✓. Ensure the name is unique.
3. Tap the *Activated* slider to turn it on.
4. Select the *Trigger.* You can either select:
   - **Period** then enter the *Interval* (in seconds), or



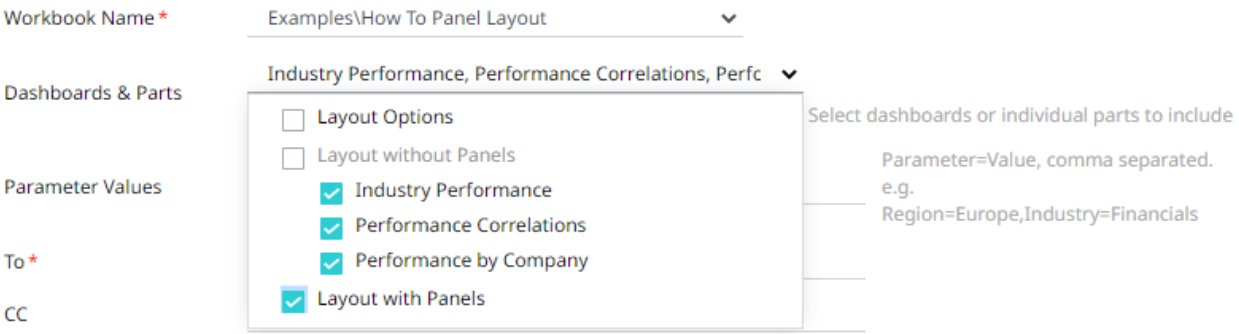
   - **CRON** then enter the CRON Expression.



5. Select the task *Type*: **Email HTML Formatted Data**.

   Upon selecting **Email HTML Formatted Data**, the *Scheduler* page changes to allow specification of the following:

   - Enter the *Description* of the task.
   - Select the *Workbook Name* in the drop-down list. These are the published workbooks available in the *Workbooks* page.
   - Select the dashboards and parts to include in the HTML formatted data file by checking their corresponding boxes in the *Dashboards & Parts* drop-down list.



6. You can also opt to enter the *Parameter Values* that will be added as parameters to the subject line of emails or as dashboard values in the MS Excel file.

   Such as `Parameter=Value`, and are comma separated. For example:

   ```
   Region=Europe,Industry=Financials
   ```

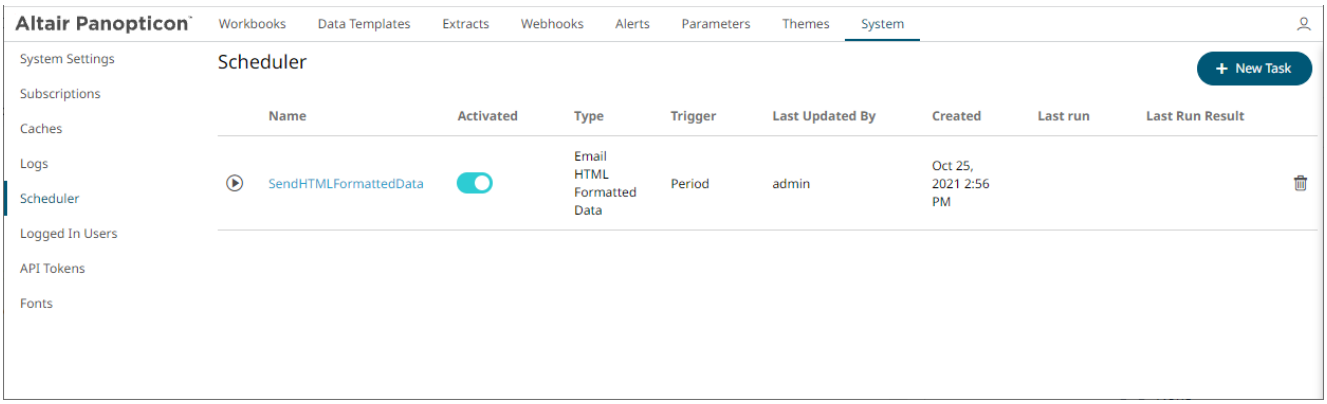
NOTE	The following Date/Time range querying parameters are also supported in the Email Formatted Data task: • CurrentTime • LastWorkDay • WeekStart • QuarterStart For example: `{CurrentTime:dd-MMM-yyyy}` However, when there is no Date/Time format supplied, the default format `yyyy-MM-dd` will be used instead.

7. Enter the email address of the recipient in the *To* field.
8. You can opt to enter the following:
   - the *CC* and/or *BCC* recipients of the email separated by a comma.
   - the mail message subject to be used in the email notifications in the *Subject* field.

NOTE	Supports dashboard parameters.

   - the content of the email in the *Body* box.
9. Click Save.

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



Create Task to Send Image File via Email

Panopticon Real Time provides the functionality to create tasks to generate and email Image files.

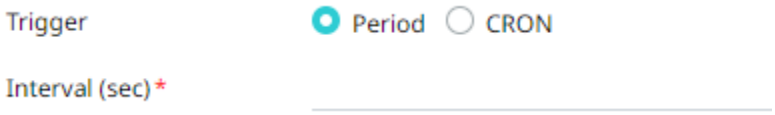
In addition, hyperlinks can also be used in email dashboard images. Hyperlinks can redirect to a workbook and a dashboard in the server.

NOTE	In cases when you schedule the emailing of dashboard images or when you are behind a proxy or load balancer, it is recommended to specify the server address in the `Panopticon.properties` file. For example: `server.host=http://www.company.com/dashboards`

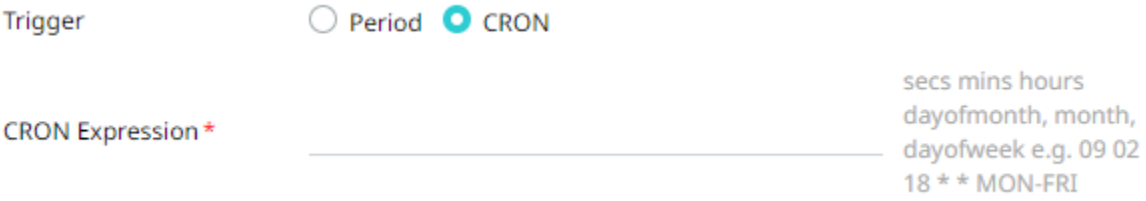
See the Email Data: Image section for more information.

Steps:

1. On the *New Task pane*, enter the *Name* of the task and click ✓. Ensure the name is unique.
2. Tap the *Activated* slider to turn it on.
3. Select the *Trigger.* You can either select:
   - **Period** then enter the *Interval* (in seconds), or



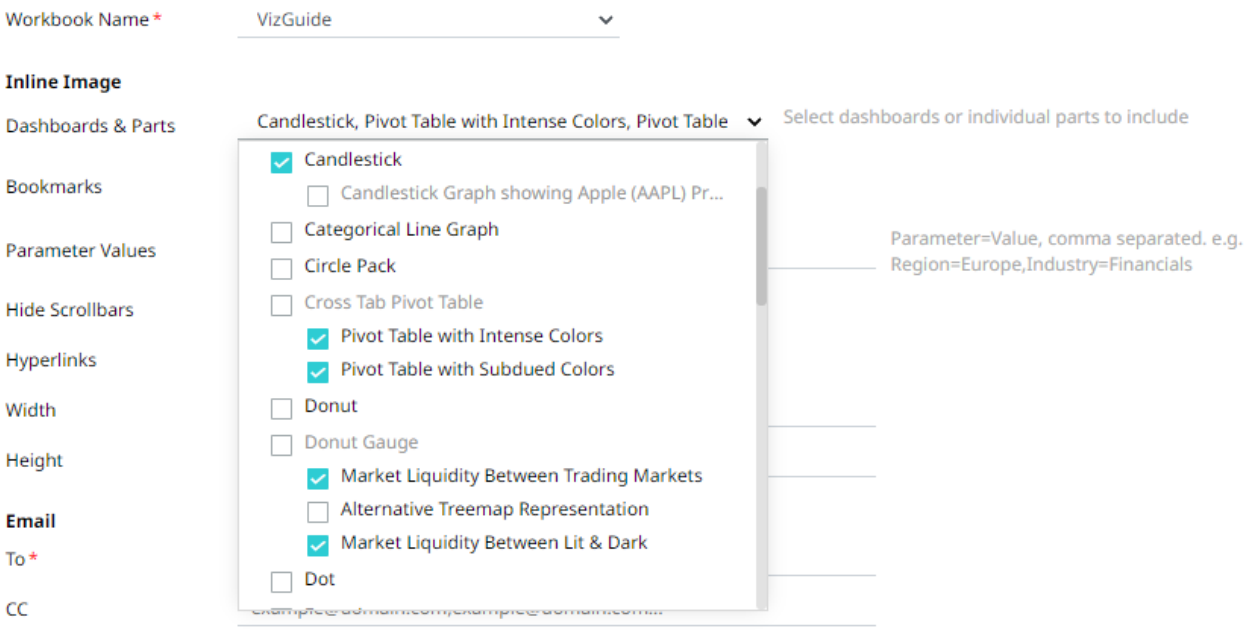
   - **CRON** then enter the CRON Expression.



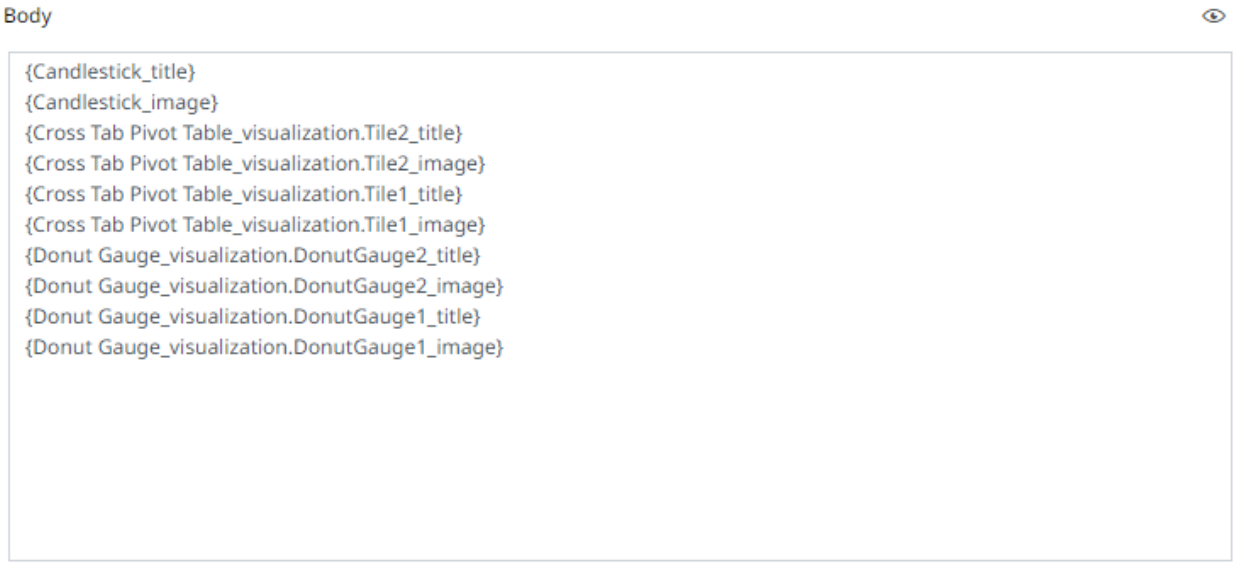
4. Select the task *Type*: **Email Image**.

   Upon selecting **Email** Image, the *Scheduler* page changes to allow specification of the following:

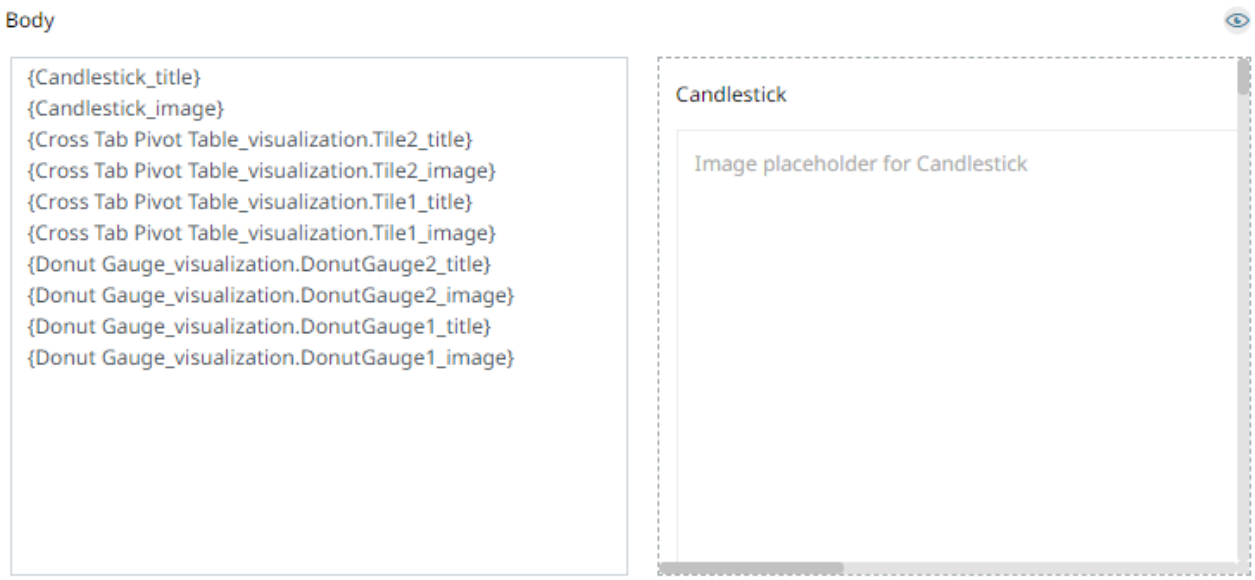
   - Enter the *Description* of the task.
   - Select the *Workbook Name* in the drop-down list. These are the published workbooks available in the *Workbooks* page.
   - Select dashboards or parts to include in the image file by checking their corresponding boxes in the *Dashboards & Parts* drop-down list.



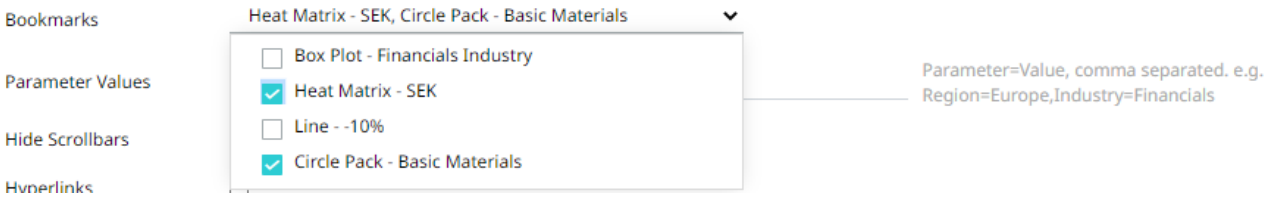
The selected dashboards or parts are inserted as parameterized text and inline images in the *Body* edit box.



Clicking the **Preview** icon displays the image placeholders for the selected dashboards or parts.



- Select bookmarks in the workbook to include in the image file by checking their corresponding boxes in the *Bookmarks* drop-down list.



5. You can also opt to:

- enter the *Parameter Values* that will be added as parameters to the subject line of emails or as dashboard values in the Image file

  Such as `Parameter=Value`, and are comma separated. For example:

  `Region=Europe,Industry=Financials`

NOTE	The following Date range querying parameters are also supported in the Email Image task: • CurrentTime • LastWorkDay • WeekStart • QuarterStart For example: `{CurrentTime:dd-MMM-yyyy}` However, when there is no Date/Time format supplied, the default format `yyyy-MM-dd` will be used instead.

- check the **Hide Scrollbars** box.
- check the **Hyperlinks** box. This makes the Image file in the email will be clickable.

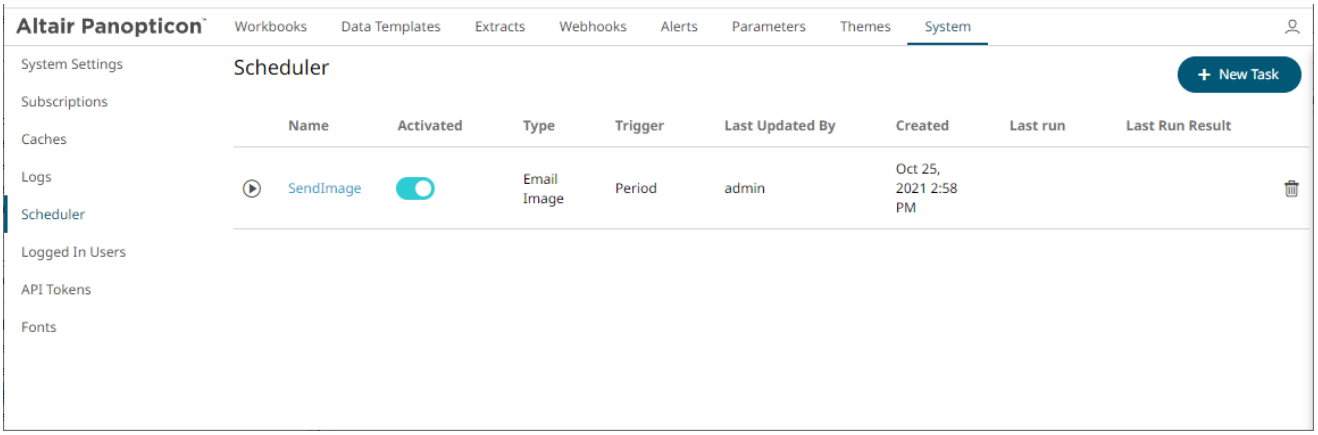
6. Enter the *Width* and *Height* of the Image file. Default values are **1024** and **768,** respectively.
7. Enter the email address of the recipient in the *To* field.
8. You can opt to enter the following:
   - the *CC* and/or *BCC* recipients of the email separated by a comma.
   - the mail message subject to be used in the email notifications in the *Subject* field.

NOTE	Supports dashboard parameters.

   - the content of the email in the *Body* box.
9. Click 

.

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

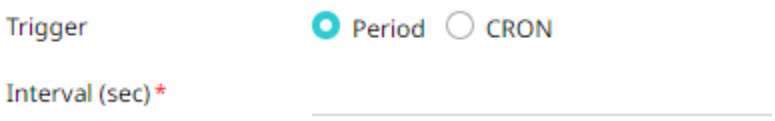


Create Task to Send PDF File via Email

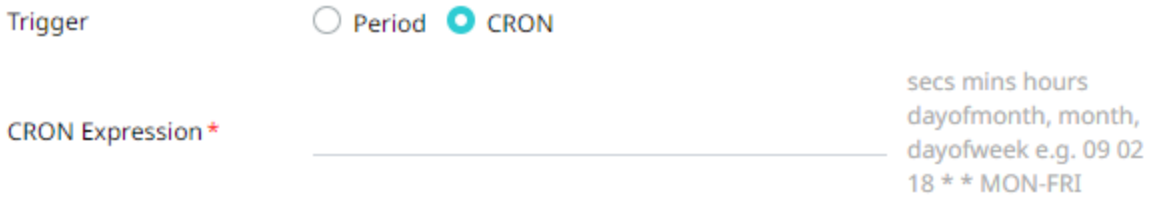
Panopticon Real Time provides the functionality to create tasks to generate and email PDF files.

Steps:

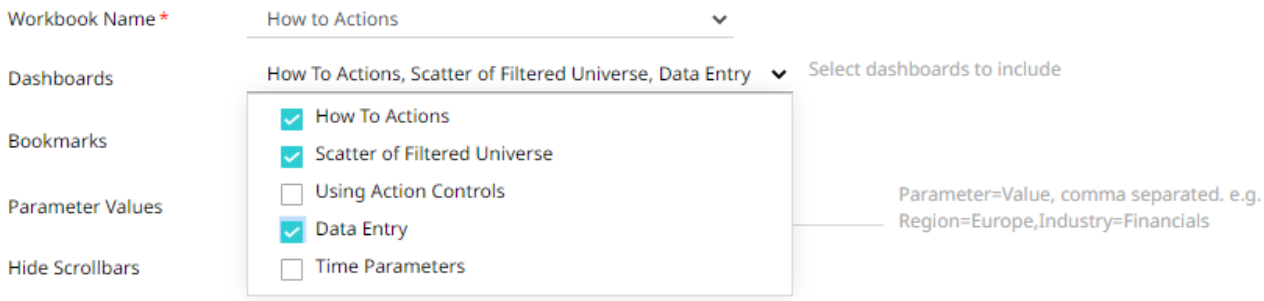
1. On the *New Task pane*, enter the *Name* of the task and click ✓. Ensure the name is unique.
2. Tap the *Activated* slider to turn it on.
3. Select the *Trigger.* You can either select:
   - **Period** then enter the Interval (in seconds), or



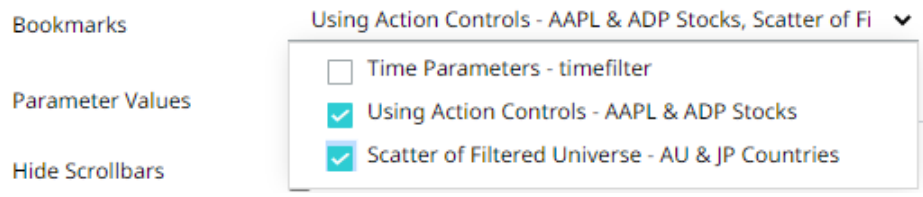
   - **CRON** then enter the CRON Expression



4. Select the task *Type*: **Email PDF**.
5. Upon selecting **Email PDF**, the *Scheduler* page changes to allow specification of the following:
   - Enter the *Description* of the task.
   - Select the *Workbook Name* in the drop-down list. These are the published workbooks available on the *Workbooks* page.
   - Select dashboards to include in the PDF by checking their corresponding boxes in the *Dashboards* drop-down list.



6. Select bookmarks in the workbook to include in the PDF by checking their corresponding boxes in the Bookmarks drop-down list.



7. You can also opt to:
   - enter the *Parameter Values* that will be added as parameters to the subject line of emails or as dashboard values in the PDF file.

     Such as `Parameter=Value`, and are comma separated. For example:

     ```
     Region=Europe,Industry=Financials
     ```

NOTE	The following Date range querying parameters are also supported in the Email Image task: • CurrentTime • LastWorkDay • WeekStart • QuarterStart For example: `{CurrentTime:dd-MMM-yyyy}` However, when there is no Date/Time format supplied, the default format `yyyy-MM-dd` will be used instead.

- check the **Hide Scrollbars** box.
- check the **Enable Pagination** box.

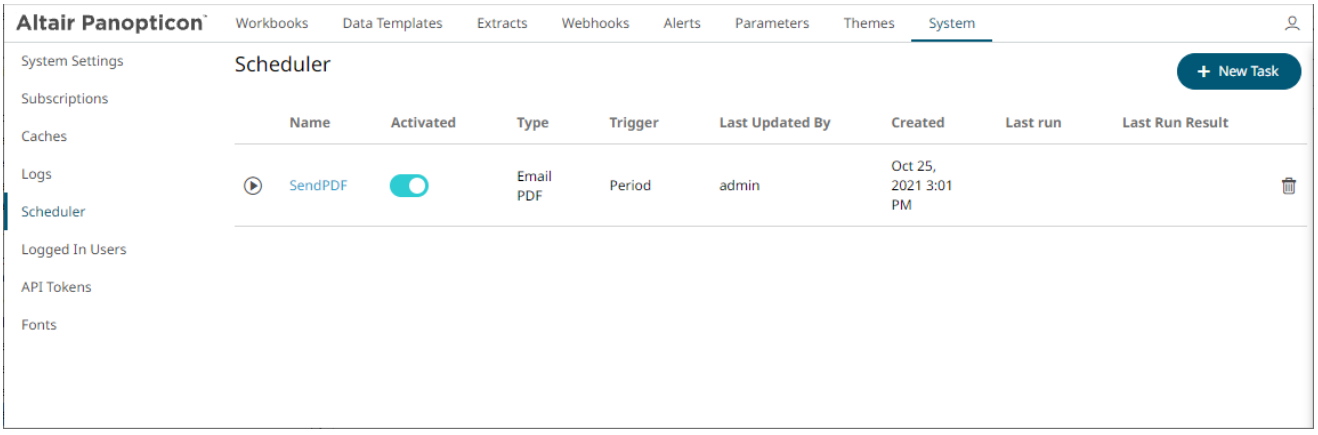
8. Enter the email address of the recipient in the *To* field.
9. You can opt to enter the following:
   - the *CC* and/or *BCC* recipients of the email separated by a comma.
   - the mail message subject to be used in the email notifications in the *Subject* field.

NOTE	Supports dashboard parameters.

   - the content of the email in the *Body* box.
10. Click 

.

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

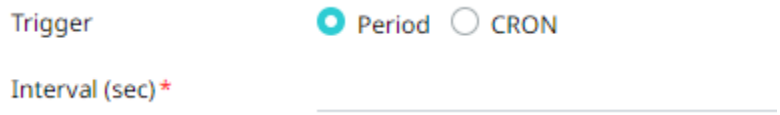


Create Task to Extract Data

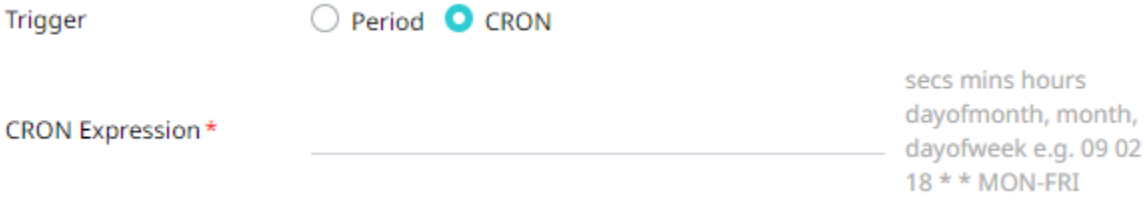
Tasks can be created to reload workbook or global extracts.

Steps:

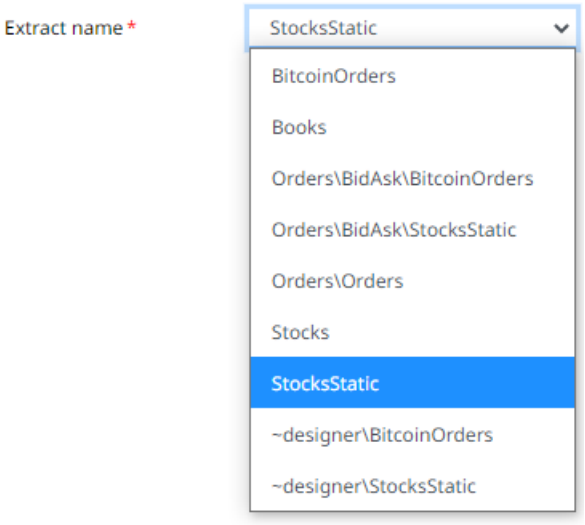
1. On the *New Task pane*, enter the *Name* of the task and click ✓. Ensure the name is unique.
2. Tap the *Activated* slider to turn it on.
3. Select the *Trigger.* You can either select:
   - *Period* then enter the *Interval* (in seconds), or



   - CRON then enter the CRON Expression



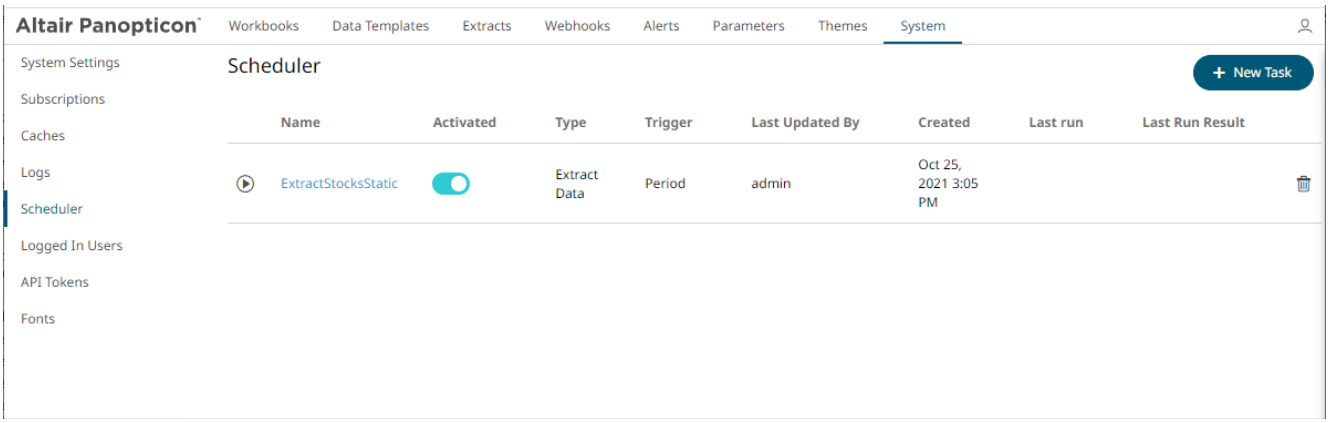
4. Select the task *Type*: **Extract Data**.
5. Enter the *Description* of the task.
6. Select the data extract to be scheduled in the *Extract Name* drop-down list box. The list is taken from the data extracts list on the **Extracts** tab.



7. Click 

.

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



Other Scheduler Tasks Operations

On the **Scheduler** tab of the *System Settings* page, you can also perform the following:

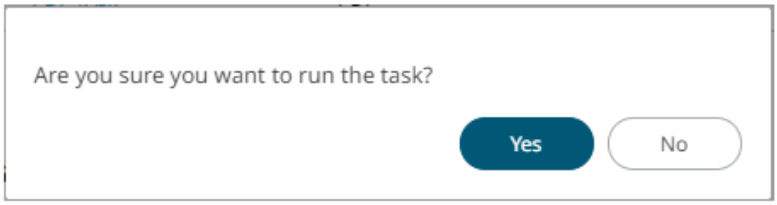
- Sort asks

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

  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.

- Manually run tasks

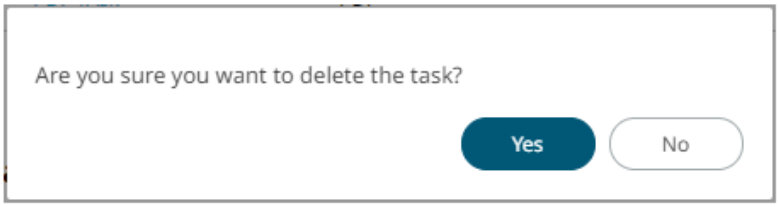
  Instead of waiting for the set Period interval or CRON Expression, you can manually execute the task by clicking ▶. A confirmation message displays.



  Click Yes.

- Modify tasks
- Delete tasks

  Click 🗑 of a task. A confirmation message displays.



Click Yes .

Modify a Scheduled Task

Steps:

1. On the **Scheduler** tab, click the link of a task to modify.

   The properties of the task are displayed.

2. Apply the desired changes.

3. Click Save .

MANAGING PANOPTICON REAL TIME USERS

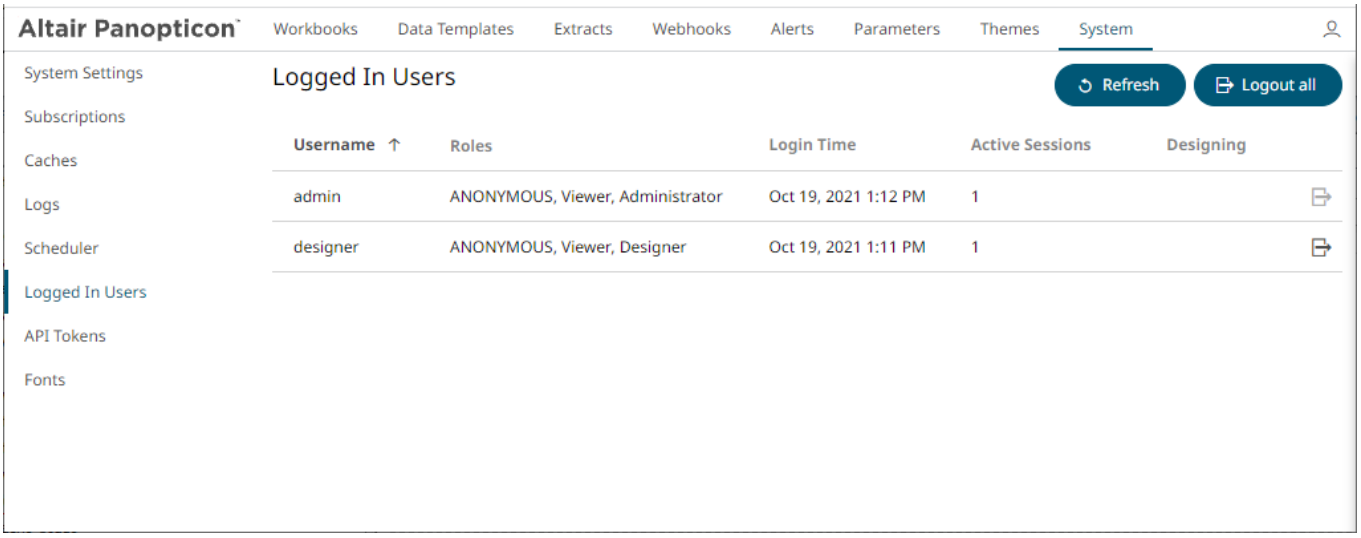
Users with an Administrator role can view the logged in users on Panopticon Real Time and log them out when necessary. The ability to manage users is beneficial in monitoring the utilization of Altair Units license.

On the **Logged In Users** tab of the *System* page, Administrators can perform the following:

- ❑ View logged in users
- ❑ Sort logged in users
- ❑ Log out users
- ❑ Refresh the *Logged In Users* list

Viewing Logged In Users

On the *System* page, click the **Logged In Users** tab. The list of logged in users is displayed.



In the list, the following properties are displayed for each user:

Property	Description
Username	Username used to the login to Panopticon Real Time.
Roles	Roles assigned to the user.
Login Time	The Date/Time the user logged in.
Active Sessions	The number of tokens a user is using. For example, if the user is logged in from two different computers, he will have two active sessions. However, if the user has two tabs on one computer, they will share a token and the active sessions will be one.
Designing	Indicates if a logged in user is currently designing a workbook.

Sorting Logged In Users

Sorting the logged in users can be done through the *Username*, *Login Time*, or *Active Sessions* column name.

Steps:

1. On the *System* page, click the **Logged In Users** tab. The list of logged in users is displayed.
2. Click on the **Username**, **Login Time**, or **Active Sessions** column header then click the *Sort Order*:
   - ↑ Ascending
   - ↓ Descending

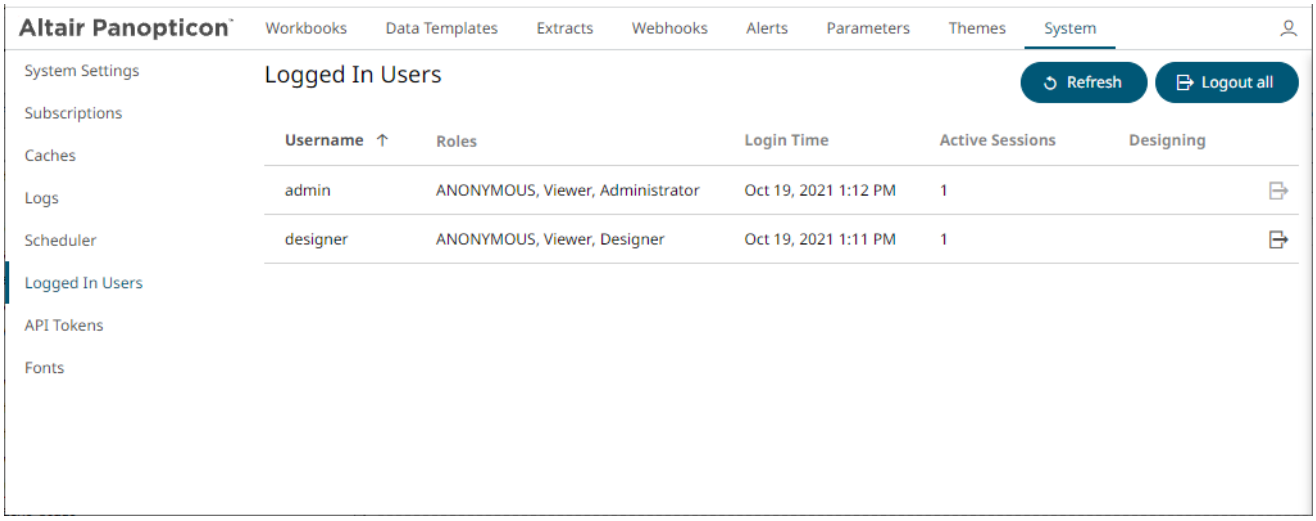
Logging Out Users

Logging out users on the server consequently deletes their tokens.

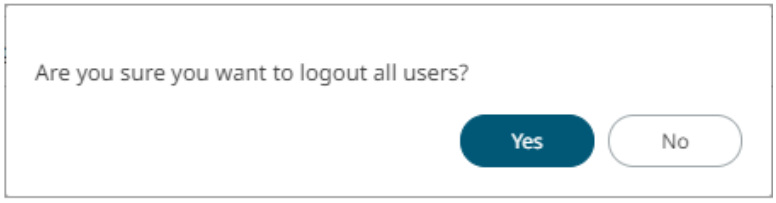
Logging Out All Users

Steps:

1. Click 



A notification message displays.

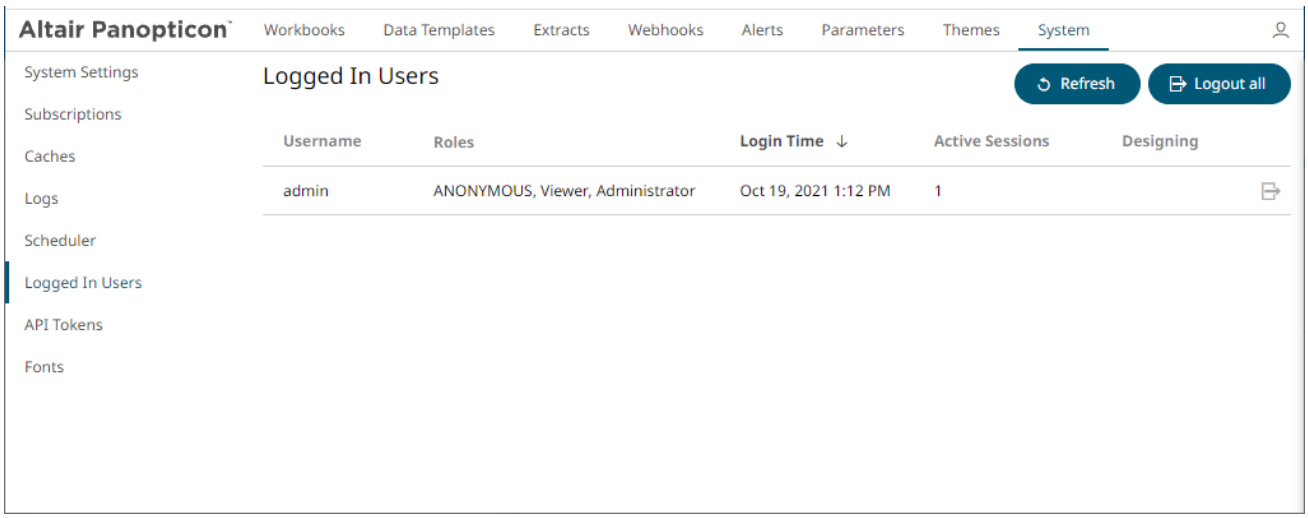


2. Click 

.

Except for the user (i.e., admin) who is calling out the logging out of the other users, all of the other users are logged out.

Also, the  button of admin is disabled.

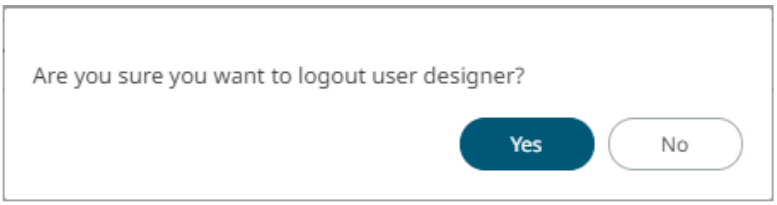


Logging Out Individual Users

Steps:

1. Click the button of a user in the list.

   A notification message displays.



2. Click Yes.

   The user is logged out and their token is deleted.

Refreshing the Logged In Users List

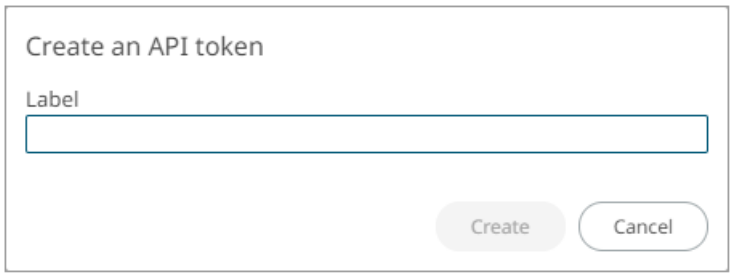
Click Refresh to refresh the list of logged in users.

MANAGING API TOKENS

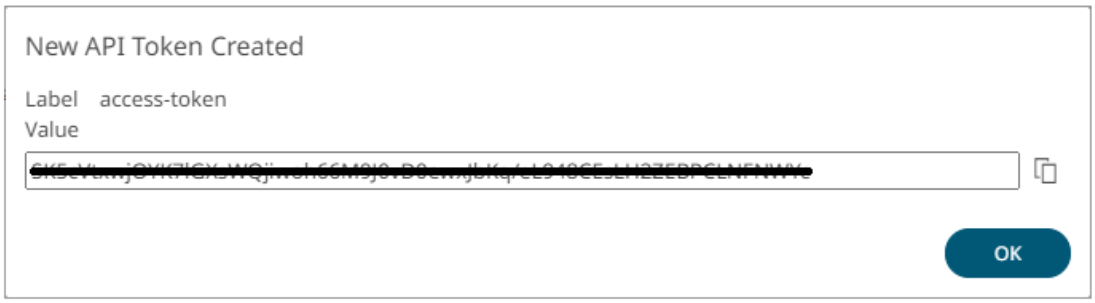
On the *API Tokens* page, an Administrator user can add API Tokens that returns a key used for authorizing requests to the server.

Steps:

1. Click + Create. The *Create an API Token* dialog displays.

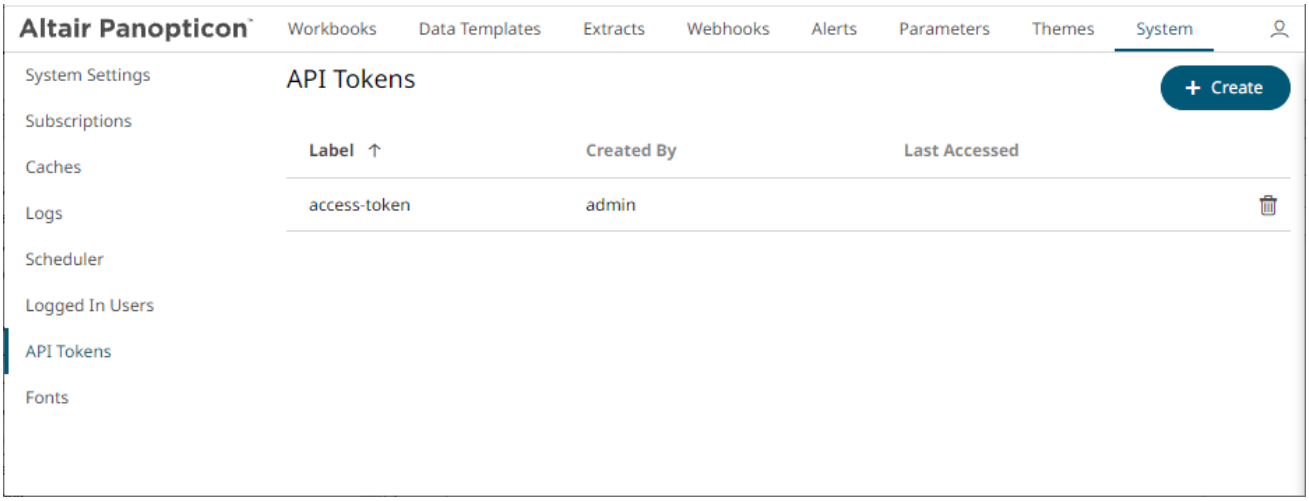


2. Enter the *Label*.
3. Click Create. The *New API Token Created* dialog displays with the auto-generated key.



NOTE	The key will not be displayed again.

4. Click  to ensure you have a copy of the key and paste in a secure location.
5. Click  . The new API Token is displayed on the list.



In the list, the following properties are displayed for each API Token:

Property	Description
Label	Label of the API Token. **NOTE:** Select a label that is easy for you to remember.
Created By	The user who created the API Token. **NOTE:** Only Administrator users can create API Tokens. However, the keys can be used by anyone as long as they are not revoked.
Last Accessed	Date/Time when the API Token was last accessed.

Click on any of these column headers then click the *Sort Order* to sort the list.

- ↑ Ascending

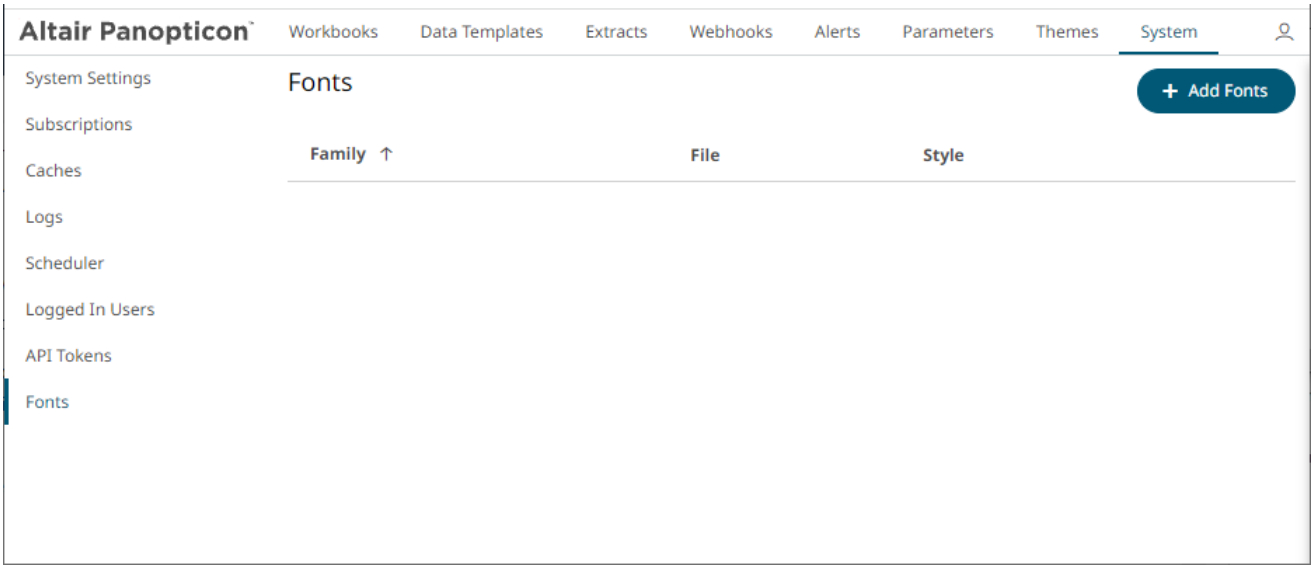
- ↓ Descending

You can also opt to click  to remove and revoke the API Token from the server.

> **NOTE** If the returned key is `key123`, then you can utilize the API services by setting an authorization header such as below:
>
> `Authorization="Bearer key123"`
>
> In the cURL, you can add a header flag such as:
>
> `-H "authorization: Bearer key123"`

ADDING CUSTOM FONTS

Users with an Administrator role are allowed to add fonts on the **System** tab in Panopticon Real Time.



The supported custom font files include the following:

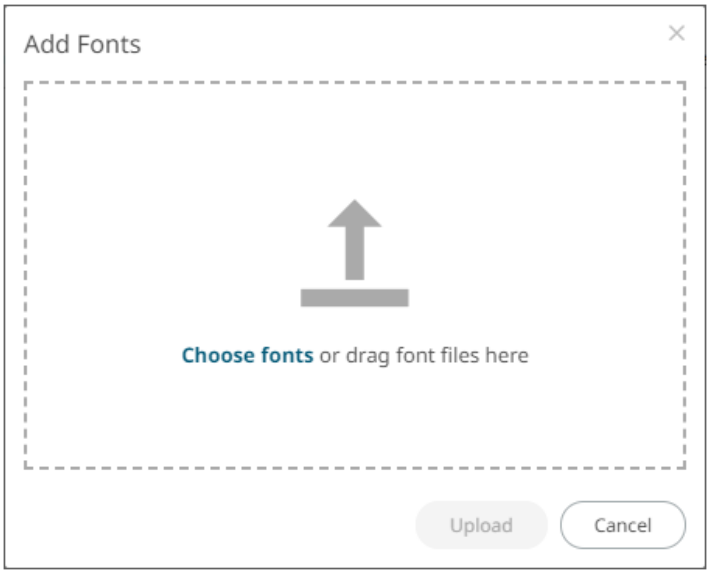
- ttf
- otf

When available on the server, the client will automatically detect and load the font and consequently, can be used in a part or workbook. Otherwise, the client will fall back to the system installed fonts.

Steps:

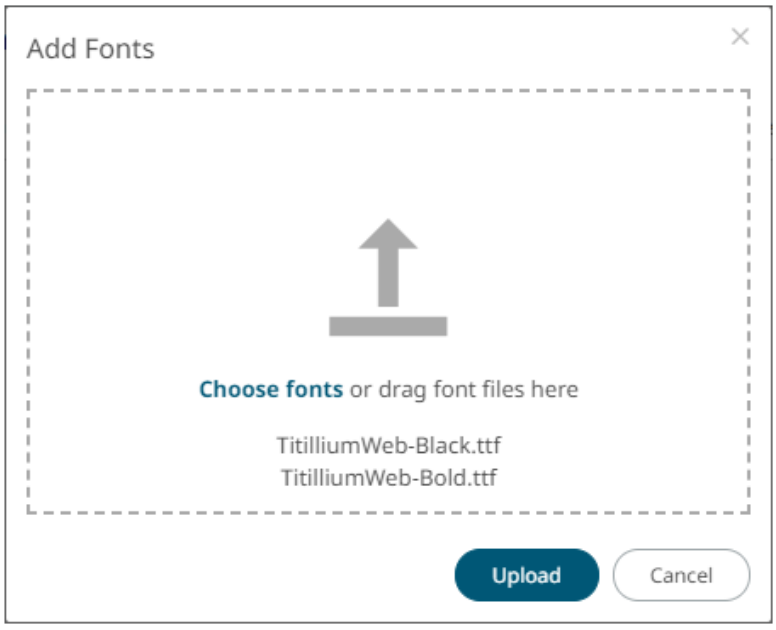
1. Click 

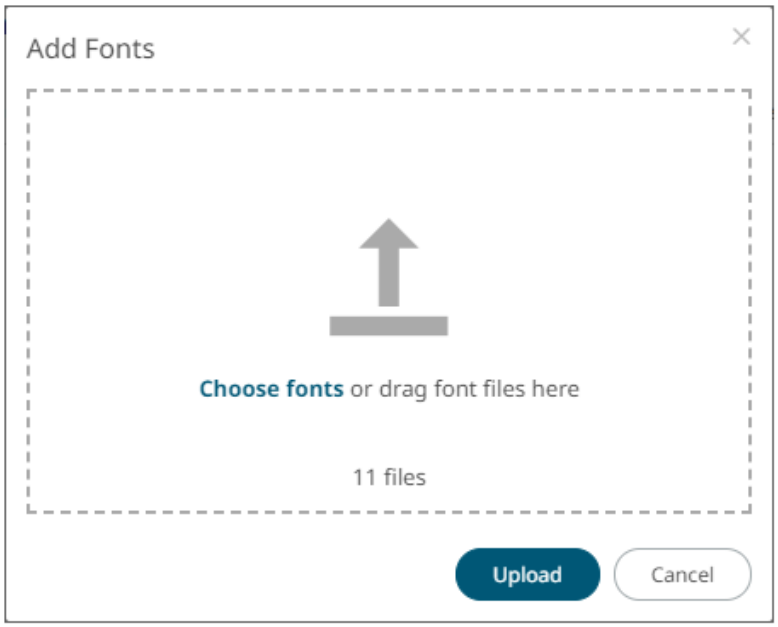
. The *Add Fonts* dialog displays.



2. To add fonts, you can either:
   - drag them from your desktop and drop on the dialog, or
   - click **Choose Fonts** and select one or more fonts on the *Open* dialog that displays.

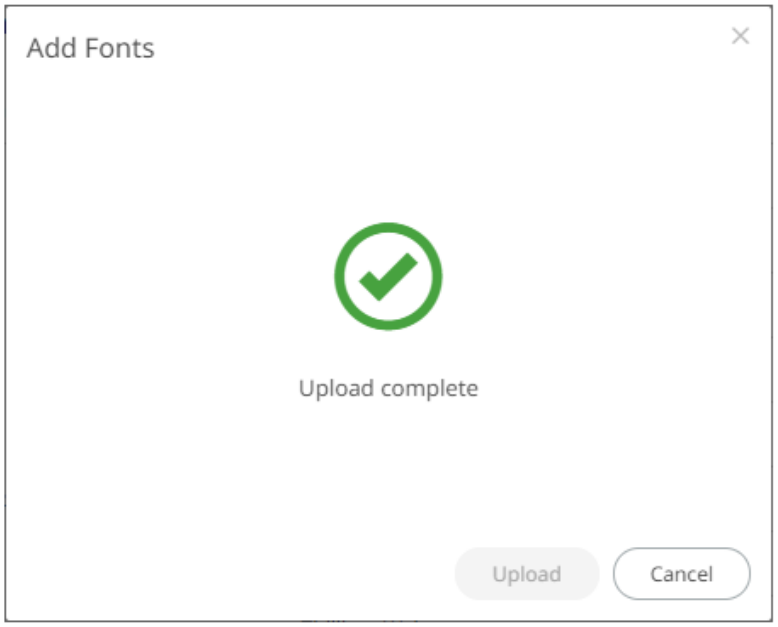
   The names or the number of fonts are displayed on the uploaded font area.



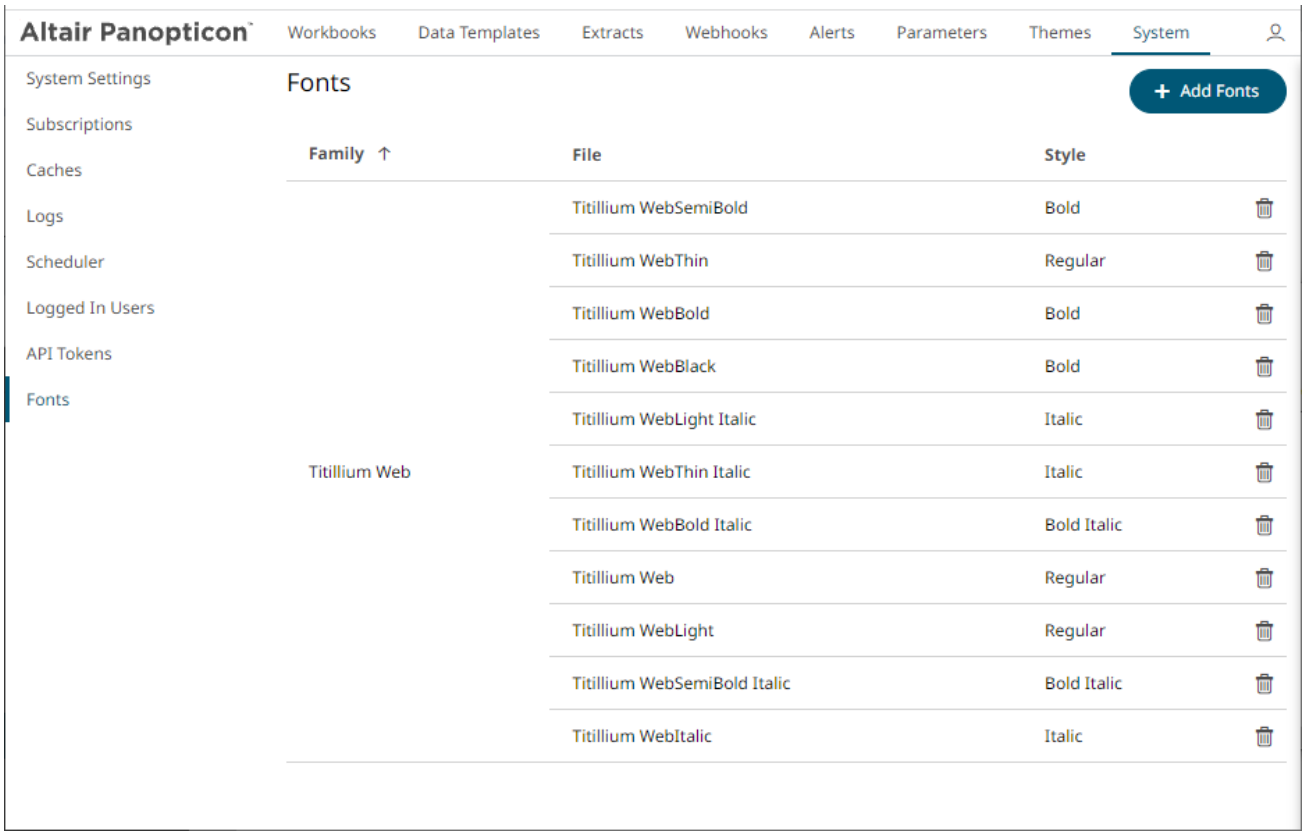


3. Click Upload .

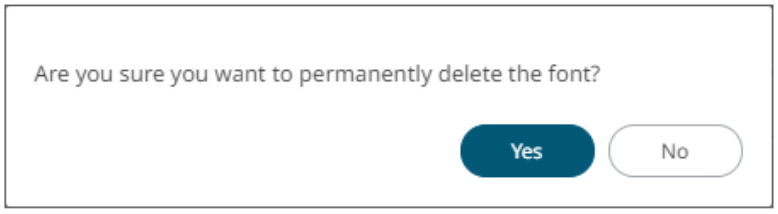
   A notification prompt will be displayed once the fonts are uploaded.



   The added custom fonts are displayed on the *Fonts* page.



To delete a font, click  . A notification is displayed.



Click  .

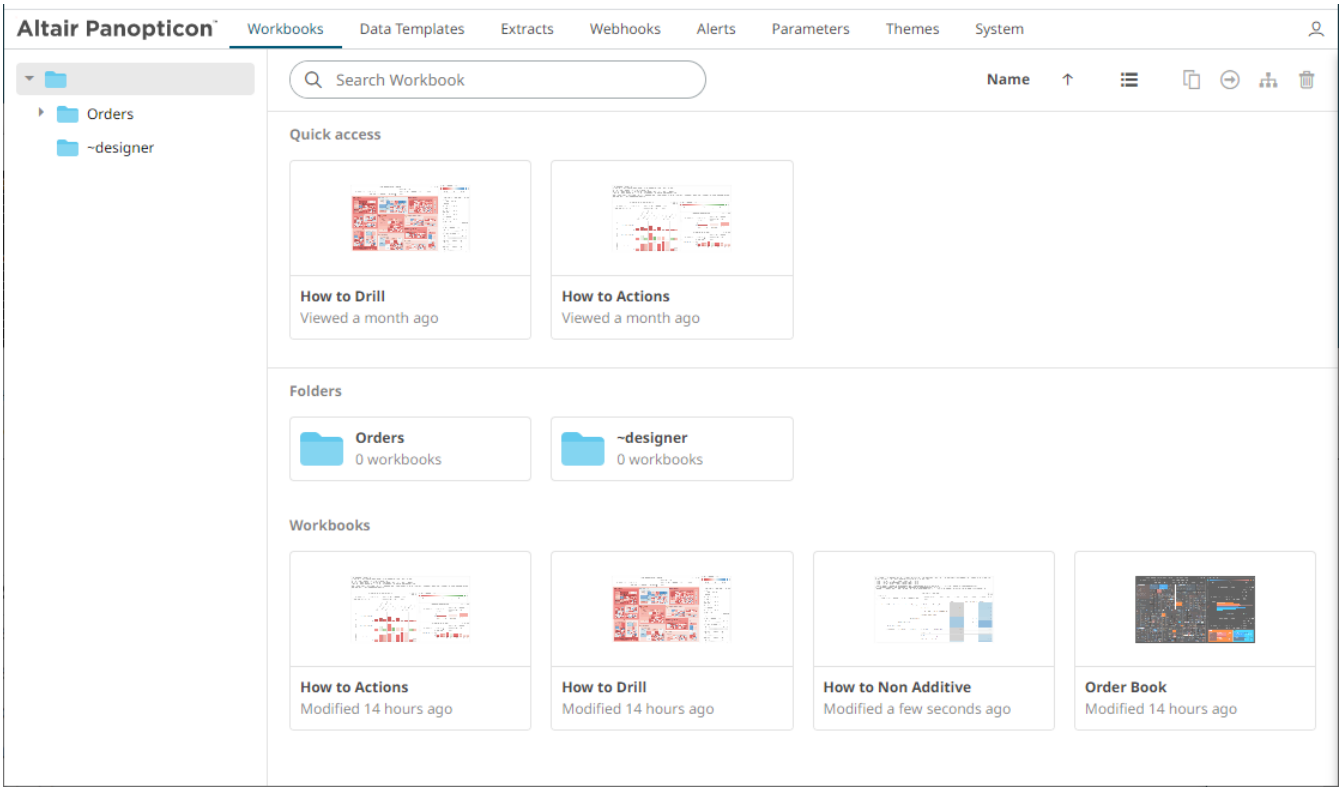
[8] VIEWING AND MANAGING WORKBOOKS

ACCESSING WORKBOOKS

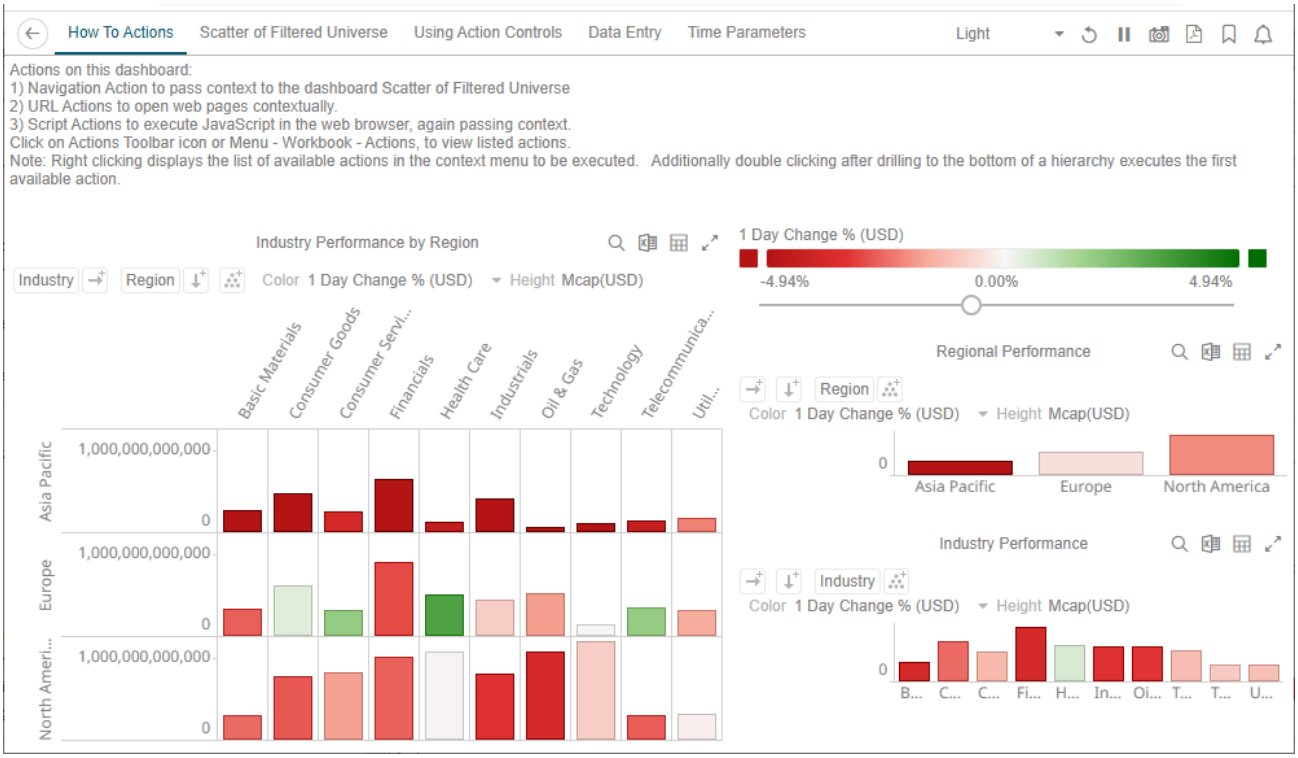
The default home page for Panopticon Real Time lists available folders and uploaded or published workbooks in *Grid View*.

- The *Folders* include their names and the number of available workbooks.
- The *Workbooks* include their titles, thumbnail images, and when they were last modified.

  Recently opened workbooks are also displayed under the *Quick Access* pane.

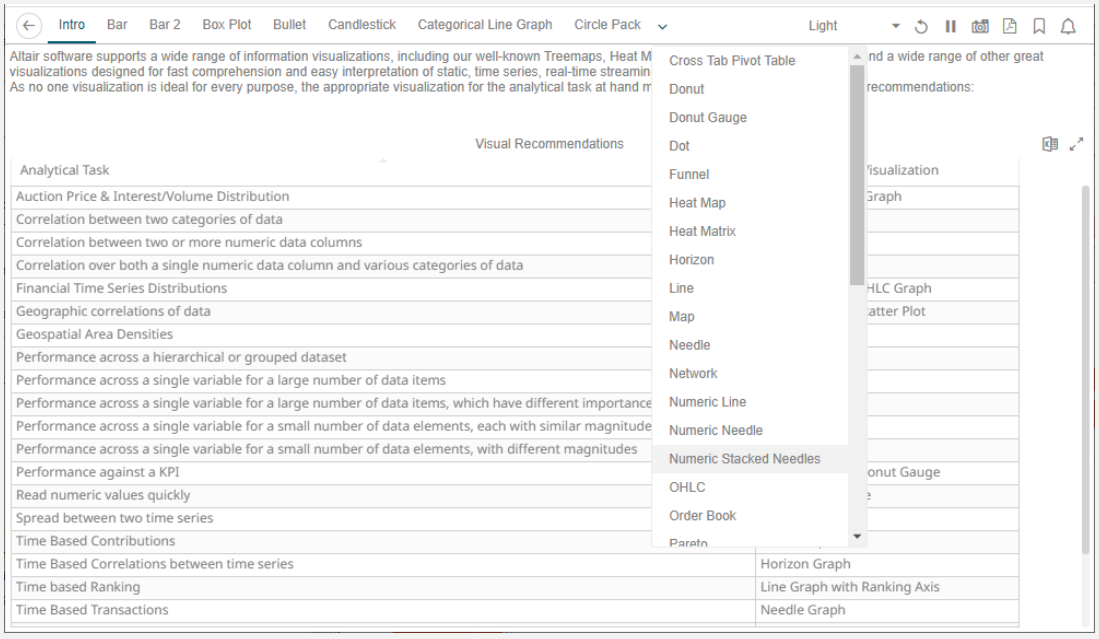


Clicking on the workbook thumbnail opens it on the web browser.



NOTE

The › signifies there are more dashboards in a workbook that can be opened. Click this icon to expand the drop-down list and display all of the available dashboards and select one to display.



NOTE

The **Back** button allows going back to the root folder. It is only available on the toolbar section of the *Open Workbook in View Mode* if `startURL` is available in the

`workbook.json` file located in `<appdata>/JavaScriptConfiguration/`.

```
{
  "baseUrl" : "..",
  "forceClientSelectionHandling" : true,
  "startUrl" : "../",
  "subscriptionCompression" : true,
  "dataLoading" : {
    "transport" : "websocket"
  },
  "webGlEnabled" : true,
  "pdfMultiplePagesEnabled" : true
}
```

However, for the **Back** button to use the browser history to navigate back despite `startUrl` being set in the file, add `useBrowserHistoryToNavigateBack` and set to **true**.

```
{
  "baseUrl" : "..",
  "forceClientSelectionHandling" : true,
  "startUrl" : "../",
  "useBrowserHistoryToNavigateBack" : true,
  "subscriptionCompression" : true,
  "dataLoading" : {
    "transport" : "websocket"
  },
  "webGlEnabled" : true,
  "pdfMultiplePagesEnabled" : true
}
```

After updating the `workbook.json` file, restart the Panopticon application.

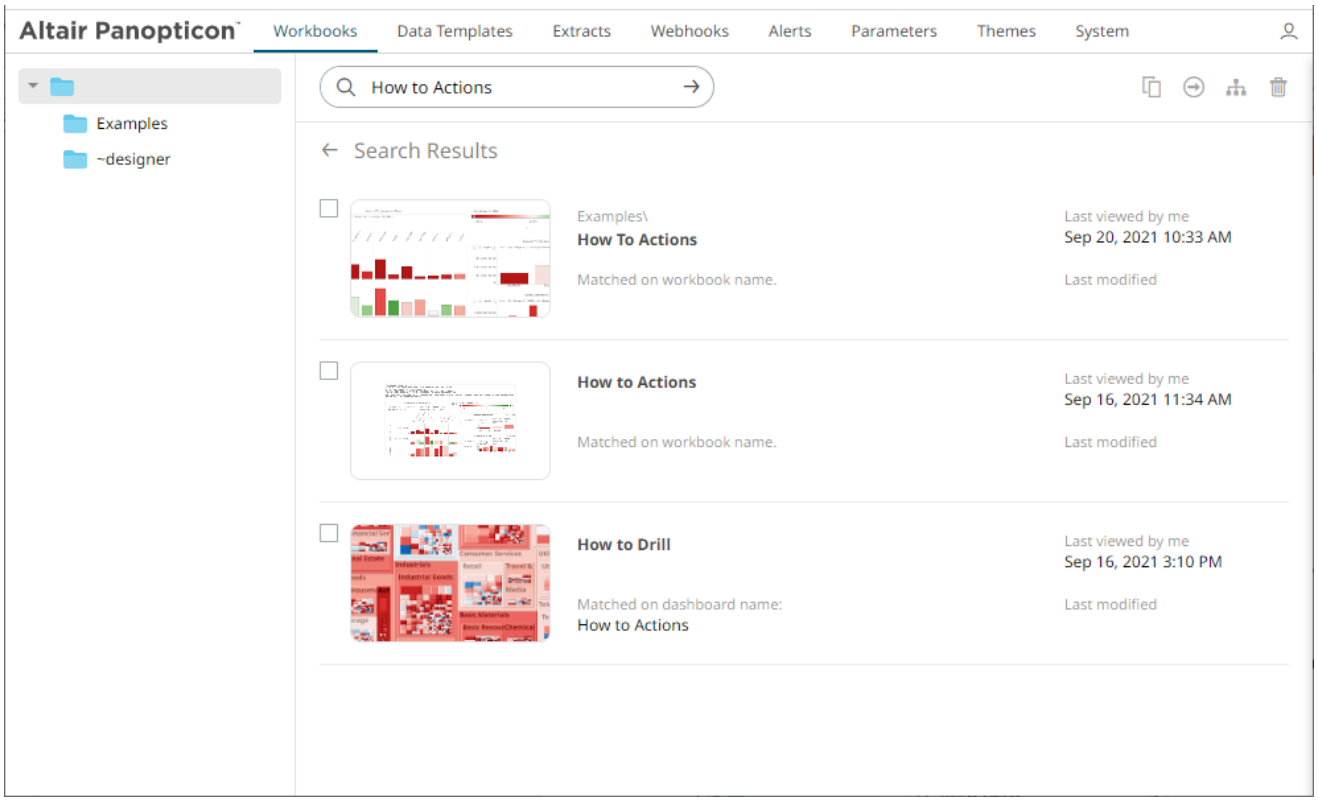
Searching for Workbooks

Search for particular workbooks that may be located in different folders and perform other operations like merge, copy, download, or remove.

Steps:

1. On the *Workbooks and Folders Summary* layout, click on a workbook folder then enter a workbook name or dashboard name in the *Search Workbook* box.

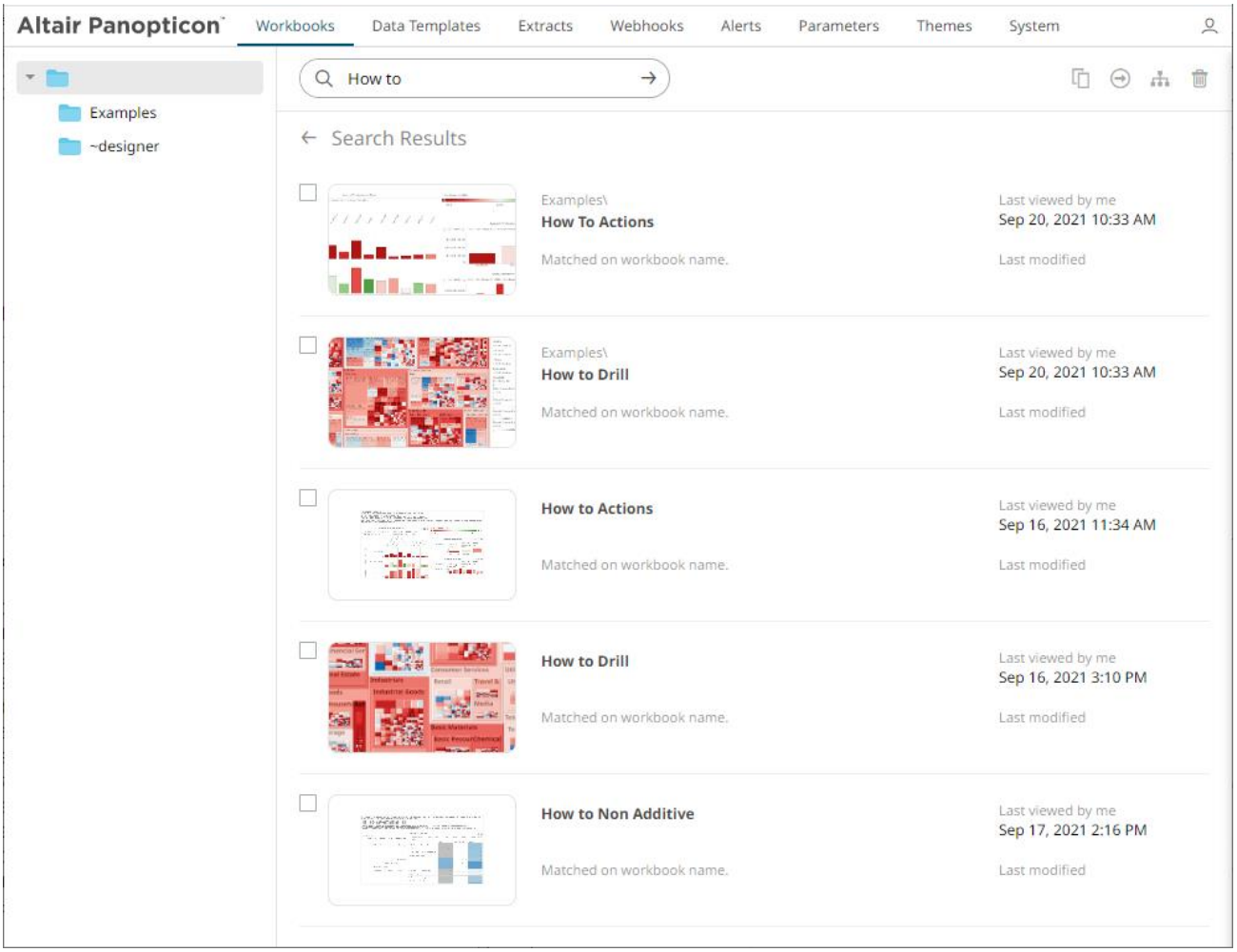
2. Click → .



The following information are displayed for each workbook:

- Folder where the workbook is located
- What the search match was based on: workbook or dashboard name
- Date/Time when the workbook was last viewed
- Date/Time when the workbook was last modified

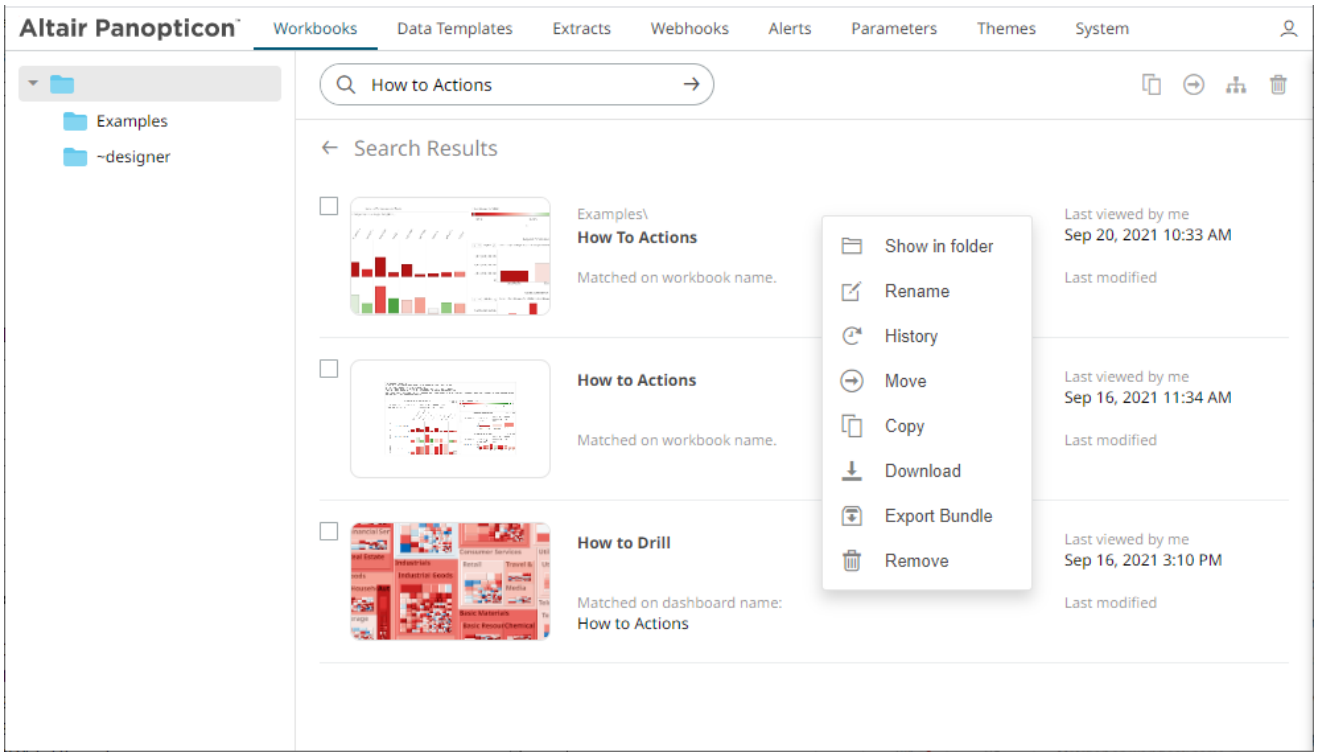
You can also enter one or more characters into the *Search Workbook* box then click **Enter**. The list of workbooks that matched the entries will be displayed.

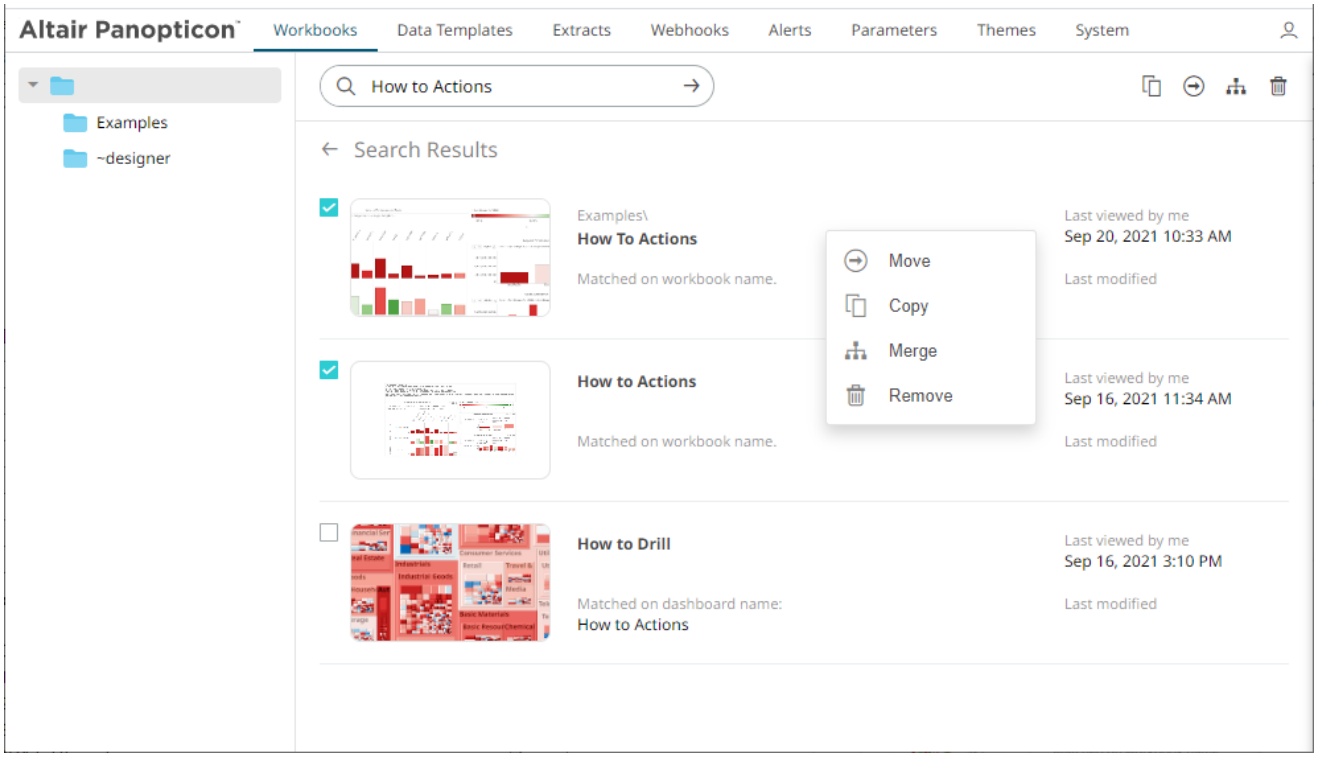


Click on a workbook thumbnail to open and display it on the web browser.

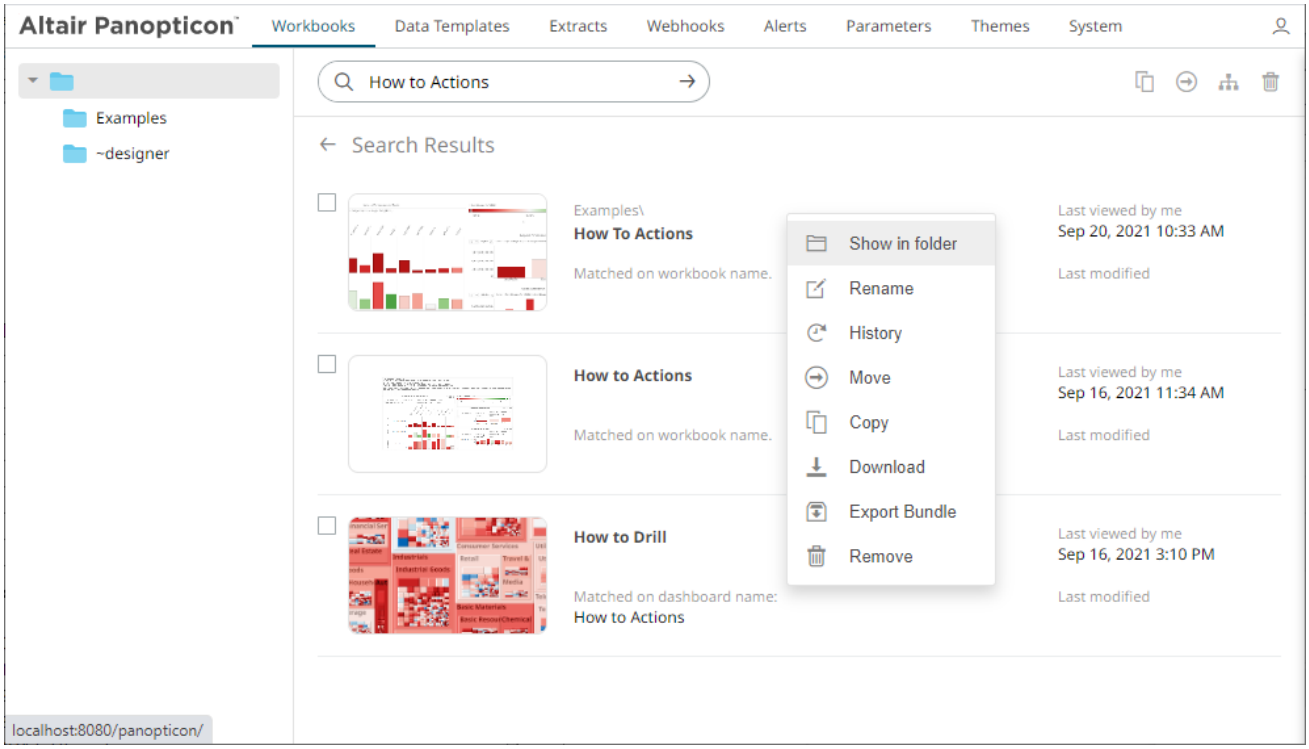
To go back to the *Workbooks and Folders Summary* layout, click ← .

You may opt to right-click on a workbook or select several workbooks to display the context menu.





To display the workbook in its location, click **Show in Folder** on the context menu.

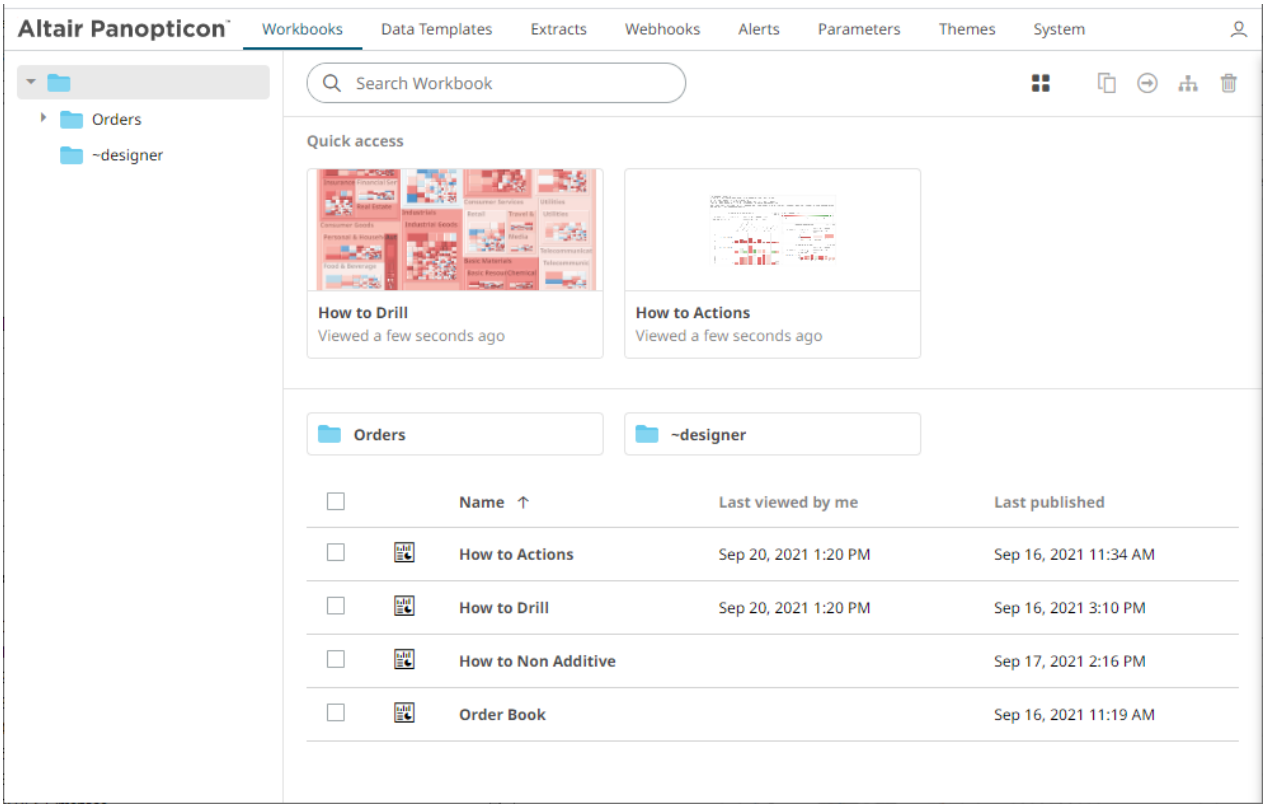


The other context menu options are discussed in the sections below.

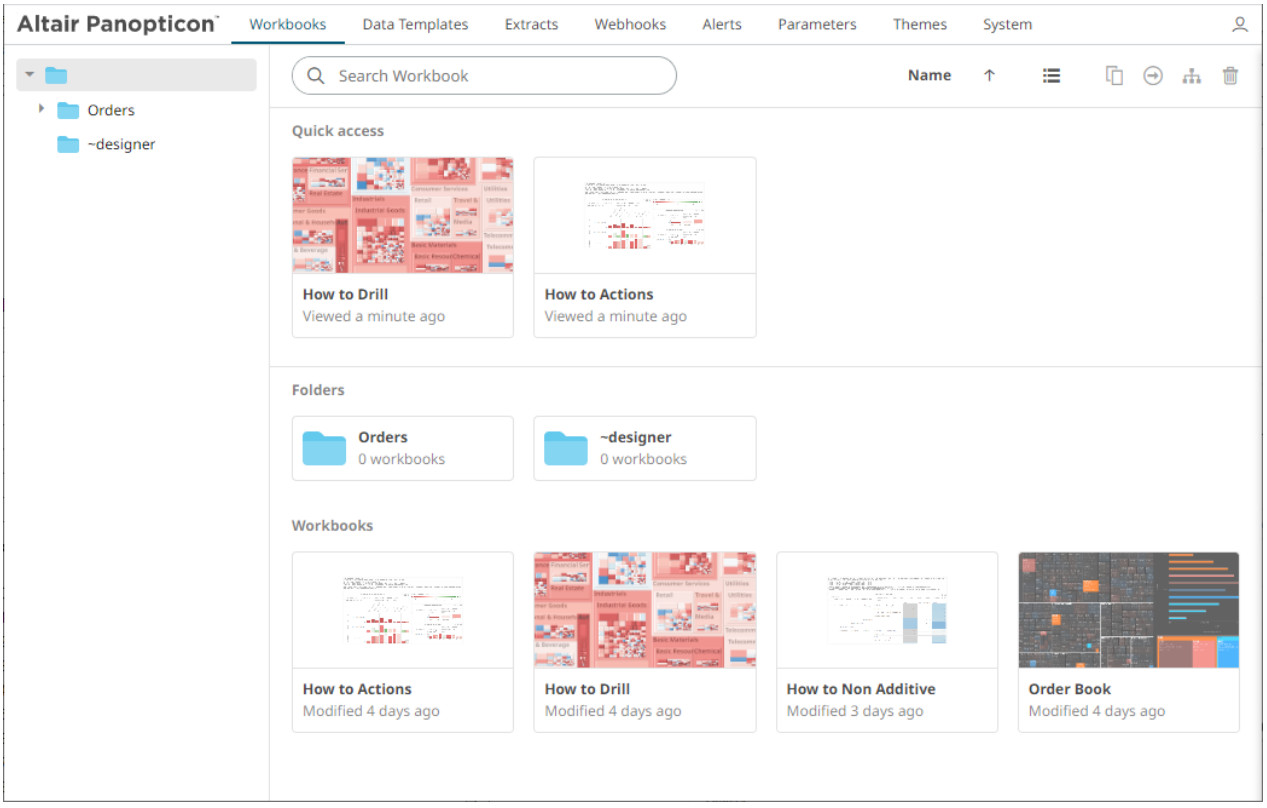
Folders and Workbooks Display View

Workbooks can be displayed either on a *List* or *Grid View*.

On the *Toolbar*, click **List View** , the folders and workbooks are displayed in a standard listing.



Or click **Grid View** . The folders and workbooks are displayed as thumbnails.



On either display view style, clicking on a workbook title or thumbnail displays the workbook on the *Open Workbook in View Mode*. For more information on how to analyze interactive dashboards, refer to the Client User Guide.

Sorting Workbooks

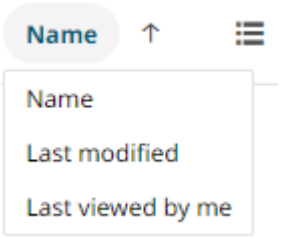
Sorting workbooks can be done by *Name*, *Last Viewed/Last Published*, or *Last Viewed by Me*.

Steps:

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

- ❑ click the **Sort By** option on the *Toolbar* of the *Grid View*

  By default, the sorting is by **Name**.

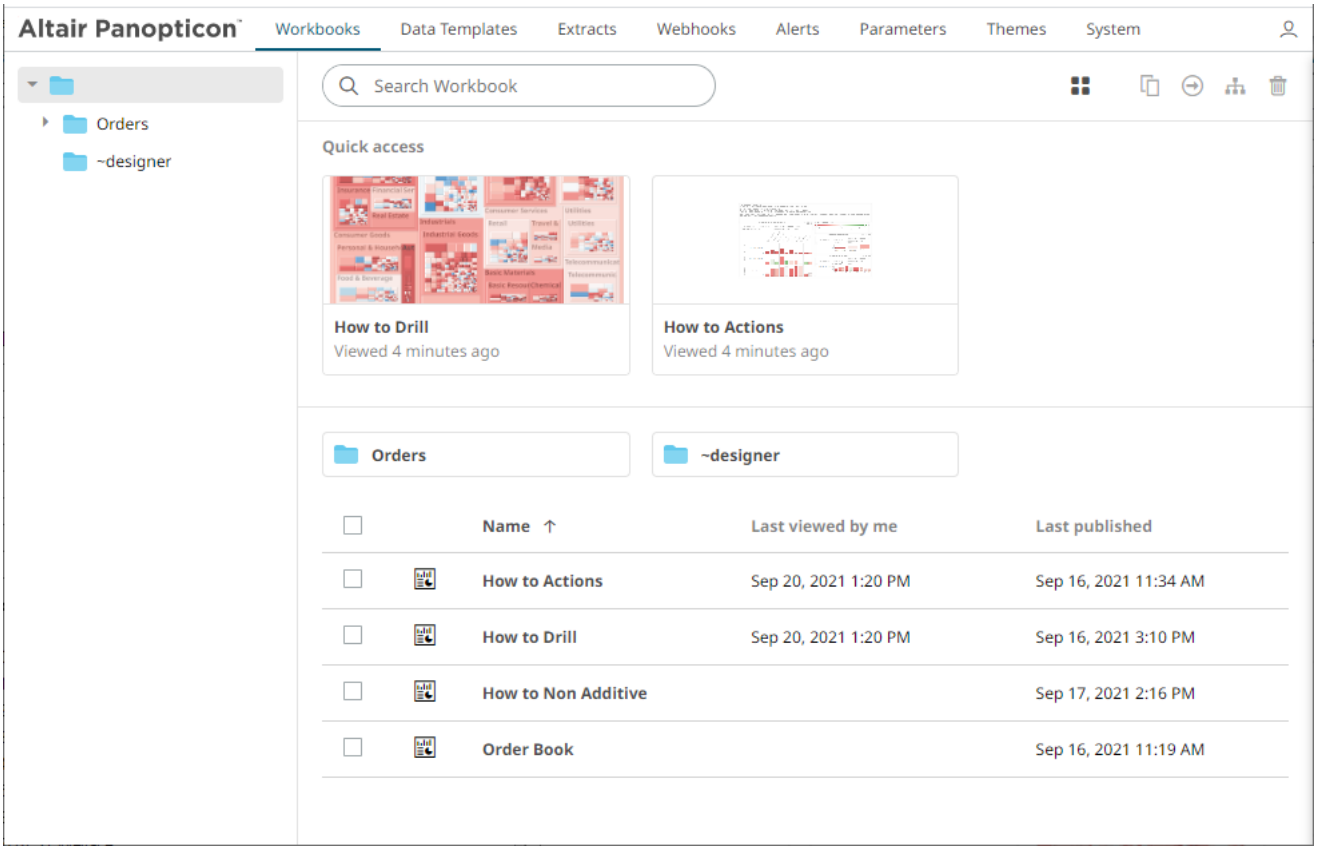
  

  - Name
  - Last Modified
  - Last Viewed By Me

  Then click the *Sort Order*:

  - Ascending
  - Descending

- ❑ click on the **Name**, **Last Viewed By Me**, or **Last Published** column header of the *List View*



Then click the *Sort Order*:

- ↑ Ascending
- ↓ Descending

Creating Workbooks

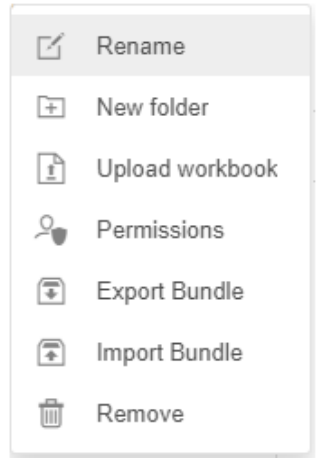
A user with a Designer role can create workbooks using the web authoring tool in Panopticon Real Time. This feature is extensively discussed in the *Panopticon Web Authoring Guide*.

Renaming Workbooks or Folders

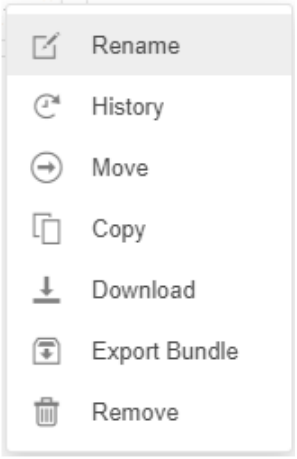
A user with an Administrator or Designer role can rename workbooks and folders.

Steps:

1. Right-click on a workbook or folder then select **Rename** on the context menu.

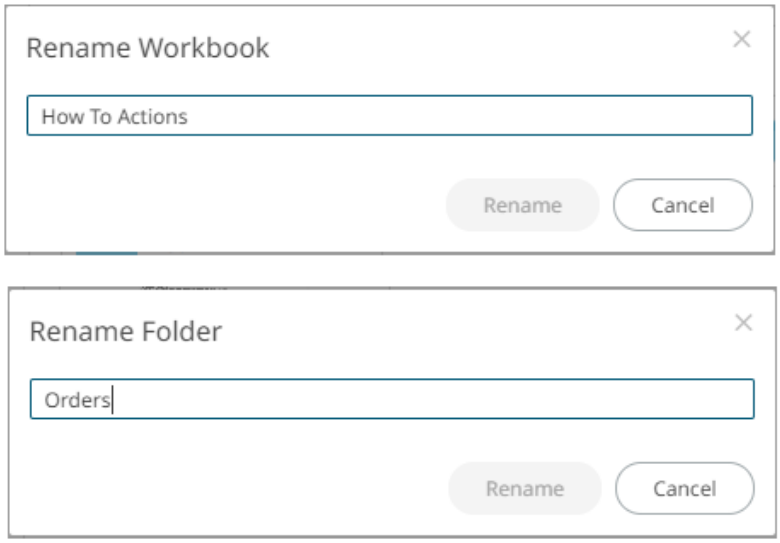


Workbook Folder or Subfolder Context Menu



Workbook Context Menu

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



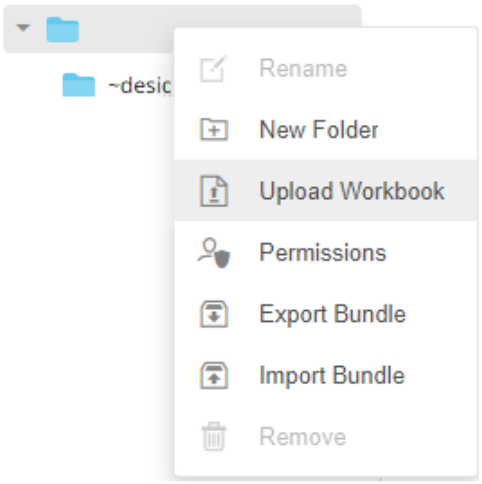
2. Enter a new name then click Rename.

Uploading Workbooks

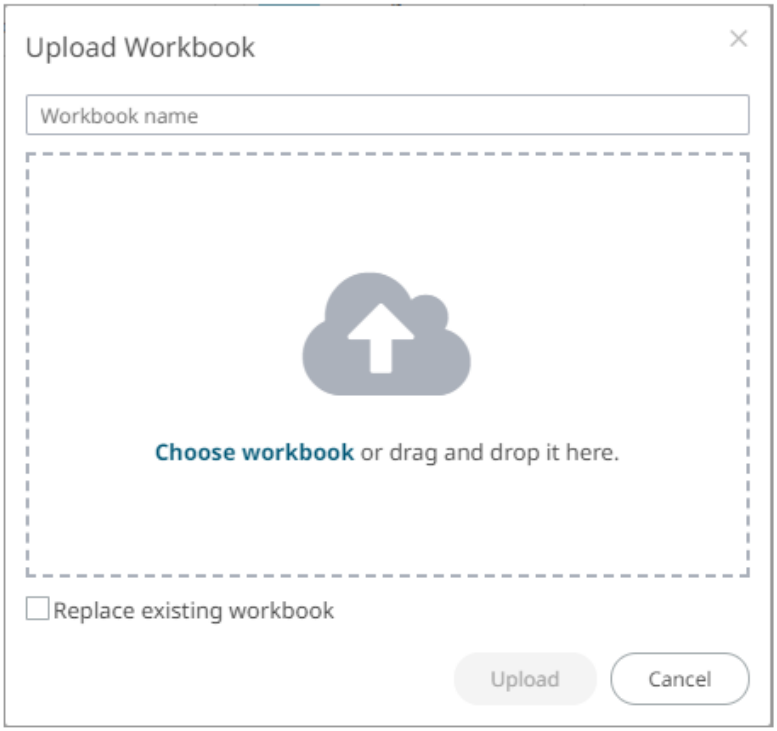
Users with an Administrator role can upload workbooks to the currently selected folder in the *Workbooks* page.

Steps:

1. On the *Workbooks* page, click on a folder or a personal folder and select **Upload Workbook**.

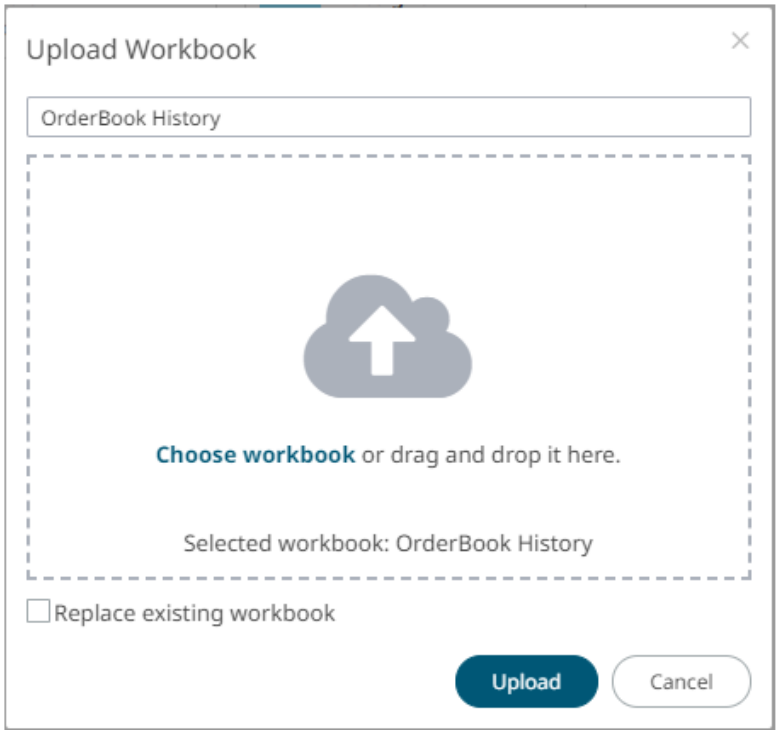


The *Upload Workbook* dialog displays.



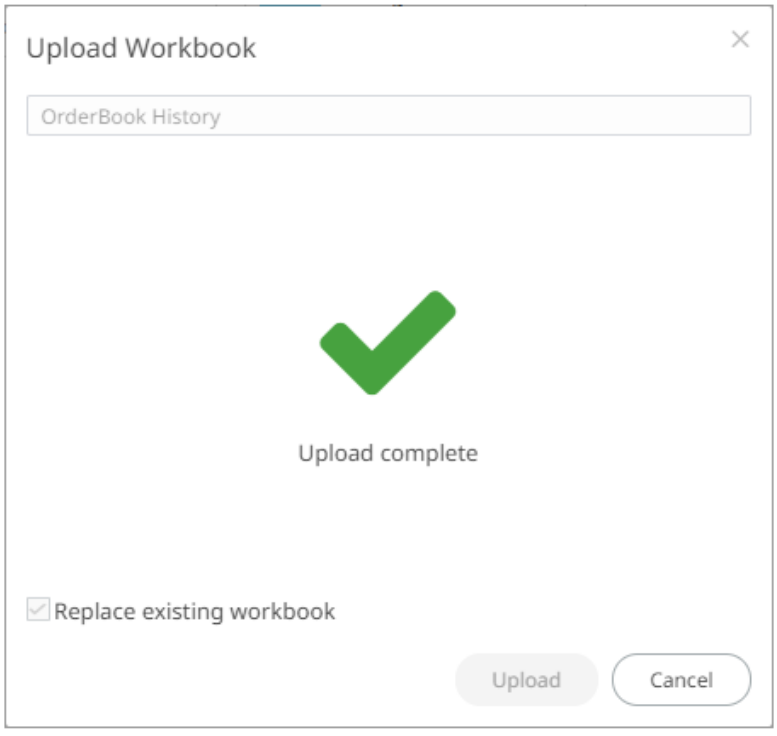
2. To upload a workbook, you can either:
   - drag it from your desktop and drop on the dialog, or
   - click **Choose Workbook** and select one on the *Open* dialog that displays.

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



3. You can opt to rename the workbook.
4. To replace an existing workbook, check the **Replace existing workbook** box.
5. Click Upload.

   You will be notified once the workbook is uploaded.



The workbook is added and displayed.

NOTE	• An error message is displayed if the data source schema of the uploaded workbook has not been updated or missing. • The uploaded workbook will not include the data source. However, if Panopticon Real Time can reach the same folder of the data source, or the workbook has been designed in the same machine, then the data can be viewed.

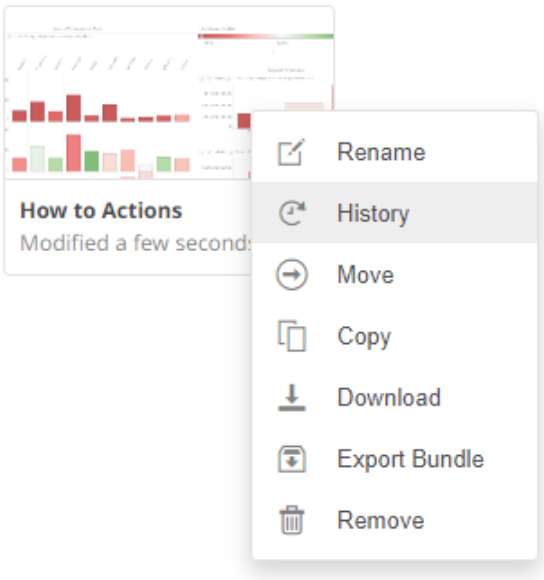
Viewing Workbook History and Republishing

Aside from opening workbooks, a user with either an Administrator or Designer role can also perform the following:

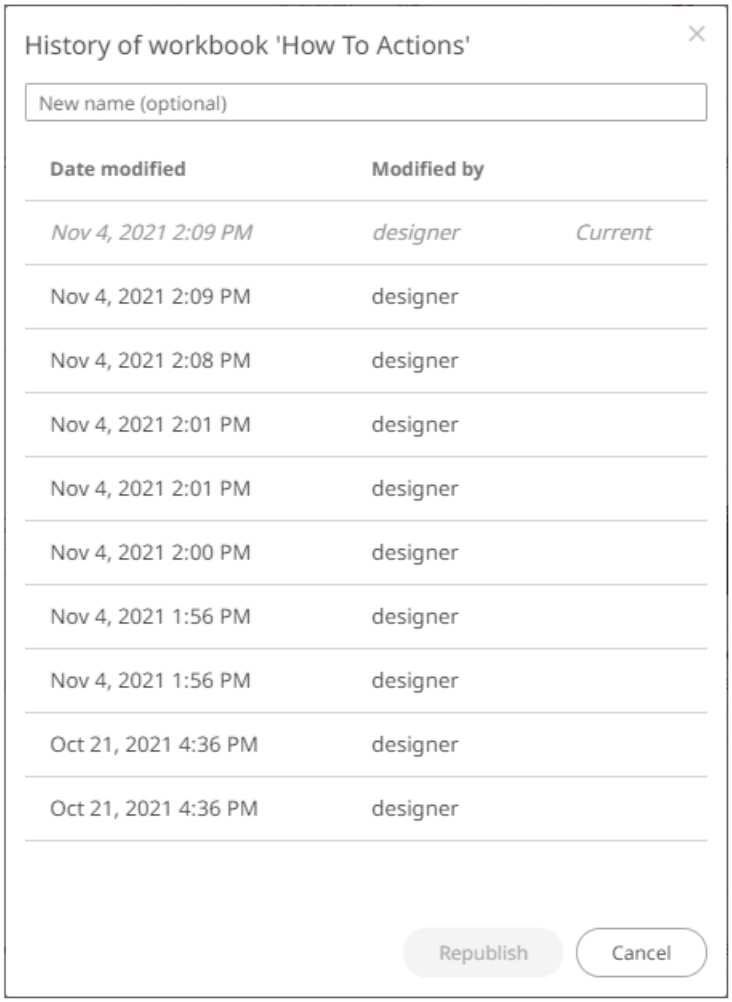
- ❑ View the change history of workbooks
- ❑ Republish an archived workbook to the recent version of Panopticon Real Time
- ❑ Rename an archived workbook

Steps:

1. On the **Workbooks** tab, right-click on a workbook and select **History** on the context menu.



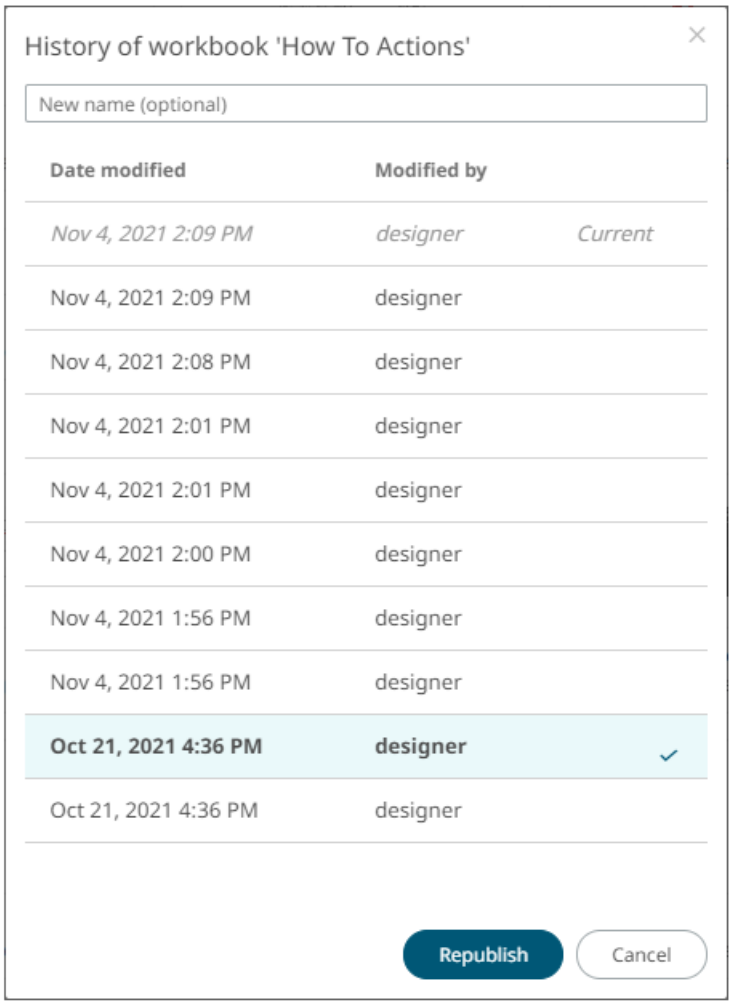
The *History of Workbook* <*Name*> dialog is displayed with the current version of the workbook indicated.



Sort the archival list either through the *Date Modified* or *Modified By* by clicking on the ↓ or ↑ button.

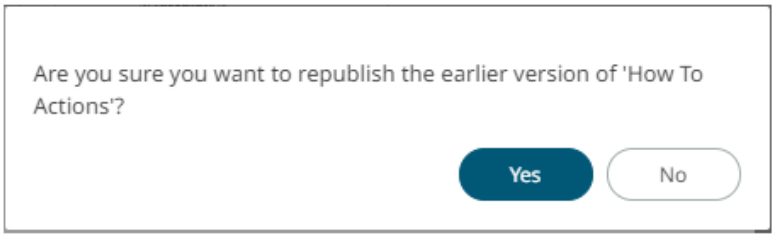
Also, move to the other pages of the list by clicking on a page or clicking the « or » button.

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



Then click 

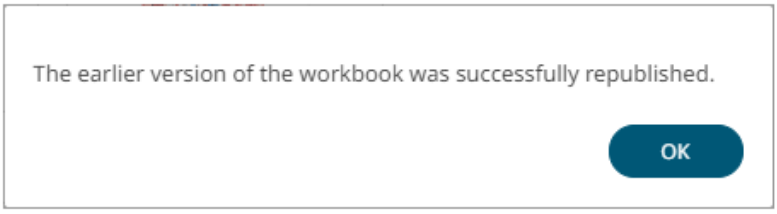
. A notification message displays.



4. Click 

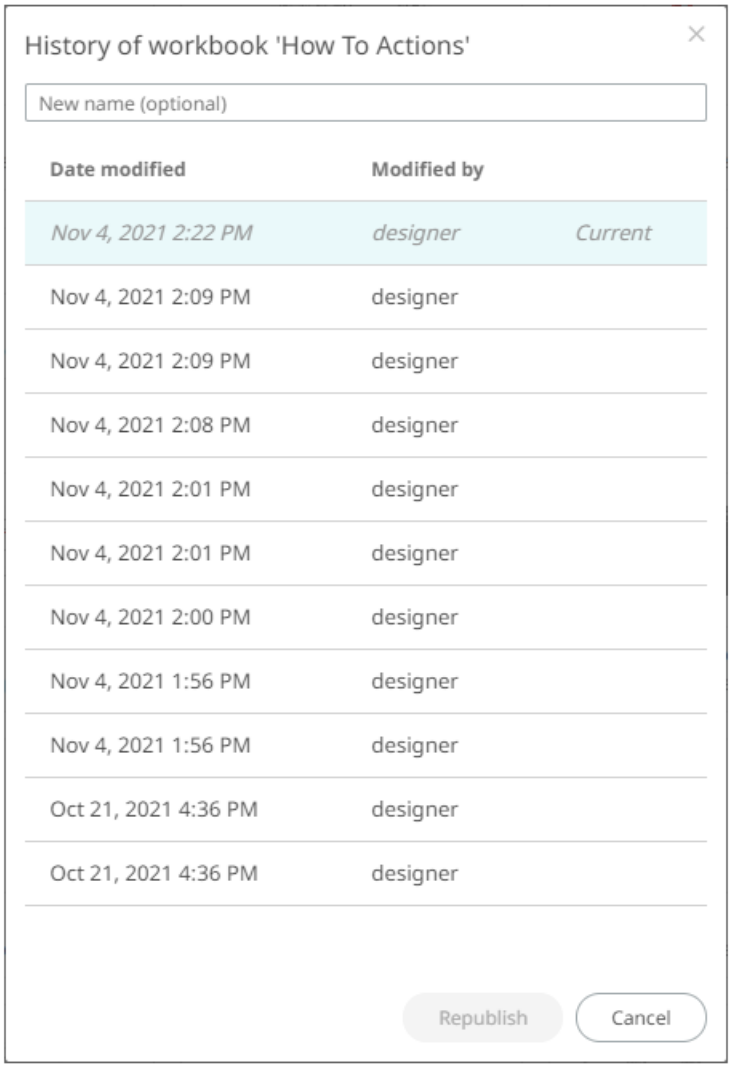
.

   A confirmation message displays.



5. Click OK .

The republished workbook version is added in the history list.

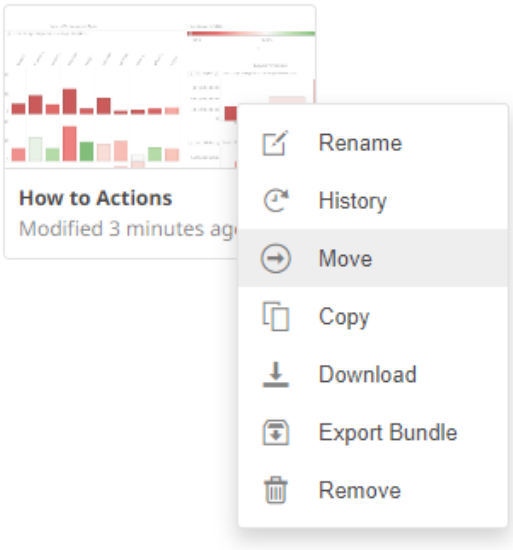


Moving Workbooks

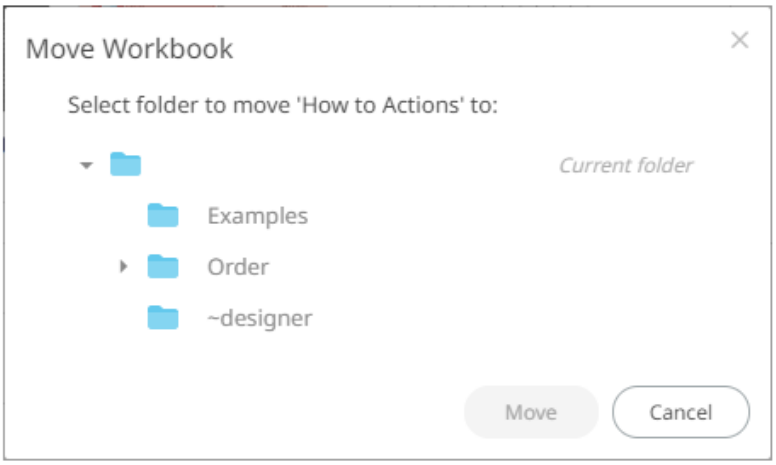
Users with Administrator or Designer role are allowed to move a workbook to another folder or subfolder they have permission to.

Steps:

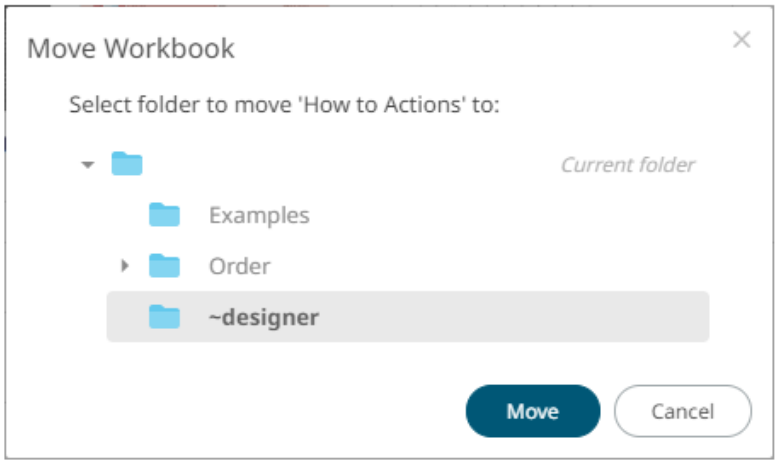
1. Right-click on a workbook and select **Move** on the context menu.



The *Move Workbook* dialog displays with the folder or subfolders the user is allowed to move the workbook.

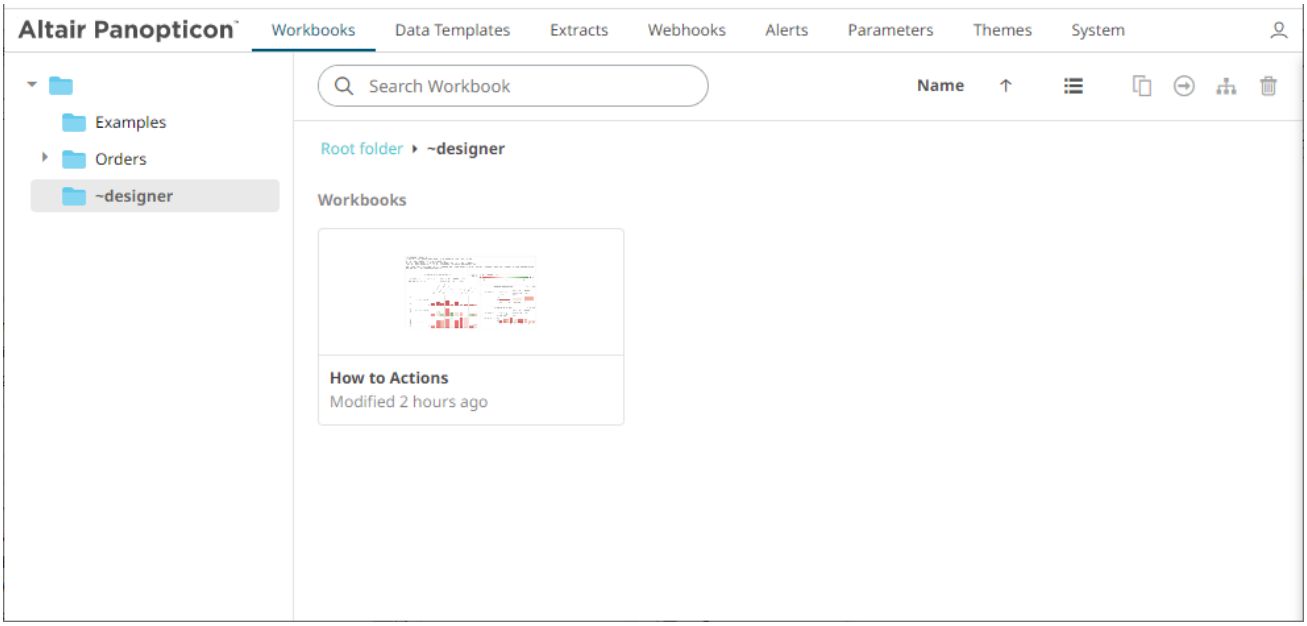


2. Select the folder or subfolder.



3. Click Move.

   The workbook is moved and displayed on the selected folder.

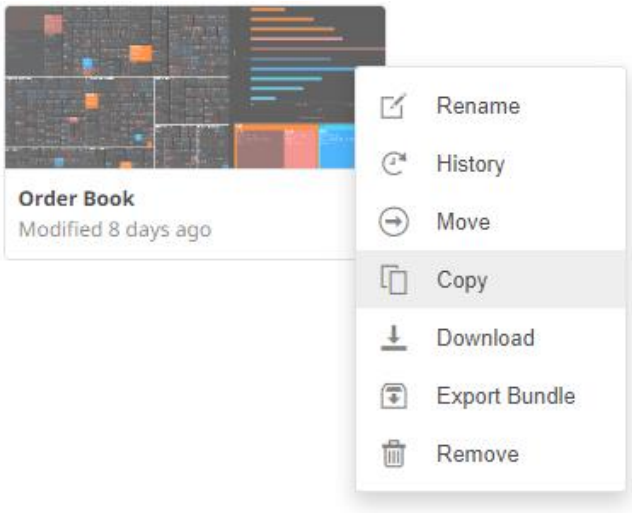


Copying Workbooks

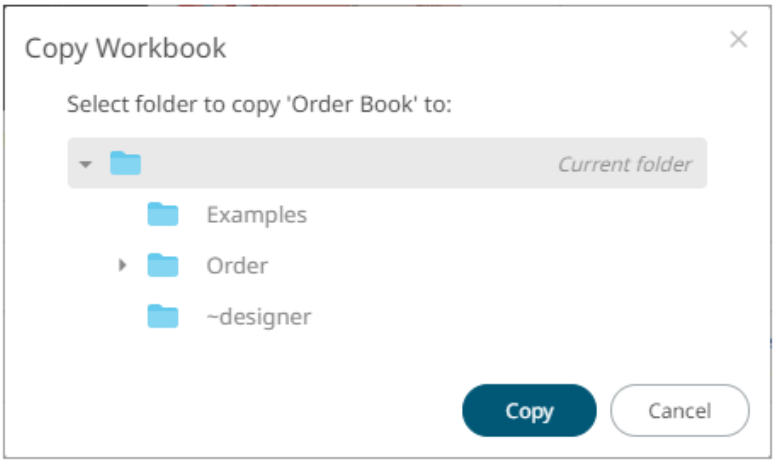
Users with Administrator or Designer role are allowed to copy a workbook to another folder or subfolder they have permission to.

Steps:

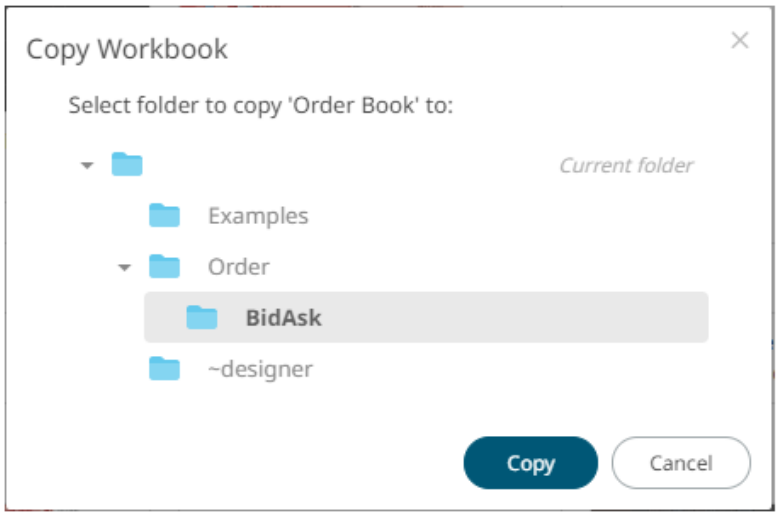
1. Right-click on a workbook and select **Copy** on the context menu.



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

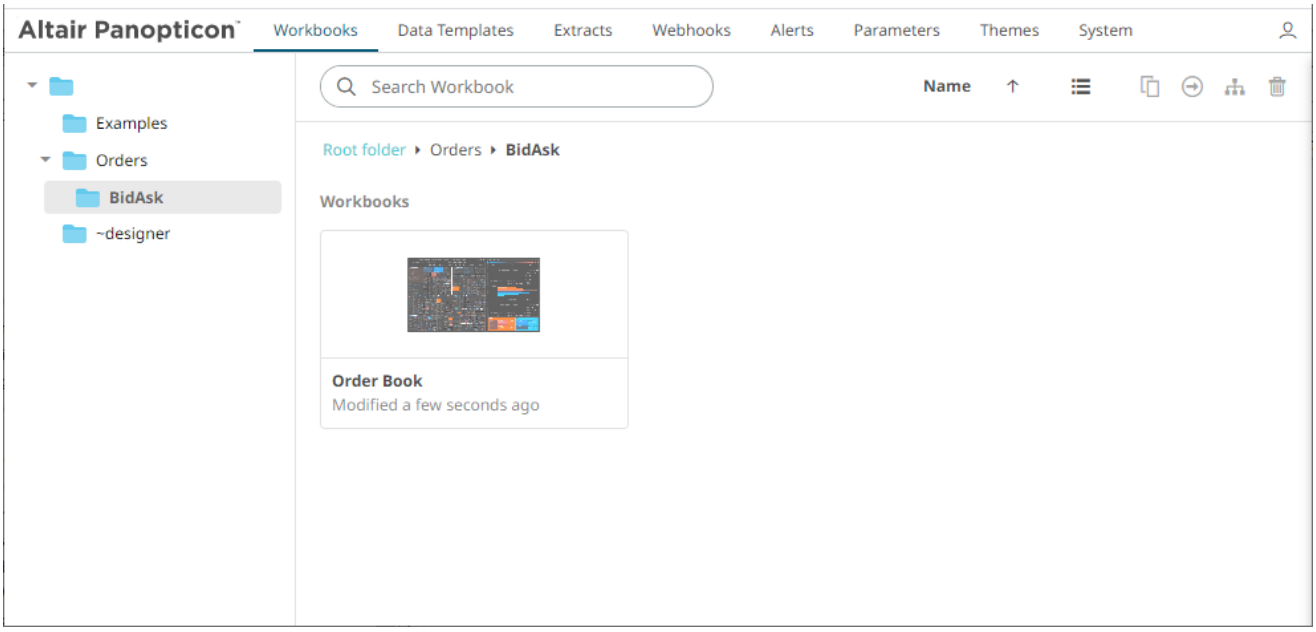


2. Select the folder or subfolder.



3. Click  .

The workbook is copied and displayed on the selected folder.



PARAMETER VALUE PASSING INTO THE WEB CLIENT

The Web client uses JSON URL query string to pass parameters.

For example:

```
/params/{"param1":"value1","param2":"value2"}
```

Again, parameter values must be URL encoded:

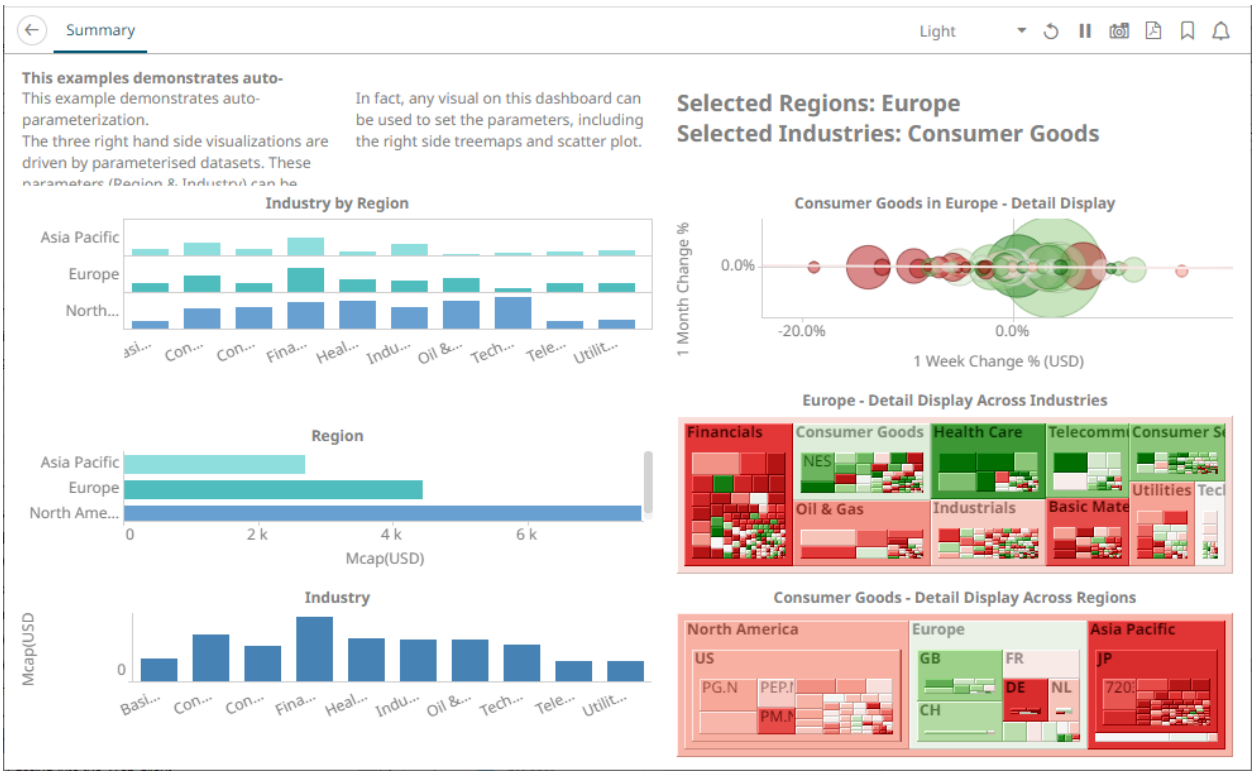
```
http://[host:port]/panopticon/workbook/#/[workbook_name]/[dashboard_name]/params/{"param1":"value1","param2":"value2"}
```

Where:

- Parameters are passed in JSON format
- Every parameter's name should be enclosed in double quotes (i.e., "")
- `/params/` sub-path should be placed in prior to JSON sections with parameters
- Special symbols in the parameter values should be URL-encoded. (Refer to *Special Symbols to Pass Parameter Values into the HTML5 Client* for more information.)

Here is an example URL with parameters that displays one of the example workbooks:

```
http://localhost:8080/panopticon/workbook/#/How%20to%20Auto%20Parameterize/Summary/params/%7B%22Region%22:%22Europe%22,%22Industry%22:%22Consumer%20Goods%22%7D
```



This workbook can also be displayed on the web browser using this URL:

```
http://localhost:8080/panopticon/workbook/#/How to Auto Parameterize/Summary/params/{"Region":"Europe","Industry":"Consumer Goods"}
```

To filter specific values, the array of values can be passed again in a standard JSON format, enclosing the array elements into square brackets:

```
{"Region":["Europe","North America"]}
```

For example:

```
http://localhost:8080/panopticon/workbook/#/How to Auto Parameterize/Summary/params/{"Region":["Europe","North America"],"Industry":"Consumer Goods"}
```

Special Symbols to Pass Parameter Values Into the HTML5 Client

When trying to pass parameters to the new HTML5 Client, you need to use URL-encoded characters.

For example, for `{"Type":[Soft/Drinks"]}` to work, it should be changed to `{"Type":[Soft%252FDrinks"]}`

Here is a list of double-encoded values you can use to replace their corresponding character.

Character	Double Encode Value
"<"	"%253C"
"/"	"%252F"
">"	"%253E"

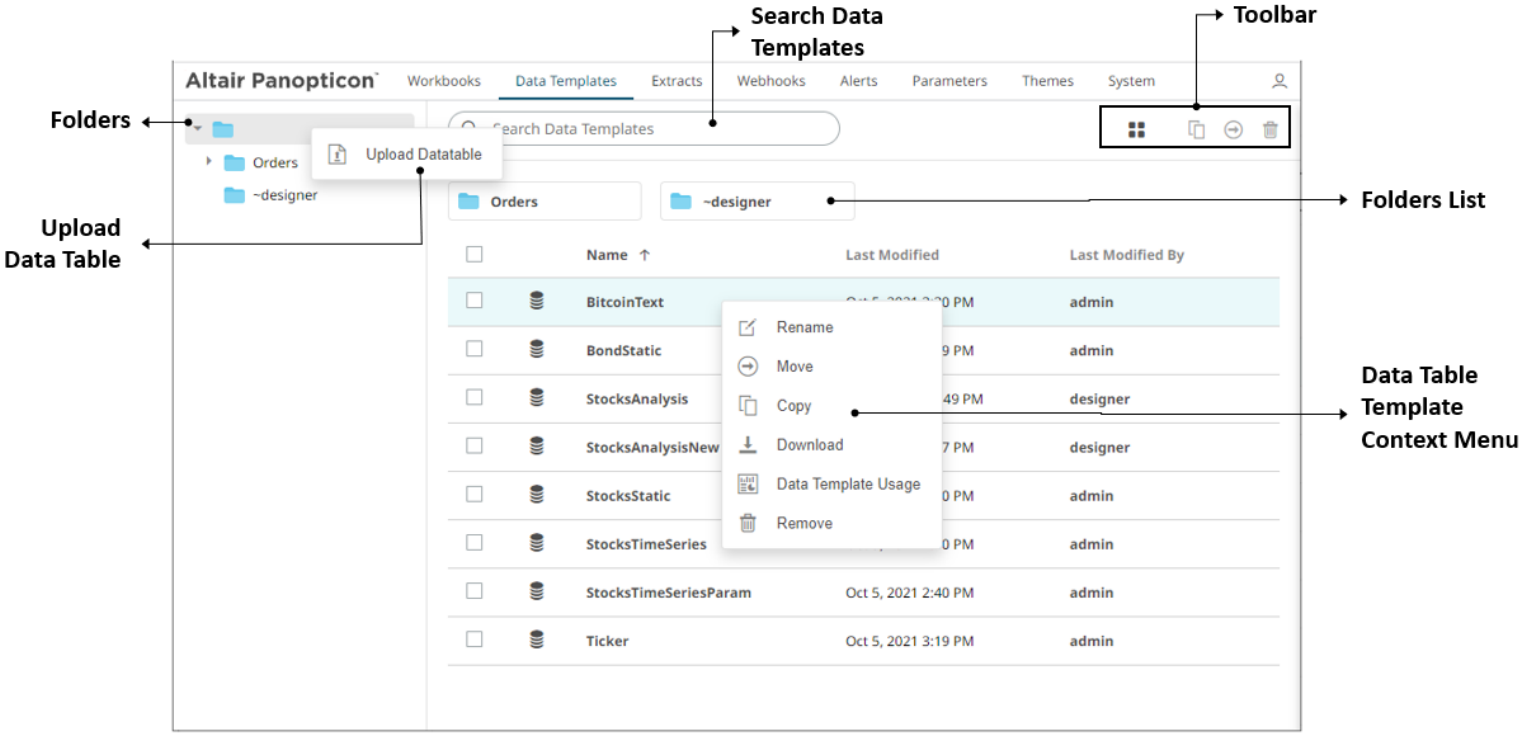
[9] DATA TEMPLATES

After the user with a Designer role successfully retrieves the data table using the Panopticon connectors and can publish it as a data template that:

- can be used by other Panopticon designers as a quick way of setting up a similar data table in any workbook.
- allows the changes in the data template to be appended to existing workbooks.

For more information on how to save and publish a data table template, refer to *Publishing a Data Table Template* section in *Altair Panopticon Web Authoring Guide*.

NOTE	Starting with version 21.2, data table templates are stored inside the repository and synced between servers in a cluster.

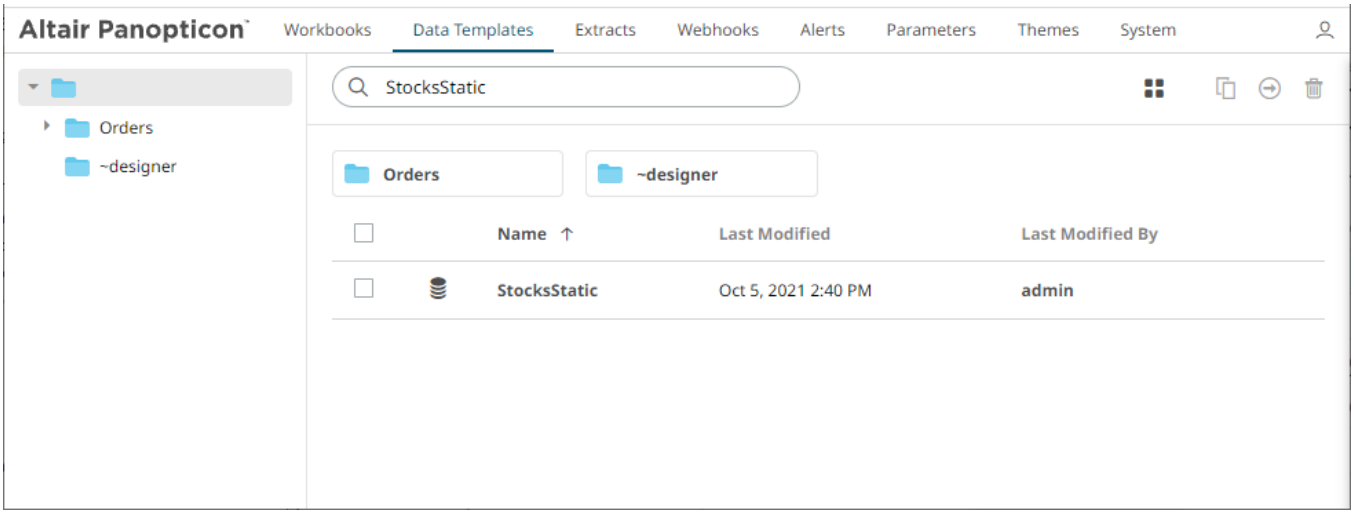


On the **Data Templates** page, Administrators can:

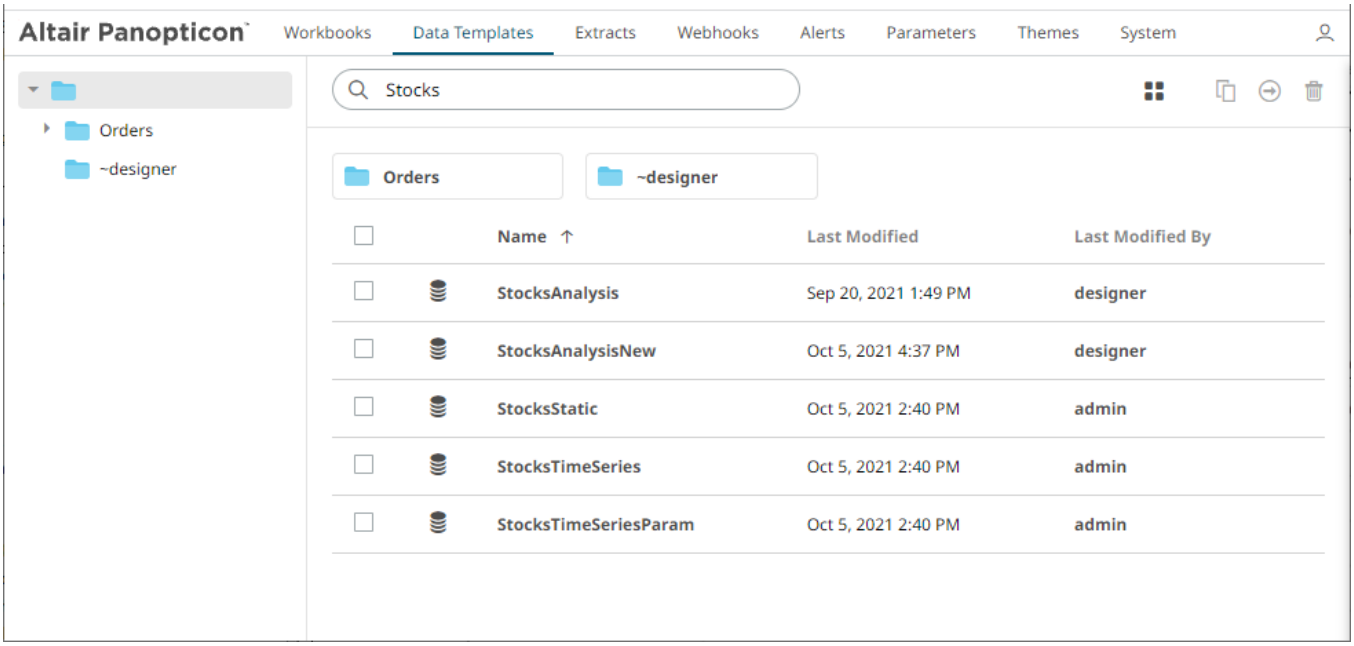
- View the list of published data table templates
- Search, sort, upload, rename, move, copy, download, or remove data table templates
- View the data template usage
- Update the referring workbook using the data template

SEARCHING FOR DATA TEMPLATES

On the *Data Templates* page, to search for a particular data template, enter it in the *Search Data Templates* box.



You can also enter one of more characters into the *Search Data Templates* box and the suggested list of data templates that matched the entries will be displayed.

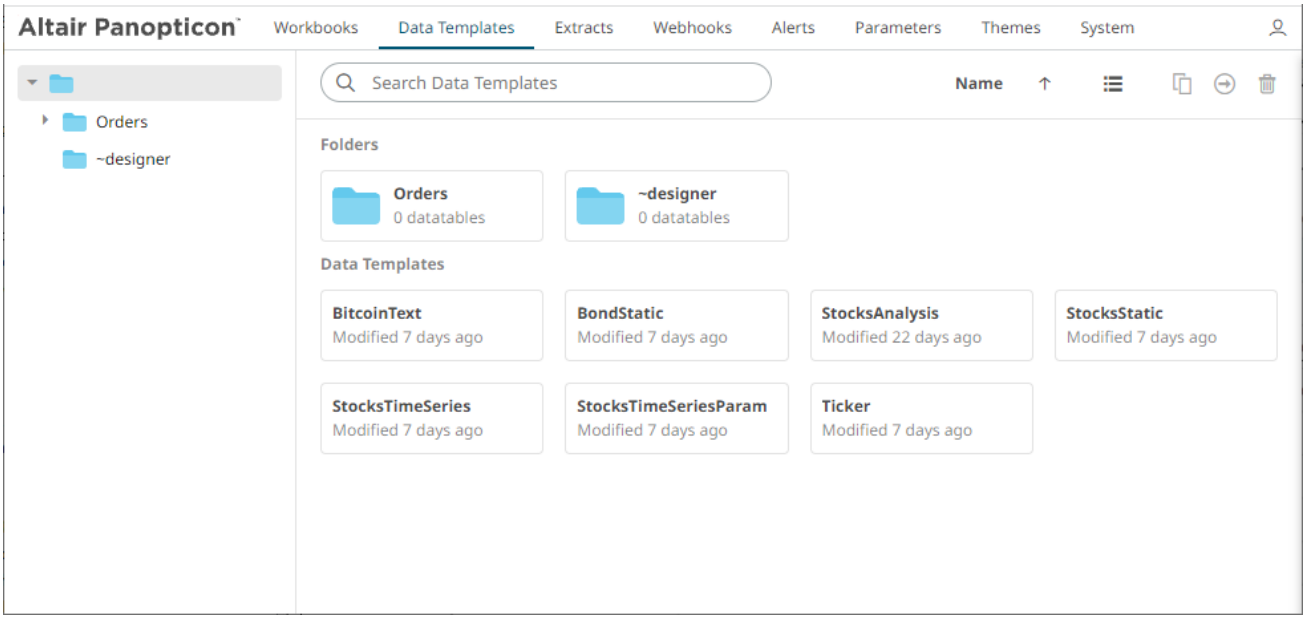


To clear the filter, delete the text entry in the *Search Data Templates* box.

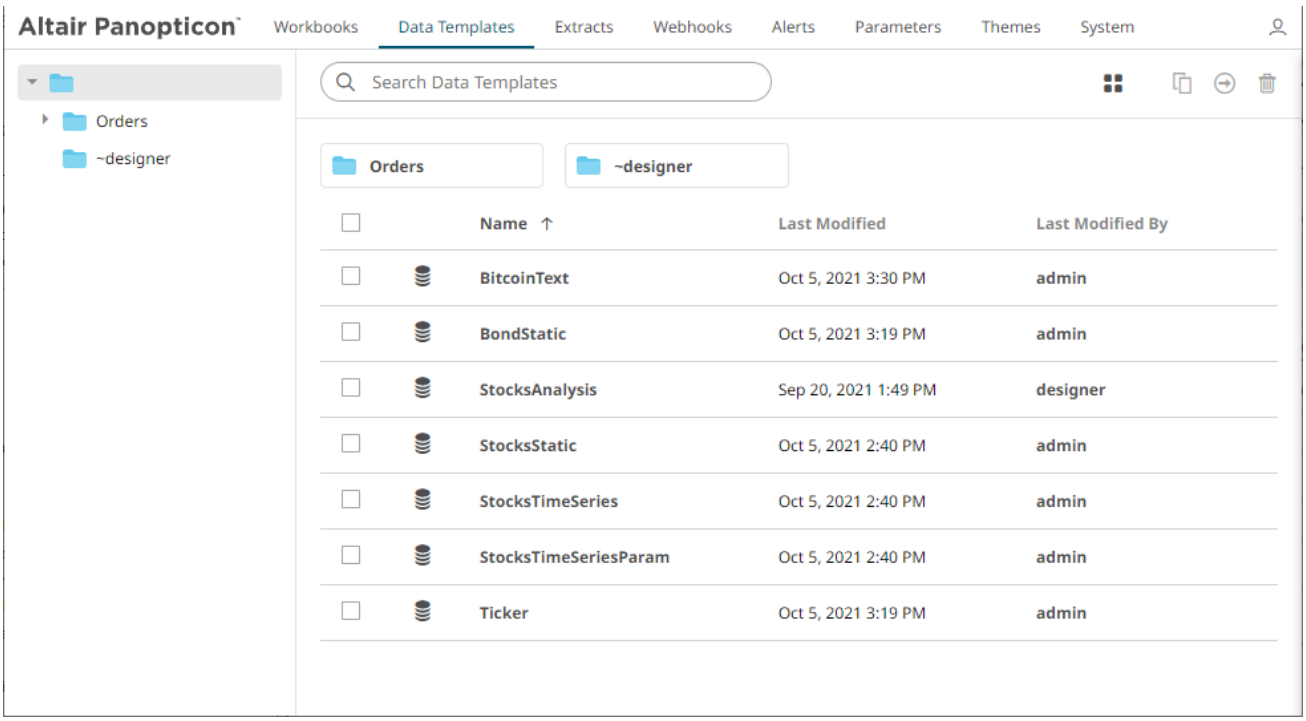
DATA TEMPLATES DISPLAY VIEW

Data templates can be displayed either on a *List* or *Grid View*.

On the *Toolbar*, click **Grid View** . The folders and data templates are displayed as thumbnails.



Or click **List View** , folders and data templates are displayed in a standard listing.



SORTING DATA TEMPLATES

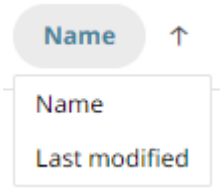
Sorting data templates can be done by *Name*, *Last Modified*, or *Last Modified By*.

Steps:

On the **Data Templates** page, either:

- ❑ click the **Sort By** option on the *Toolbar* of the *Grid View*

  By default, the sorting is by **Name**.

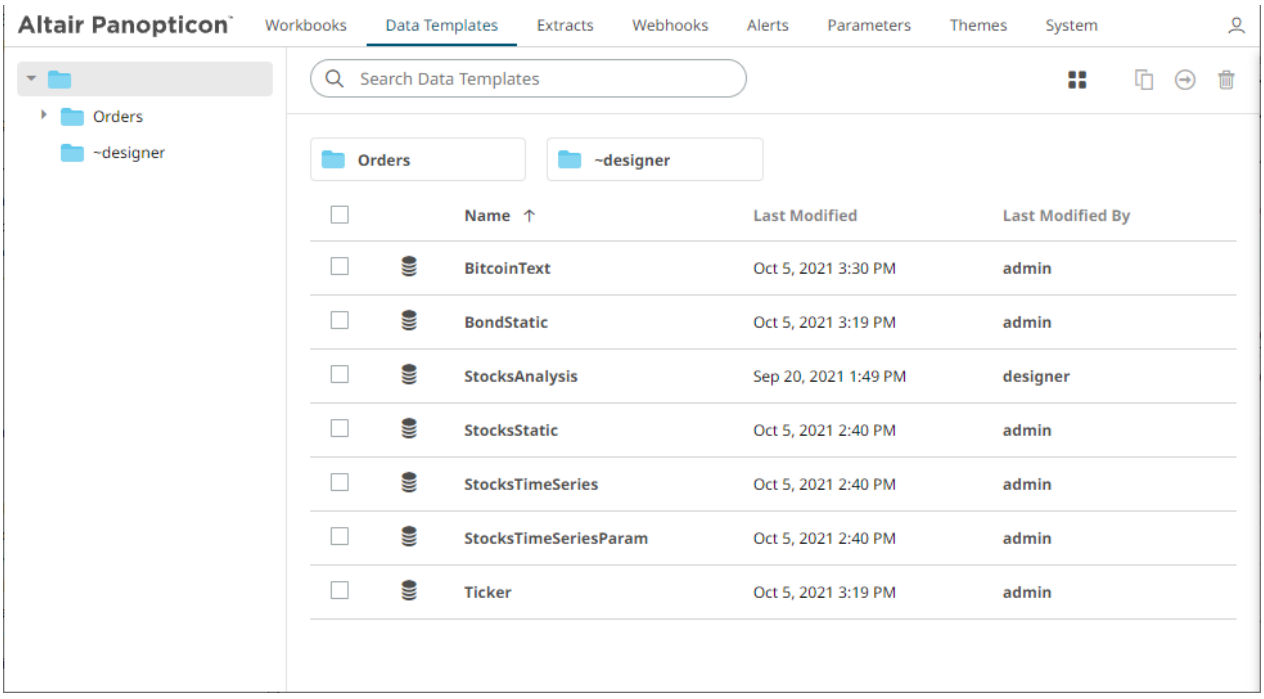
  

  - Name
  - Last Modified

  Then click the *Sort Order*:

  - Ascending
  - Descending

- ❑ click on the **Name**, **Modified**, or **Last Modified By** column header of the *List View*



Then click the *Sort Order*:

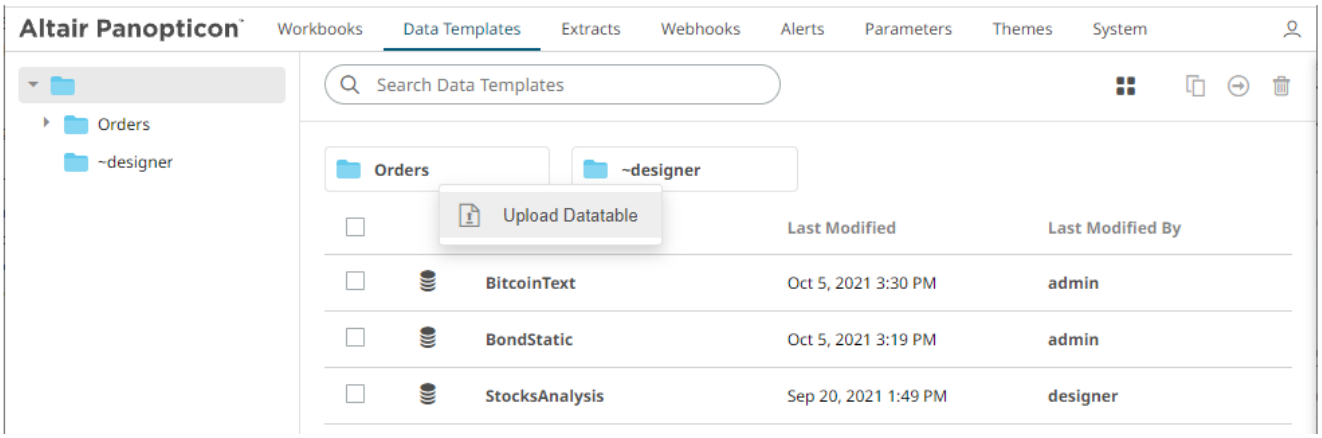
- ↑ Ascending
- ↓ Descending

UPLOADING A DATA TABLE TEMPLATE

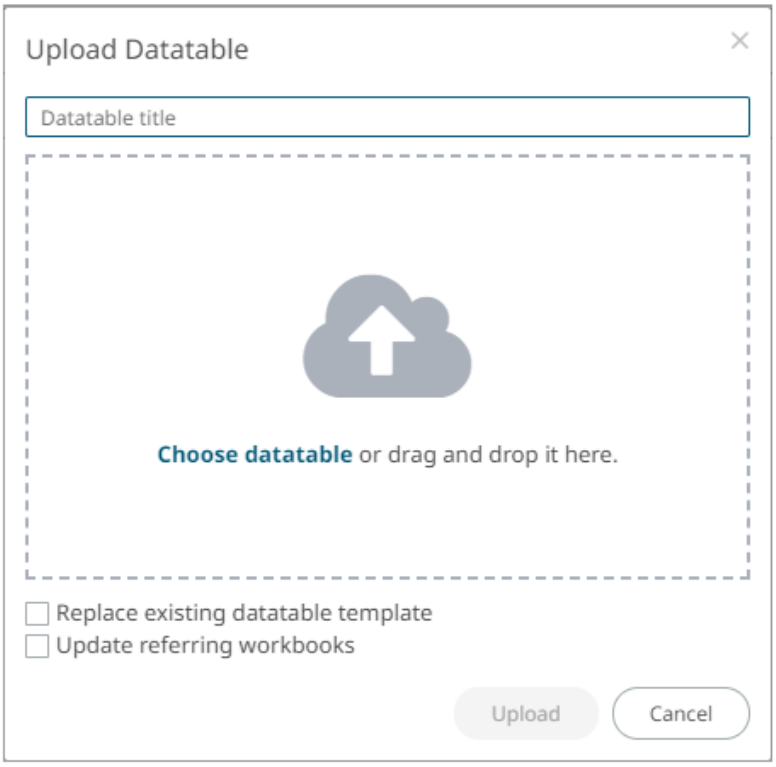
Users with an Administrator role can upload data table templates in the folders where they have permission.

Steps:

1. You can right-click on a folder either on the *Folders* pane or *Data Templates* pane and select **Upload Datatable** on the context menu.

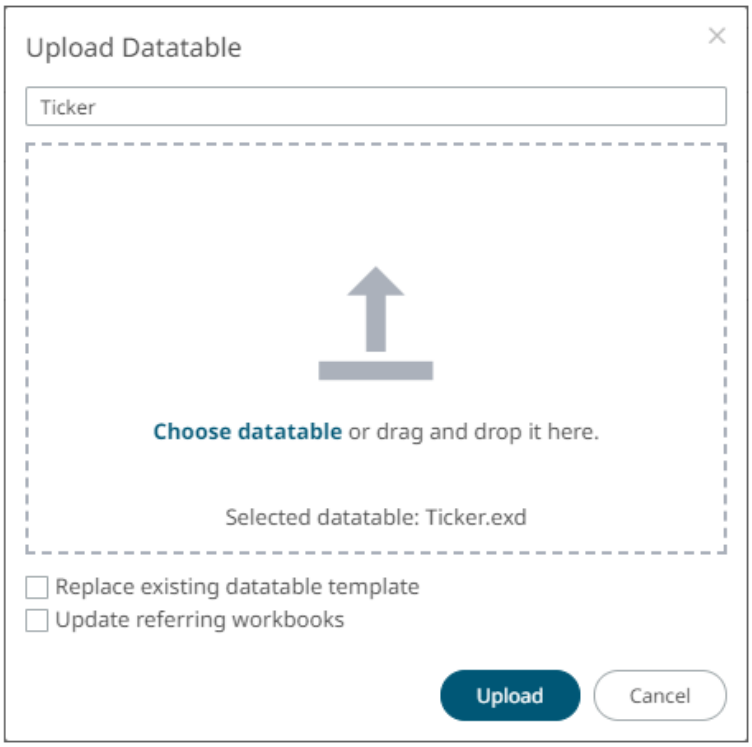


   The *Upload Datatable* dialog displays.



2. To upload a data table template, either:
   - drag the file from your desktop and drop on the dialog, or
   - click **Choose datatable** and then browse and select one on the Open dialog that displays.

   The name of the data table template is displayed on the uploaded data table area and in the *Name* box.



3. You can opt to rename the uploaded data table template.
4. Check the **Replace existing datatable template** box to replace the previously published data template with the same name to the new one.
5. Check the **Update referring workbooks** box to update the workbooks associated with the data template.
6. Click Upload .

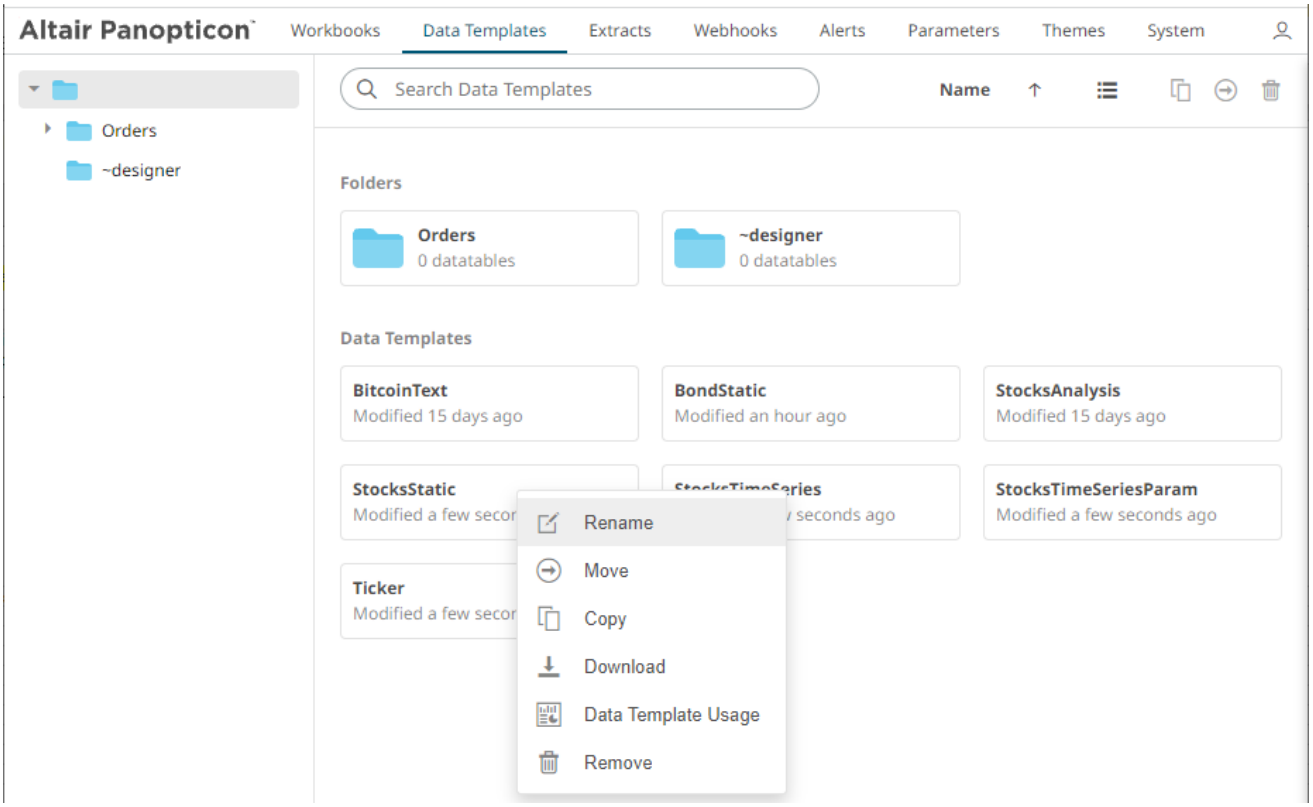
   The uploaded data table template is added in the list.

RENAMING DATA TEMPLATES

Users with an Administrator role can rename data table templates.

Steps:

1. On the *List* of *Grid* view, right-click on a data template then select **Rename** on the context menu.



   The *Rename Datatable* dialog displays.

2. Enter a new name then click Rename .

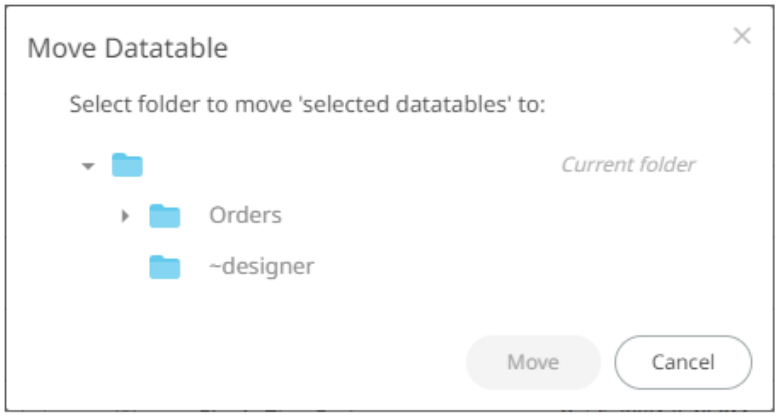
MOVING DATA TEMPLATES

Users with an Administrator role are allowed to move data table templates to another folder or subfolder where they have permission.

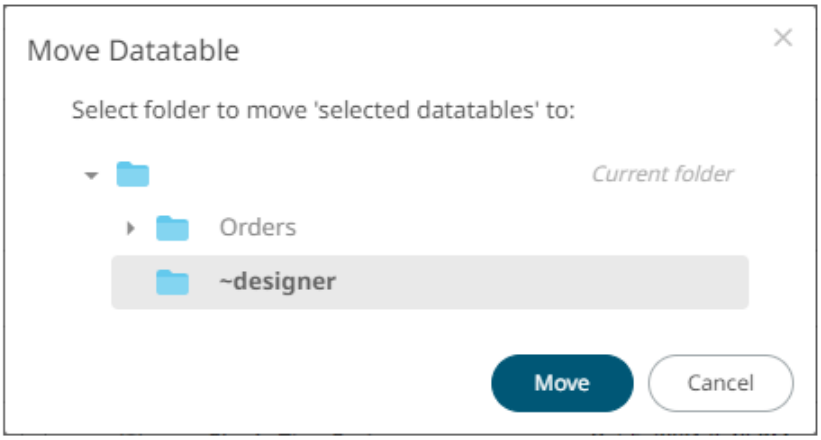
Steps:

1. On the *List* or *Grid* view, select one or several data templates then:
   - right-click and select **Move** on the context menu, or
   - click the **Move** icon on the toolbar.

   The *Move Datatable* dialog displays with the folder or subfolders the user is allowed to move the data template.

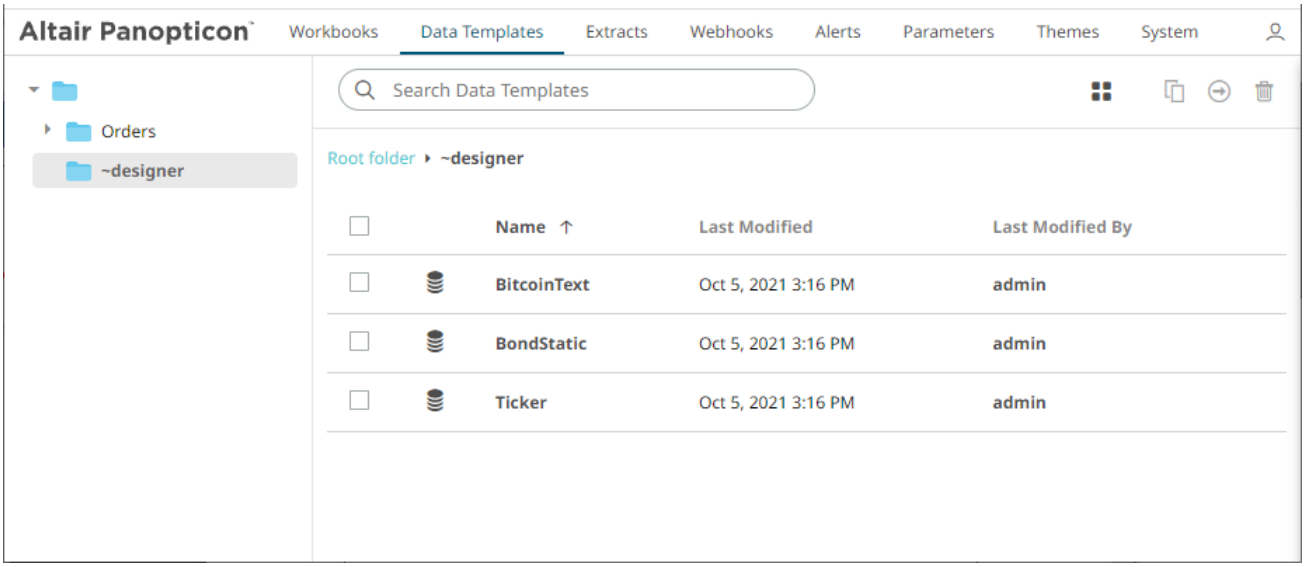


2. Select the folder or subfolder.



3. Click Move .

   The data templates are moved and displayed on the selected folder.



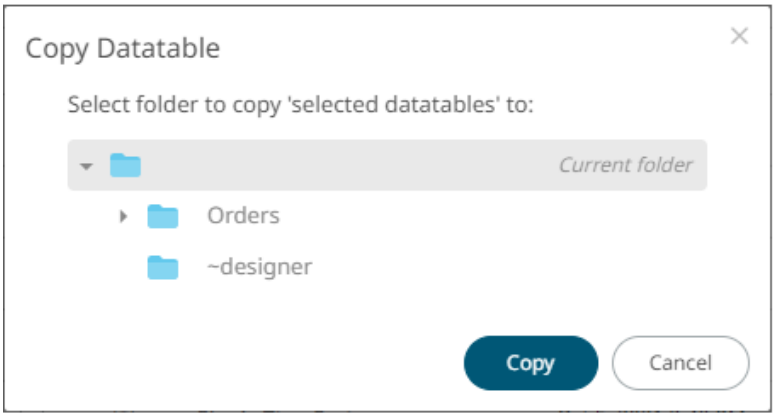
COPYING DATA TEMPLATES

Users with an Administrator role are allowed to copy data templates to another folder or subfolder where they have permission.

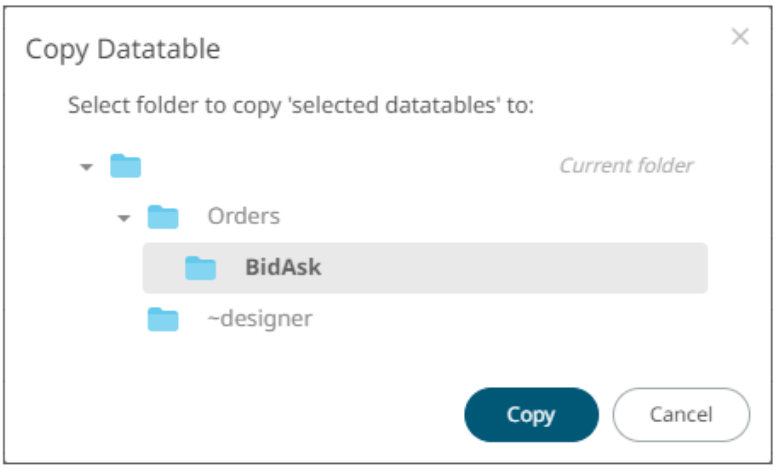
Steps:

1. On the *List* or *Grid* view, select one or several data templates then:
   - right-click and select **Copy** on the context menu, or
   - click on the **Copy** icon on the toolbar.

The *Copy Datatable* dialog displays with the folder or subfolders the user is allowed to copy the data template.



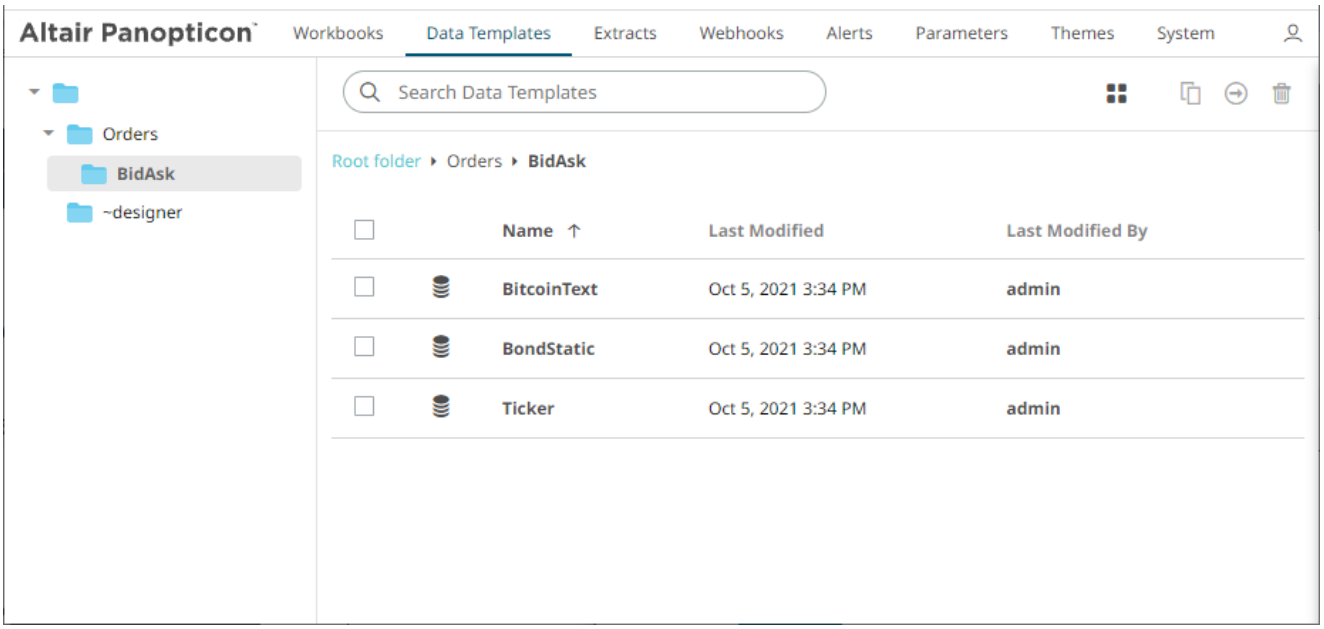
2. Select the folder or subfolder.



3. Click 

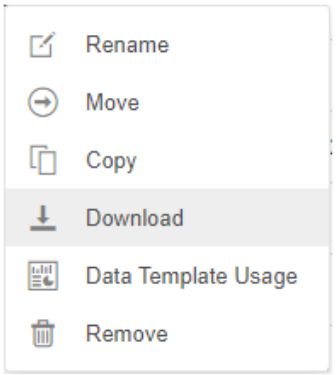
.

The data templates are copied and displayed on the selected folder.



DOWNLOADING A DATA TEMPLATE

On the *List* or *Grid* view, right-click on a data template and select **Download** on the context menu.



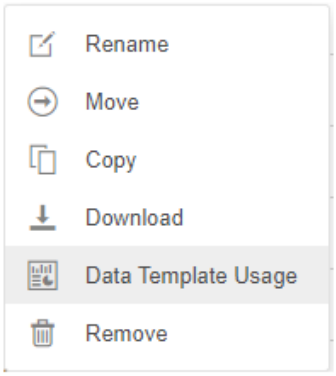
A copy of the data table template is downloaded.

VIEWING DATA TABLE TEMPLATE USAGE AND UPDATING THE REFERRING WORKBOOK

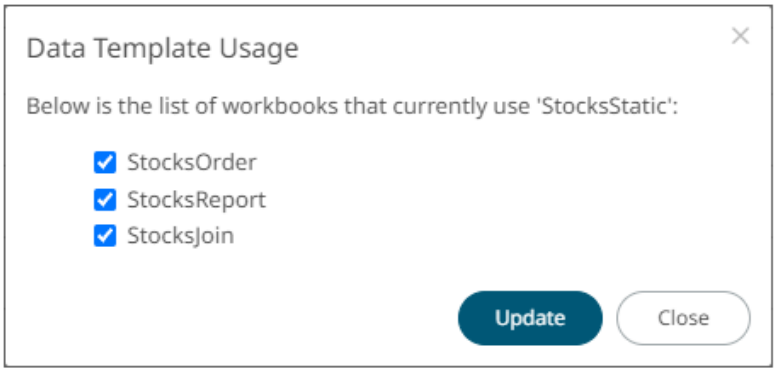
Users with an Administrator Role can view and update the workbooks that currently use the data table template.

Steps:

1. On the *List* of *Grid* view, right-click on a data table template then select **Data Template Usage** on the context menu.



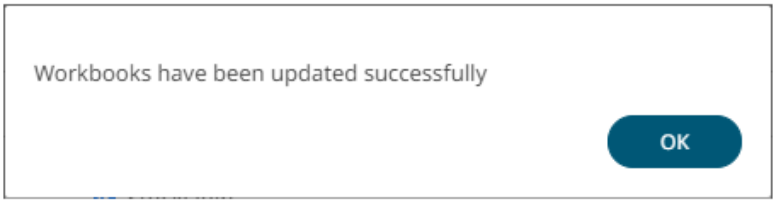
The *Data Table Template Usage* dialog displays.



2. Check the workbook boxes that will be updated based on the changes made on the associated data table template.
3. Click 

.

A notification message displays.



4. Click 

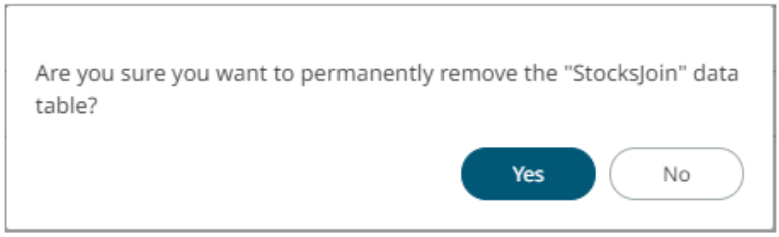
.

DELETING DATA TEMPLATES

Steps:

1. On the *List* or *Grid* view, select one or several data templates then:
   - right-click and select **Remove** on the context menu, or
   - click on the **Copy**  icon on the toolbar.
   -

   A confirmation message displays.

2. Click 

   .

   The data table template is removed.

DATA TABLE REGRESSION TESTING

If the `startup.regression.datatable.runonce` property is set to **true** in `Panopticon.properties`, the server will run a data table regression test the next time it starts up. The server goes through all workbooks and tries to load the data tables in them one by one. This can take some time. When the test has completed, the server proceeds to start up as usual. The `startup.regression.datatable.runonce` property is reset to **false** after a test run, so you need to change it each time you want to run the test.

You can configure the server to only test data tables in workbooks in particular folders with the `startup.regression.datatable.include.folders` and `startup.regression.datatable.exclude.folders` properties. This configuration can be useful if you have numerous workbooks, or have workbooks that are "work in progress" in some folders.

Any errors in the data table loading are written to a CSV file in `<appdata>/Reports/`. The errors include the following information:

- ❑ date and time when the test ran
- ❑ the full workbook path
- ❑ the data table title
- ❑ the error message
- ❑ the `server.id`, if the property is set

If there are no errors in any data table, then there will not be a file written.

The errors are also written to the server logs, as usual, so more information may be found there.

For data tables that have parameters, the default values -- as declared on the data table -- will be used. This means that a data table may fail in the regression test if you normally initialize a parameter using an action dropdown, the URL, the API, or any other means.

The server does not treat empty results as an error, and this will not be reported. For real-time data tables, it starts the subscription and then immediately stops it, so most of these will not have any rows. The regression test is primarily intended for configuration issues (e.g., forgot to change a host name, renamed a table in a database), not to check data health.

[10] CONNECTIVITY AND INTEGRATION

THIRD PARTY SOFTWARE DEPENDENCY INSTALLATION

Some data connectors require additional third-party software installation to be enabled which typically requires adding JAR files to the `Lib` folder of the Tomcat installation and then restarting Tomcat.

Common additions include:

- JDBC Drivers
- Advanced Message Processing System (AMPS)

  The latest version for AMPS can be downloaded from the 60East Technologies official website: http://www.crankuptheamps.com/amps/

  Copy `amps_client.jar`, `amps_client-javadoc.jar` and `amps_client-sources.jar` into the Tomcat `lib` folder.

  The pre-compiled JAR files are located in the `api\client\java\dist\lib\` directory, which contains the JAR files mentioned above.

  All of the above-mentioned java dependency files can be found after downloading and installing the AMPS Java Evolution Kit.

  If a user has Linux machine available, install the AMPS distribution. Otherwise, download the AMPS Evolution Virtual Machine.

NOTE	To effectively use the `.jar` files, unblock these files by right-clicking on the File and selecting Properties. On the General tab, click Unblock.

- Elasticsearch connectors

  Dependencies for each supported Elasticsearch version are included in Panopticon Real Time installation:

  - Elastic_6X_Dependencies.zip
  - Elastic_7X_Dependencies.zip

  Select the target Elasticsearch version and unzip the contents of the appropriate dependency zip into the Tomcat `lib` folder.

- JMX

  Use the following java options to enable JMX monitoring for the JMX plugin:

  Enable JMX remote connection: `(-Dcom.sun.management.jmxremote)`

  Disable JMX authentication: `(-Dcom.sun.management.jmxremote.authenticate=false)`

  Set remote port for jmx: `(-Dcom.sun.management.jmxremote.port=number)`

NOTE	Providing invalid parameters into JMX connection string may cause a number of exceptions and make the server inaccessible. Make sure you are using the syntax provided above.

- OneMarketData OneTick / OneTick CEP

  This connector requires that the following JAR be added:

  `jomd.jar`

  Which is retrieved from the OneTick `bin` folder:

  For example:

  `C:\omd\one_market_data\one_tick\bin`

  Additionally, the following environment variables **MUST** be configured:

  PATH

  To include the OneTick `bin` folder.

  For example:

  `C:\omd\one_market_data\one_tick\bin`

  ONE_TICK_CONFIG

  To reference the OneTick configuration file.

  For example:

  `C:\omd\client_data\config\one_tick_config.txt`

  Plus, the Tomcat configuration should include the following Java option:

  `-Djava.library.path=C:\omd\one_market_data\one_tick\bin`

  The OneTick configuration file should have entries for Windows OS time zone mapping and information.

  Example:

  ```
  WINDOWS_TZ_MAPPING_FILE="C:/OMD/one_market_data/one_tick/config/windows_tz_mapping.dat"
  ```

  ```
  WINDOWS_ZONEINFO_PATH="C:/OMD/one_market_data/one_tick/config/zoneinfo"
  ```

  Additionally, the OneTick client folder should be set to have the same permissions as those running the Tomcat process. Please check that the OneTick Java API is operational, before accessing workbooks through the server that utilize OneTick connectivity. This can be easily achieved by running one of the OneTick Java API examples.

NOTE	
	• The OneTick JAR must be updated to match the version of the OneTick client installation. • For version 16.7.0, the OneTick connector is built and tested against version 1.17 of the OneTick Client.

- SAP Sybase ESP

Manually copy the following dependency files from the Sybase ESP installation folder (e.g., `C:\Sybase\ESP-5_1\libj):`

- commons-codec-1.3.jar
- log4j-1.2.16.jar
- streaming-client.jar
- streaming-system.jar
- ws-commons-util-1.0.2.jar
- xmlrpc-client-3.1.3.jar
- xmlrpc-common-3.1.3.jar

NOTE	
	Make sure the dependency files are copied to the appropriate `WEB-INF` folder in Apache Tomcat: • For 64-bit: `C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\panopticon\WEB-INF\lib` • For 32-bit: `C:\Program Files (x86)\Apache Software Foundation\Tomcat 9.0\webapps\panopticon\WEB-INF\lib`

- StreamBase CEP

This connector requires the following JAR to be added:

`sbclient.jar`

Which is retrieved from the StreamBase `Lib` folder.

For example: `C:\TIBCO\sb-cep\7.5\lib`

- StreamBase LiveView

This connector requires the following JAR files to be added:

- `sbclient.jar`, `lv-client.jar`, `lv-client-wwwdeps.jar`

  Which are retrieved from the StreamBase `Lib` folder.

  For example: `C:\TIBCO\sb-cep\7.5\lib`

Plus, the JARS from the LiveView installation:

- `lv-compiler.jar, jyaml-1.3.jar`

  Which are retrieved from the LiveView `Lib` folder.

  For example: `C:\TIBCO\sb-cep\7.5\liveview\lib`

DATABASE

There are two ways of connecting to a database from Altair Panopticon Real Time.

a. Use the Listed Data Connector for the specific Database (if available).

   Includes: Cassandra, Elasticsearch 6.x, Elasticsearch 7.x, InfluxDB, Kx kdb+, ksqlDB, LivySpark, MongoDB, OneTick, OneTick Cloud, Panopticon Data Extract, Splunk.

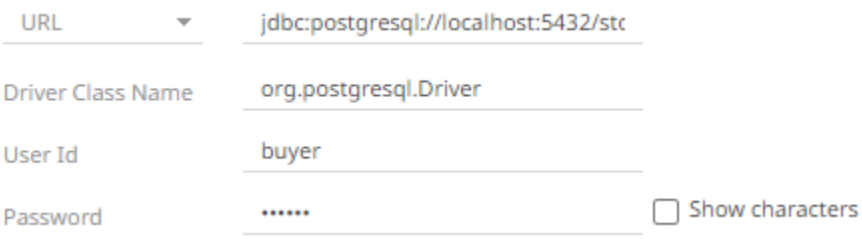
b. Use the JDBC connector.

   This requires:

   1. Addition of the JDBC JAR(s) for the required Database into `Tomcat/Lib`.
   2. For JNDI:

      Update of the server configuration file: `panopticon.xml` to include the new JNDI resource name.

   3. For URL: Use the *URL* specific to the database's JDBC driver, the *Driver Class Name* specific to the driver, and the *Username* and *Password*.



JDBC Driver Installation

Install the relevant JDBC driver(s) on the system where you are running Tomcat and Panopticon Real Time. The exact installation procedure depends on the JDBC driver. Follow the instructions given by the provider of the JDBC driver and by the provider of your Java application server (for example, Apache Tomcat). In almost every case, a JDBC driver is installed by placing one or several jar-files in the `lib` folder of your Tomcat installation.

JNDI Connection Details

JNDI Connection details are specified in Panopticon Real Time configuration file `panopticon.xml`.

Each connection has the following structure:

```
<Resource name="jdbc/[Unique Name]"
       auth="Container"
```

```
        type="javax.sql.DataSource"
        maxActive="100"
        maxIdle="30"
        maxWait="10000"
        username="[User Name]"
        password="[Password]"
        driverClassName="[Class Name]"
        url="[URL]"
/>
```

Where:

- **Unique Name**: Defines the unique JNDI resource name to be used.
- **User Name**: The username to authenticate to the database.
- **Password**: The password to authenticate to the database.
- **Class Name**: The Class Name specific to the Database's JDBC Driver.
- **URL**: The URL specific to the Database's JDBC Driver, and selected Server instance and database.

Additionally, other key attributes of the JNDI resource are:

- **maxActive:** The maximum number of active connections that can be allocated from this pool.
- **maxIdle:** The maximum number of connections that will be kept active even when there are no requests.
- **maxWait:** Maximum time in milliseconds to wait for a database connection to become available.

Common Databases and their JNDI Configurations

Database	Description
Oracle 11	Using `ojdbc6.jar` • `driverClassName="oracle.jdbc.OracleDriver"` • `url="jdbc:oracle:thin:@[HostName]:1521:[DatabaseName]"/>`
MS SQL Server	Using `sqljdbc4.jar` • `driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"` • `url="jdbc:sqlserver://[Server]\[Instance];databaseName=[Database Name]"/>`
Sybase ASE	Using `jconn4.jar` • `driverClassName="com.sybase.jdbc4.jdbc.SybDriver"` • `url="jdbc:sybase:Tds:[HostName]:5000/[DatabaseName]"`
PostgreSQL	Using `postgresql-9.4.1208.jar` • `driverClassName="org.postgresql.Driver"` • `url="jdbc:postgresql://[HostName]:5432/[DatabaseName]"`
MySQL	Using `mysql-connector-java-5.1.38-bin.jar` • `driverClassName="com.mysql.jdbc.Driver"` • `url="jdbc:mysql://[HostName]:3306/[DatabaseName]"/>`

R AND PYTHON TRANSFORM SUPPORT

R and Python connectivity and transforms occur over TCP/IP network links.

- For R, Rserve is used.
- For Python, Pyro (Python Remote Data Objects) is used.

R Integration

To enable R connectivity:

1. Download R, install it, and the R Console (http://cran.rstudio.com/).
2. Open the R Console.
3. Install Rserve using the following command from within the R Console:

   ```
   install.packages("Rserve")
   ```

4. Initiatiate the Rserve library using the following command:

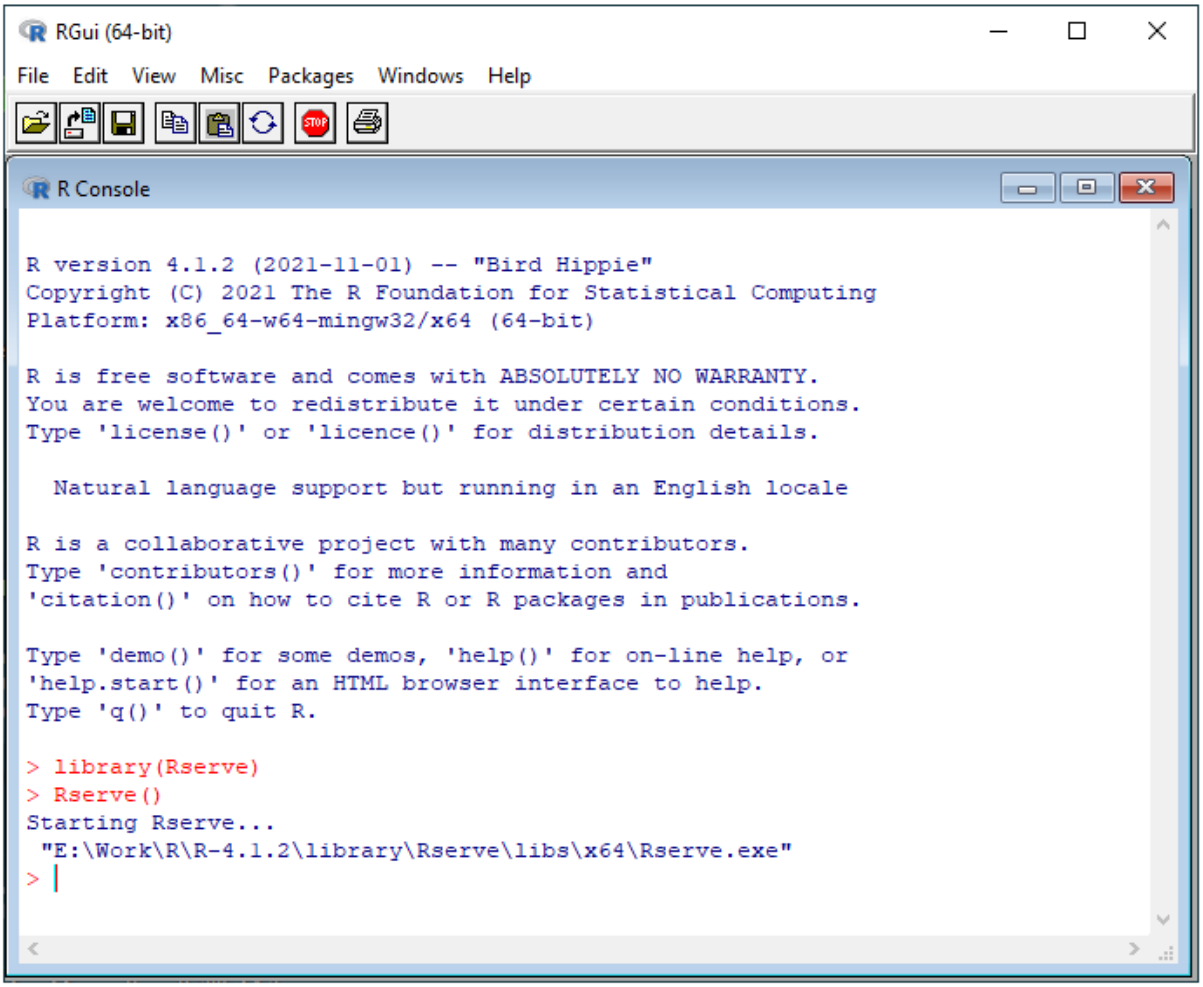
   ```
   library(Rserve)
   ```

5. Run Rserve by executing the following command:

   ```
   Rserve()
   ```

Only steps 2, 4 & 5 need to be repeated when R connectivity is required.

Example:



NOTE	Connectivity by default is over Port 6311.

To enable authentication across the Rserve TCP/IP link

- create a password file (pwdfile.pwd)

  Each line of the file should have the user and then the password.

  Example:

  ```
  user1 password1
  user2 password2
  ```

- Create a configuration file with following parameters (`rconfig.conf`)

  ```
  auth required
  pwdfile [path of password file]
  ```

Example:

```
remote enable
auth required
port 6311
pwdfile C:\\RIntegration\\pwdfile.pwd
```

- load the created configuration file (the default Rserve configuration file is still loaded, but its settings have lower priority) and run Rserve:

```
Rserve(args="--RS-conf [path of configuration file]")
```

Example:

```
Rserve(args="--RS-conf C:\\RIntegration\\rconfig.conf")
```

Python Integration

To enable Python connectivity:

1. Download and install Python.

   For ease of deployment we suggest the Anaconda Python distribution.

   https://www.anaconda.com/distribution/

2. Add Python installation folder to your **Path** environment variable.

> **NOTE** On Windows, when using the Anaconda distribution of Python, you should add the following folder paths to your Windows System Environment variable Path:
>
> - \Anaconda3\
>
>   The folder where Python is installed.
>
> - \Anaconda3\Library\bin
>
>   The folder containing DLLs related to the installed Python packages.

3. Download the latest version of Pyro4.

   https://pypi.org/project/Pyro4/

   For Windows, download the Wheel: `Pyro4-4.79-py2.py3-none-any.whl`

4. Open the Anaconda Prompt. Navigate to the location of `Pyro4-4.79-py2.py3-none-any.whl` and run:

   ```
   Pip install Pyro4-4.79-py2.py3-none-any.whl
   ```

   Pyro is now installed into the Python site packages.

5. Copy the `start_Python_connectivity.bat, start_Python_connectivity.sh,` and `pyro.py` files from Panopticon Real Time Installation zip to a new location.

6. Depending on your operation system:
   - Linux – run the `start_Python_connectivity.sh` file
   - Windows – run the `start_Python_connectivity.bat` file

Only step 6 needs to be repeated when Python connectivity is required.

NOTE	To uninstall prior versions of Pyro: `pip uninstall Pyro4`

The default password to secure connectivity is set to `password.`

To change this:

- For Linux

  Modify `start_Python_connectivity.sh`, editing the line:

  ```
  python -m Pyro4.naming -n $1 -k "password"&
  ```

- For Windows

  Modify `start_Python_connectivity.bat`, editing the line:

  ```
  start python -m Pyro4.naming -k "password"
  ```

Modify `pyro.py`, editing the line:

```
ns = Pyro4.locateNS(host=host, hmac_key="password")
```

Pyro is set to listen on a specific host/IP, which by default is localhost.

This can be modified through editing the batch file that instantiates Pyro.

```
@echo on
start python -m Pyro4.naming -n "[Required HostName]" -k "password"
ping localhost
cd "%~dp0"
python pyro.py "[Required HostName]"
pause
```

To enable the Pickle serialization, modify the `configuration.py` file located in `..\Anaconda3\Lib\site-packages\Pyro4` to specify the serialization to be used.

For example, if **Pickle** is selected, `self.SERIALIZER` value should be changed to **pickle** and `self.SERIALIZERS_ACCEPTED` value should be changed to include **pickle**:

```
def reset(self, useenvironment=True):
        """
        Set default config items.
        If useenvironment is False, won't read environment variables
settings (useful if you can't trust your env).
        """
        self.HOST = "localhost"  # don't expose us to the outside
world by default
        self.NS_HOST = self.HOST
        self.NS_PORT = 9090  # tcp
        self.NS_BCPORT = 9091  # udp
        self.NS_BCHOST = None
        self.NATHOST = None
        self.NATPORT = 0
        self.COMPRESSION = False
        self.SERVERTYPE = "thread"
        self.COMMTIMEOUT = 0.0
        self.POLLTIMEOUT = 2.0  # seconds
        self.SOCK_REUSE = True  # so_reuseaddr on server sockets?
        self.SOCK_NODELAY = False  # tcp_nodelay on socket?
        self.THREADING2 = False  # use threading2 if available?
        self.ONEWAY_THREADED = True  # oneway calls run in their own
thread
        self.DETAILED_TRACEBACK = False
        self.THREADPOOL_SIZE = 16
        self.AUTOPROXY = True
        self.MAX_MESSAGE_SIZE = 0  # 0 = unlimited
        self.BROADCAST_ADDRS = "<broadcast>, 0.0.0.0"  # comma
separated list of broadcast addresses
        self.FLAME_ENABLED = False
        self.PREFER_IP_VERSION = 4  # 4, 6 or 0 (let OS choose
according to RFC 3484)
        self.SERIALIZER = "pickle"
        self.SERIALIZERS_ACCEPTED = "pickle,marshal,json"   # these
are the 'safe' serializers
        self.LOGWIRE = False  # log wire-level messages
        self.PICKLE_PROTOCOL_VERSION = pickle.HIGHEST_PROTOCOL
        self.METADATA = True  # get metadata from server on proxy
connect
        self.REQUIRE_EXPOSE = False  # require @expose to make
members remotely accessible (if False, everything is accessible)
        self.USE_MSG_WAITALL = hasattr(socket, "MSG_WAITALL") and
platform.system() != "Windows"      # not reliable on windows even
though it is defined
        self.JSON_MODULE = "json"
        self.MAX_RETRIES = 0
```

LOAD CUSTOM DATA PLUGINS

Panopticon Real Time will load a file named **Plugins.xml** during startup. The file contains class names of all the data plugins that will be loaded and applied to the server. However, the `Plugins.xml` file can be replaced in case the user wants to have a custom setup and load their own plugins or if they want to disable certain data plugins from being loaded. This is achieved by creating a new `plugins.xml` file and placing it in the `AppData` folder (e.g., `c:\vizserverdata`).

The original `plugins.xml` file is always distributed with the `panopticon.war` file. From the `.war` file, copy the `plugins.xml` file from the root folder to your `AppData` (i.e., `c:\vizserverdata`) folder. Then open `plugins.xml` and add or remove items to either enable or disable certain plugins.

NOTE	New data plugins are constantly being developed and distributed. Therefore, it is recommended that you revisit the shipped `plugins.xml` file after each release if you have replaced the default `plugins.xml` file.

[11] CACHING

Panopticon Real Time supports five levels of caching:

- Data extracts, offline and manually created
- Real-time subscription cache
- Data source cache
- Data table cache
- Query result cache

All of which are optional. If caching is specifically not desired, data requests can always be forwarded to the underlying data repository.

The subscription cache describes the cache used for streaming subscriptions. This cache is used to ensure subscriptions are not duplicated by the server, and that instead the server manages duplicate end client subscription requests. Subscriptions are started when the server receives a valid client request and can be set to stop when users are no longer watching data from them or be kept alive until the server is stopped.

The data caches simply keep corresponding data sources and tables in memory to avoid unnessary reloads from the underlying data repositories. Neither is used for real-time data, but the data source cache helps with real-time data is joined to standing data. The cache entries are keyed on:

- The workbook
- The data table
- (The data source)
- Parameter values

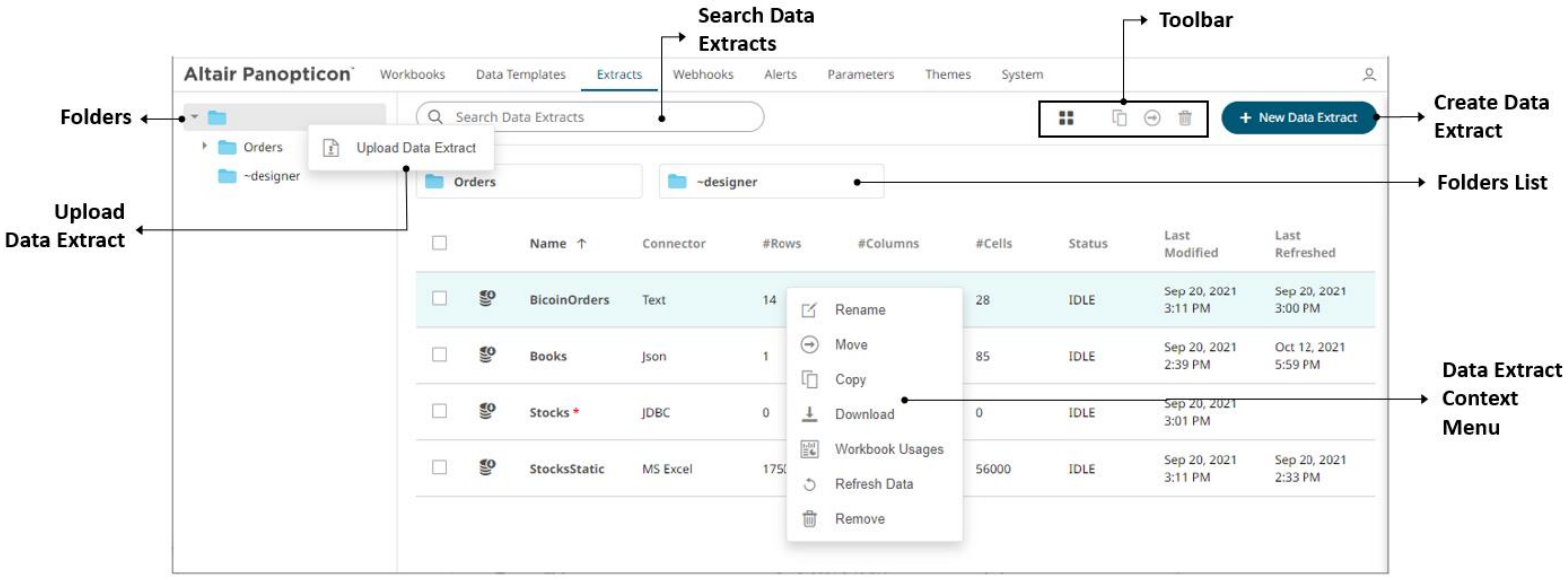
The time-to-live (TTL) for entries is based on the auto refresh period set on the data table.

The query result cache stores the result of a query from an individual visualization, filter, or legend on a dashboard. It is useful if many users are viewing the same dashboard, when many identical queries will be sent in parallel to the server. It also caches real-time data for this purpose.

DATA EXTRACTS

Starting with version 21.0, data extracts are stored in the repository. If the `repository.migrate.data.extracts.path` property in the `Panopticon.properties` file (located in the `AppData` folder) is set to **GlobalCaches**, or to an absolute path, the server will migrate data extracts into the repository on start up as long as the repository does not contain any previous data extracts.

In addition, data extracts are now associated with a folder to allow access control. The data extracts folder structure is a copy of the workbook folder and is kept in sync with folder changes made to to that folder tree.



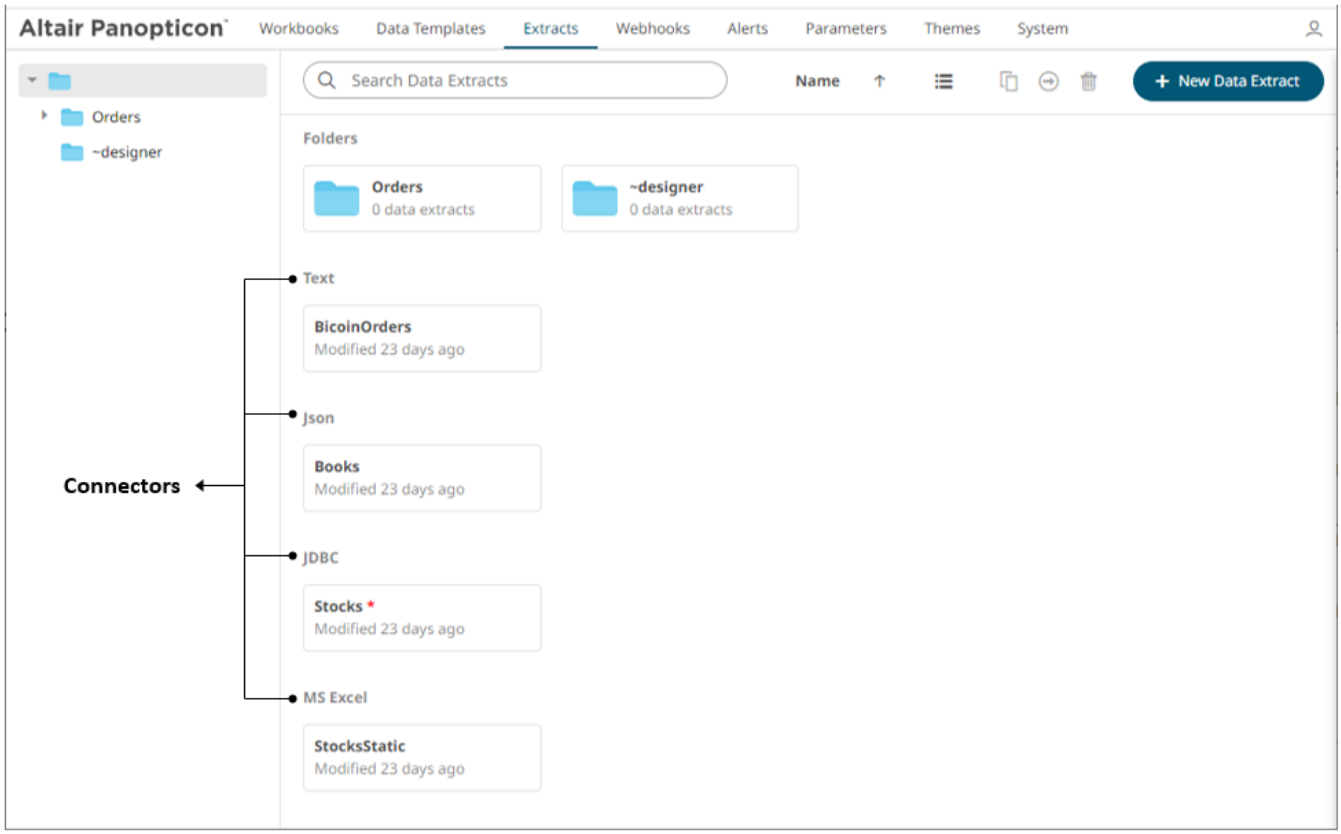
Property	Description
Folders	List of folders where data extracts can be uploaded, moved, or copied.
Folder Context Menu	Allows uploading of data extracts to folders.
Search Data Extract	Entering text will filter the returned data extracts.
Toolbar	Allows copying, moving, and removing of data extracts. Also, to display the data extracts list either on *List View* or *Grid View*.
Create Data Extract	Allows to create a new data extract.
Folders List	Available folders on *List View*.
Data Extract Context Menu	Allows renaming, moving, copying, deleting, and downloading of a data extract. Also, viewing of workbook usages and refreshing of data extract.

Folders and Data Extracts Display View

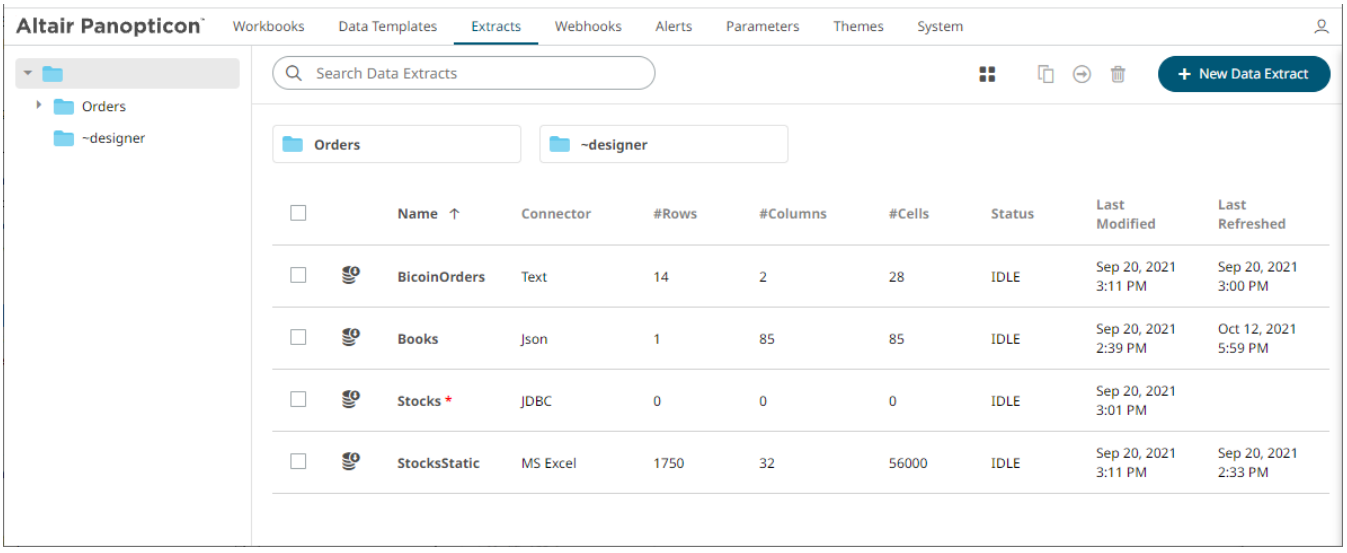
Data extracts can be displayed either on a *List* or *Grid View*.

On the *Toolbar*, click **Grid View**  . The folders and data extracts are displayed as thumbnails.

NOTE	Data extracts are placed under their corresponding connector.



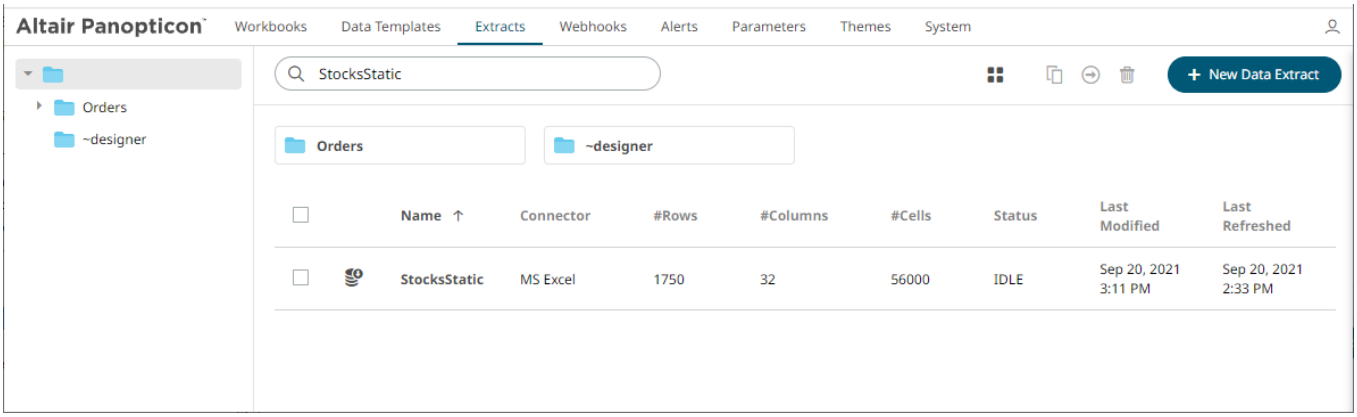
Or click **List View** , the folders and data extracts are displayed in a standard listing.



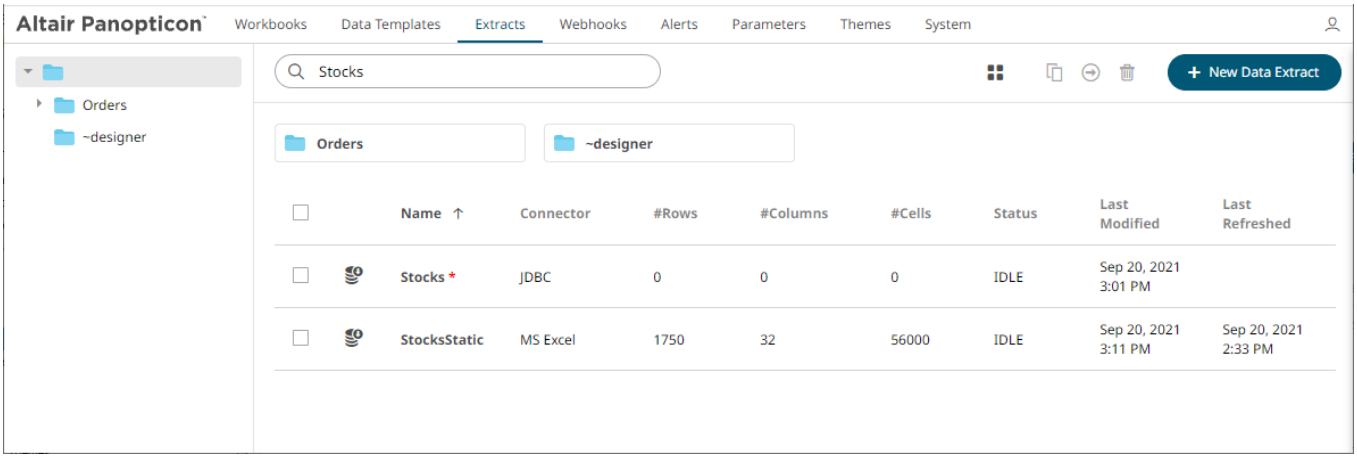
On either display view style, clicking on a data extract title or thumbnail displays the *Extracts* page.

Searching for Data Extracts

On the *Extracts* page, to search for a particular data extract, enter it in the *Search Data Extracts* box.



You can also enter one of more characters into the *Search Data Extracts* box and the suggested list of data extracts that matched the entries will be displayed.



Click on a data extract to open and display.

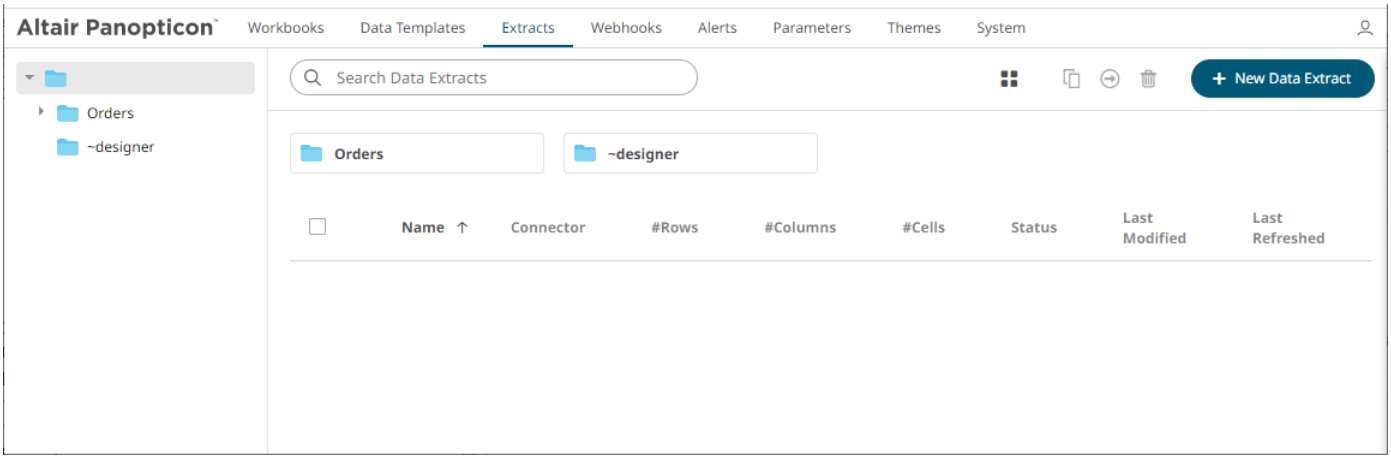
To clear the filter, delete the text entry in the *Search Data Extracts* box.

CREATING DATA EXTRACTS

One of the methods in accessing data is by retrieving only the required results into memory, by querying on demand, pushing aggregation and filtering tasks to underlying big data repositories, or queryable data extracts.

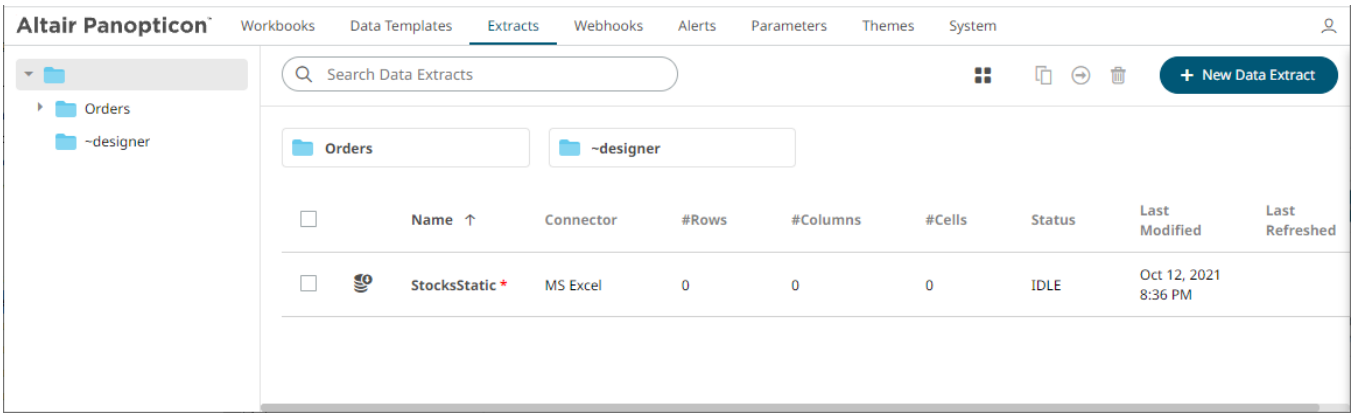
This is commonly known as a ROLAP implementation, where the product is dynamically writing data queries to the underlying data repository and retrieving aggregated and filtered datasets. Given the on-demand nature of this method it is more suitable to exploratory data analysis but requires dynamic query generation.

Data extracts are created in the server administration environment. These server-based data extracts can then be retrieved and used through the *Panopticon Data Extract* connector.



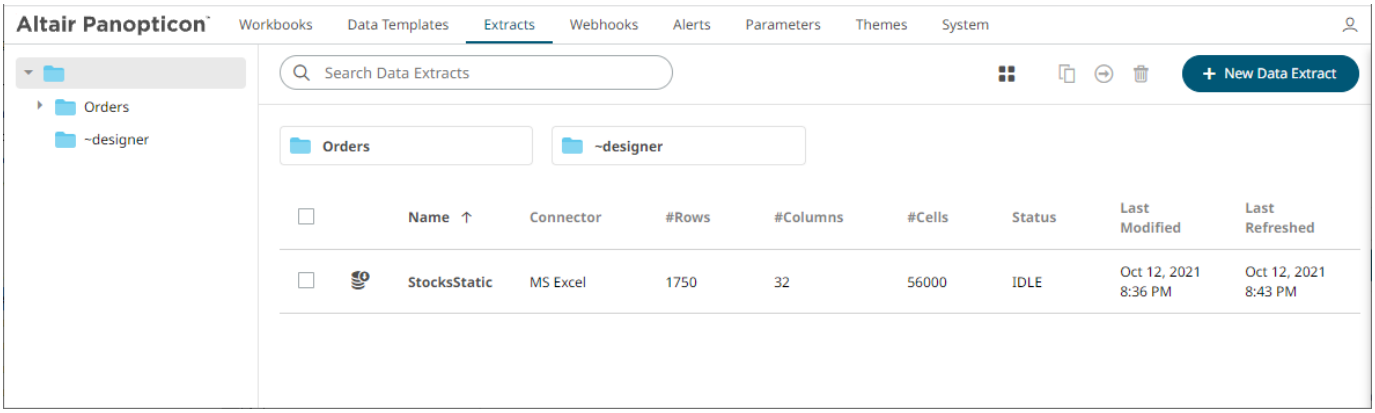
Data Extract Property	Description
Name	Data extract name.
Connector	Data connector used in the extract.
#Rows	Number of rows of the data extract.
#Columns	Number of columns of the data extract.
#Cells	Number of cells of the data extract.
Status	Status of the data extract. Values include **IDLE** or **RUNNING**. When saving a new data extract, it is first run and the status changes to **RUNNING**. Consequently, the **Refresh Data** icon is disabled. When the data extract is complete, the status changes to **IDLE** and the **Refresh Data** icon is enabled.
Last Updated	The Date/Time when the data extract is last completed or updated.

A sample data extract using the MS Excel connector before refreshing the data:



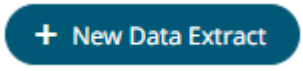
NOTE A * symbol appears beside a data extract that is not yet refreshed.

After refreshing the data:



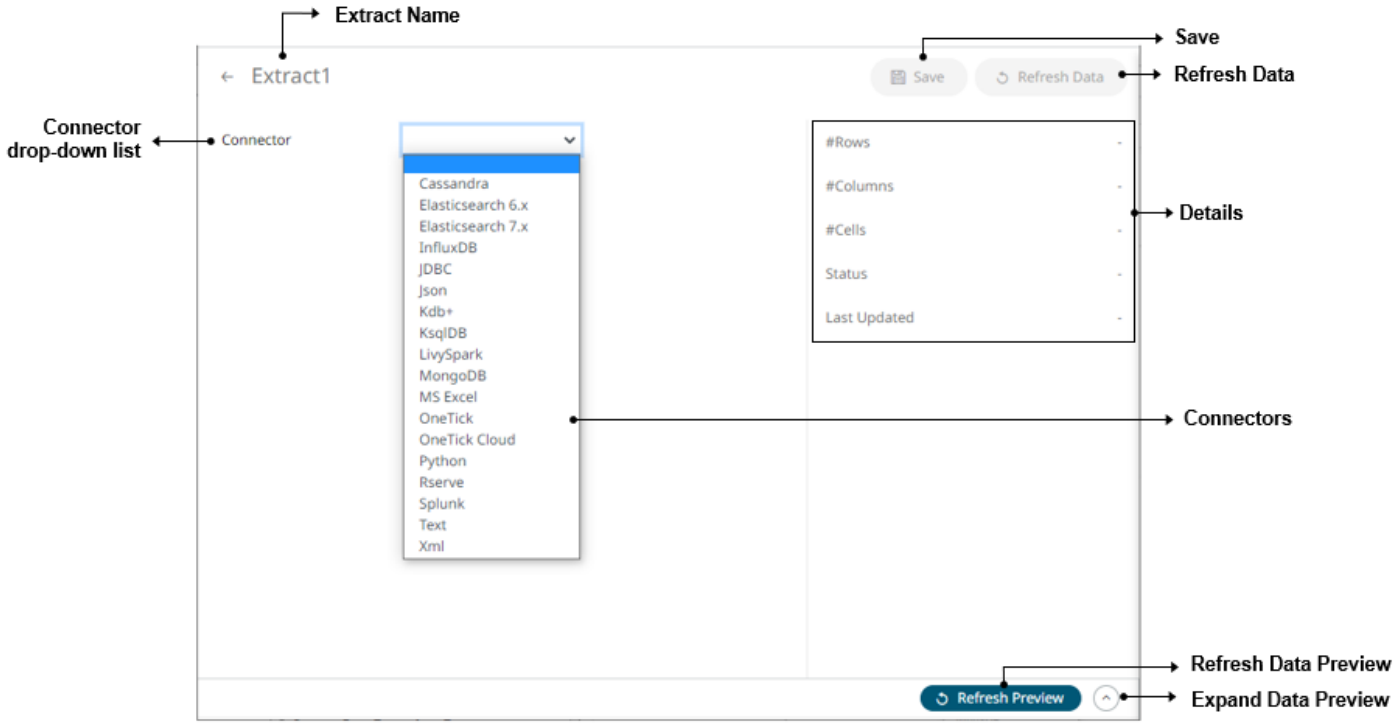
This section discusses the steps and guidelines on how to create data extracts.

Steps:

1. On the **Extracts** tab, click 

.

   The **Extracts** tab displays with the following sections:



Section/Panel	Description
Extract Name	Name of the data extract. Click the ← button to go back to the *Data Extracts* listing page.
Connector drop-down list	Includes the non-streaming connectors to extract data from.
Save	Save the changes made on the **Extracts** tab.
Refresh Data	Refresh the data after modifying and saving changes on the *Data Extract* page. You can also opt to click **Cancel Refresh Data**  .
Details	Display the details of the data extract including the number of rows, columns, cells, status, and the last time it was updated.
Refresh Data Preview	Refresh the data preview.
Expand Data Preview	Expand the *Data Preview* pane.

2. Enter the *Name* of the data extract. This should be unique and should only contain letters (a to Z), numbers (0 to 9), and underscores.
3. Click ✓ or press **Enter** to apply the name.
4. Select any of the following non-streaming connectors:
   - Cassandra
   - Elasticsearch 6.x

- Elasticsearch 7.x
- InfluxDB
- JDBC Database
- JSON
- Kx kdb+
- ksqlDB
- Livy Spark
- MongoDB
- MS Excel
- OneTick
- OneTick Cloud
- Python
- Rserve
- Splunk
- Text
- XML

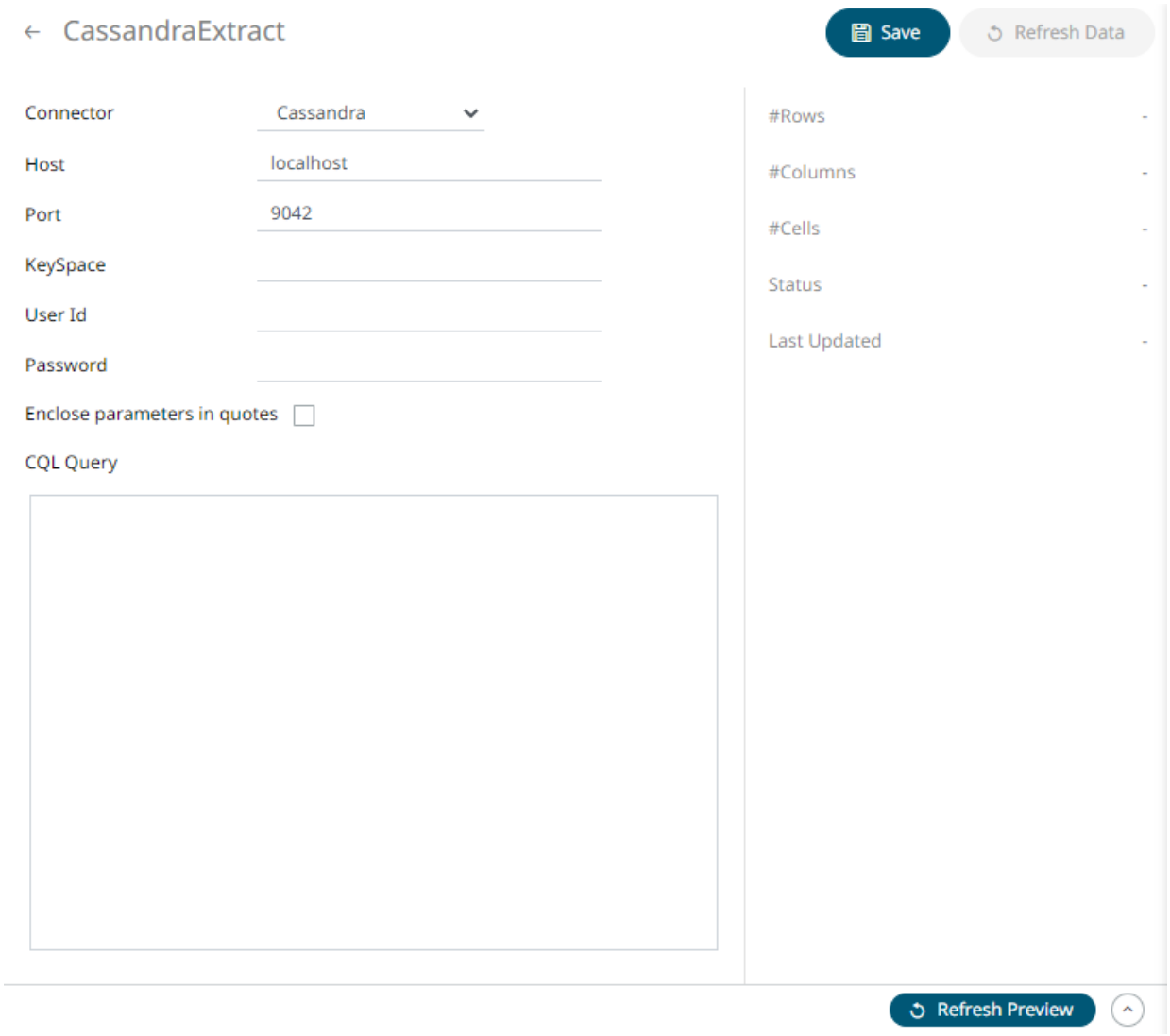
  The tab page changes depending on the selected connector. Refer to the sections below for more information.

Creating Data Extract from Apache Cassandra

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 Extract* page, select **Cassandra** in the *Connector* drop-down list.



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 checking the **Enclose parameters in quotes** box.
4. Enter the *CQL Query*, which can contain parameters in a similar manner to the database connector.
5. Click 

to save and display the details of the data extract.

6. Click  then  to display the data preview.

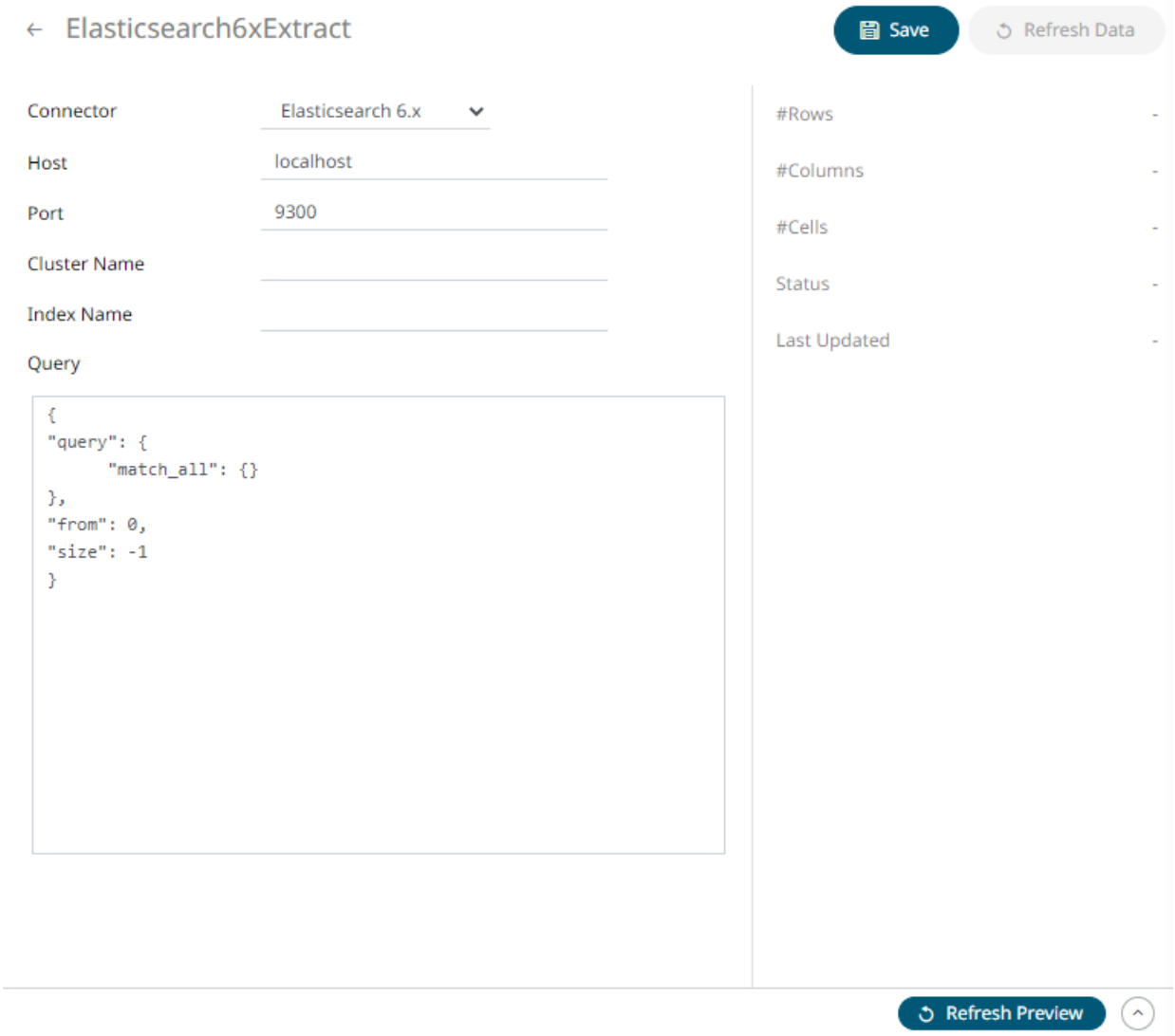
Creating Data Extract from Elasticsearch 6.x

The Elasticsearch 6.x connector allows you to connect and access data from an Elasticsearch cluster using Transport Client.

NOTE	• The Elasticsearch 6.x connector supports Elasticsearch 6.x versions. • Elasticsearch 6.x and 7.x connectors will not work in a single Panopticon Real Time instance due to conflicting Elasticsearch API dependencies.

Steps:

1. On the *New Data Extract* page, select **Elasticsearch 6.x** in the *Connector* drop-down list.



2. Enter the following information:

Property	Description
Host	The hostname of any node in your Elasticsearch cluster, or localhost for a node on your local machine.
Port	The port running the Elasticsearch HTTP service (default is **9300**). If the port you wish to use is different from the default port, change the value to the correct one.
Cluster Name	The cluster name that can be used to discover and auto-join nodes.
Index Name	The Index name in Elasticsearch. This is some type of data organization mechanism that allows partition of data in a certain way.

3. Enter an optional JSON-encoded request body in the *Query* box.
4. Click 

to save and display the details of the data extract.
5. Click  then 

to display the data preview.

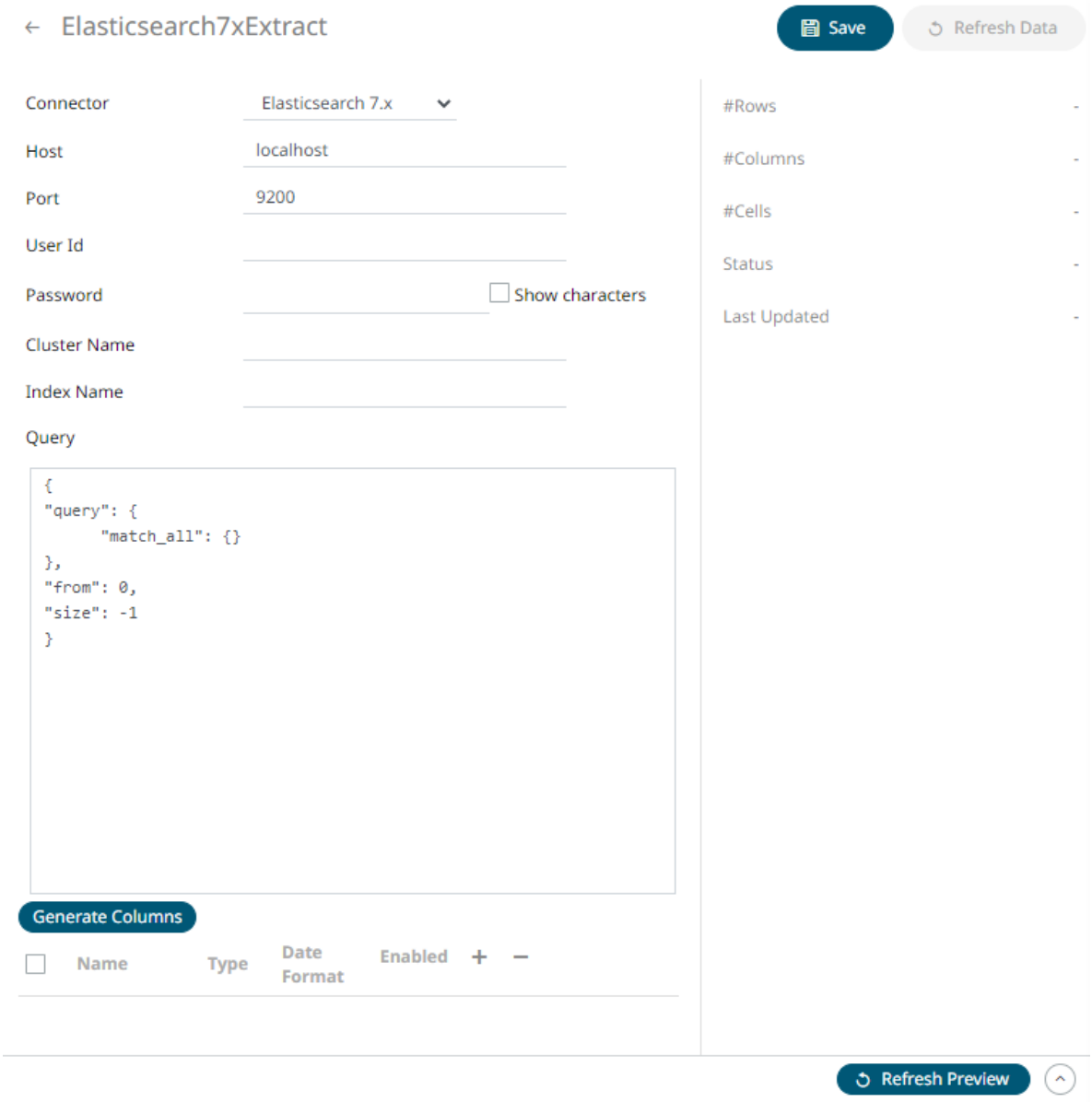
Creating Data Extract from Elasticsearch 7.x

The Elasticsearch 7.x connector allows you to connect and access data from an Elasticsearch cluster using Java High Level REST Client.

NOTE	Similar to Elasticsearch 6.x connector but uses Java High Level REST Client.

Steps:

1. On the *New Data Extract* page, select **Elasticsearch 7.x** in the *Connector* drop-down list.



2. Enter the following information:

Property	Description
Host	The hostname of any node in your Elasticsearch cluster, or localhost for a node on your local machine.
Port	The port running the Elasticsearch HTTP service (default is **9300**). If the port you wish to use is different from the default port, change the value to the correct one.
User Id	The username used to connect to the Elasticsearch 7.x service.
Password	The password used to connect to the Elasticsearch 7.x service. Check the *Show Characters* box to display the entered password characters.
Cluster Name	The cluster name that can be used to discover and auto-join nodes.

Index Name	The Index name in Elasticsearch. This is some type of data organization mechanism that allows partition of data in a certain way.

3. Enter an optional JSON-encoded request body in the *Query* box.
4. Click Generate Columns. The columns populate the *Output Column* section.
5. Click + to add columns and specify their 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 field should be processed.

   To delete a column, check its ☐ or all the column entries, check the topmost ☐, then click −.

6. Click Save to save and display the details of the data extract.
7. Click ^ then Refresh Preview to display the data preview.

Elasticsearch Connectors Dependency Installation

Dependencies for each supported Elasticsearch version are included in Panopticon Real Time zip as individual zip archive files:

- Elastic_6X_Dependencies.zip
- Elastic_7X_Dependencies.zip.

Steps:

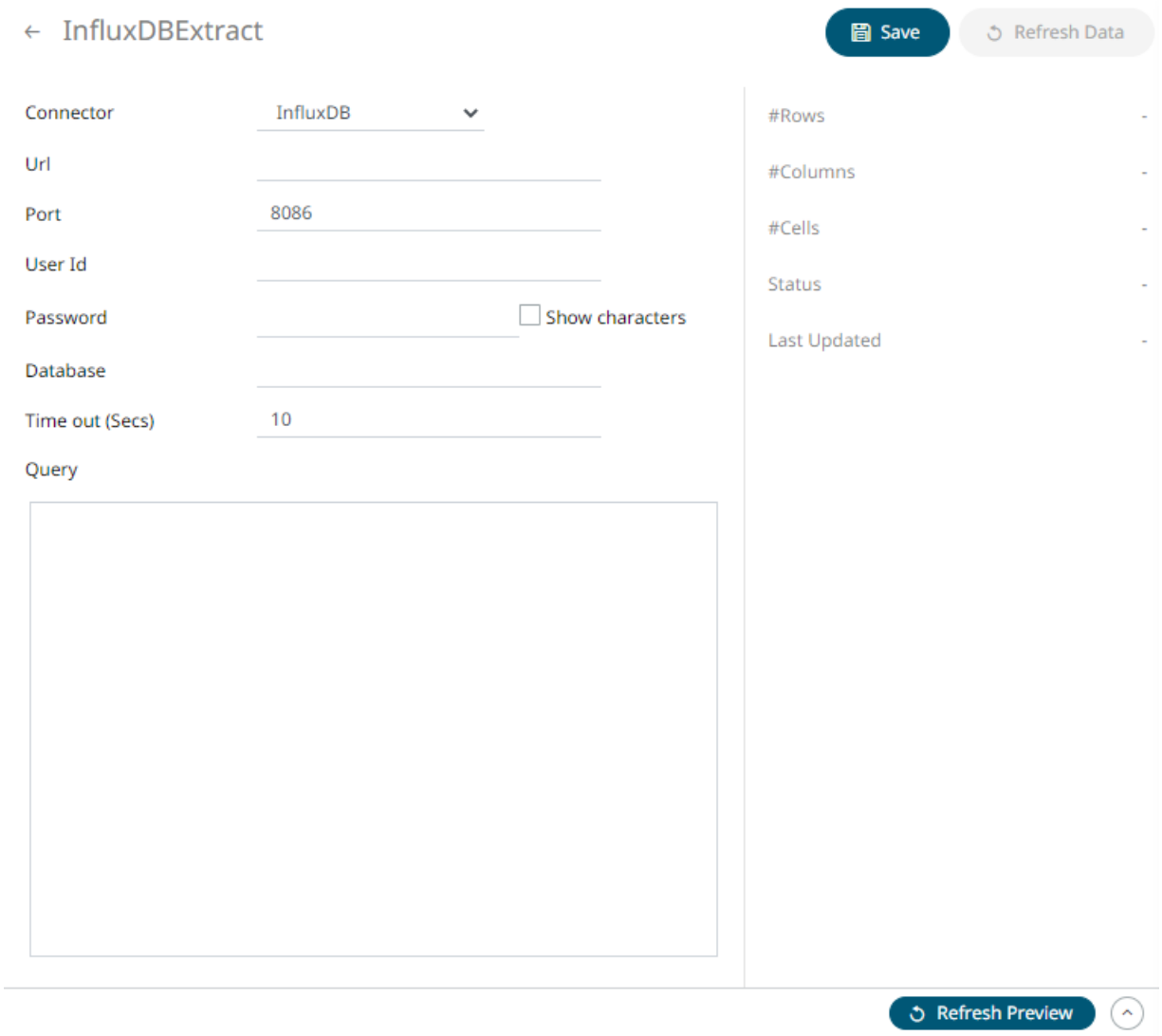
1. Select the target Elasticsearch version and unzip the contents of the appropriate dependency zip into the `tomcat/webapps/panopticon/WEB-INF/lib` folder to enable connectivity for a specific server instance.
2. Restart Tomcat.

Creating Data Extract from InfluxDB

The InfluxDB connector allows for the retrieval of a JSON data set from the InfluxDB. The database communicates over HTTP(S) where you can define a query on the URL to return the desired data.

Steps:

1. On the *New Data Extract* page, select **InfluxDB** in the *Connector* drop-down list.



2. Enter the following information:

Property	Description
URL	InfluxDB host address.
Port	InfluxDB host port. Default is **8086**.
User Id	The user Id that will be used to connect to the InfluxDB service.
Password	The password to connect to the InfluxDB service. Check the **Show Characters** 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.

4. Click  to save and display the details of the data extract.

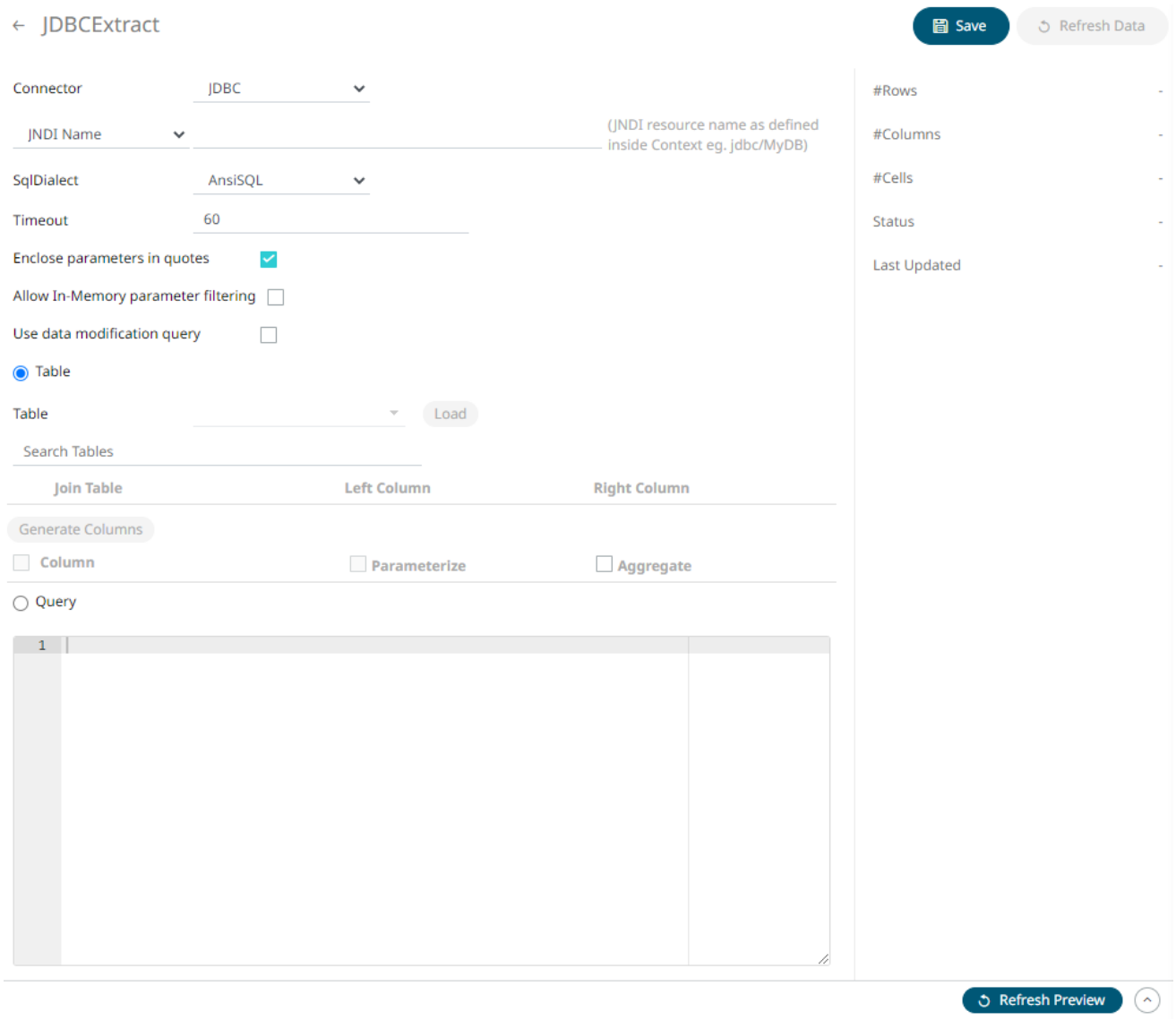
5. Click  then  to display the data preview.

Creating Data Extract from JDBC Database

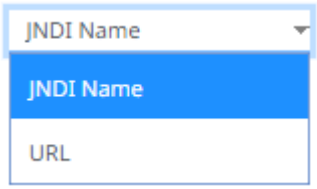
IMPORTANT	For DolphinDB, the query builder is not supported, only the query mode.

Steps:

1. On the *New Data Extract* page, select **JDBC** in the *Connector* drop-down list.



2. You can either select:

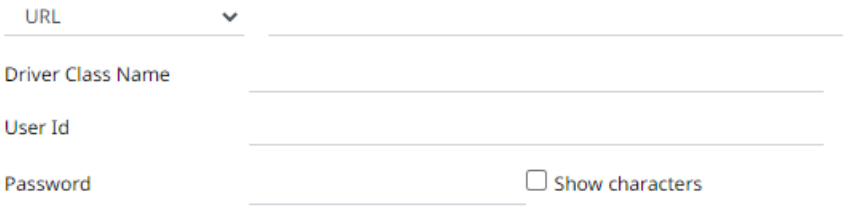


- JNDI Name

Enter the *JNDI resource name* to be used.

NOTE	The JNDI resource name needs to be on the form: `jdbc/[resourcename]`

- URL



Enter the *URL* specific to the database's JDBC driver, the *Driver Class Name* specific to the driver, and the *Username* and *Password*.

Check the **Show Characters** 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, Access/Excel, MySQL, Oracle, SQL Server, Sybase IQ/ASA, Sybase ASE, Netezza, Vertica, SQLite, HadoopHive, KxQ, DB2, PostgreSQL, Impala, Redshift, Informix, Teradata, dBase, SparkSQL.

   Default is **AnsiSQL.**

4. Enter the *Timeout*. Default is **60**.
5. Check any of the following options when building the query:
   - Enclose parameters in quotes

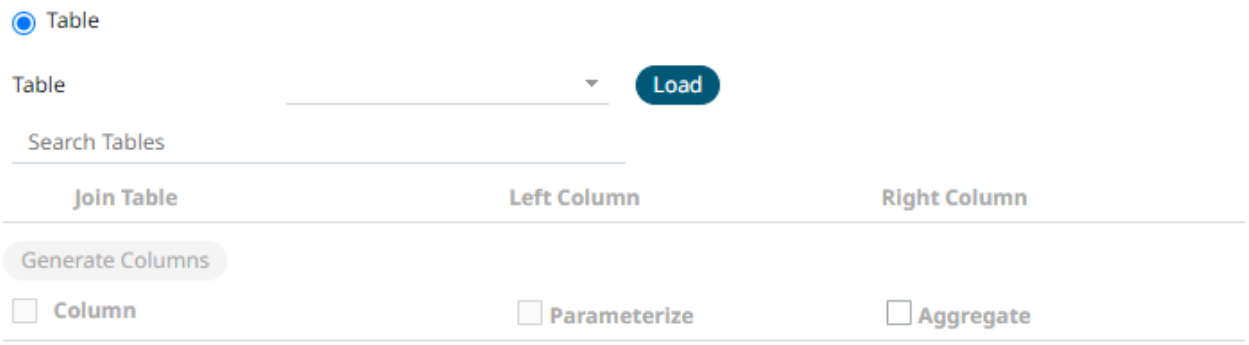
     By default, this option is checked, as the common use case for parameters is a filter `WHERE` clause.

   - 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.

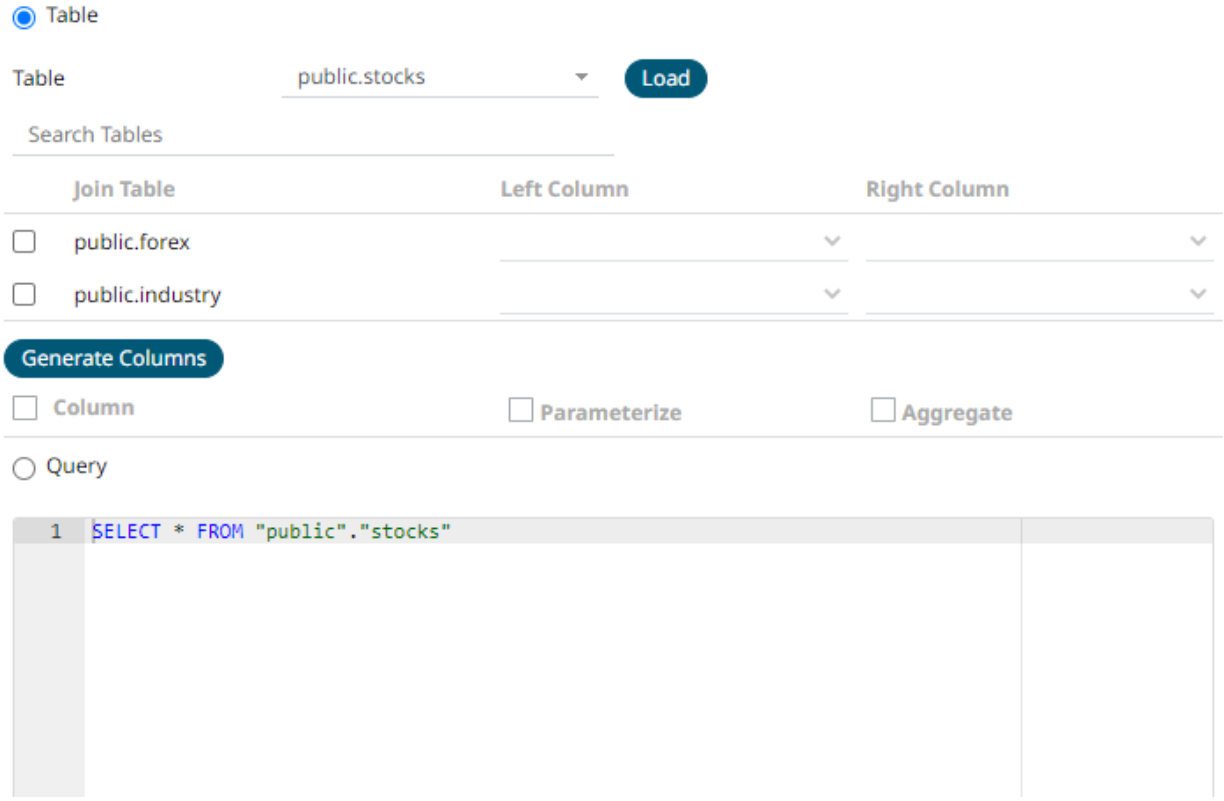
   - Use data modification query

     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:



7. On the *Table* field, click Load to populate the drop-down list with tables. Select a table.

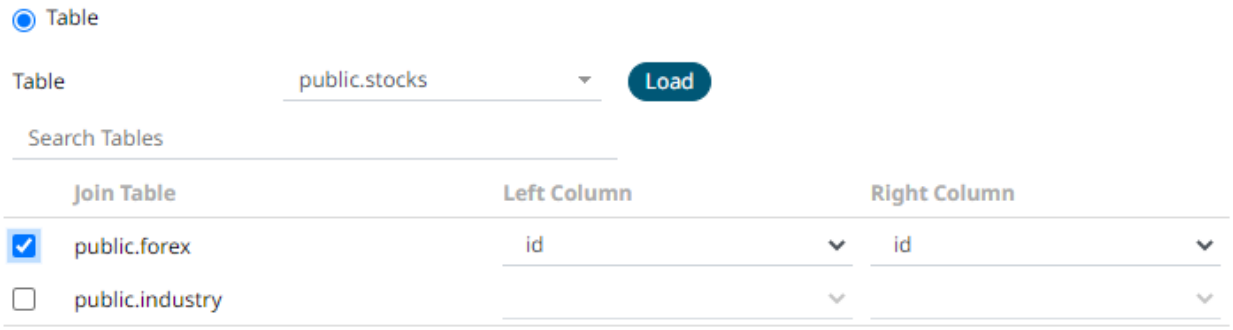
   The list of tables that you can join is displayed. Also, the SQL query is generated and displayed in the *Query* text box.



   Use *Search Tables* to filter the list.

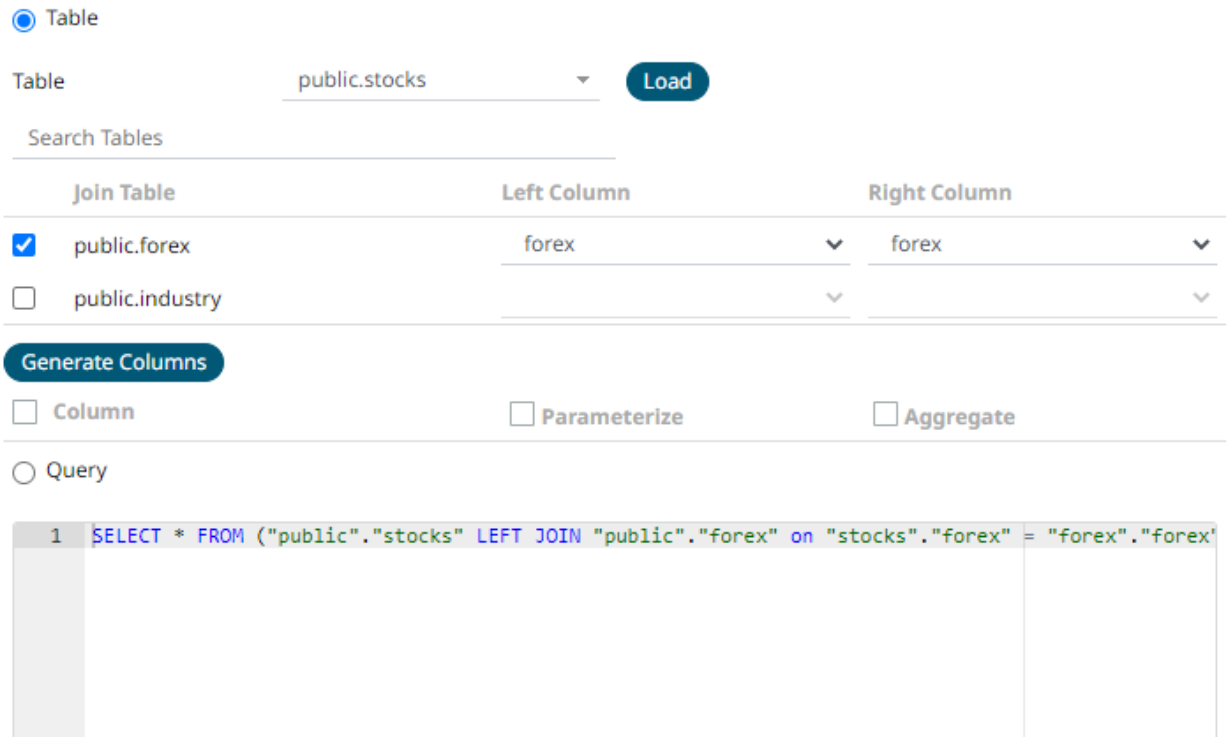
8. Perform a join by checking one or more tables in the list.

   The *Left Column* and *Right Column* fields are automatically filled out with the common fields.



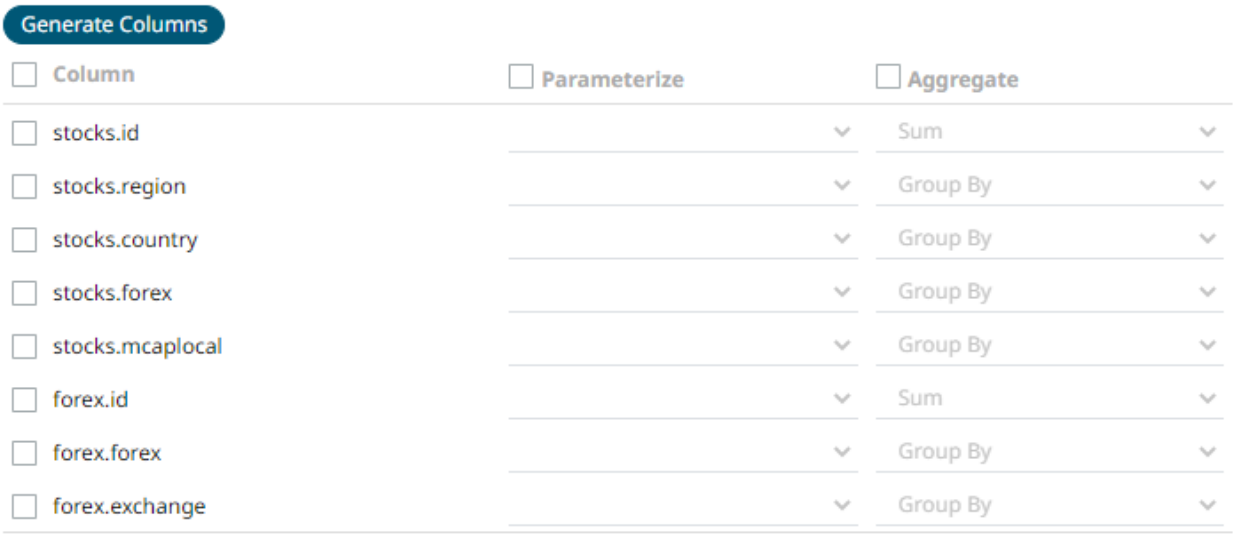
You can also opt to select other common fields.

The SQL query is generated and displayed in the *Query* text box.



9. Click 

. The columns populate the *Output Column* section.



10. Individual columns can be added by checking the corresponding *Column* box in the *Output Column* listing. To select all of the columns, check the topmost 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. Check the **Parameterize** 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. Click the **Query** radio button to enable the text box and modify the SQL-like query language.

14. Click 

to save and display the details of the data extract.

15. Click  then 

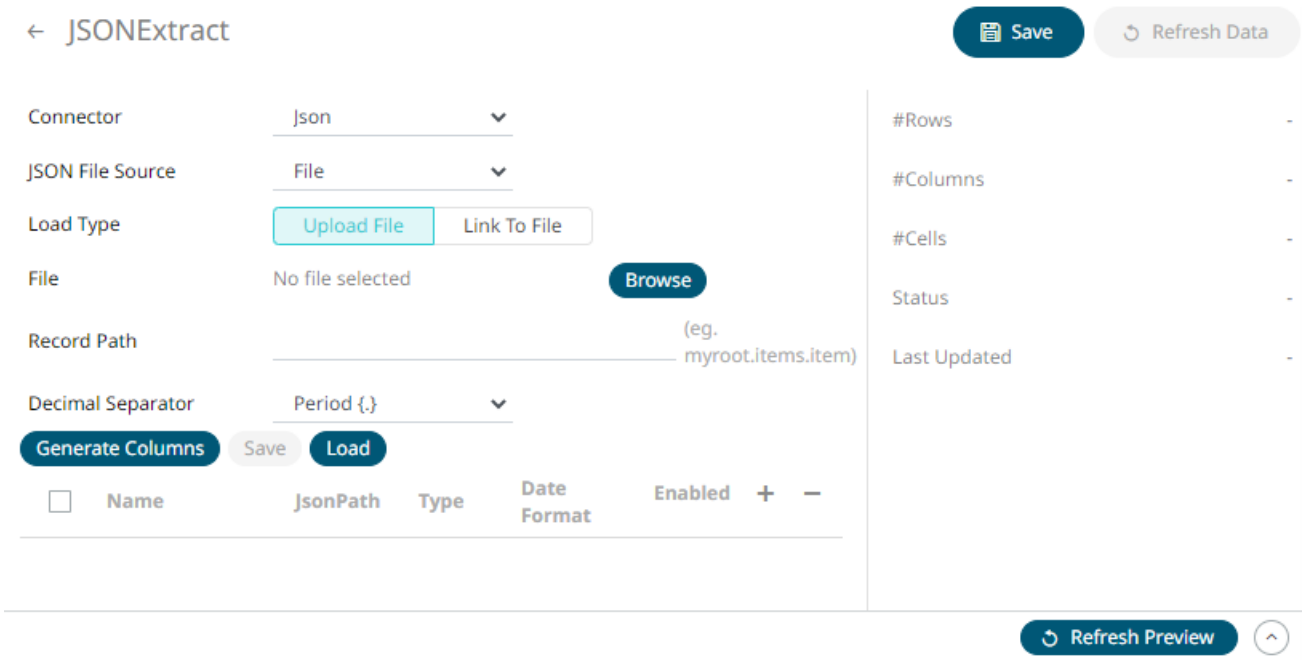
to display the data preview.

Creating Data Extract from JSON

The JSON connector allows the retrieval and processing of JSON files, either from a disk, a Text, or from a defined URL.

Steps:

1. On the *New Data Extract* page, select **JSON** in the *Connector* drop-down list.



2. Select the JSON *File Source*.
3. Enter the *Record Path* (e.g., `myroot.items.item`).
4. Select either the dot (.) or comma (,) as the *Decimal Separator*.
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. 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.

To delete a column, check its ☐ or all the column entries, check the topmost ☐, then click −.

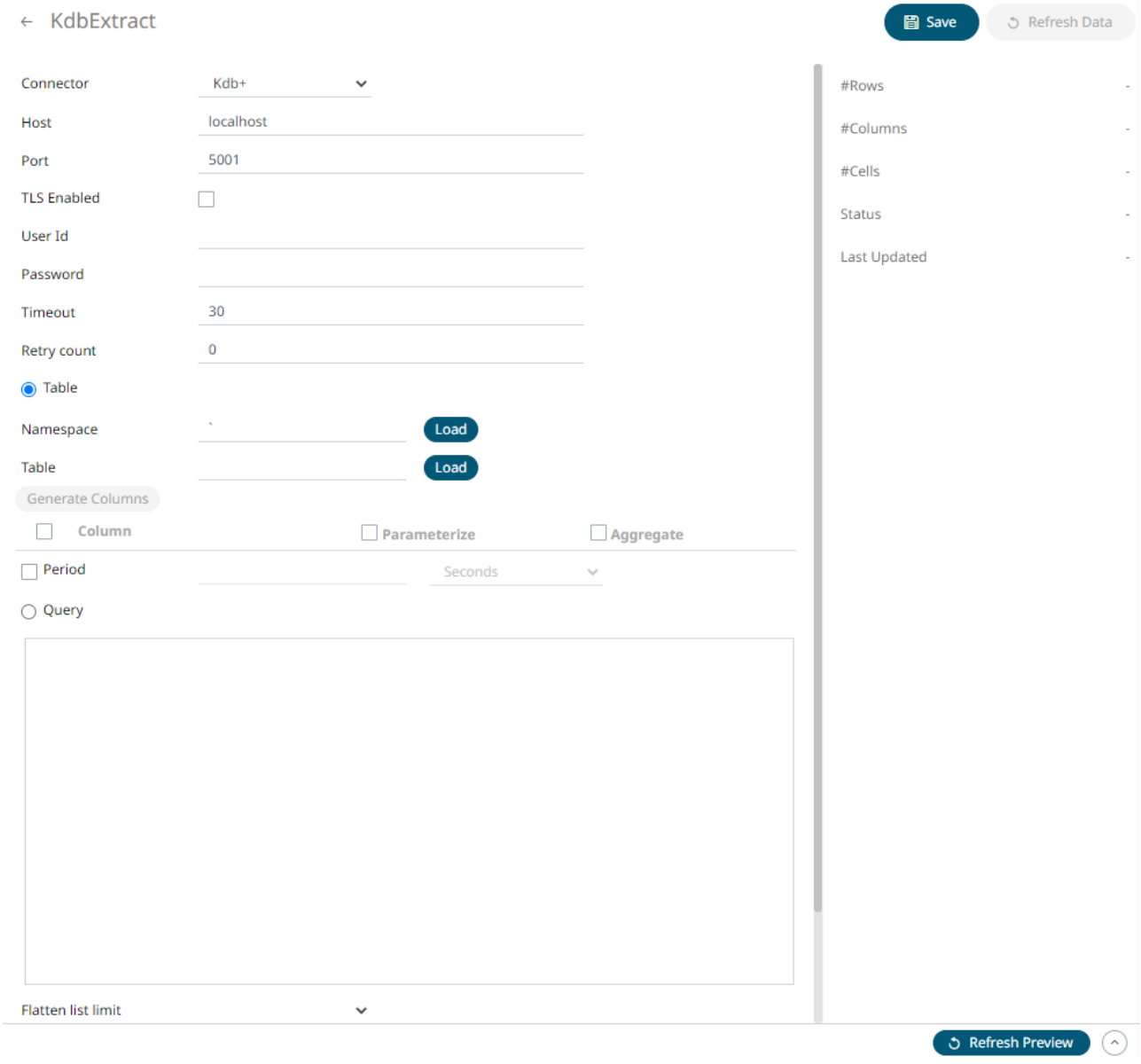
8. Click Save to save and display the details of the data extract.
9. Click ^ then Refresh Preview to display the data preview.

Creating Data Extract from Kx kdb+

The Kx kdb+ connector allows connection to the Kx kdb+ databases on a polled basis.

Steps:

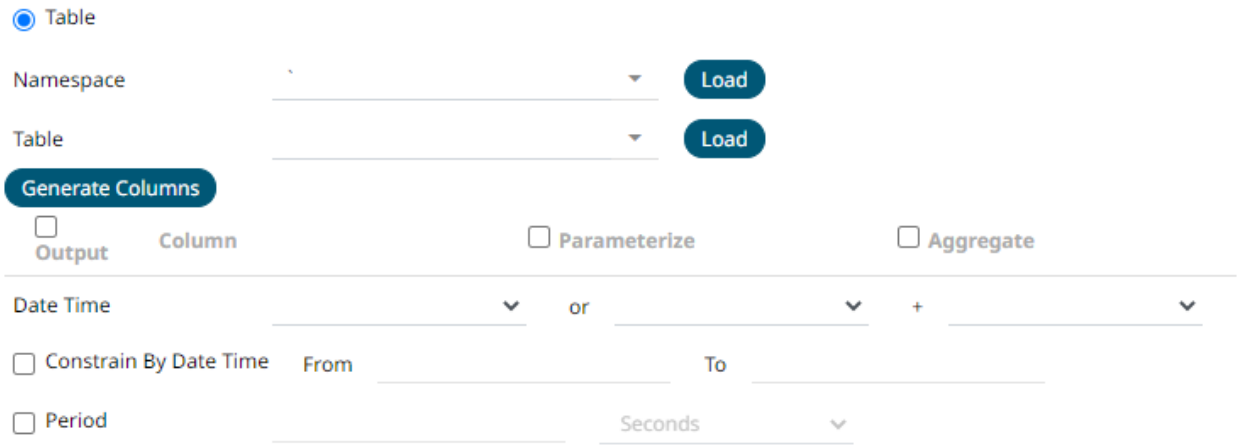
1. On the *New Data Extract* page, select **Kdb+** in the *Connector* drop-down list.



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. 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.

You can either:

- click  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  to populate the drop-down list with tables and views. Select a table or view.
5. Click  . The columns of the selected table or view populates the *Output Column* section.
6. Individual columns can be added by checking the corresponding *Column* 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:

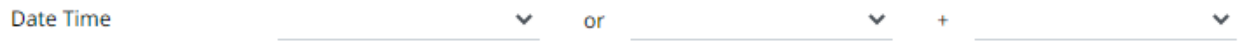
   - Text Columns: Group By
   - Date Columns: Count, Min, Max, Group By
   - Numeric Columns: Sum, Count, Min, Max, Group By

   In addition, the qSQL query is generated and displayed on the *Query* text box.

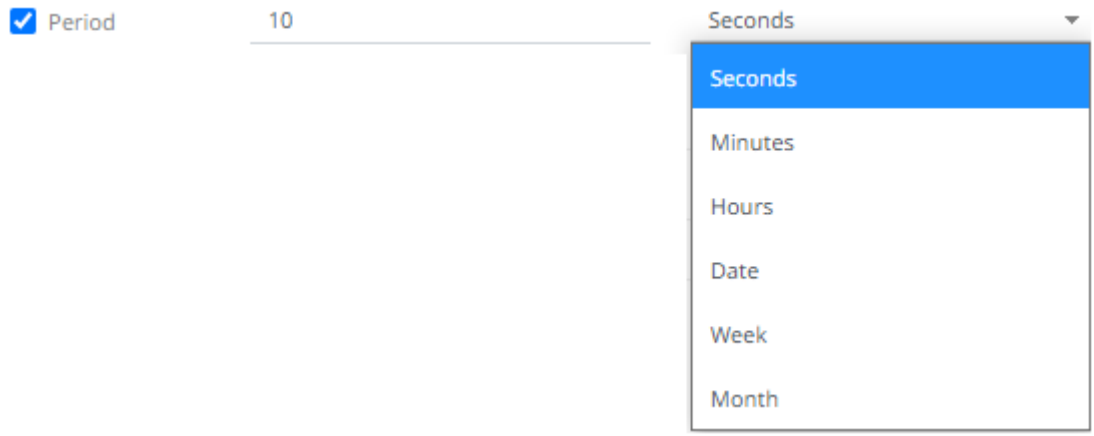
8. Check the **Parameterize** checkbox and match the parameter to the appropriate column. By default, they will be matched by name.

   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. Check the **Constrain by Date Time** 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). Check the **Period** 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.

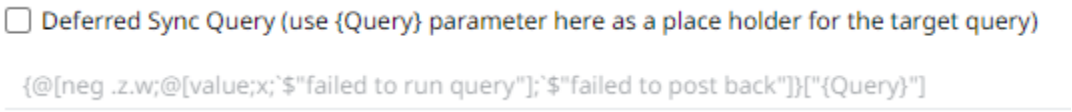
14. Check **Pass to function** box to activate a connection to a server using a proxy. Enter the value.
15. You may also define a Deferred Sync Query.
16. Click 

to save and display the details of the data extract.
17. Click  then 

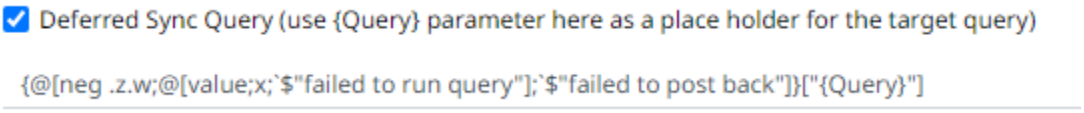
to display the data preview.

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.



Checking the *Deferred Sync Query* box would enable the query box:



The {Query} parameter is used as a place holder for the target query that is defined in the *Query* builder.

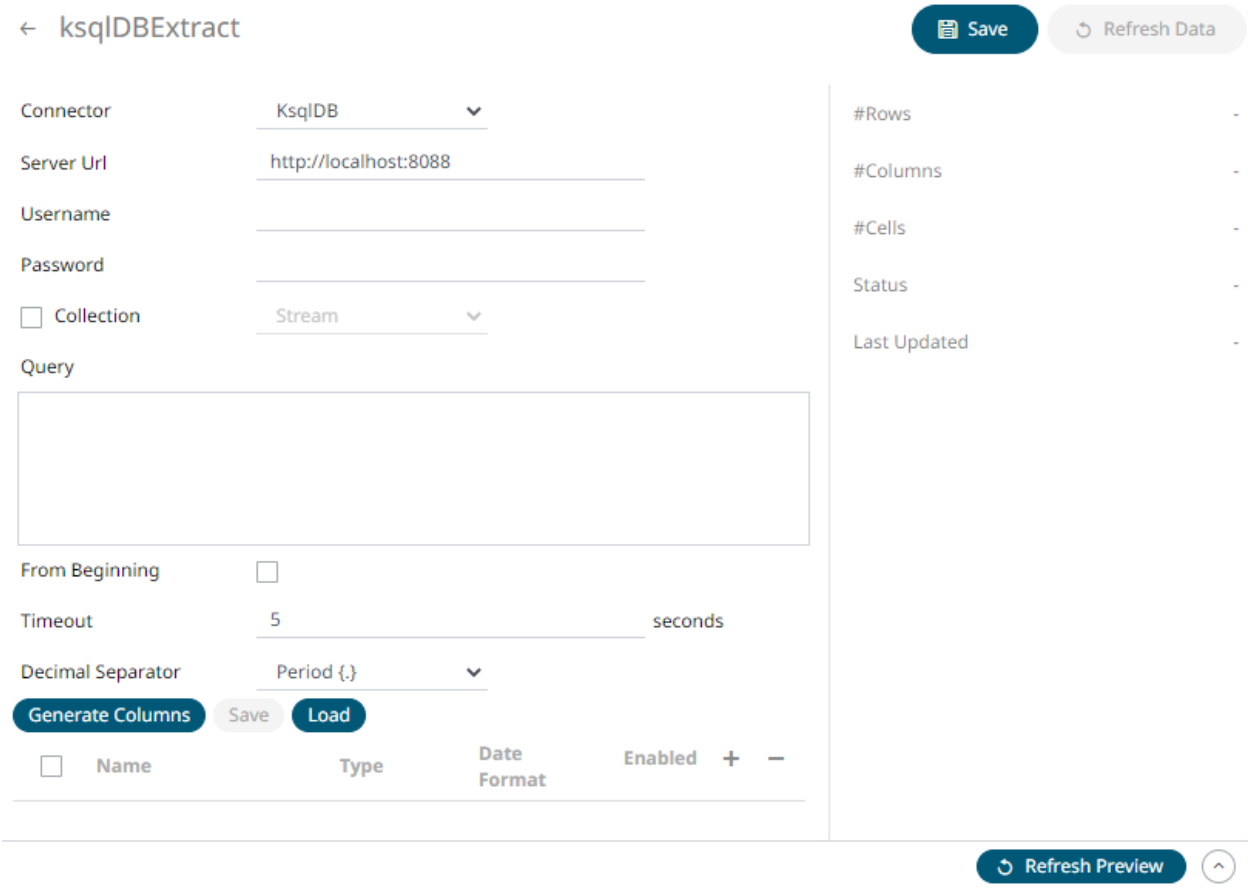
Creating Data Extract from ksqlDB

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. On the *New Data Extract* page, select **ksqlDB** in the *Connector* drop-down list



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. Check the **Collection** 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. Check the *From Beginning* box to subscribe from the beginning to the latest messages.

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 to save and display the details of the data extract.
13. Click ^ then Refresh Preview to display the data preview.

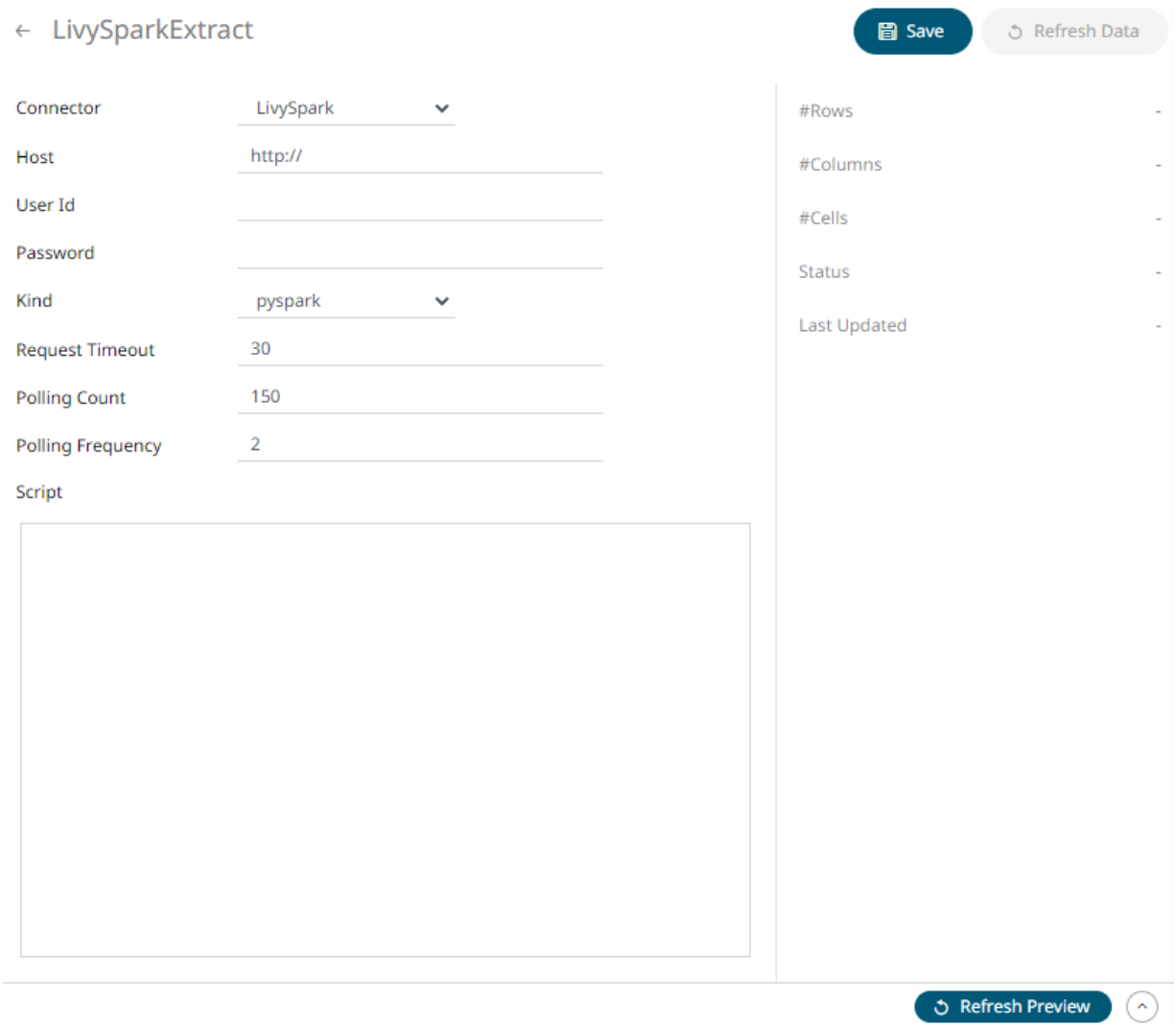
Creating Data Extract from Livy Spark

Livy is an open source REST interface for interacting with Apache Stark. It supports executing snippets of code or programs such as Scala, Python, Java, and R in a Spark context that runs locally or in Apache Hadoop YARN.

The Livy Spark connector allows you to run these codes and fetch the data in Panopticon Real Time.

Steps:

1. On the *New Data Extract* page, select **LivySpark** in the *Connector* drop-down list.



2. Enter or select the following properties:

Property	Description
Host	Livy Spark host address.
User Id	User Id that will be used to connect to Livy Spark.
Password	Password that will be used to connect to Livy Spark.
Kind	Currently, the supported kind of connection to be used is **pyspark (**Interactive Python Spark session).
Request Timeout	Length of time to wait for the server response. Default is **30**.
Polling Count	The number of polling done to the Livy Spark server to check if the status of the app is successful. Default limit is **150**.
Polling Frequency (in seconds)	Frequency of the polling. Default is **2**.
Script	The script to use.

3. Click  to save and display the details of the data extract.

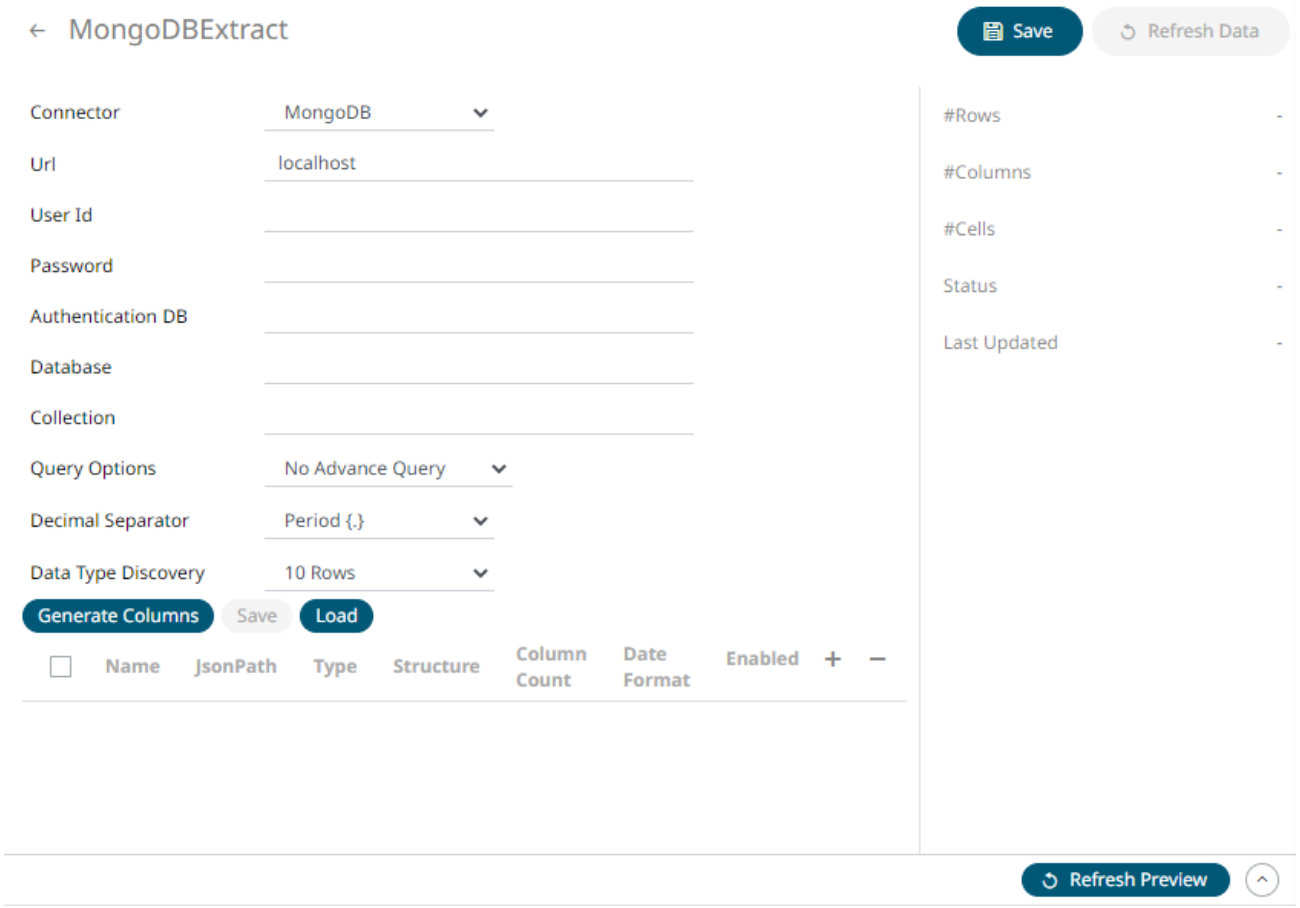
4. Click  then  to display the data preview.

Creating Data Extract from MongoDB

The MongoDB connector is an interface used to import MongoDB's schema-less BSON documents into a table schema that Panopticon Real Time can interpret and analyze. It uses many BSON structure types and MongoDB query features.

Steps:

1. On the *New Data Extract* page, select **MongoDB** in the *Connector* drop-down list.



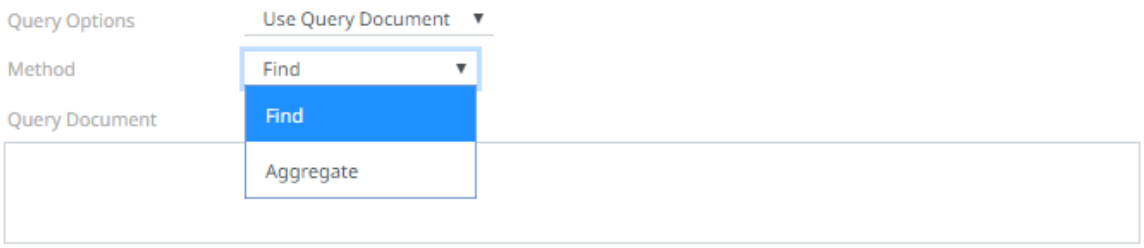
2. Enter the following properties:

Property	Description
URL	Enter either: • **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. You can also opt to make the Query Document feature of MongoDB to be available in Panopticon Real Time. Select **Use Query Document** in the *Query Options* drop-down list.

   This also displays the *Method* drop-down. Select either **Find** (Default) or **Aggregate**.



   When **Aggregate** is selected, you can add all the columns generated by aggregation in the schema.

   In addition, the MongoDB command line interface displays query operations with a JSON style syntax.

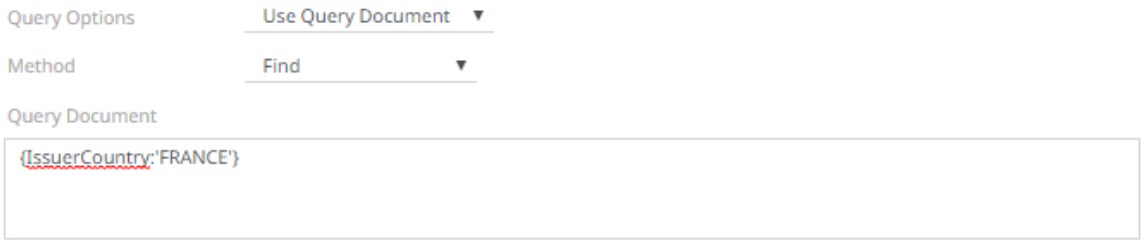
   Enter your desired JSON query document. Refer to http://docs.mongodb.org/manual/tutorial/query-documents/ for more information on the Query Documents feature on MongoDB.

   For example:

   Queries from the documentation look like this: `db.inventory.find ( {type: "snacks"} )`. The database and collection are already defined in the UI and the *Find* operation is handled in the code. The user only needs to enter the query document:

   { type : “snacks” }

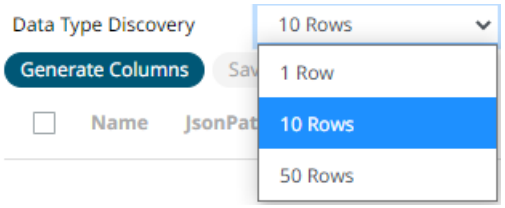
   This query must include surrounding curly braces as well as matching internal braces should the user decide to make a more advanced query.



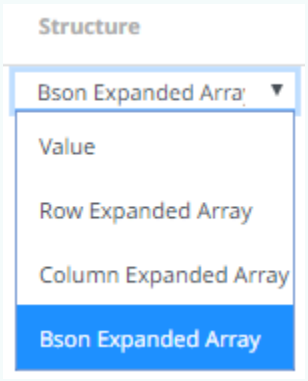
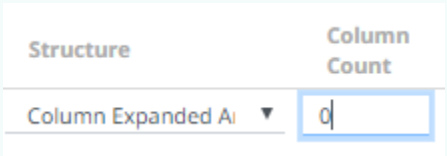
4. Instead of using **Use Query Document**, select the **Parameterize** query option.

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 dot (.) or comma (,) as the *Decimal Separator*.
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.



7. You can also opt to load or save a copy of the column definition.
8. 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 sections. **Value** is the default structure and will always display data regardless of actual structure. 
Column Count	Enabled when **Column-Expanded Array** structure is selected.  Enter the number of columns for the plugin to generate as columns for that array.
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`
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 Save to save and display the details of the data extract.
10. Click ^ then Refresh Preview to display the data preview.

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.

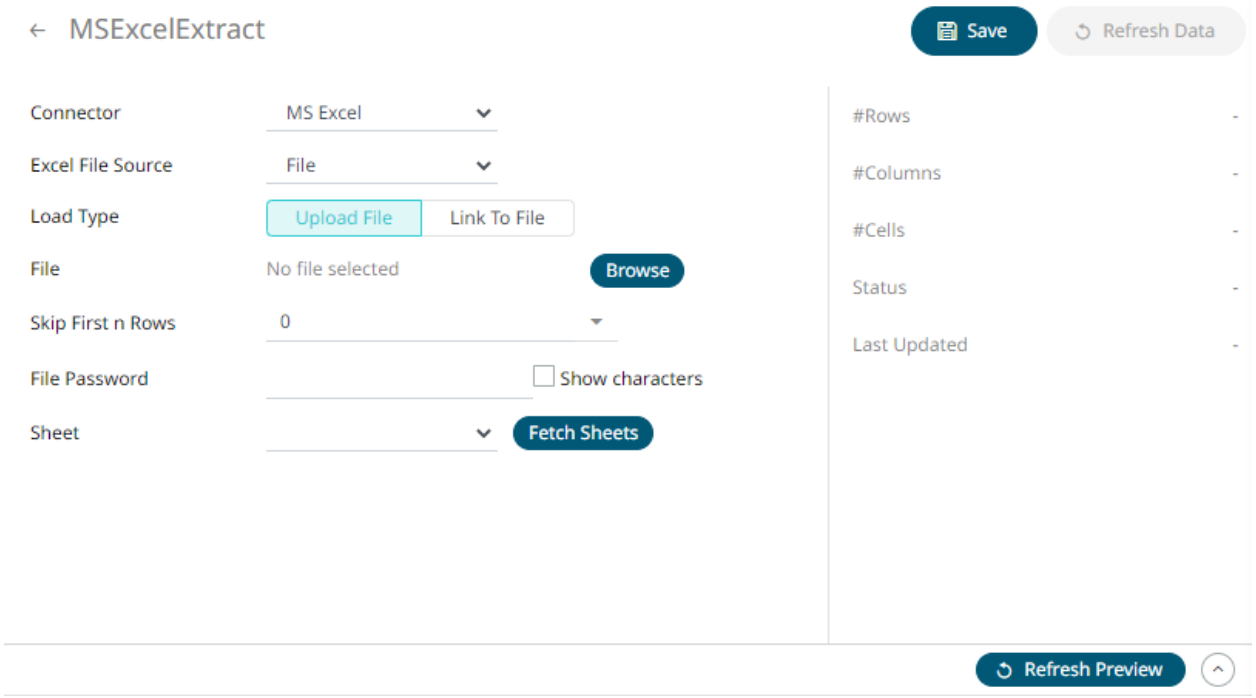
Creating Data Extract from MS Excel

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.

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. A workaround could be to set up a Data Extract with scheduled refresh for each of the datasets in the Excel file, and then let the data tables in your workbook load the data from the Data Extracts.

Steps:

1. On the *New Data Extract* page, select **MS Excel** in the *Connector* drop-down list.



2. Select the MS Excel *File Source*.
3. Select the number of rows that will be skipped in the Excel file from the *Skip First n Rows* drop-down list.
4. If the MS Excel file is password-protected, enter the *File Password*.

   Check the **Show Characters** box to display the entered password characters.

NOTE	The password is case-sensitive.

   Otherwise, proceed to step 5.

5. Click 

. This will populate the *Sheet* drop-down list box.
6. Select the required sheet.
7. Click 

to save and display the details of the data extract.
8. Click  then 

to display the data preview.

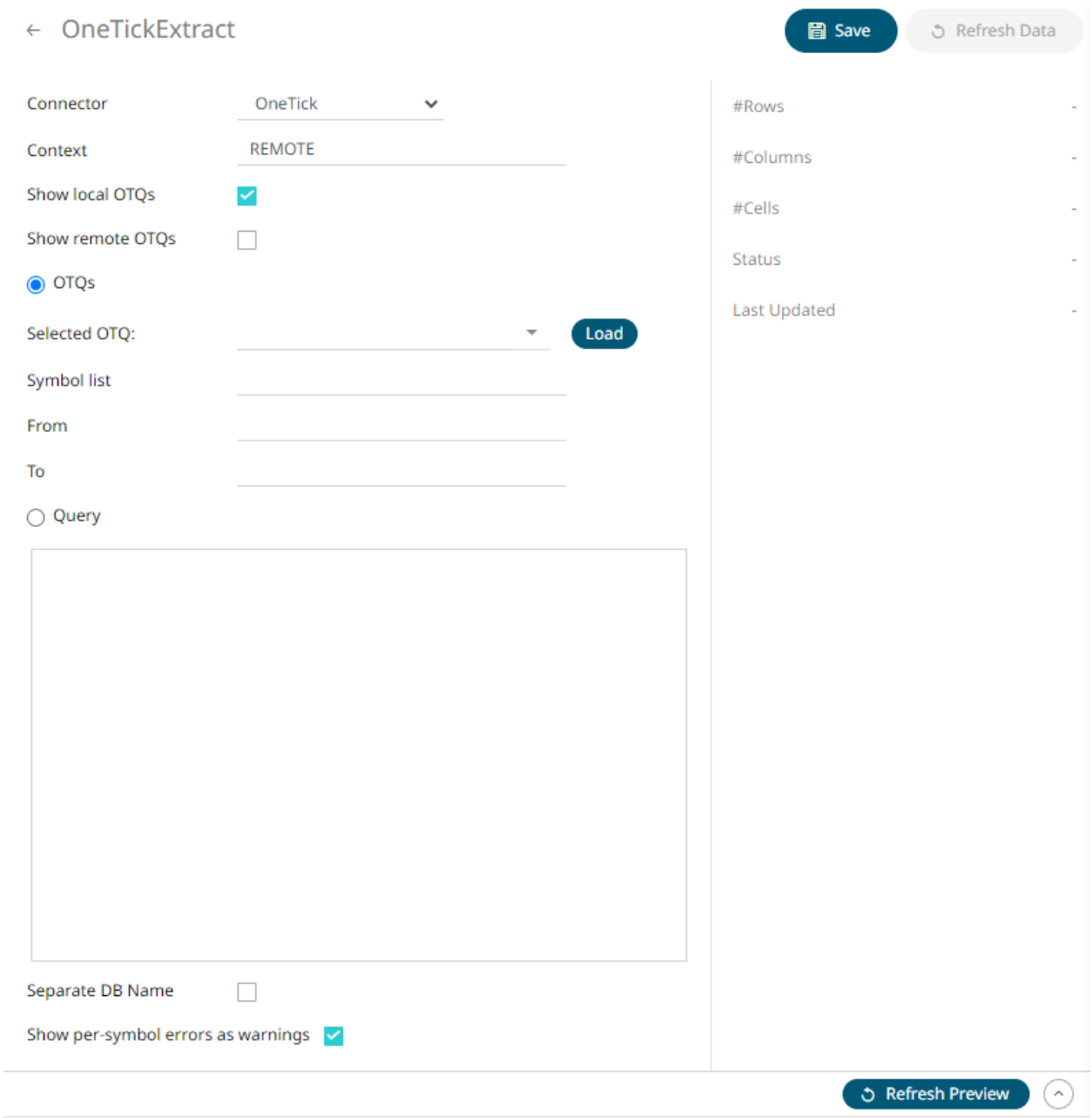
Creating Data Extract from OneTick

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:

1. On the *New Data Extract* page, select **OneTick** in the *Connector* drop-down list.



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 

to save and display the details of the data extract.
8. Click  then 

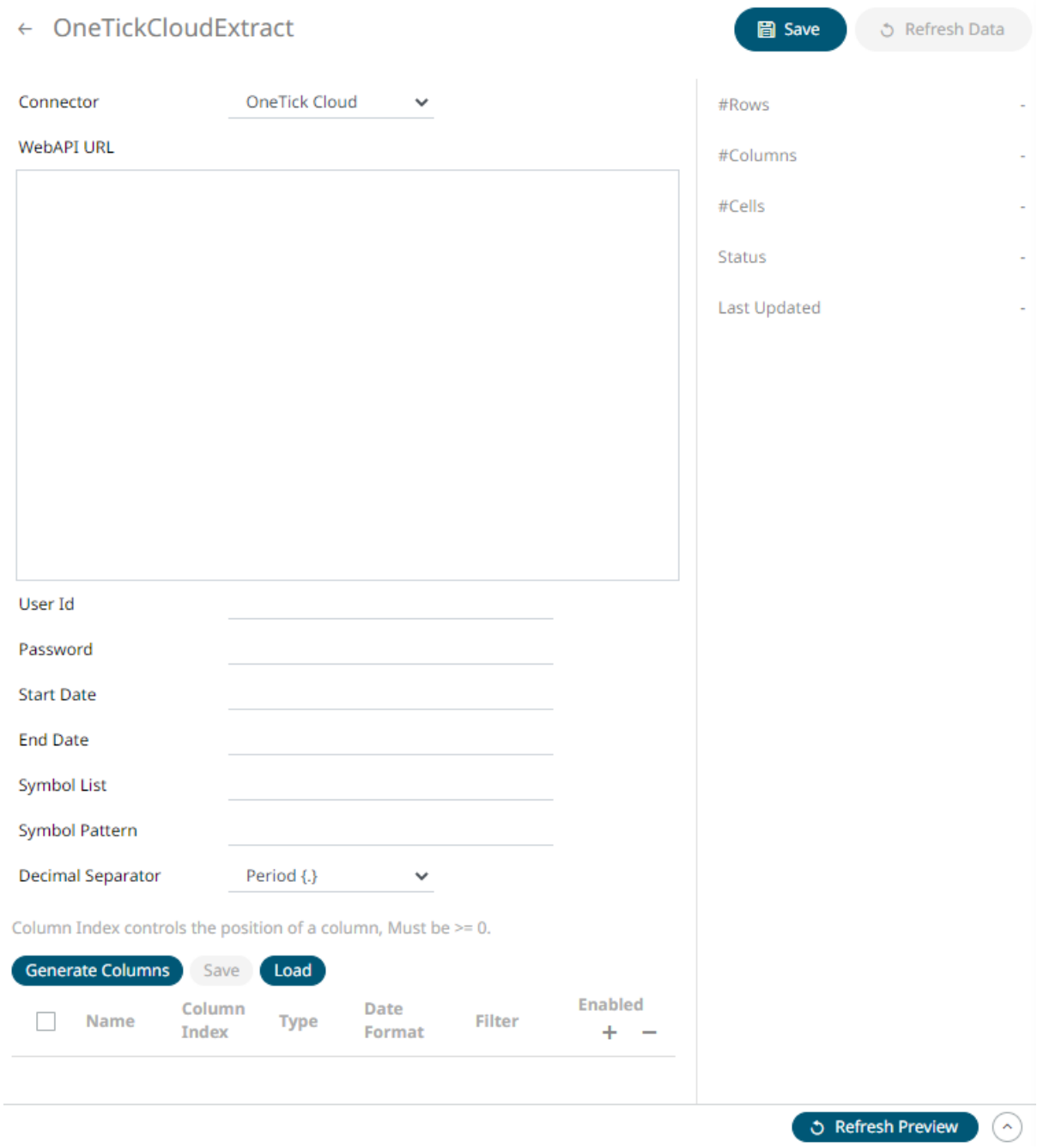
to display the data preview.

Creating Data Extract from OneTick Cloud

The OneTick Cloud connector allows access to historic market data with no software dependencies by using the OneTick Cloud and their web API.

Steps:

1. On the *New Data Extract* page, select **OneTick Cloud** in the *Connector* drop-down list.



2. Enter the OneTick Cloud WebAPI URL into the *WebAPI URL* box with the following form:

```
http://<host>/omdwebapi/rest/?params={"context":"DEFAULT","query_type":"otq",
"otq":"1/12/otq/71b50459-8431-48dc-829f
"s":"20150305130802",
"e":"20150305140805",
"timezone":"America/New_York", "response":"csv",
"compression":"gzip"}
```

Where:

- s, e, timezone – the start and end time of the query YYYYMMDDhhmmss form. The timezone used to interpret this value is taken from the timezone parameter.
- response – the supported response format is csv.
- compression – if available, this option enables gzip compression of the results stream. Large data should always be pulled with compression on.

3. Enter the *User Id* (email) and *Password* to execute the query and retrieve the data. Note that the *User Id* is case sensitive.
4. Enter the time window *Start Date* and *End Date*.
5. Enter the *Symbol List.* This value filters the query output with matching symbols.

   To make it work, ensure to include `Symbol` in the *Query URL*. Consequently, the data will be filtered out for the input (Symbols) provided in the *Symbol List* field.
6. Enter the *Symbol Pattern*. This value filters the query output with the data for all the symbols with matching pattern.

   To make it work, ensure to include `Symbol_Pattern` in the *Query URL*. Consequently, the data will be filtered (for all the Symbols) with matching pattern provided in the *Symbol Pattern* field.
7. Select either the dot (.) or comma (,) as the *Decimal Separator*.
8. 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.
9. You can also opt to load or save a copy of the column definition.
10. 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**.
Filter	Defined parameters that can be used as filter.
Enabled	Determines whether the message should be processed.

To delete a column, check its  or all the column entries, check the topmost  , then click .

11. Click 

to save and display the details of the data extract.
12. Click  then 

to display the data preview.

Creating Data Extract from Python

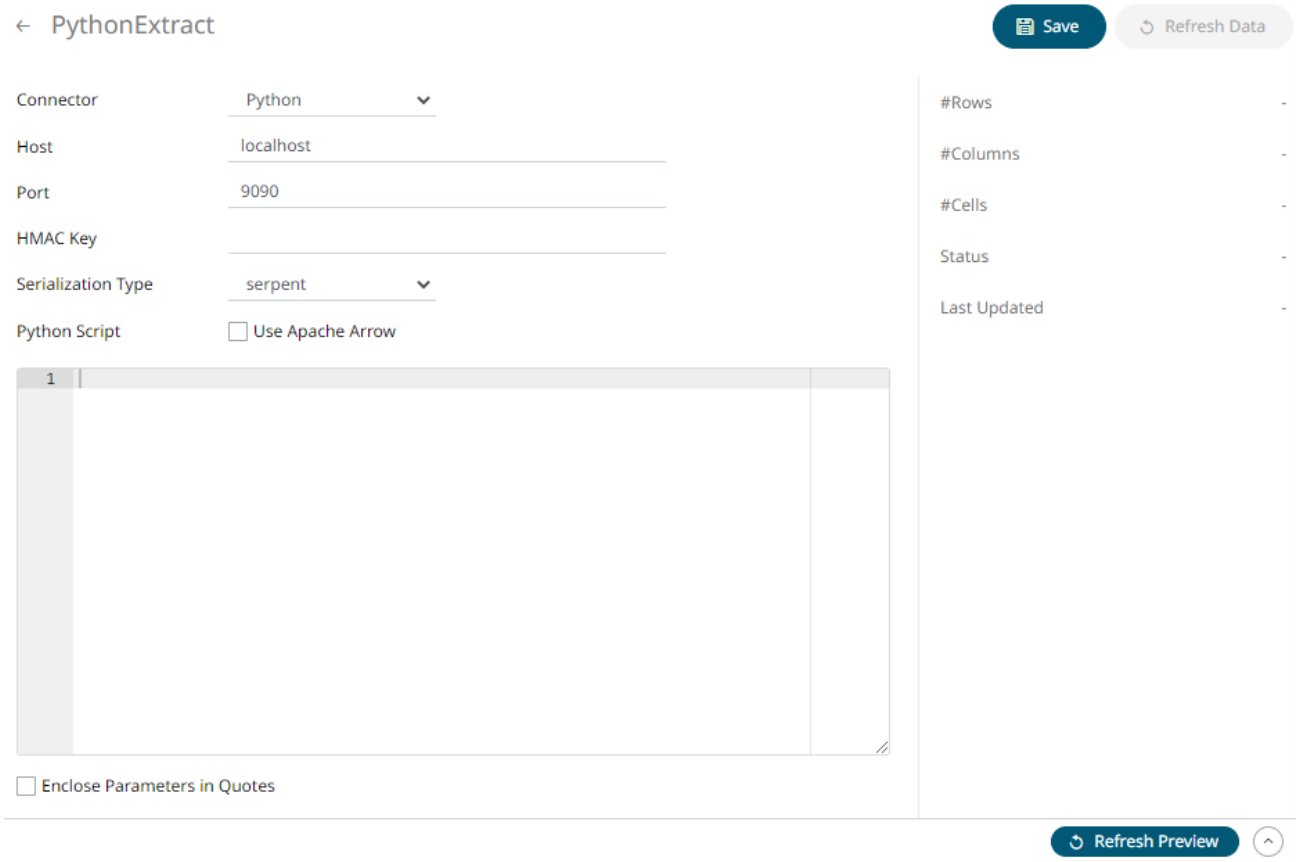
The Python connector allows the retrieval of output data from a Python Pyro (Python Remote Objects) process.

For Python connectivity, Python must be first installed, together with the latest version of Pyro4. In addition, Pyro must be initiated manually or through using the batch file **start_Python_connectivity.bat**.

If the scripts utilize additional modules such as Numpy & Scipy in the shipped example, these also need to be installed into the existing Python installation.

Steps:

1. On the *New Data Extract* page, select **Python** in the *Connector* drop-down list.



2. Enter the following properties:

Property	Description
Host	Python Pyro instance host address.
Port	Python Pyro host port. Default is **9090**.
HMAC Key	Set to **password**.

3. Select the *Serialization Type*: **Serpent** or **Pickle.**
   - Serpent – simple serialization library based on `ast.literal_eval`
   - Pickle – faster serialization but less secure

Modify the `configuration.py` file located in `..\Anaconda3\Lib\site-packages\Pyro4` to specify the serialization to be used.

```
def reset(self, useenvironment=True):
        """
        Set default config items.
        If useenvironment is False, won't read environment variables settings
(useful if you can't trust your env).
        """
        self.HOST = "localhost"  # don't expose us to the outside world by
default
        self.NS_HOST = self.HOST
        self.NS_PORT = 9090  # tcp
        self.NS_BCPORT = 9091  # udp
        self.NS_BCHOST = None
        self.NATHOST = None
        self.NATPORT = 0
        self.COMPRESSION = False
        self.SERVERTYPE = "thread"
        self.COMMTIMEOUT = 0.0
        self.POLLTIMEOUT = 2.0  # seconds
        self.SOCK_REUSE = True  # so_reuseaddr on server sockets?
        self.SOCK_NODELAY = False  # tcp_nodelay on socket?
        self.THREADING2 = False  # use threading2 if available?
        self.ONEWAY_THREADED = True  # oneway calls run in their own thread
        self.DETAILED_TRACEBACK = False
        self.THREADPOOL_SIZE = 16
        self.AUTOPROXY = True
        self.MAX_MESSAGE_SIZE = 0  # 0 = unlimited
        self.BROADCAST_ADDRS = "<broadcast>, 0.0.0.0"  # comma separated list
of broadcast addresses
        self.FLAME_ENABLED = False
        self.PREFER_IP_VERSION = 4  # 4, 6 or 0 (let OS choose according to RFC
3484)
        self.SERIALIZER = "pickle"
        self.SERIALIZERS_ACCEPTED = "pickle,marshal,json"   # these are the
'safe' serializers
        self.LOGWIRE = False  # log wire-level messages
        self.PICKLE_PROTOCOL_VERSION = pickle.HIGHEST_PROTOCOL
        self.METADATA = True  # get metadata from server on proxy connect
        self.REQUIRE_EXPOSE = False  # require @expose to make members remotely
accessible (if False, everything is accessible)
        self.USE_MSG_WAITALL = hasattr(socket, "MSG_WAITALL") and
platform.system() != "Windows"      # not reliable on windows even though it is
defined
        self.JSON_MODULE = "json"
        self.MAX_RETRIES = 0
```

For example, if **Pickle** is selected, `self.SERIALIZER` value should be changed to **pickle** and `self.SERIALIZERS_ACCEPTED` value should be changed to include **pickle**:

4. Enter the required *Python script* to execute on the active Pyro instance.
5. Check the **Use Apache Arrow** box to enable fast serialization of data frames.
6. Select whether the parameters should be automatically enclosed in quotes by checking the **Enclose Parameters in Quotes** box.

7. Click  to save and display the details of the data extract.

8. Click  then  to display the data preview.

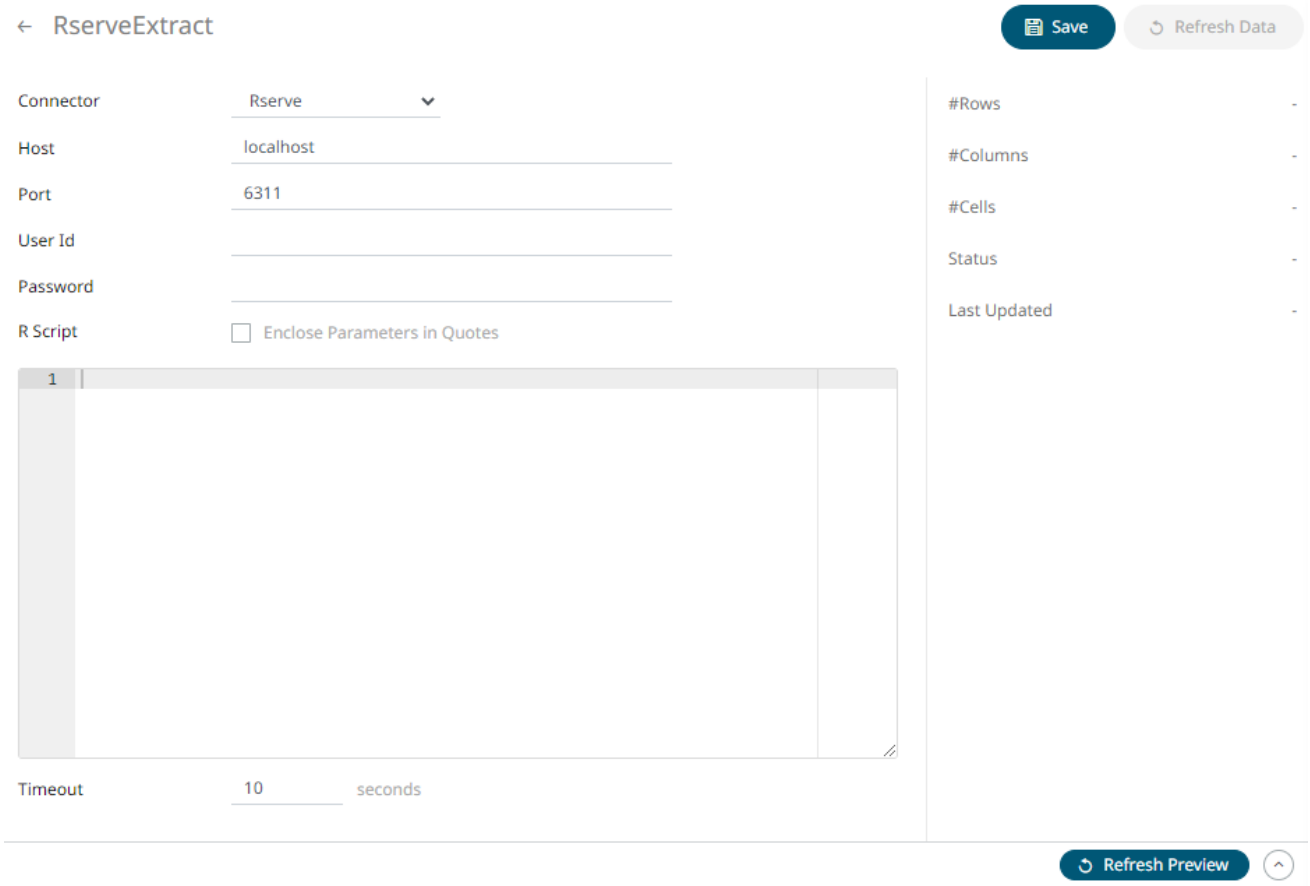
Creating Data Extract from Rserve

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. On the *New Data Extract* page, select **Rserve** in the *Connector* drop-down list.



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. Enter the *Timeout*. Default is **10** (in seconds).
5. Select whether the parameters should be automatically enclosed in quotes by checking the *Enclose parameters in quotes* box.
6. Click 

to save and display the details of the data extract.
7. Click  then 

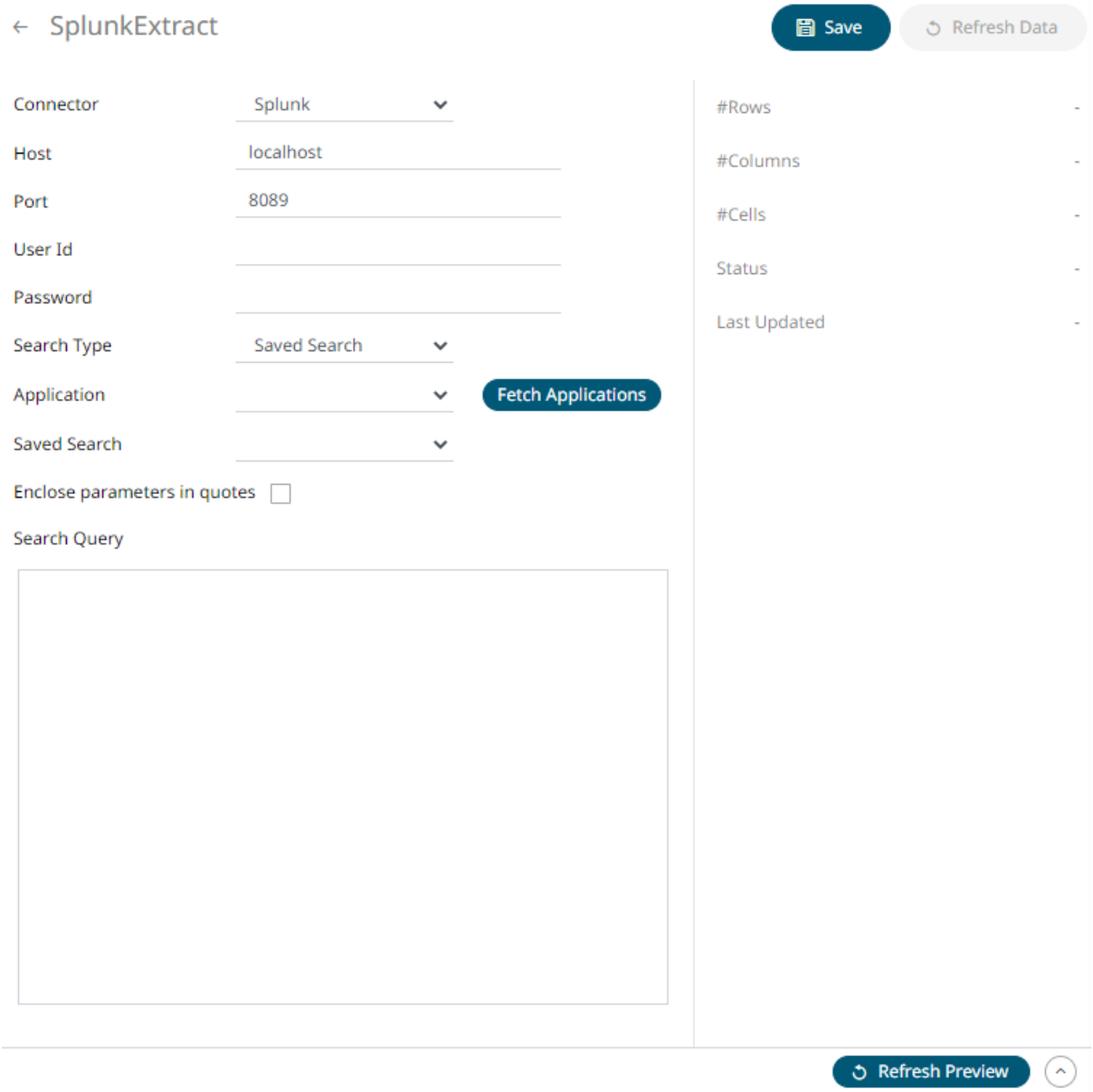
to display the data preview.

Creating Data Extract from Splunk

The Splunk connector allows the retrieval of data from a Splunk instance.

Steps:

1. On the *New Data Extract* page, select **Splunk** in the *Connector* drop-down list.



2. Enter the following properties:

Property	Description
Host	Splunk host address.
Port	Splunk host port. Default is **8089**.
User Id	The user Id that will be used to connect to the Splunk service.
Password	The password that will be used to connect to the Splunk service.

3. Select the *Search Type*:
   - Manual

     Proceed to step 6 to define a new search query.

- Saved Search

  Allows you to select in the *Saved Search* drop-down list.

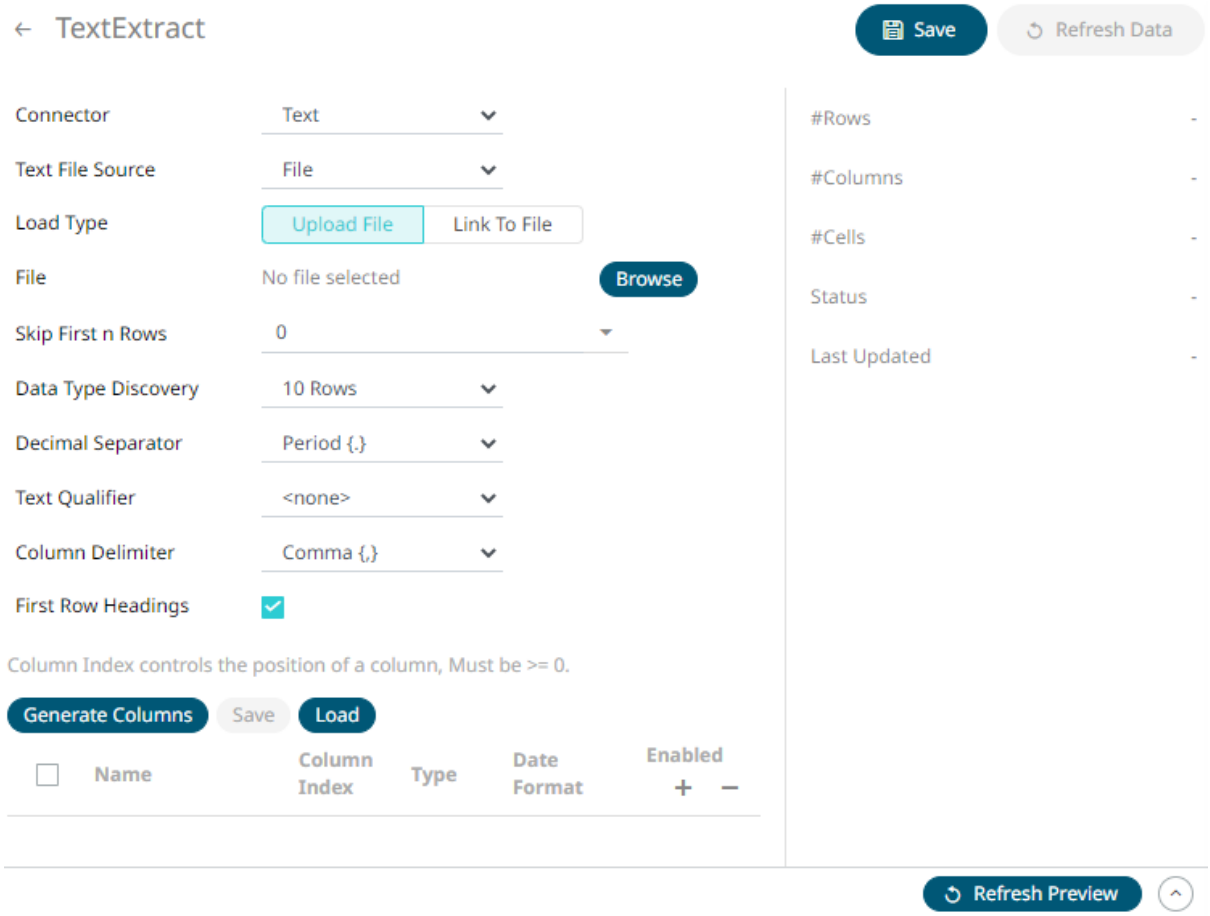
4. Click Fetch Applications to populate the *Application* drop-down list and select one.
5. Select whether the parameters should be automatically enclosed in quotes by checking the **Enclose parameters in quotes** box.
6. Enter a *Search Query*.
7. Click Save to save and display the details of the data extract.
8. Click then Refresh Preview to display the data preview.

Creating Data Extract from Text

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:

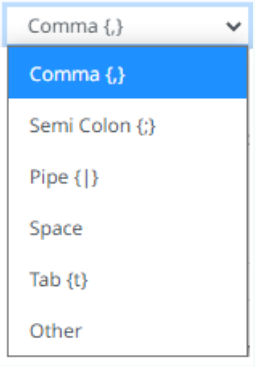
1. On the *New Data Extract* page, select **Text** in the *Connector* drop-down list.



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 dot (.) 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.
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. 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 − .

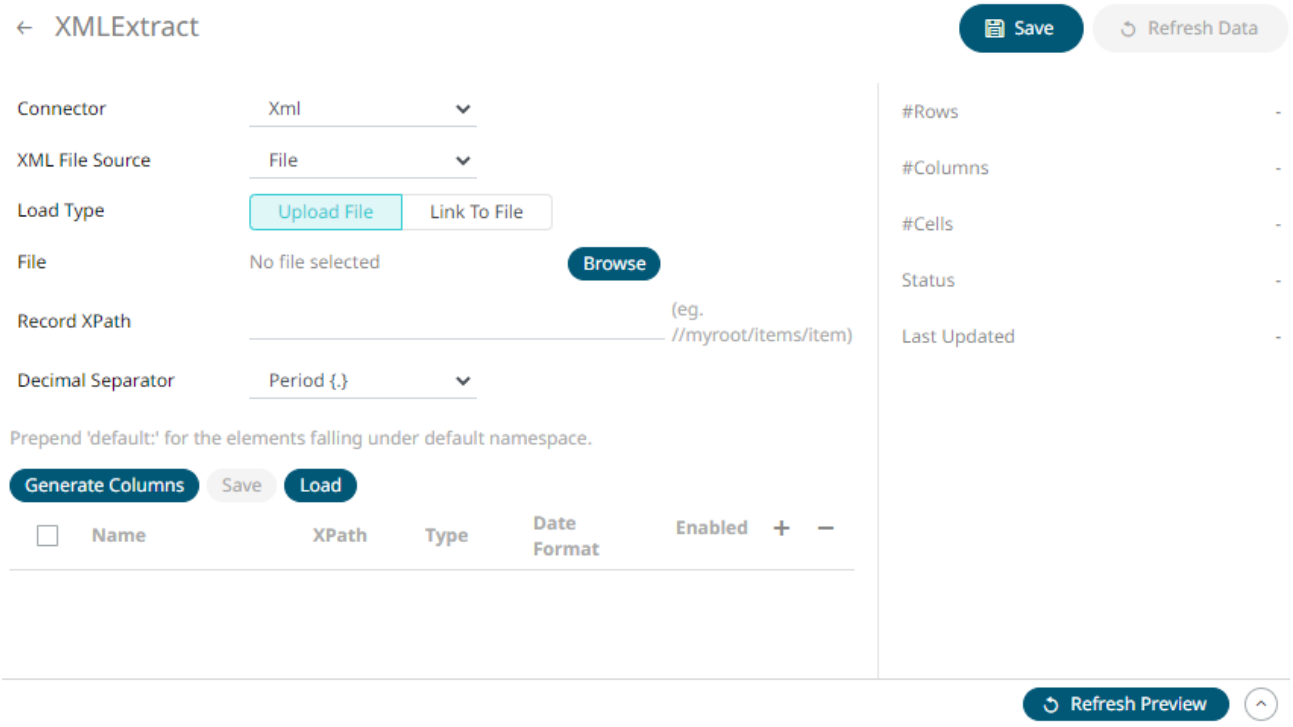
6. Click Save to save and display the details of the data extract.
7. Click ^ then Refresh Preview to display the data preview.

Creating Data Extract from XML

The XML connector allows the retrieval and processing of XML files, either from a disk, a Text, or from a defined URL.

Steps:

1. On the *New Data Extract* page, select **Xml** in the *Connector* drop-down list.



2. Select the XML *File Source*.
3. Enter the *Record XPath* (e.g., `//myroot/items/item`).
4. Select either the dot (.) 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. 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 −.

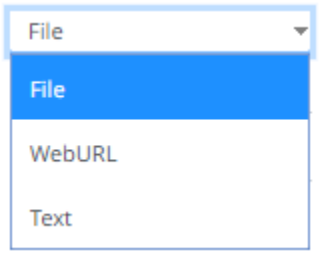
8. Click Save to save and display the details of the data extract.
9. Click ^ then Refresh Preview to display the data preview.

Selecting and Defining the Data Connector File Source

Several connectors including JSON, MS Excel, Text, and, XML, allow selection from a File, Web URL, or Text source.

Steps:

Select the connector file source:

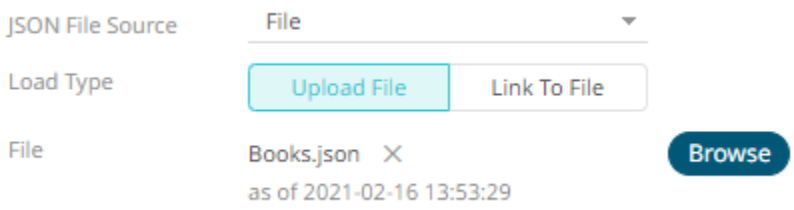


- File

  You can either:

  - Upload a data source snapshot by clicking **Upload File** Upload File then **Browse** Browse to browse to the file source.

    After selecting the file, it is displayed with the timestamp of the snapshot.



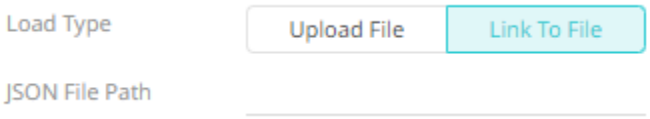
    The data source is placed in the repository and locked, synchronized, and bundled with the workbook version.

To change the data source, click ✕ then **Browse** 

to browse to a new version of the file, which is uploaded into the repository, and also create a new version of the workbook that reads it.

- Link to a data source file by clicking **Link to File** 

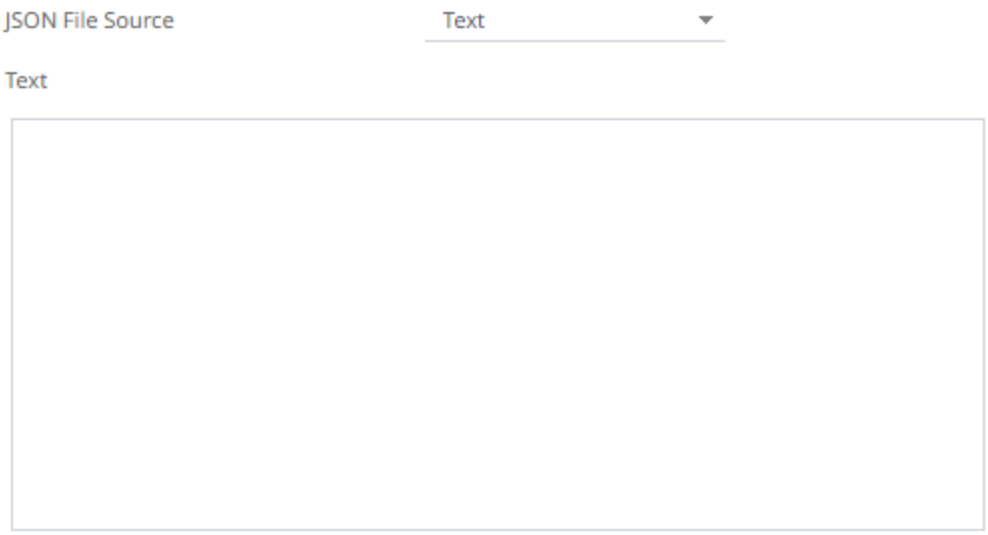
and entering a *File Path*.



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.

- Text

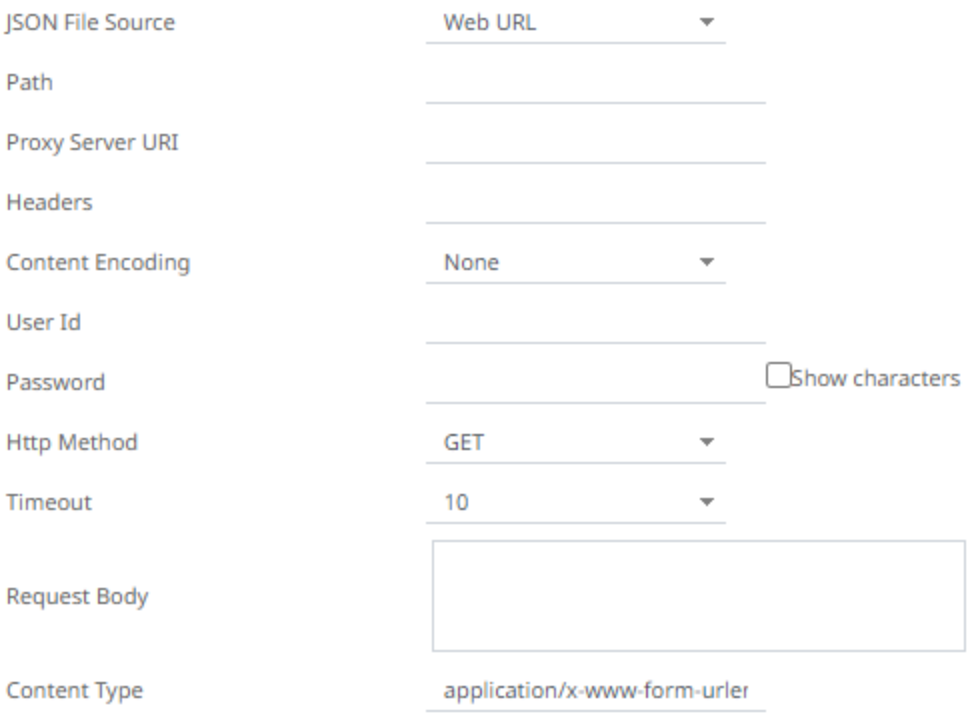
Then enter the text block to be parsed.



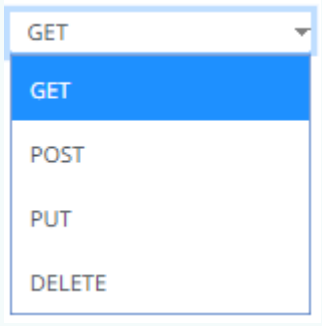
NOTE	The Text file source is not available for the MS Excel connector.

- Web URL

The dialog changes to allow specification of the following:



Property	Description
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	• Headers are separated by a comma • Each Header is entered as **Name = Value**, where *Name* and *Value* can be enclosed in double quotes to allow inclusion of any character except for double quotes • *Name* and *Value* can also be left unquoted, in which case they may not include comma or equals characters
Content Encoding	Select the *Content Encoding* 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. Check the **Show Characters** box to display the entered characters.
HTTP Method	Select the appropriate HTTP method for the request from the following options:

	 • 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**).

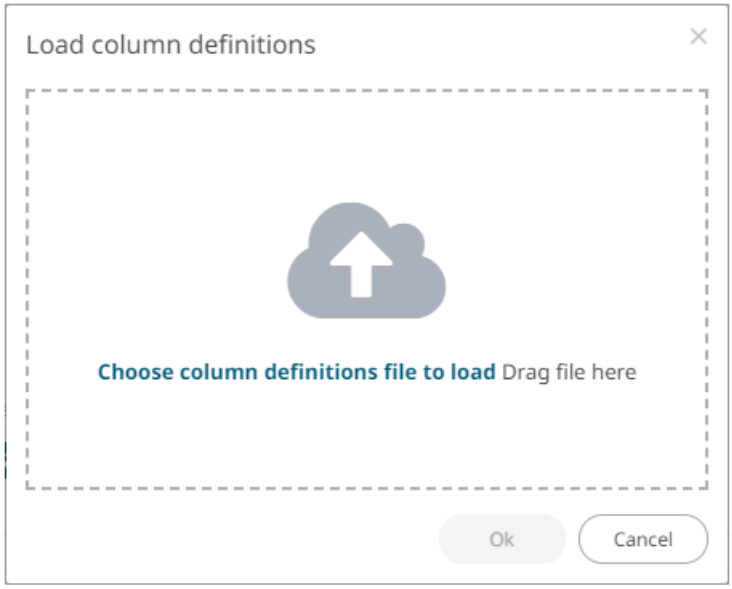
Saving or Loading Column Definitions in the Data Sources

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.

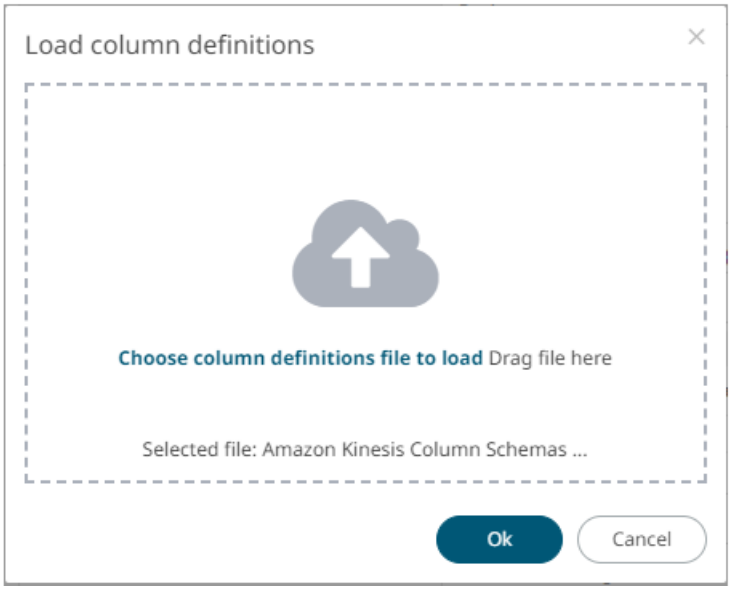
   The *Load Column Definitions* dialog displays.



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.

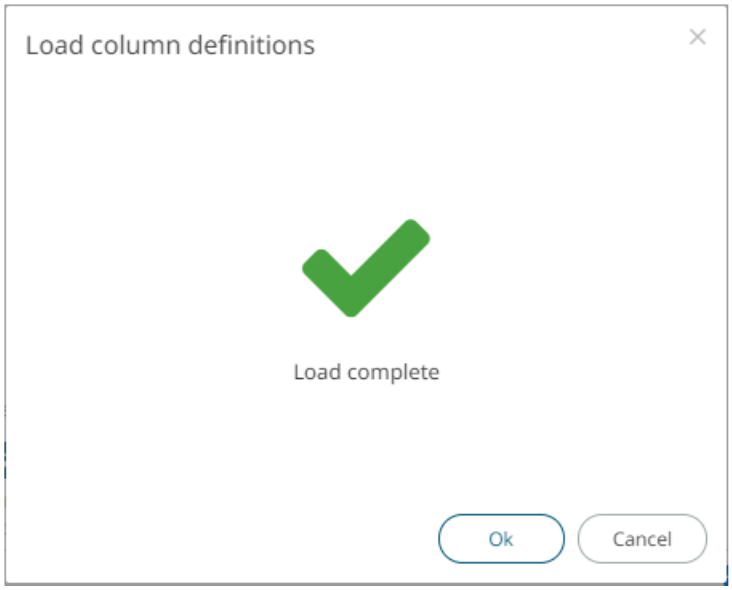
  The name of the column definitions is displayed on the loaded column definitions area.



2.2. Click 

.

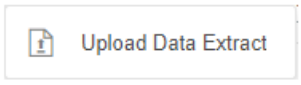
A notification displays when the file is loaded.



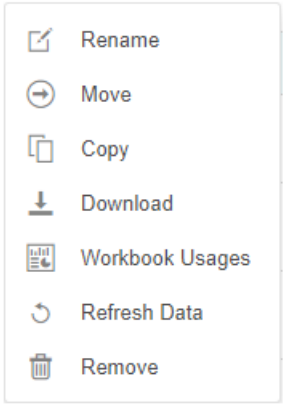
This populates the list of columns from the .exs file.

DATA EXTRACT AND FOLDER CONTEXT MENU

The *Data Extracts* page provides context menu in each folder and the data extract.



Folder Context Menu



Data Extract Context Menu

The *Data Extract* context menu options include:

Menu Option	Description
Rename	Rename the data extract.
Move	Move a data extract to another folder where the user has permission.
Copy	Copy a data extract to another folder where the user has permission.
Download	Download a copy of the data extract definition.

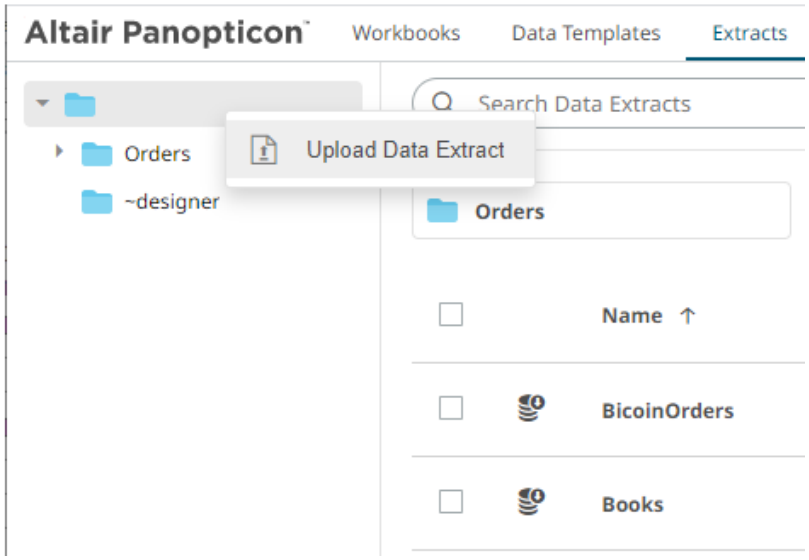
Workbook Usages	View the list of workbooks currently using the data extract.
Refresh Data	Refresh the data extract.
Remove	Delete the data extract.

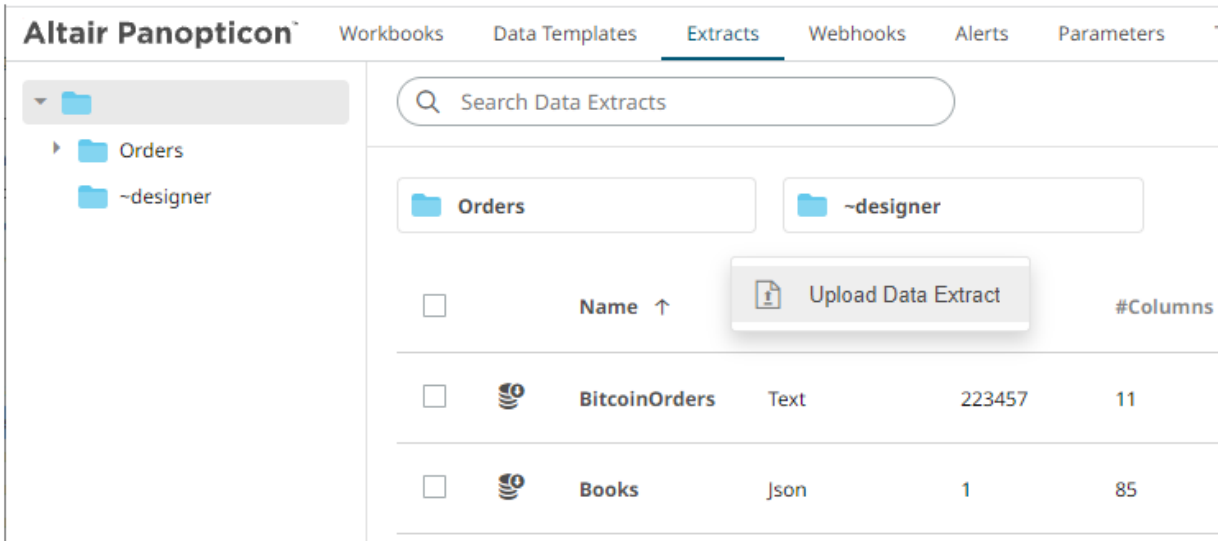
Uploading Data Extracts

Users with an Administrator role can upload data extracts to folders where they have permission.

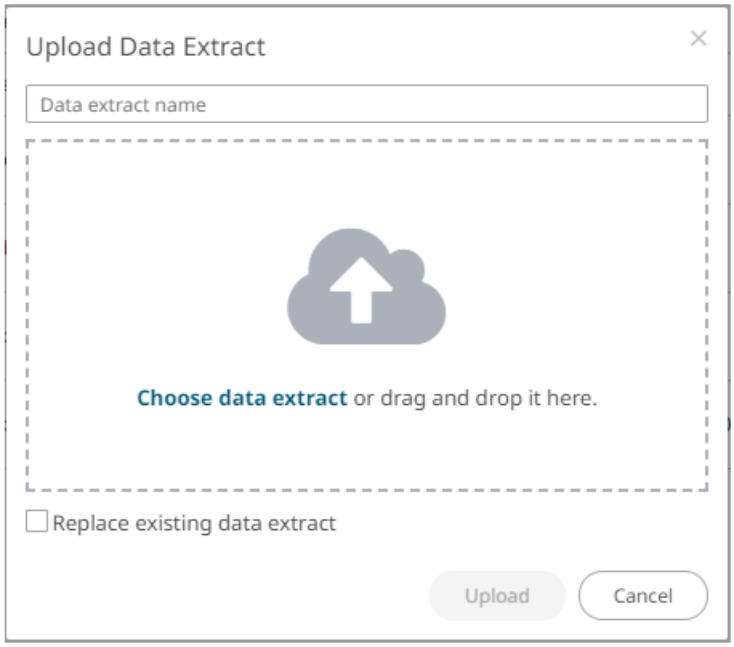
Steps:

1. On the *Data Extracts* page, click on a folder or a personal folder and select **Upload Data Extract**.



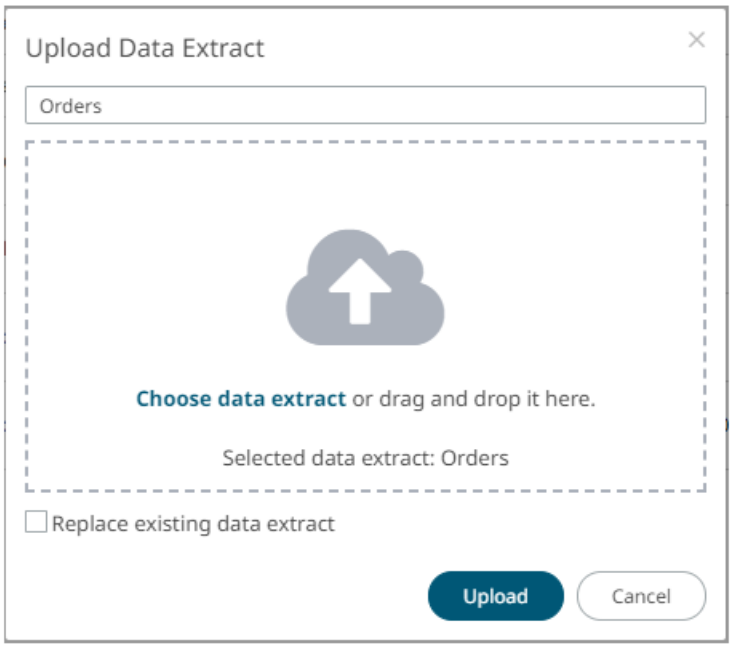


The *Upload Data Extract* dialog displays.



2. To upload a data extract, you can either:
   - drag it from your desktop and drop on the dialog, or
   - click **Choose Data Extract** and select one on the *Open* dialog that displays.

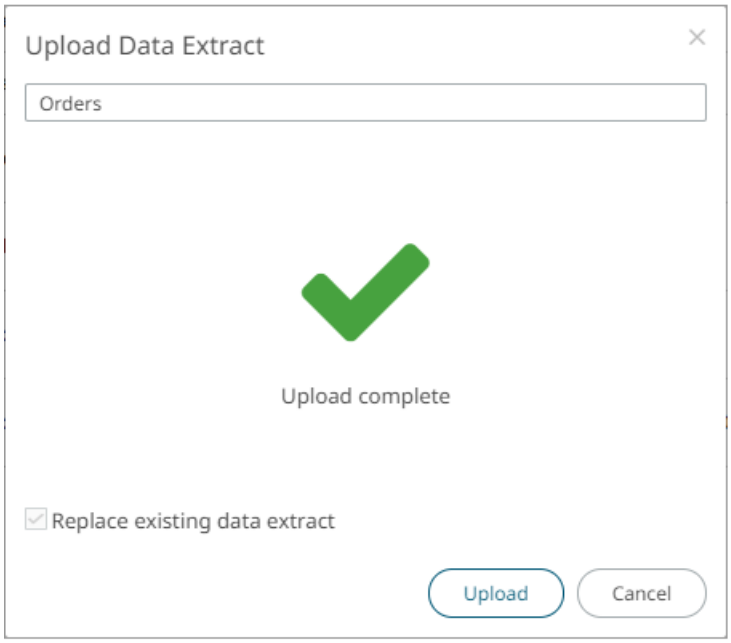
   The name of the data extract is displayed on the uploaded data extract area and in the *Name* box.



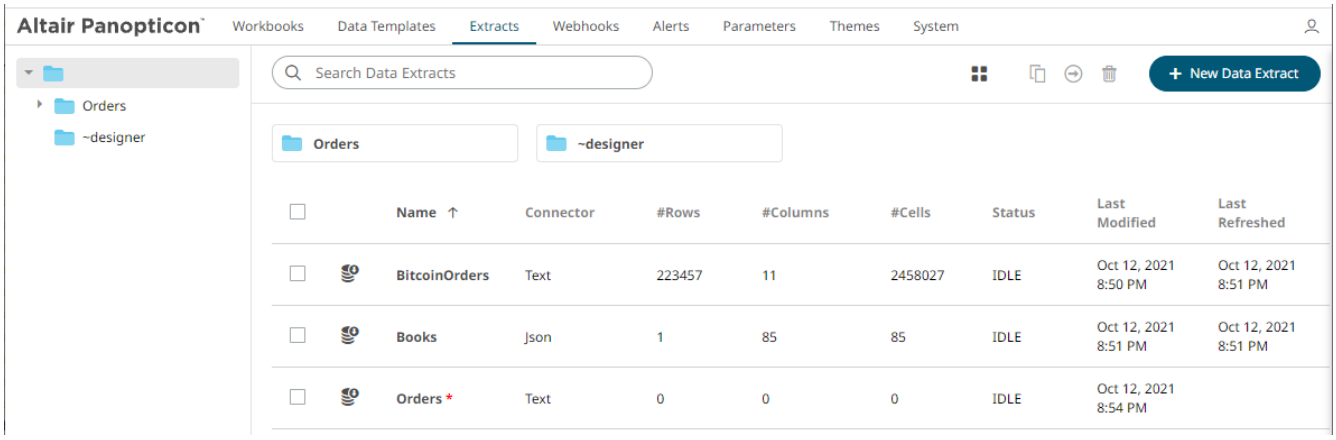
3. You can opt to rename the data extract.
4. To replace an existing data extract, check the *Replace existing data extract* box.
5. Click 

.

   You will be notified once the data extract is uploaded.



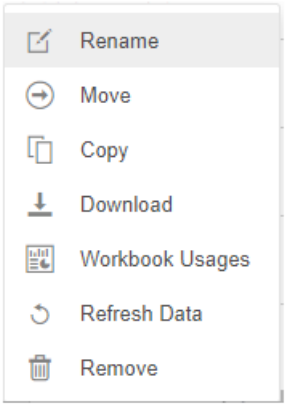
The data extract is uploaded and added in the designated folder.



Renaming a Data Extract

Steps:

1. Right-click on a data extract then select **Rename** on the context menu.



The *Rename Data Extract* dialog displays.

2. Enter a new name then click 

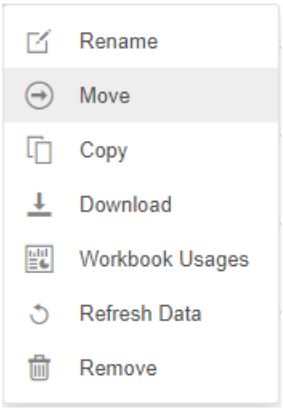
.

Moving a Data Extract

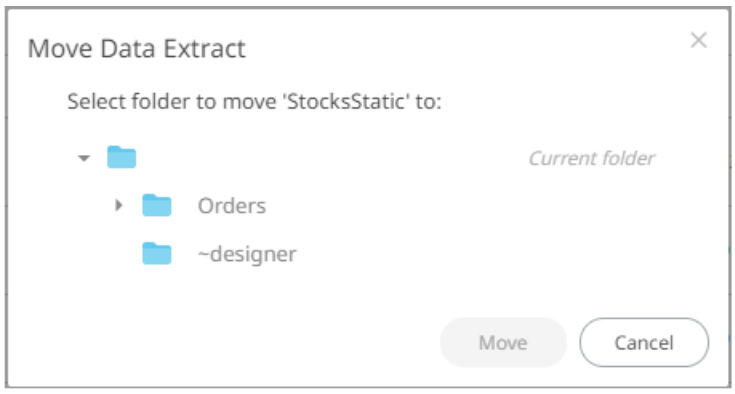
Users with a Designer role are allowed to move a data extract to another folder where they have permission.

Steps:

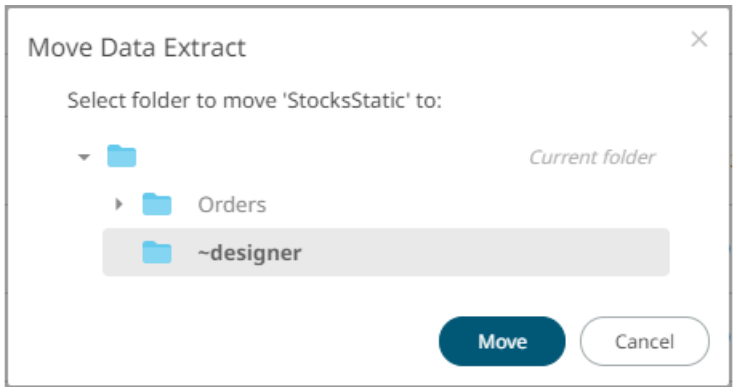
1. Right-click on a data extract and select **Move** on the context menu.



The *Move Data Extract* dialog displays with the folders that the user is allowed to move the data extract.



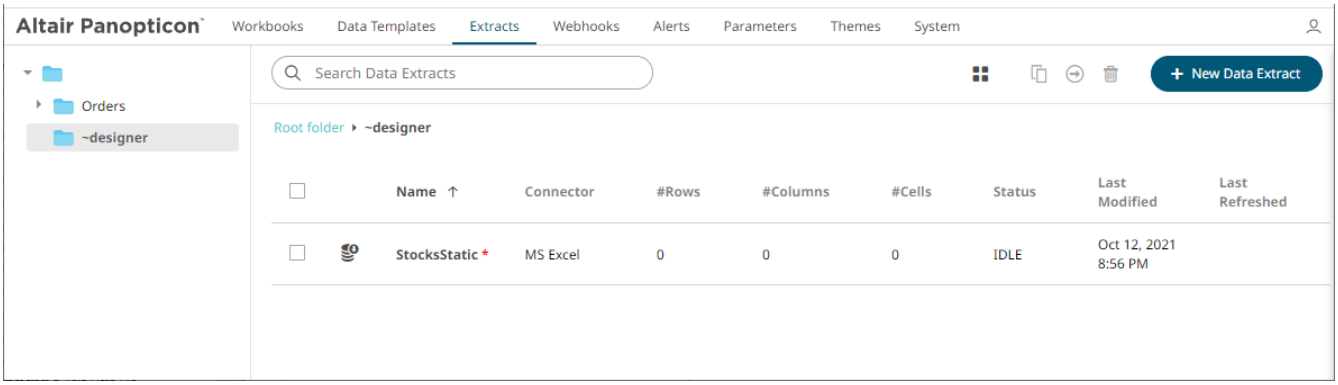
1. Select the folder or subfolder.



2. Click 

.

   The data extract is moved and displayed on the selected folder.

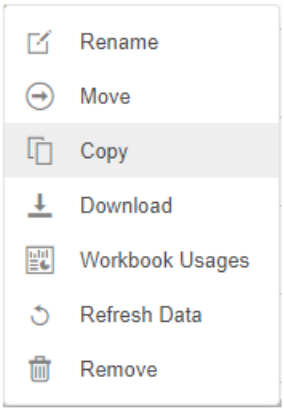


Copying a Data Extract

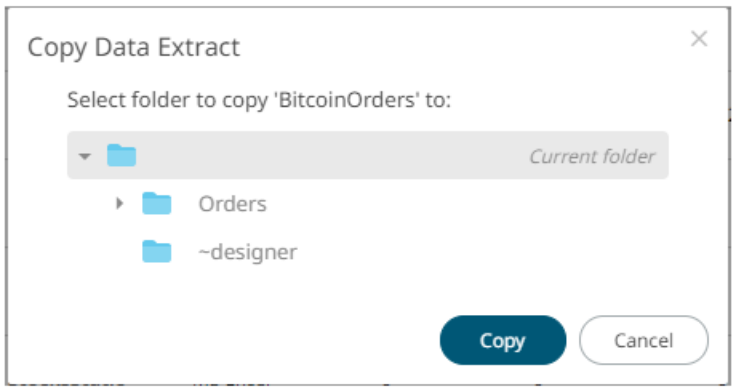
Users with a Designer role are allowed to copy a data extract to another folder where they have permission.

Steps:

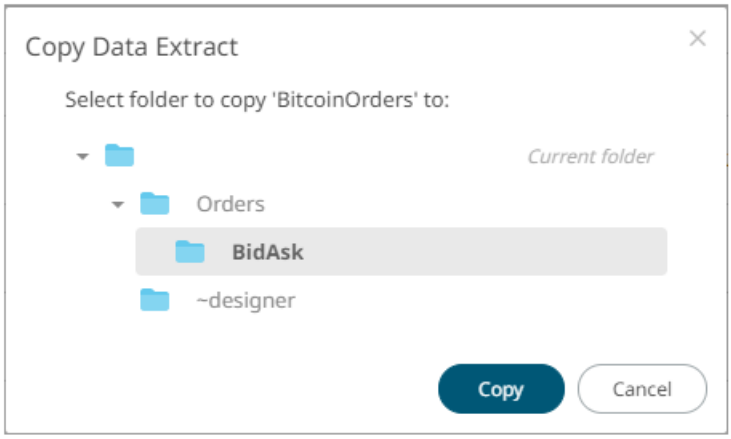
1. Right-click on a data extract and select **Copy** on the context menu.



The *Copy Data Extract* dialog displays with the folder or subfolders the user is allowed to copy the data extract to.



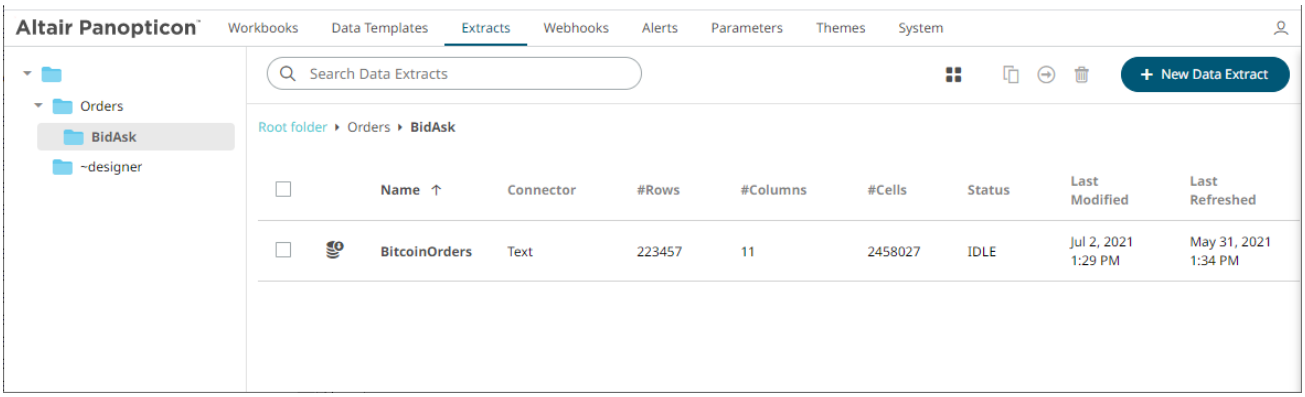
1. Select the folder or subfolder.



2. Click 

.

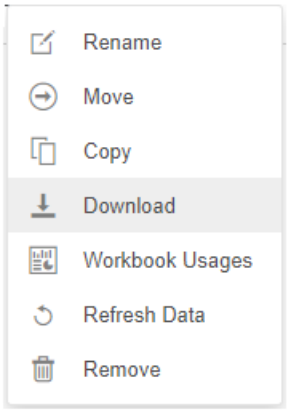
The data extract is copied and displayed on the selected folder.



Downloading a Data Extract

A user with a Designer role with READ + WRITE permission to the folder is allowed to download a copy of a data extract available in it.

Right-click on a data extract and select **Download** on the context menu.



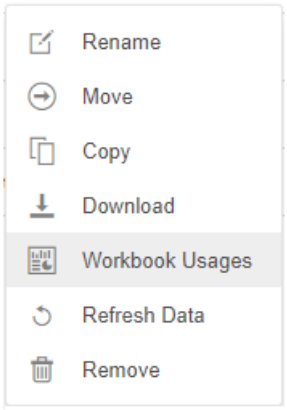
A copy of the data extract is downloaded.

Viewing the Data Extract Usage

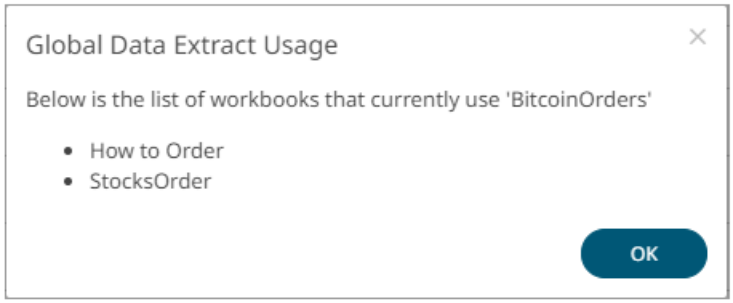
View the list of workbooks that is currently using the data extract.

Steps:

1. Right-click on a data extract and select **Workbook Usage** on the context menu.



The list of workbooks that is currently using the data extract is displayed in the *Data Extract Usage* dialog.



2. Click OK.

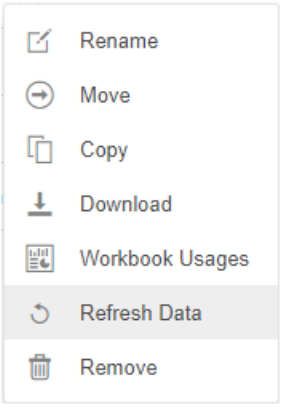
Refreshing the Data Extract

After saving the modifications in the data extract settings, the extract is flushed. Refresh the data extract to run it. Consequently, the extract meta data details are displayed.

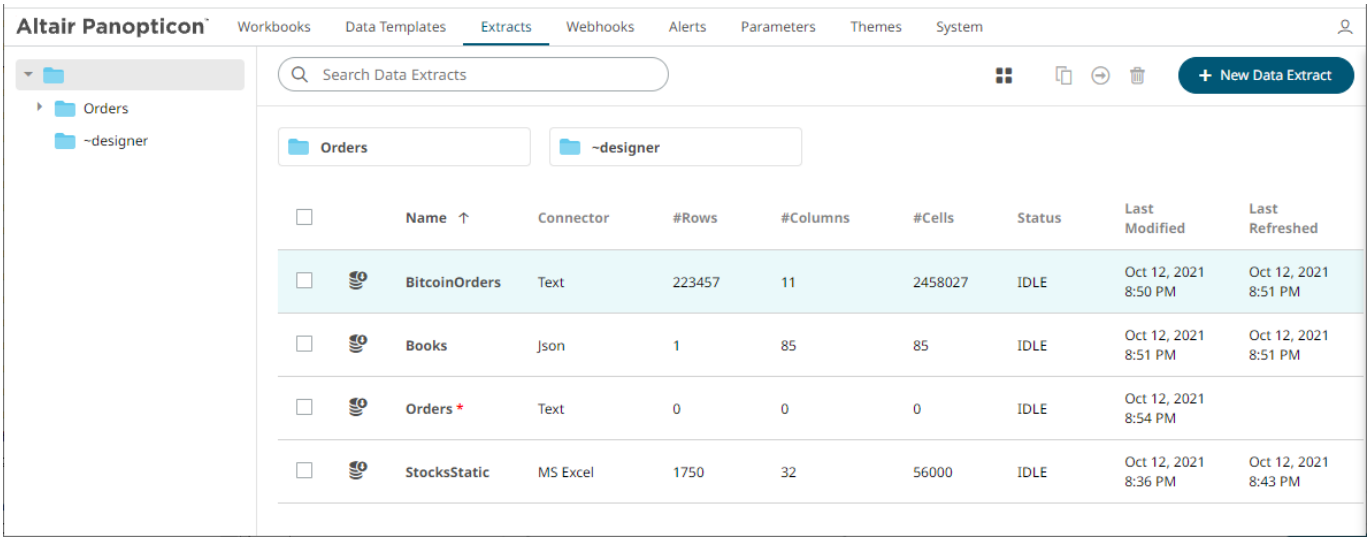
NOTE	A * symbol appears beside a data extract that is not yet refreshed.

Steps:

Right-click on a data extract and select **Refresh Data** on the context menu.



The data extract is first ran and the status changes to **RUNNING**. When the data extract is complete, the status changes to **IDLE**. Also, the number of *Rows*, *Columns*, *Cells*, *Status*, and the last Date/Time it was updated and refreshed are displayed.



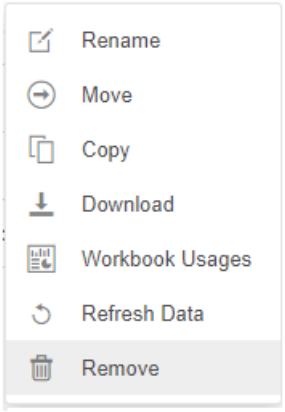
NOTE	A * symbol appears beside a data extract that is not yet refreshed.

Deleting a Data Extract

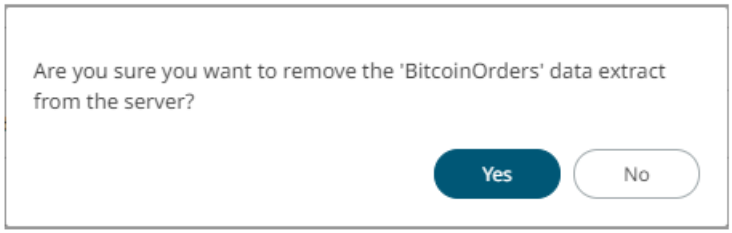
Users with a Designer role have the ability to remove a data extract.

Steps:

1. Right-click on a data extract and select **Remove** on the context menu.



A notification message displays.



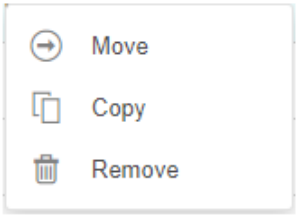
2. Click 

to remove.

DATA EXTRACTS TOOLBAR AND CONTEXT MENU

Moving, copying, and removing data extracts can either be done using:

- Context menu



- Toolbar

List View

Grid View

The *Data Extracts* toolbar options include:

Toolbar Option	Description
Sort By / Sort Order	Allows sorting data extracts by *Name, Last Modified,* or *Connector*.
Display View	Display data extracts either by *List View* or *Grid View*.
Copy	Copy data extracts to another folder or subfolder where the user has permission.
Move	Move data extracts to another folder or subfolder where the user has permission.
Remove	Remove data extracts.

The *Context Menu* options include:

Toolbar Option	Description
Copy	Copy data extracts to another folder or subfolder where the user has permission.
Move	Move data extracts to another folder or subfolder where the user has permission.
Remove	Remove data extracts.

Sorting Data Extracts

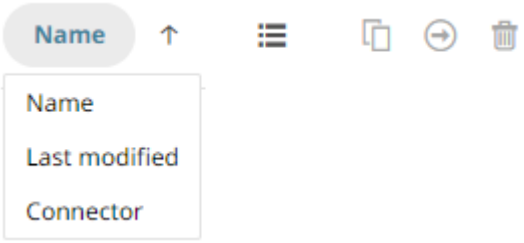
Sorting data extracts can be done by **Name, Last Modified**, or **Connector**.

Steps:

On the *Data Extracts* page, either:

- click the **Sort By** option on the *Toolbar* of the *Grid View*.

  By default, the sorting is by **Name**.

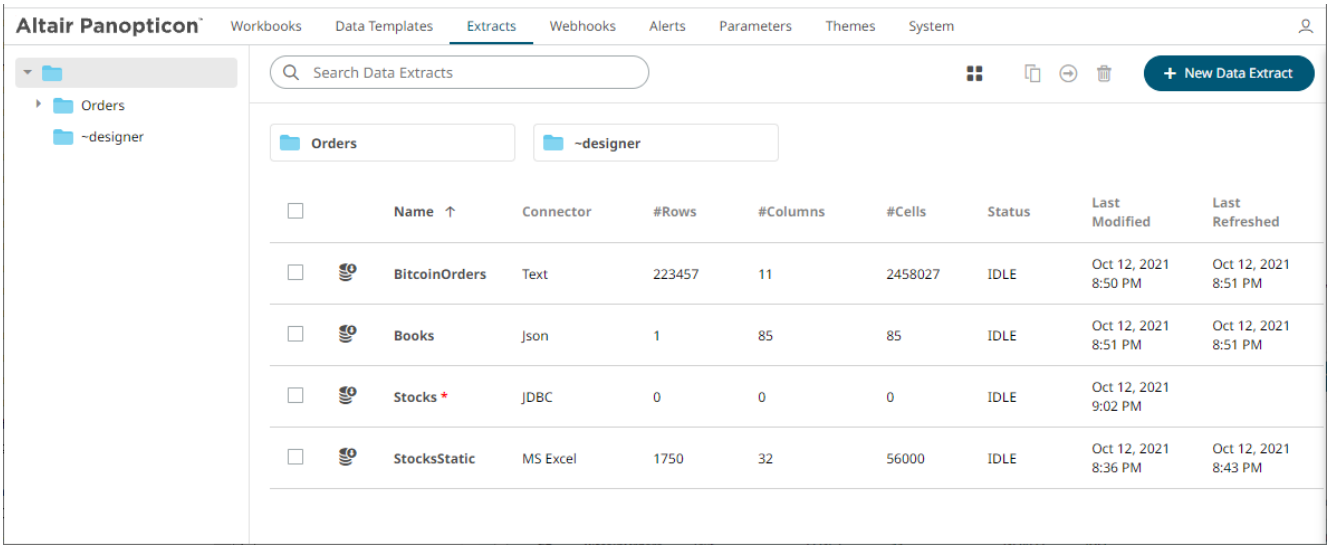


  - Name
  - Last Modified
  - Connector

  Then click the *Sort Order*:

- Ascending
- Descending

- click on the **Name**, **Connector**, **#Rows**, **#Columns**, **#Cells**, **Status**, **Last Modified**, or **Last Refreshed** column header of the *List View*



Then click the *Sort Order*:

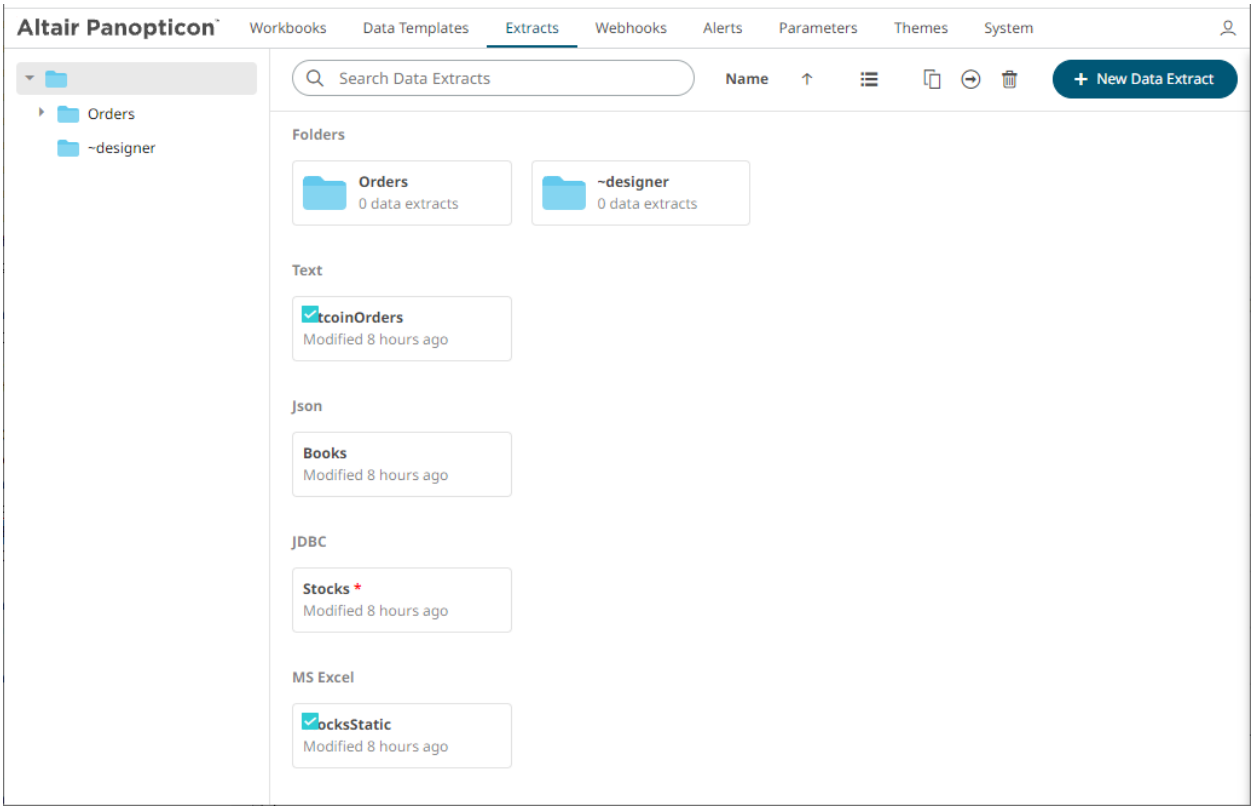
- Ascending
- Descending

Copying Data Extracts Using the Toolbar

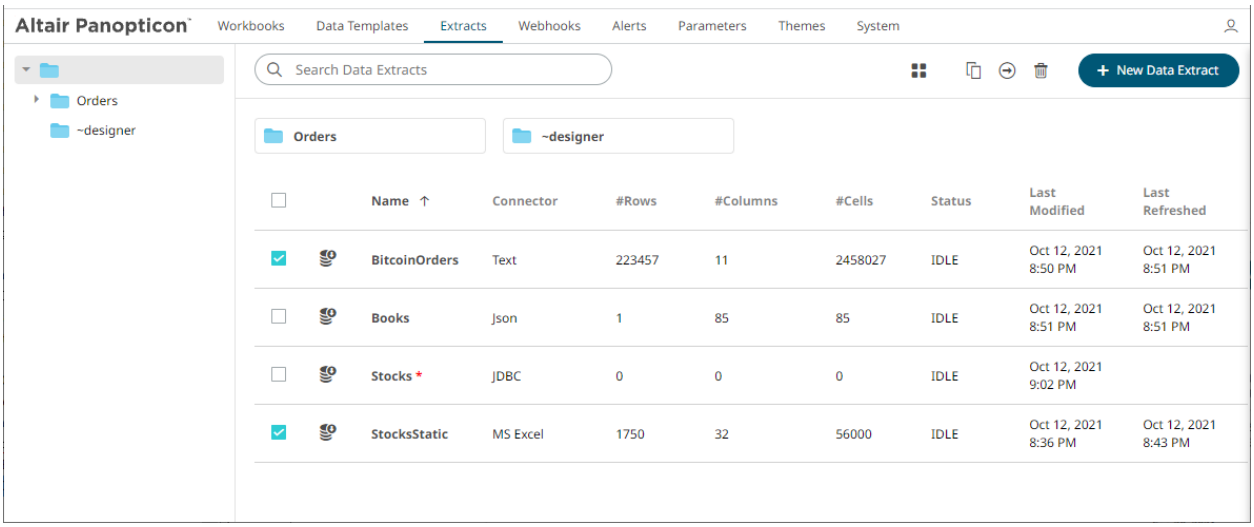
Users with a Designer role are allowed to copy data extracts to another folder or subfolder where they have permission.

Steps:

1. Check the box of data extracts either:
   - on the *Grid View,* or

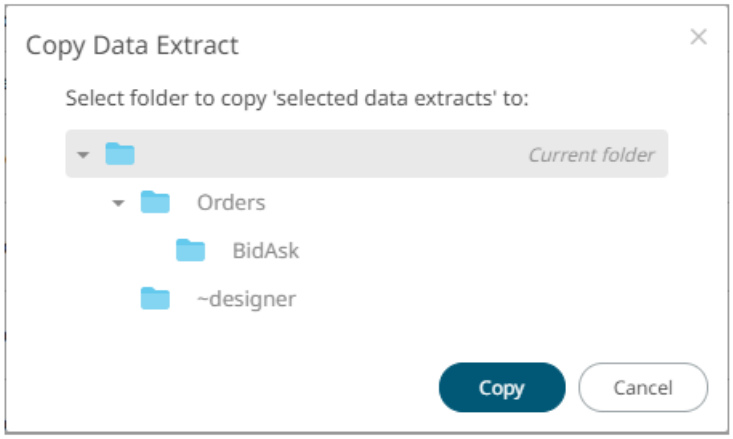


- on the *List View*

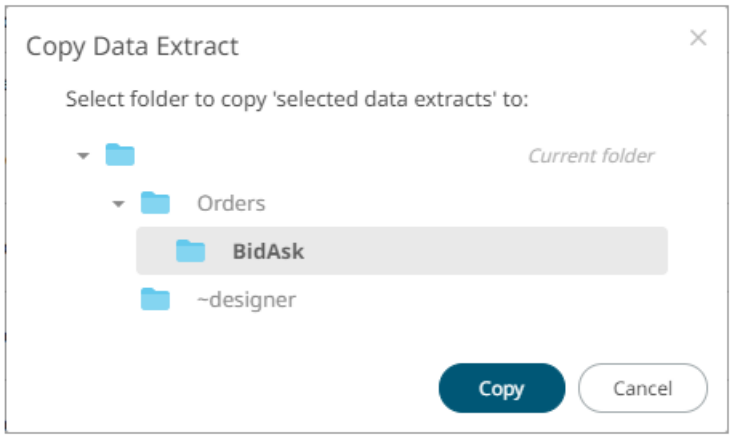


2. Click the **Copy**  icon.

   The *Copy Data Extract* dialog displays with the folder or subfolders the user is allowed to copy the data extracts to.

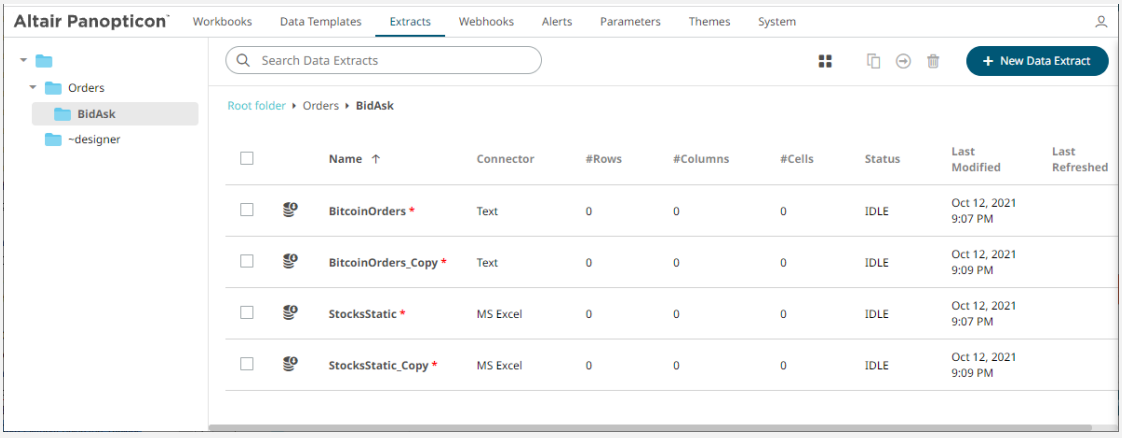


3. Select the folder or subfolder.

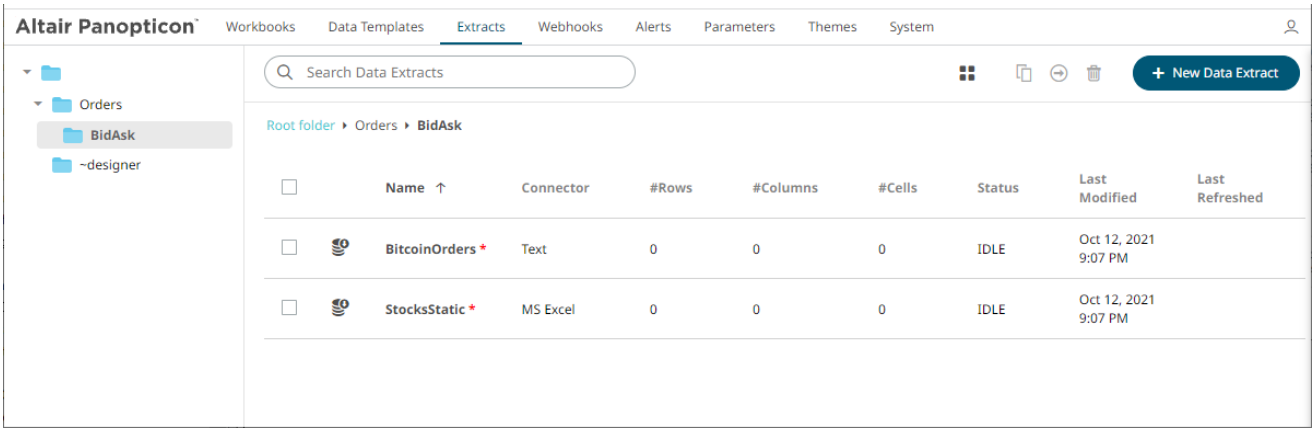


4. Click Copy .

NOTE **If** data extracts with the same name are already in the selected folder, a copy of the data extracts are added.



The data extracts are copied and displayed on the selected folder.

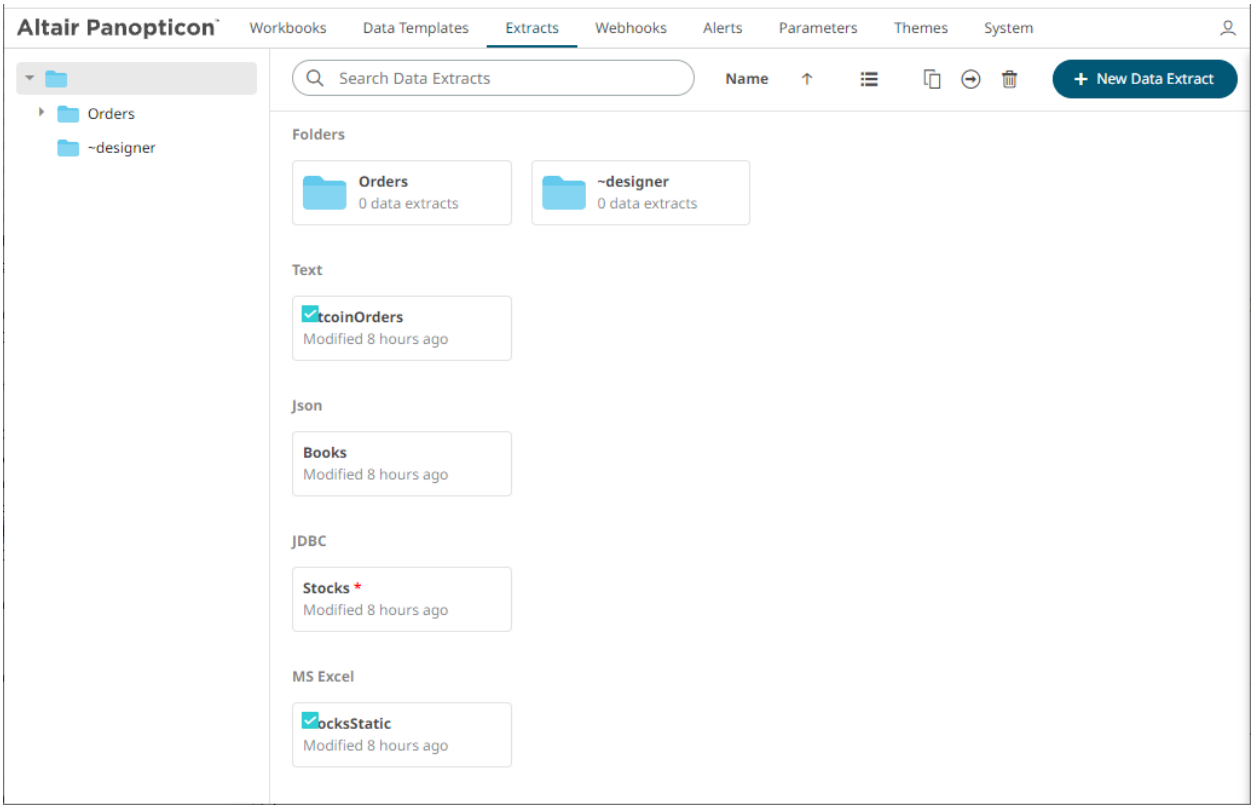


Moving Data Extracts Using the Toolbar

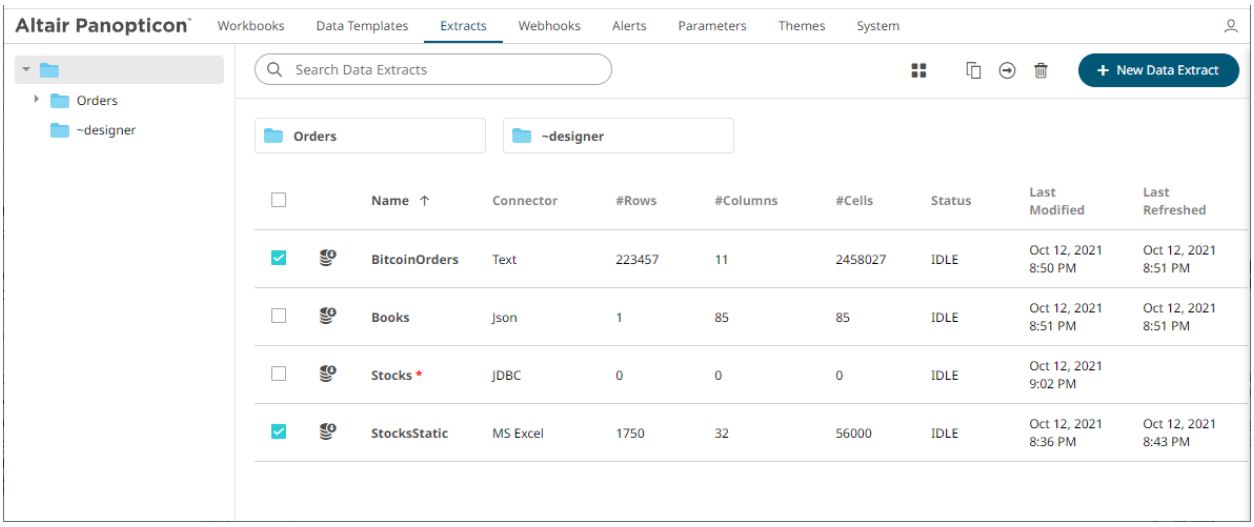
Users with a Designer role are allowed to move data extracts to another folder or subfolder where they have permission.

Steps:

1. Check the box of data extracts either:
   - on the *Grid View,* or

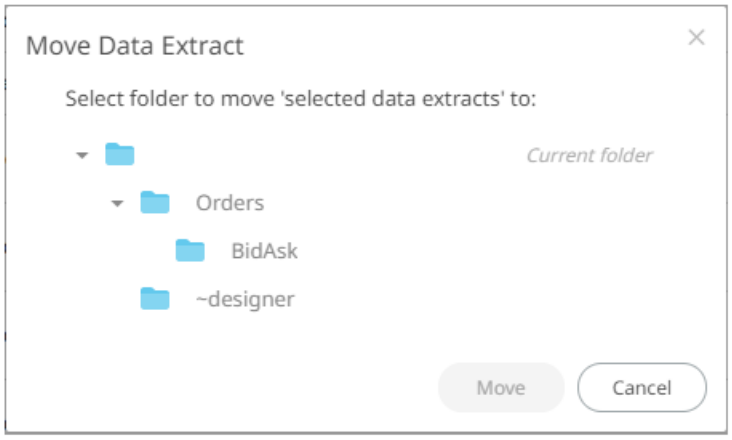


   - on the *List View*

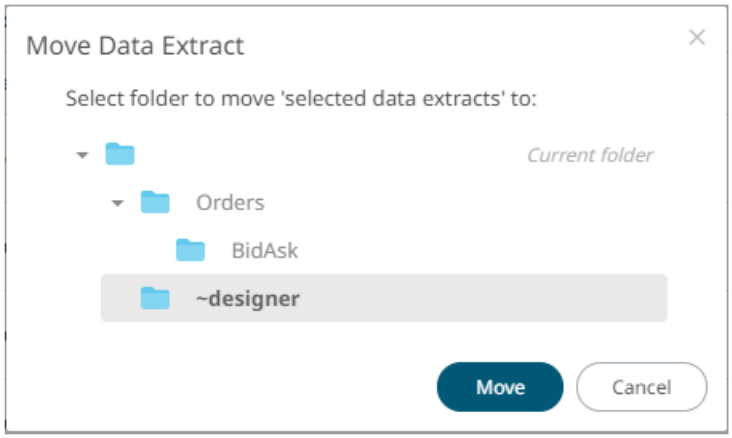


2. Click the **Move** icon.

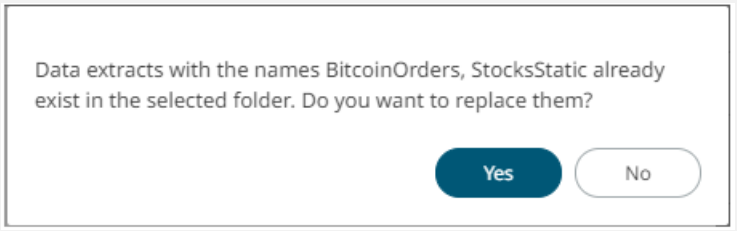
   The *Move Data Extract* dialog displays with the folder or subfolders that the user is allowed to move the data extracts.



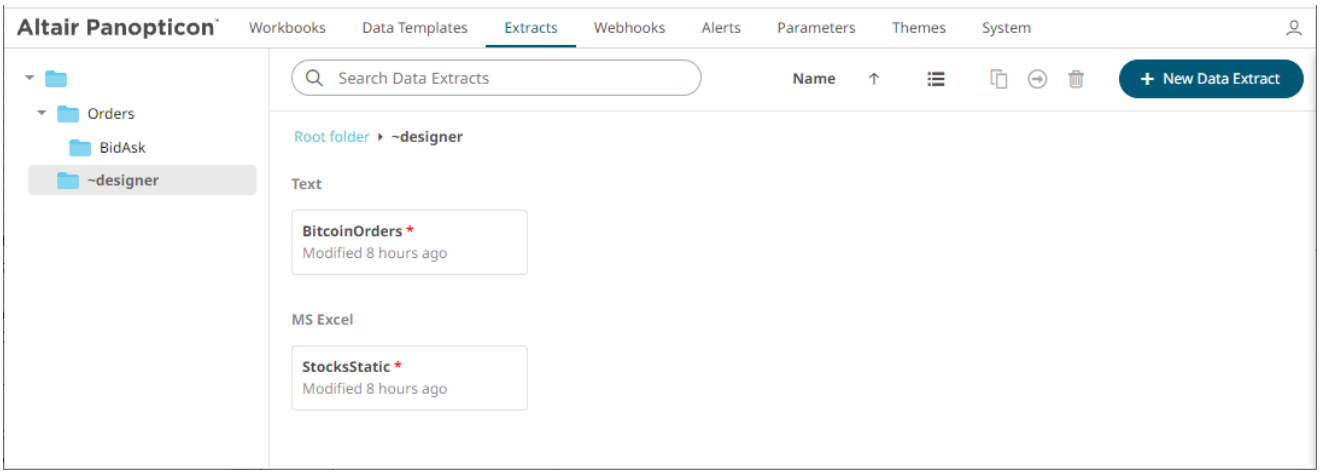
3. Select the folder or subfolder.



4. Click Move .

NOTE	If data extracts with the same name are already in the selected folder, a notification message displays if they will be replaced.  Click **Yes** to replace a copy of the same data extracts.

The data extracts are moved and displayed on the selected folder.

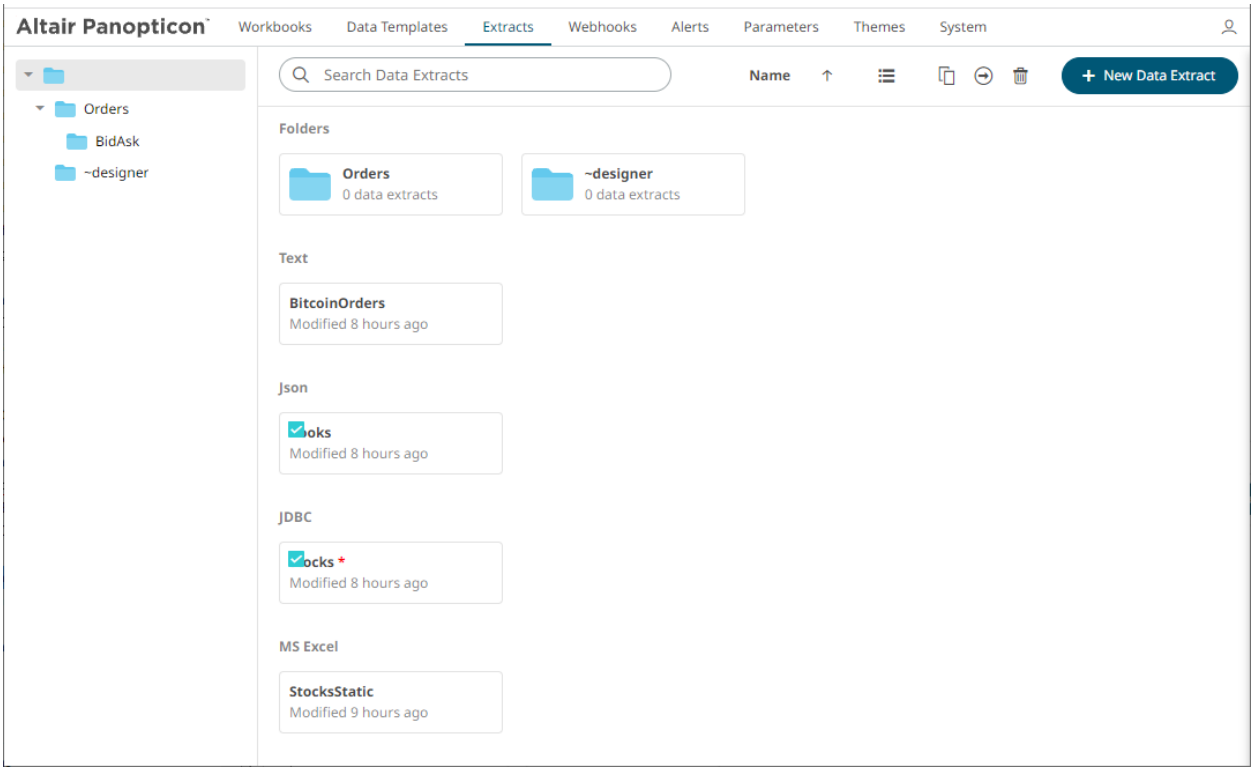


Deleting Data Extracts Using the Toolbar

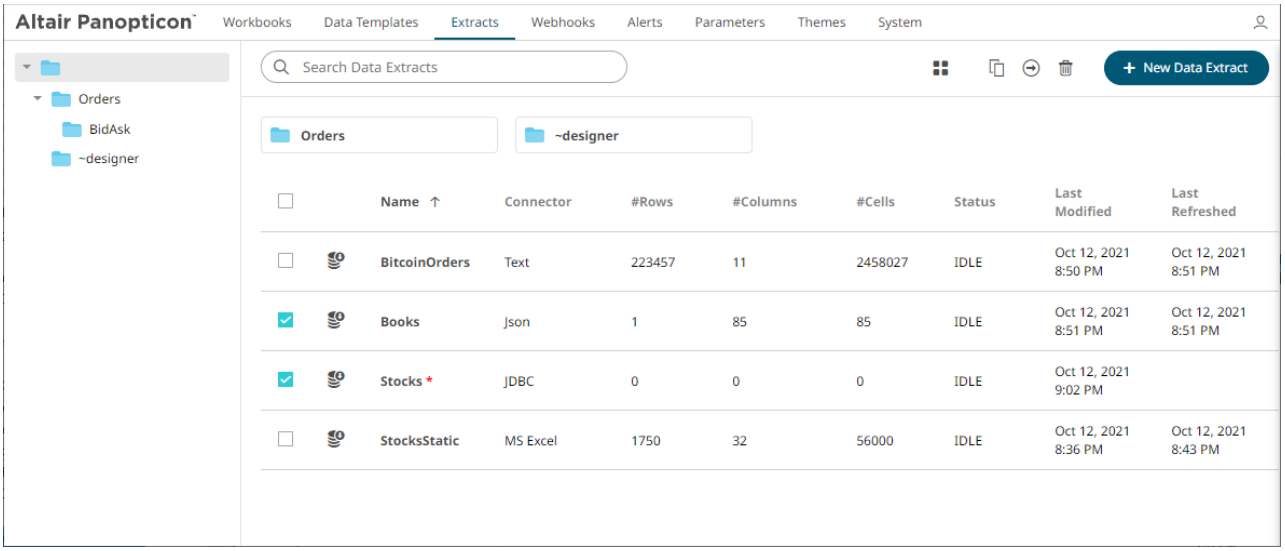
Users with a Designer role have the ability to remove data extracts.

Steps:

1. Check the box of data extracts either:
   - on the *Grid View,* or

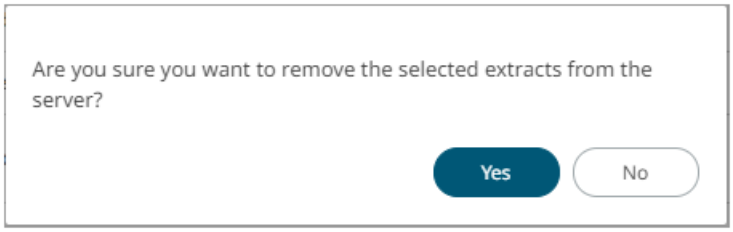


- on the *List View*



2. Click  on the toolbar.

   A notification message displays.



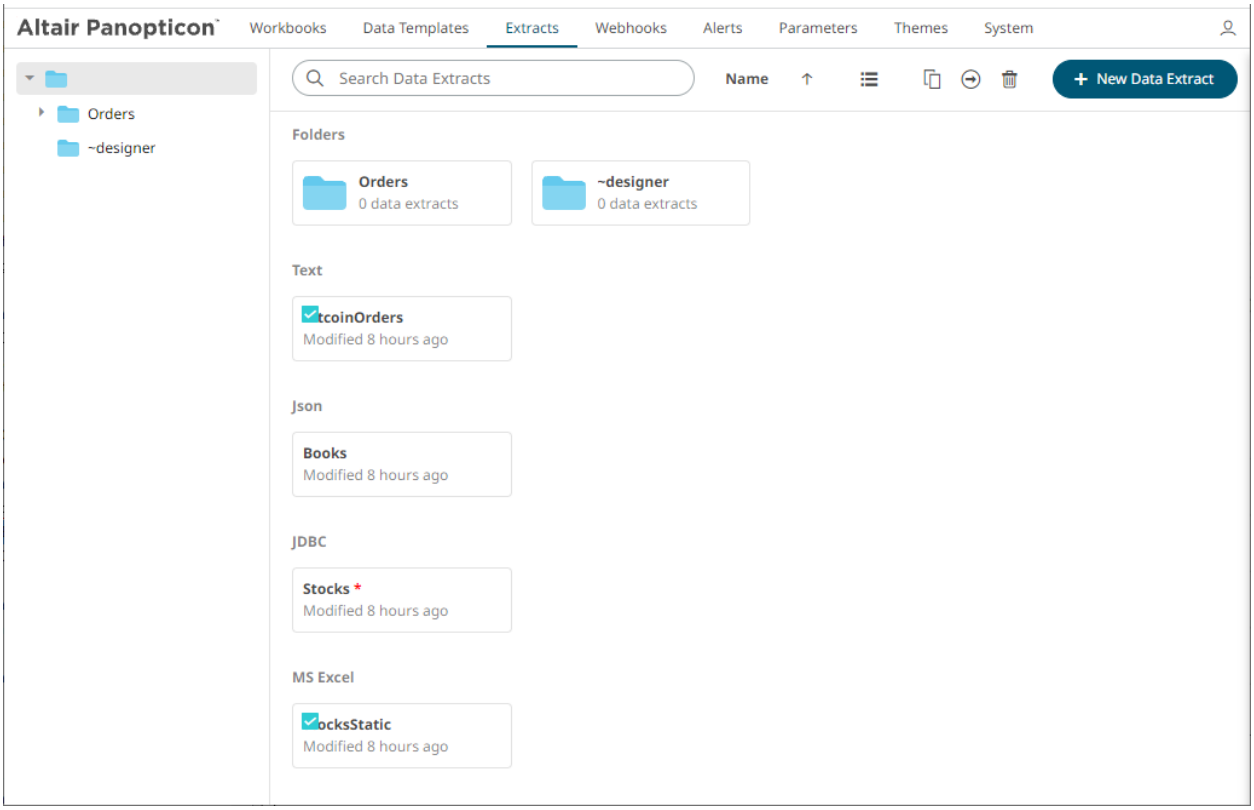
3. Click  to remove.

Copying Data Extracts Using the Context Menu

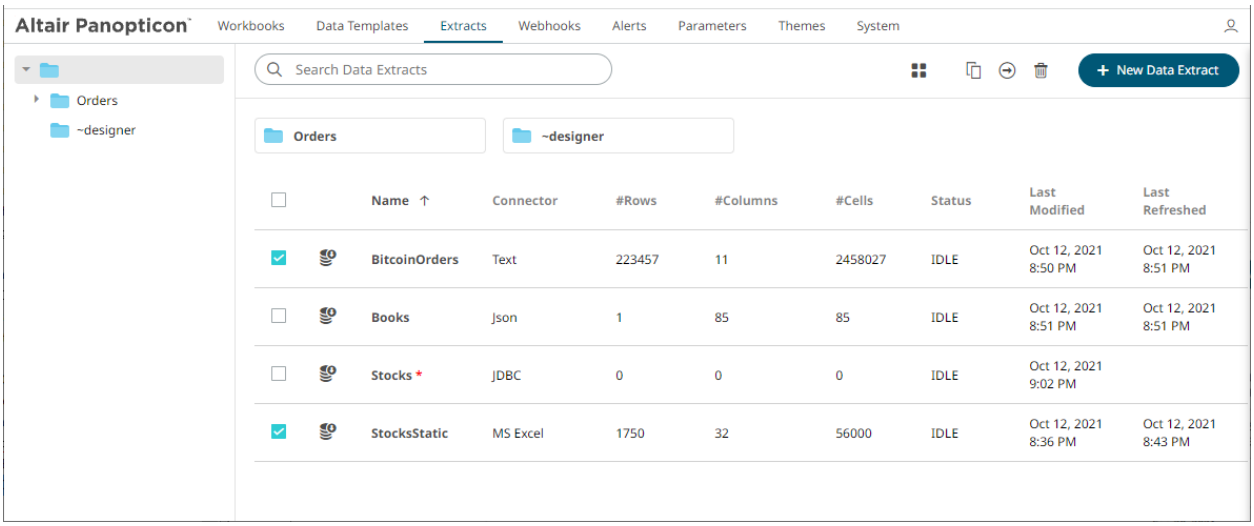
Users with a Designer role are allowed to copy data extracts to another folder or subfolder where they have permission.

Steps:

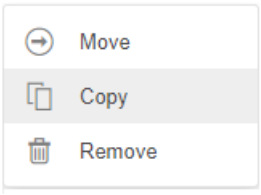
1. Check the box of data extracts either:
   - on the *Grid View,* or



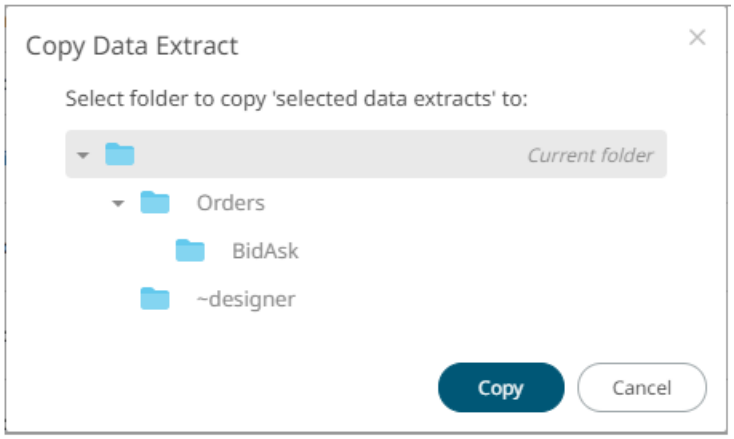
- on the *List View*



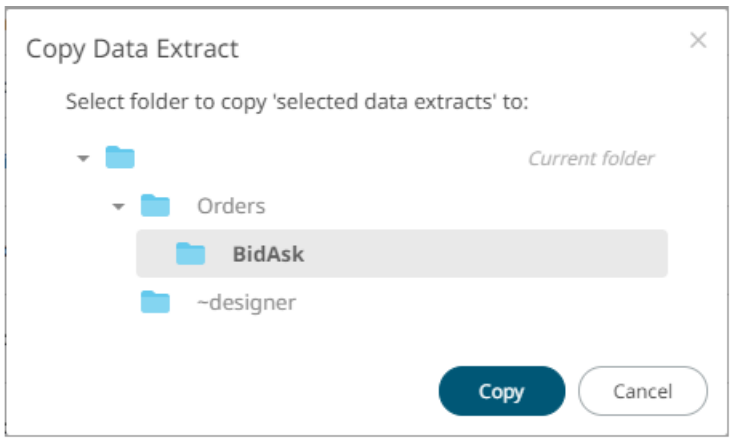
2. Select **Copy** on the *Context Menu*.



The *Copy Data Extract* dialog displays with the folder or subfolders the user is allowed to copy the data extracts to.

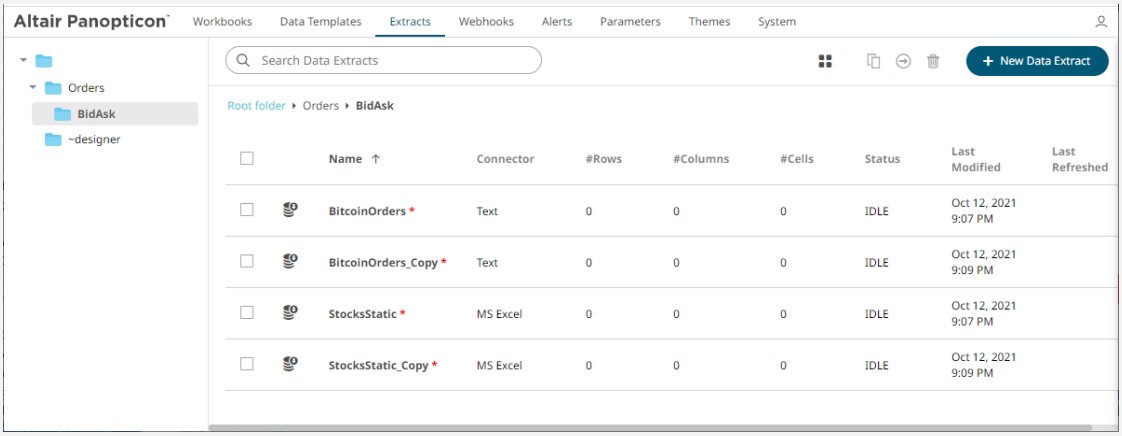


3. Select the folder or subfolder.

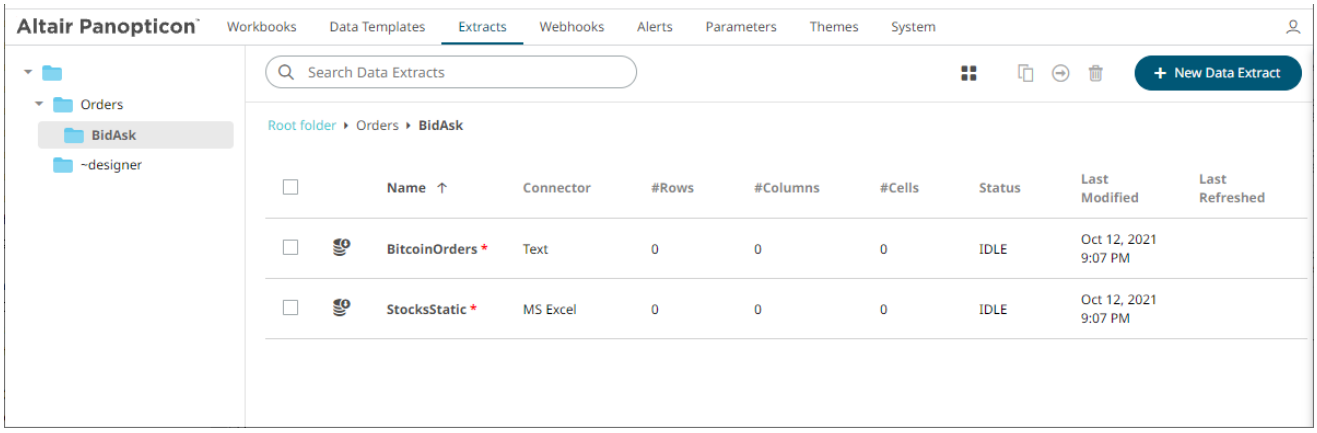


4. Click Copy .

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



The data extracts are copied and displayed on the selected folder.

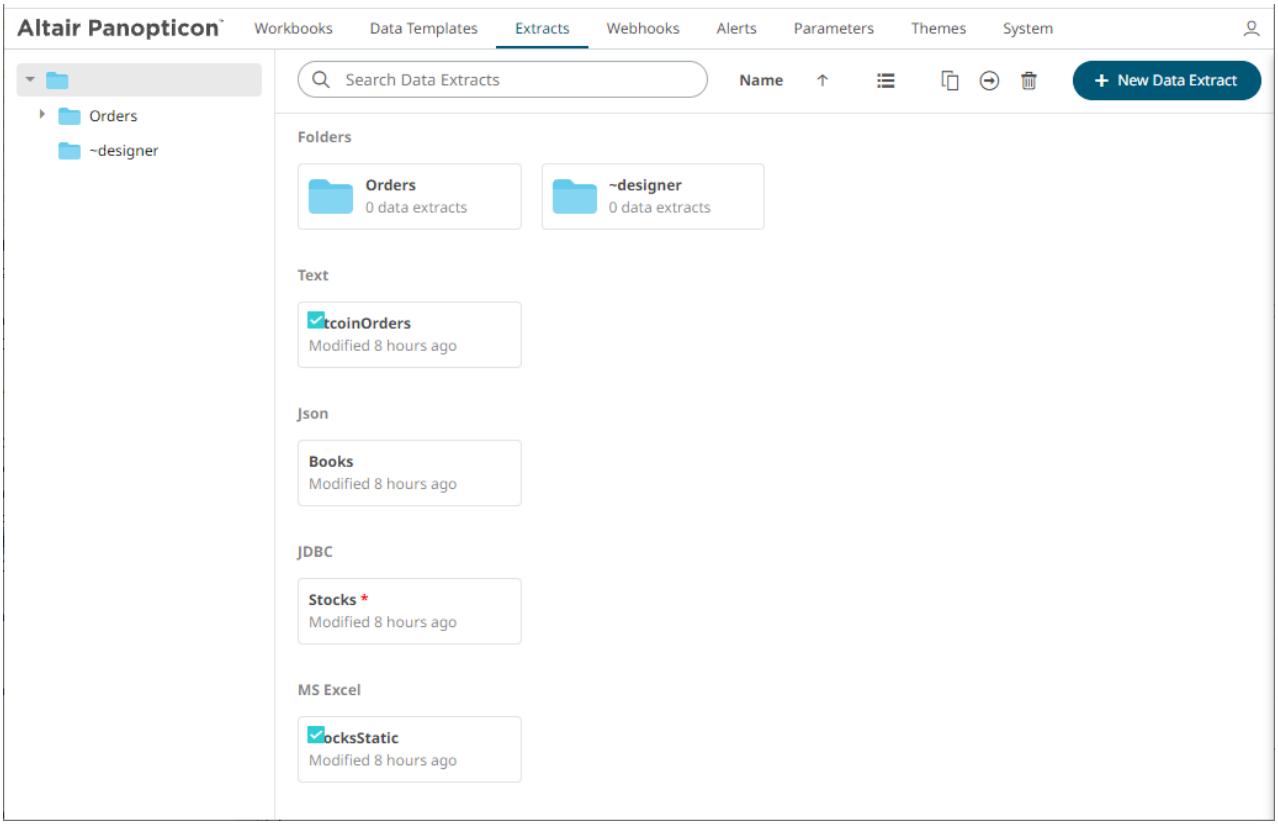


Moving Data Extracts Using the Context Menu

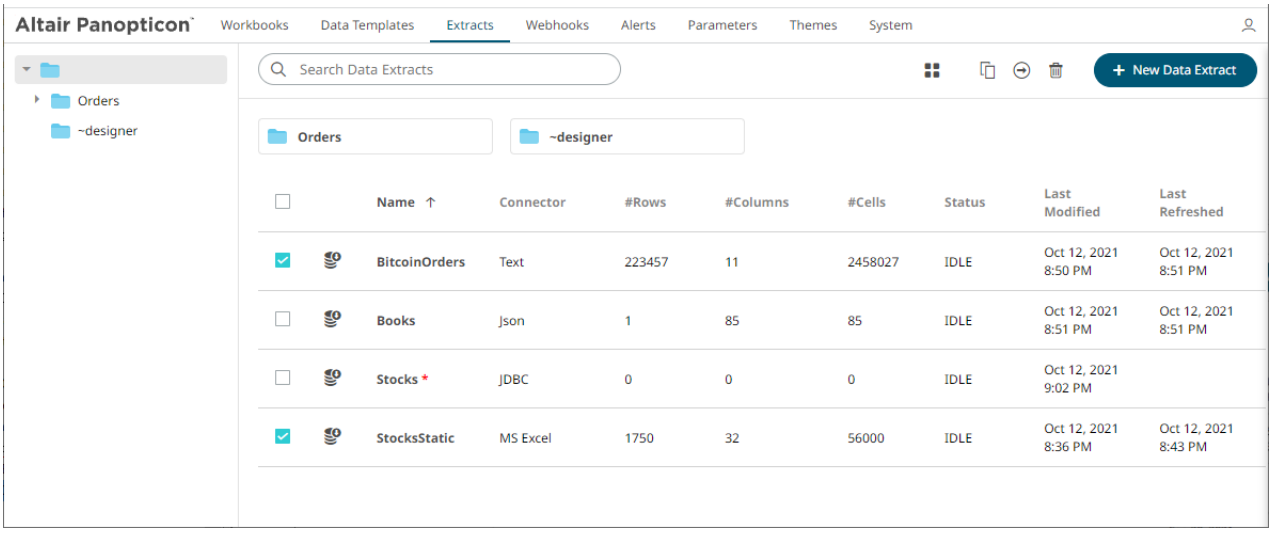
Users with a Designer role are allowed to move data extracts to another folder or subfolder where they have permission

Steps:

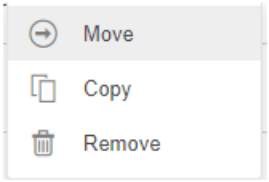
1. Check the boxes of data extracts either:
   - on the *Grid View,* or



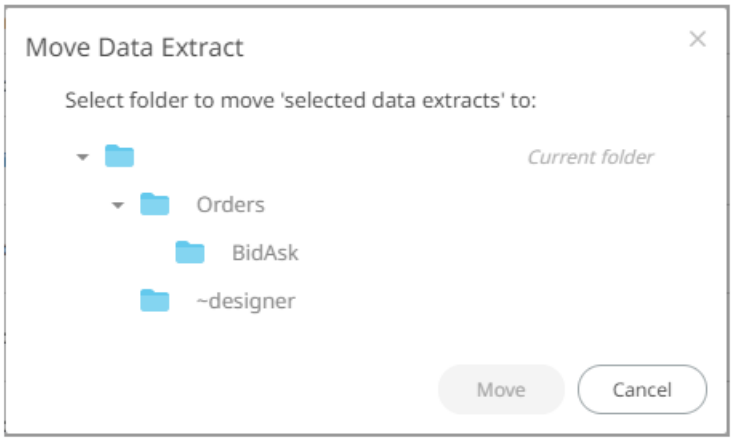
   - on the *List View*



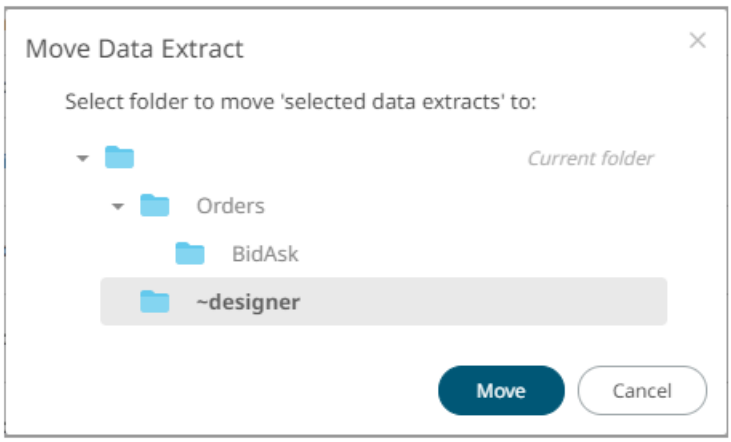
2. Select **Move** on the *Context Menu*.



The *Move Data Extract* dialog displays with the folder or subfolders that the user is allowed to move the data extracts.



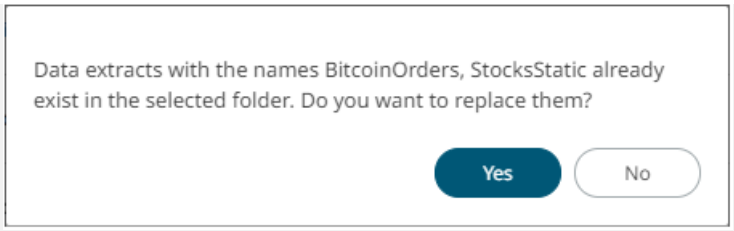
3. Select the folder or subfolder.



4. Click 

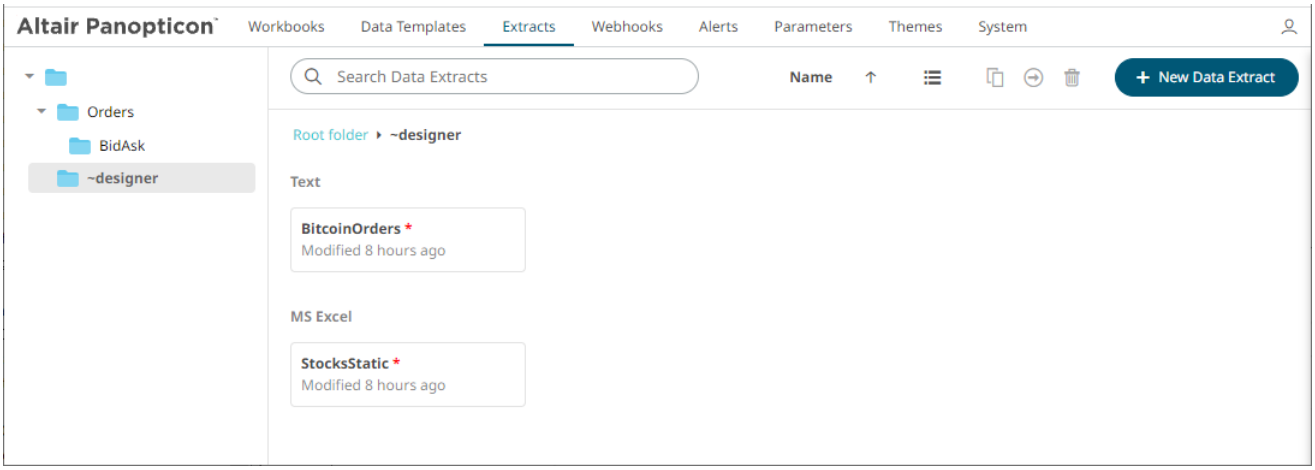
.

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



Click **Yes** to replace a copy of the same data extracts.

The data extracts are moved and displayed on the selected folder.

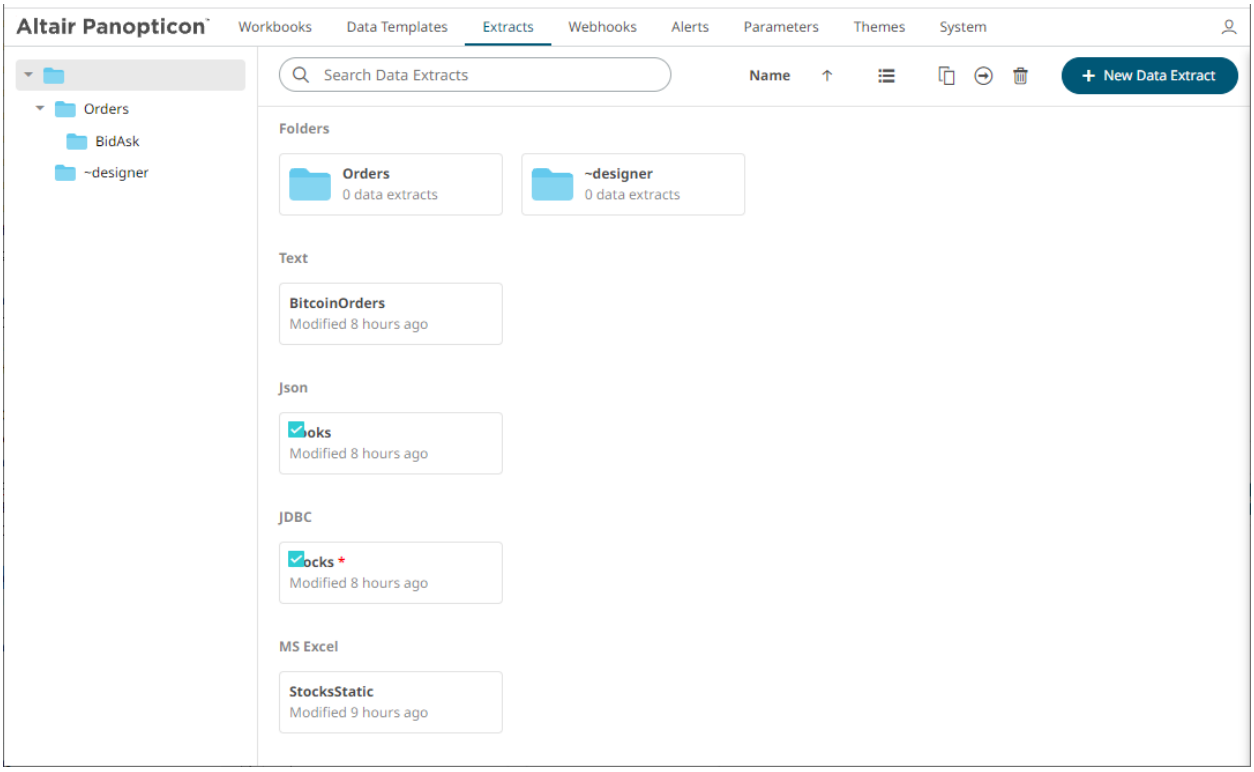


Deleting Data Extracts Using the Context Menu

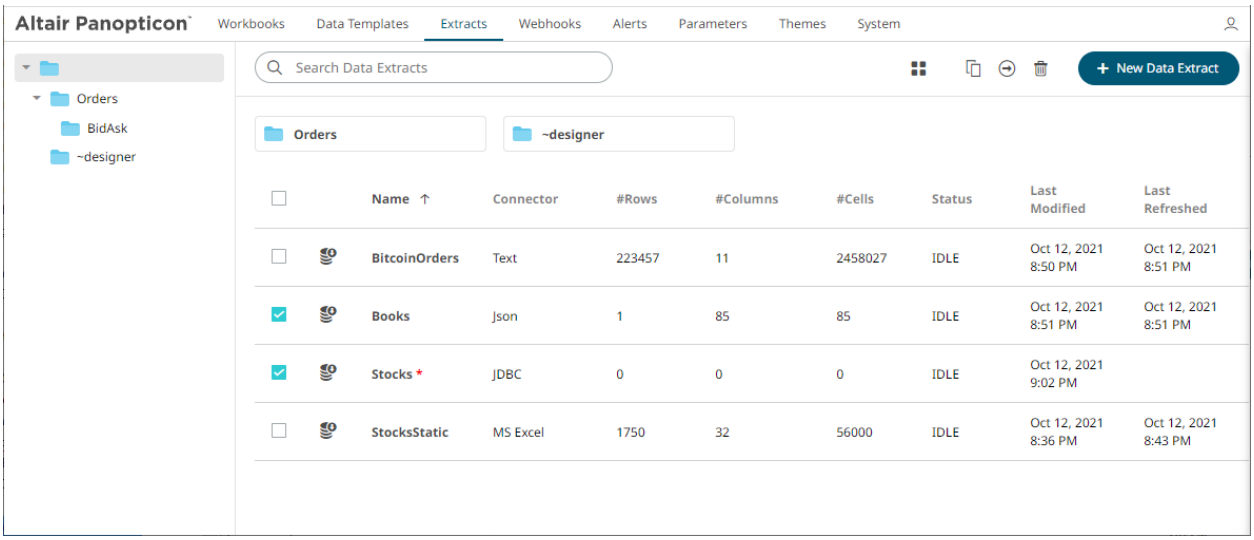
Users with a Designer role have the ability to remove data extracts.

Steps:

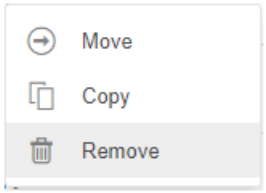
1. Check the boxes of data extracts either:
   - on the *Grid View,* or



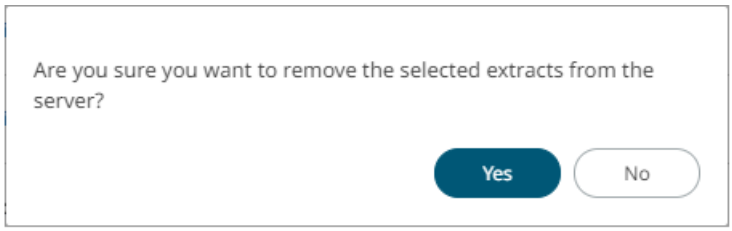
- on the *List View*



2. Click **Remove** on the *Context Menu*.



A notification message displays.



3. Click  to remove.

[12] WEBHOOKS

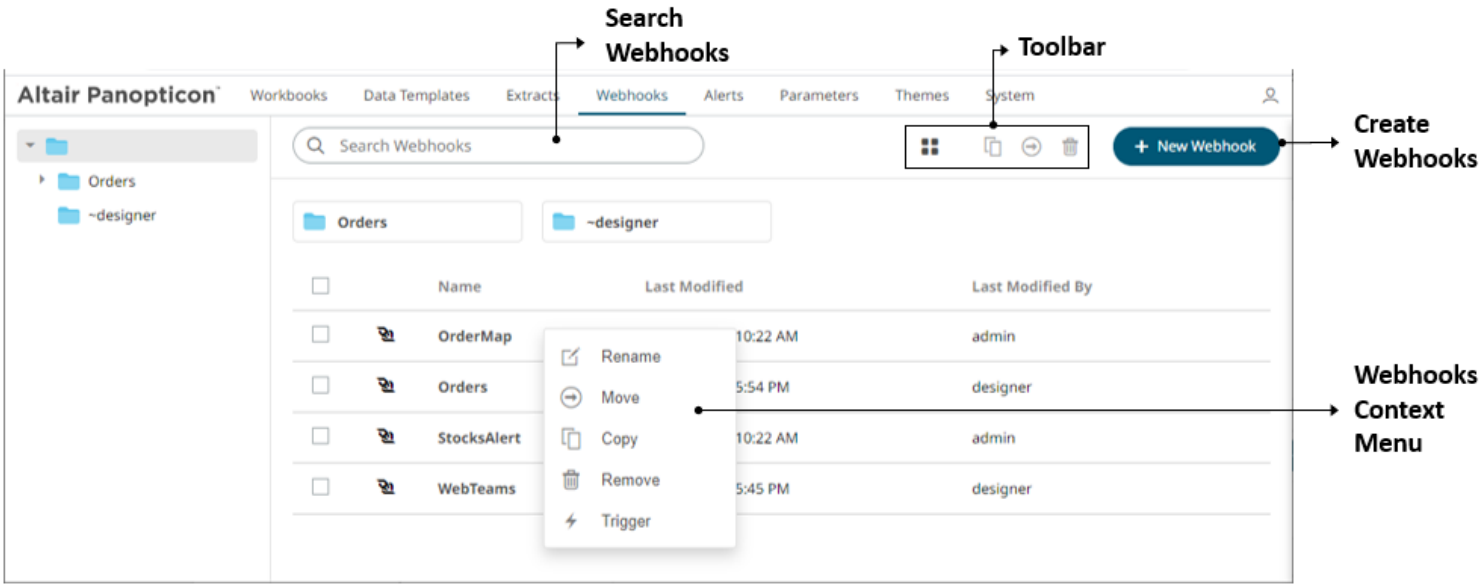
A webhook is a special URL that makes it possible to send a message from other systems into the system that issued the webhook. Webhook URLs should be treated with care and not shared publically, since anyone with knowledge about the webhook URL will be able to use it.

Collaboration platforms such as Microsoft Teams, Slack and many others all have support for creating incoming webooks. In Panopticon, outgoing webhooks can be added (based on incoming webhook URLs from other systems) and used as a channel for sending messages about triggered alerts, similar to how such messages can also be sent by email. Webhooks added to Panopticon are stored in the server folder structure and are subject to the same permissions model as workbooks.

An outgoing webhook in Panopticon can be used as the message channel for multiple different alerts in multiple different workbooks, due to the parameterization of the webhook request body. The exact structure and content that you should create in the request body of a webhook will be specified in the documentation of the system that issued the webhook.

NOTE	Do not expect that the example request body shown below, will work as is.

Starting with version 21.1, users with an Administrator role can create and manage webhooks.



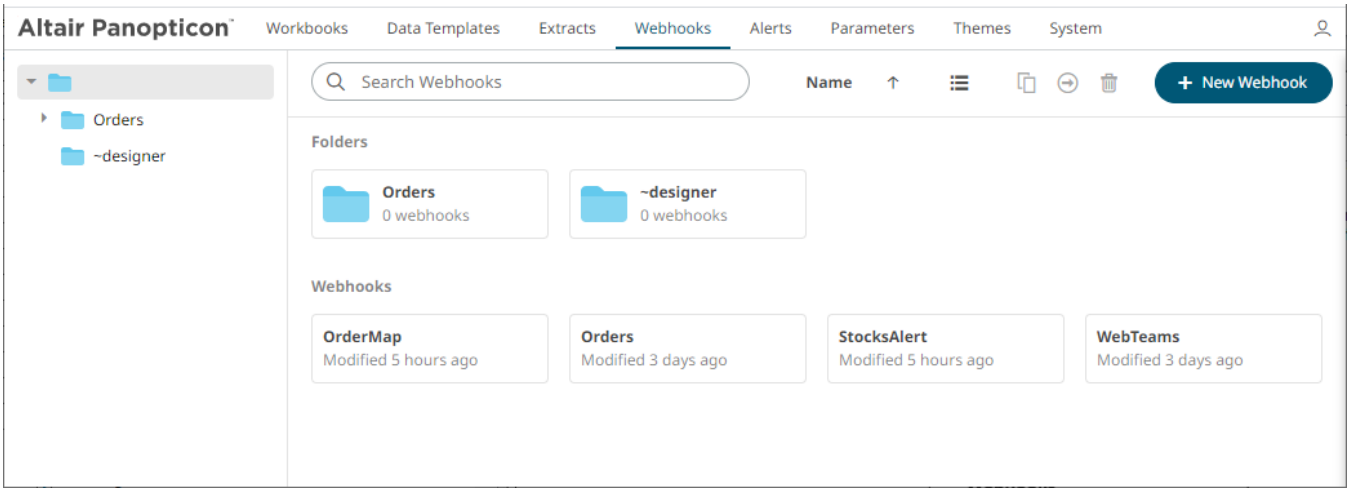
Property	Description
Search Webhooks	Entering text will filter the webhooks.
Toolbar	Allows copying, moving, and removing of webhooks. Also, to display the webhooks list either on *List View* or *Grid View*.
Create Webhooks	Allows creating new webhooks.

Webhooks Context Menu	Allows renaming, moving, copying, deleting, and enabling of the trigger of webhooks.

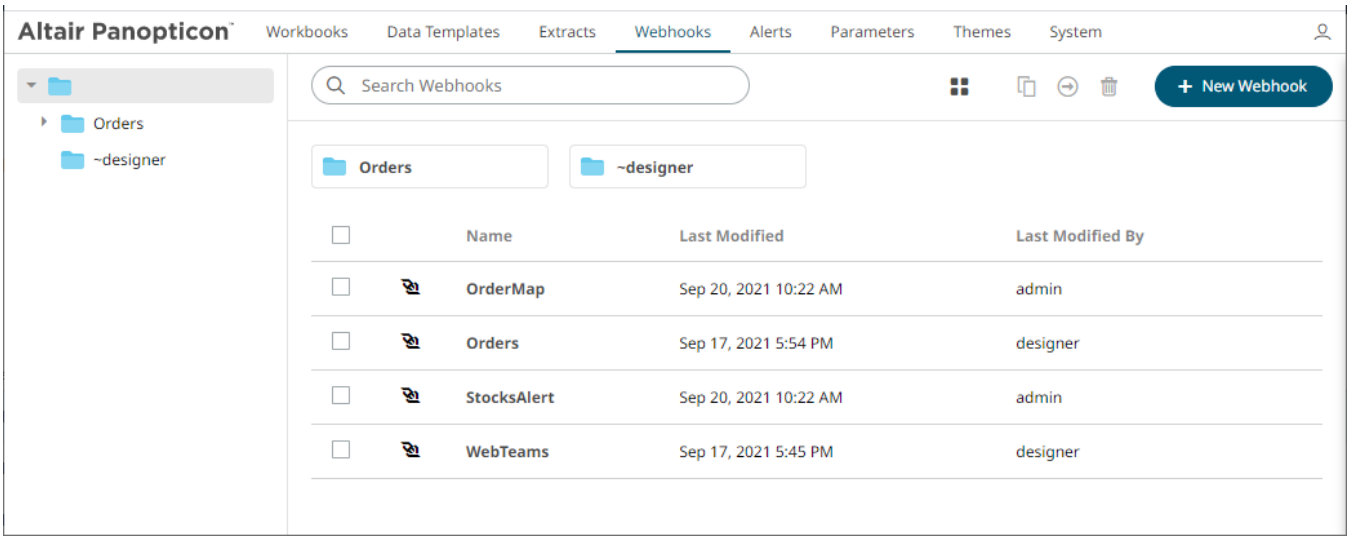
Folders and Webhooks Display View

Webhooks can be displayed either on a *List* or *Grid View*.

On the *Toolbar*, click **Grid View** . The folders and webhooks are displayed as thumbnails.



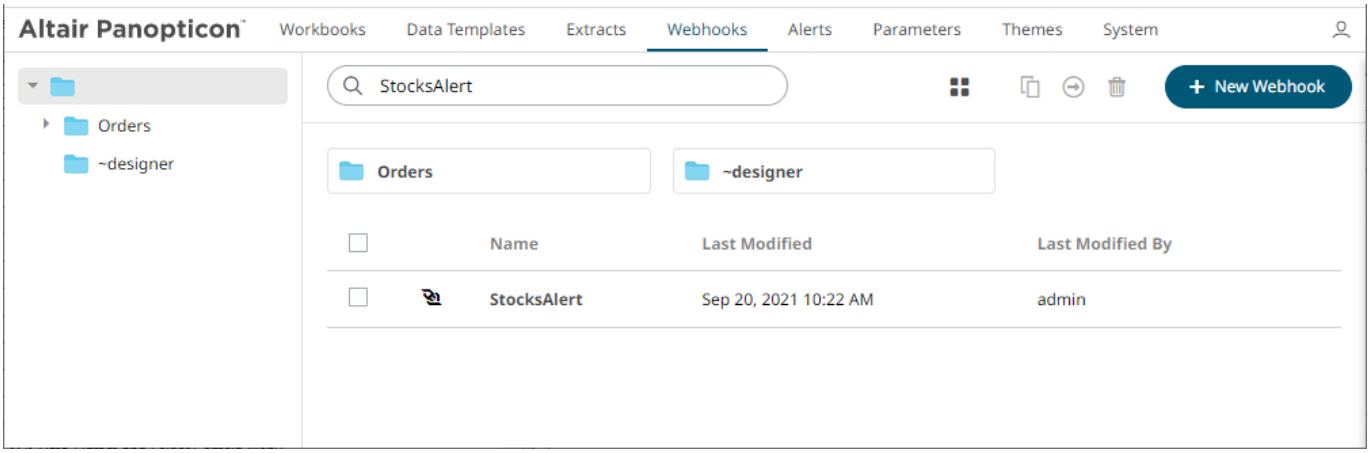
Or click **List View** , the folders and webhooks are displayed in a standard listing.



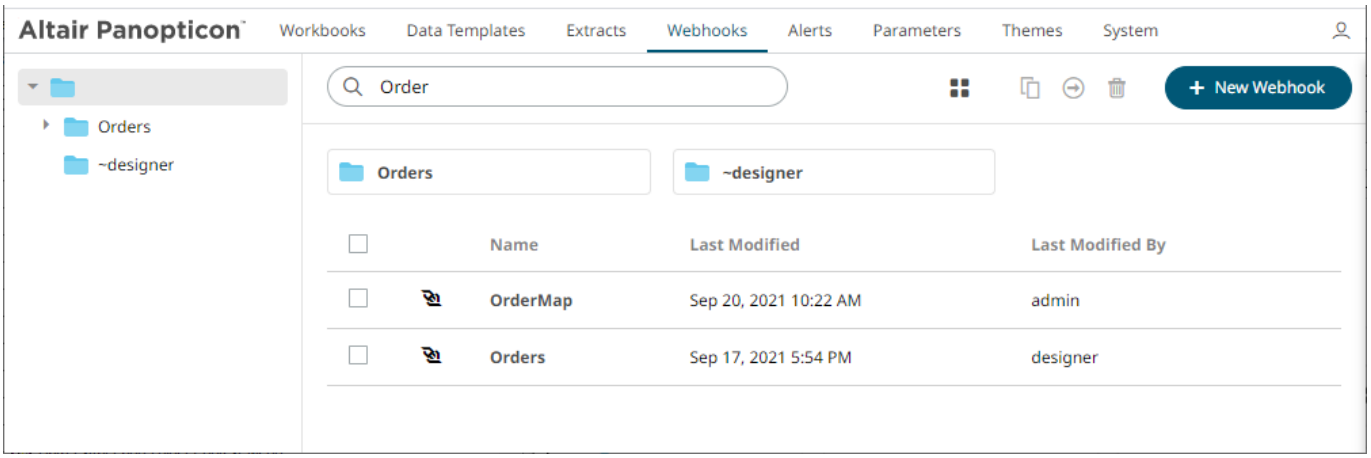
On either display view style, clicking on a webhooks title or thumbnail displays the *Webhooks* page.

Searching for Webhooks

On the *Webhooks* tab, to search for a particular webhook, enter it in the *Search Webhooks* box.



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



Click on a webhooks to open and display.

To clear the filter, delete the text entry in the *Search Webhooks* box.

CREATING WEBHOOKS

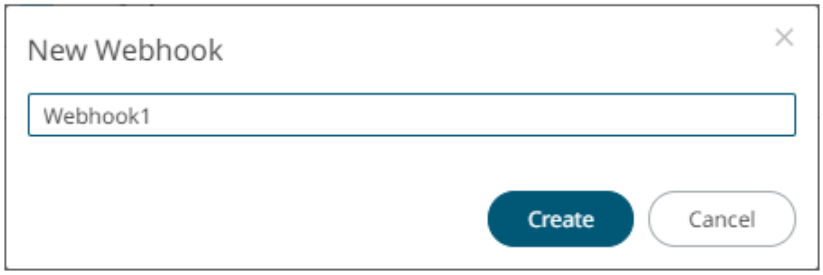
This section discusses the instructions and guidelines to create webhooks.

Steps:

1. On the **Webhooks** tab, click on a folder then 

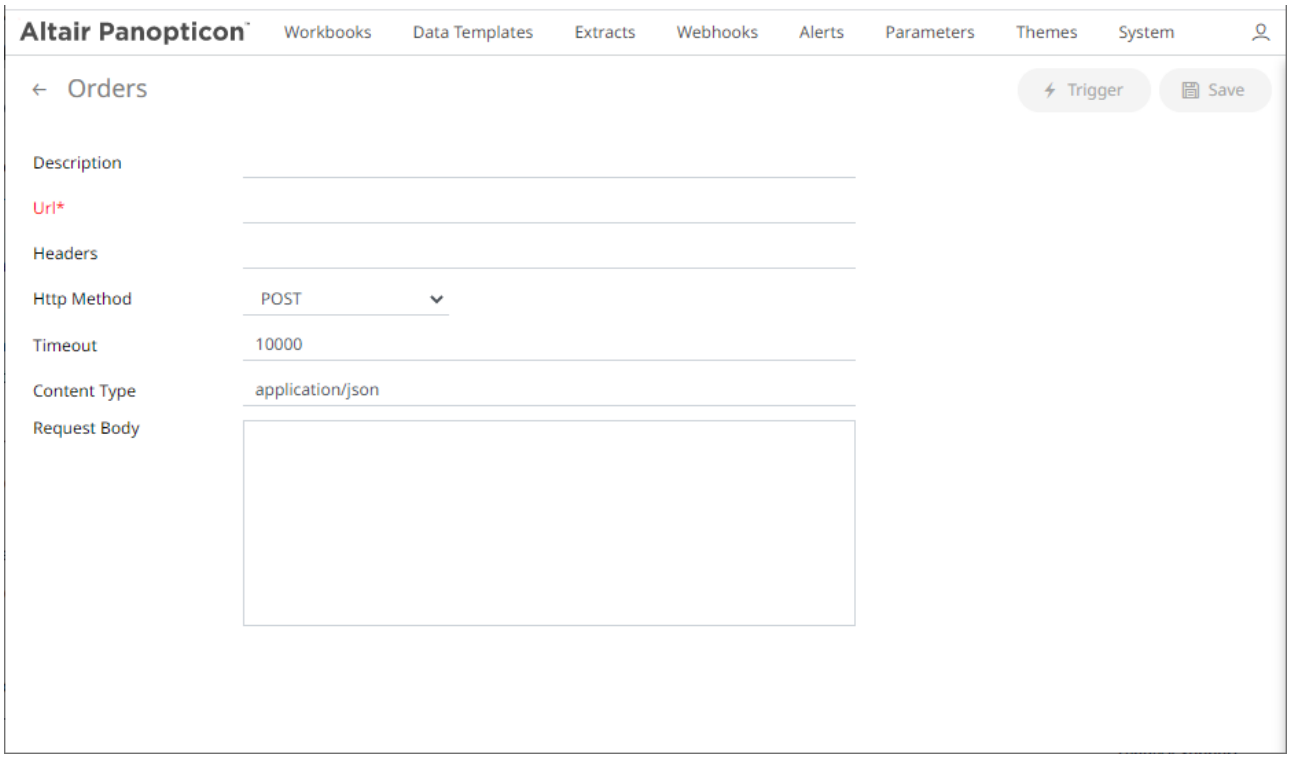
.

   The *New Webhook* dialog displays.



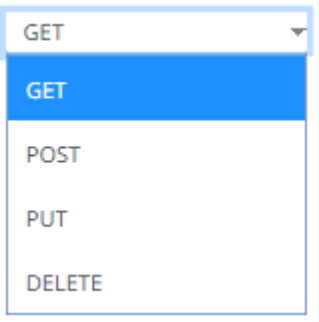
2. Enter the name of the webhook then click Create.

   The new webhook is displayed on the *Webhook* page.



3. Enter or select the following webhook properties:

Property	Description
Description	Description of the webhook.
URL	URL of the webhook. This property is required.
Headers	A comma separated list of name=value pairs representing HTTP headers.
HTTP Method	Select the appropriate HTTP method for the request from the following options:

	 • GET – retrieve data • POST – add new data • PUT – replace existing data • DELETE – remove existing data
Timeout	Timeout (in ms) for reading a response from the URL.
Content Type	The content type of the request body. Default is **application/json.**

Request Body	The request body to be supplied to the HTTP call.
	For example:
	`{`
	`'Alert title': '{_alert_title}',`
	`'Alert dashboard URL': '{_alert_dashboard_url}',`
	`'Alert description': '{_alert_description}',`
	`'Alert reason': '{_alert_reason}',`
	`'Triggering items': '{_alert_triggering_items}',`
	`'Timestamp': '{_current_time}',`
	`'Folder': '{_workbook_folder}',`
	`'Workbook': '{_workbook_name}',`
	`'Dashboard': '{_dashboard_name}'`
	`}`

NOTE	*URL*, *Headers*, and *Request Body* fields can be parameterized (i.e., special server parameters, alert parameters, and global parameters).

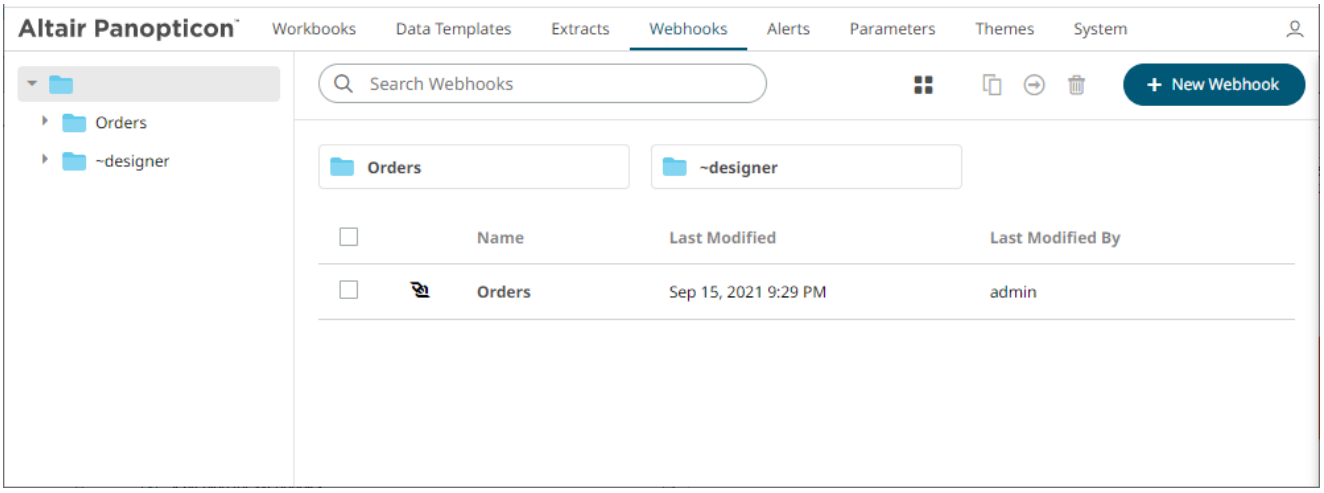
4. Click  to save the new webhook.

5. You may opt to click  to trigger the webhook. Any parameter in the request body will be replaced by its value when triggering the webhook request.

   For example:

   `{_current_time} - 2021-07-01T12:34:56Z`

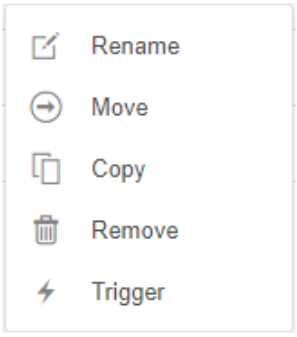
6. Click  to go back to the *Folders and Webhooks* list. The new webhook is added on the list.



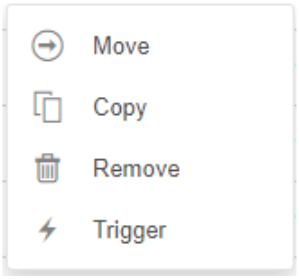
WEBHOOKS TOOLBAR AND CONTEXT MENU

Moving, copying, and removing webhooks can either be done using:

- Context menu



Webhook Context Menu



Webhooks Folder Context Menu

- Toolbar

List View

Grid View

The *Webhooks* toolbar options include:

Toolbar Option	Description
Sort By / Sort Order	Allows sorting webhooks by *Name, Last Modified,* or *Last Modified By*.
Display View	Display webhooks either by *List View* or *Grid View*.

Copy	Copy webhooks to another folder or subfolder where the user has permission.
Move	Move webhooks to another folder or subfolder where the user has permission.
Remove	Remove webhooks.

The *Context Menu* options include:

Toolbar Option	Description
Rename	Rename the webhook.
Move	Move webhooks to another folder or subfolder where the user has permission.
Copy	Copy webhooks to another folder or subfolder where the user has permission.
Remove	Remove webhooks.
Trigger	Trigger the webhook.

Sorting Webhooks

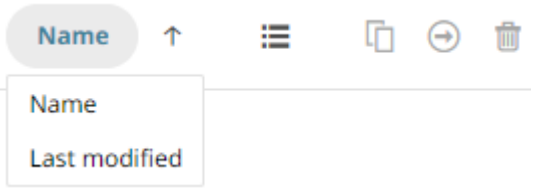
Sorting webhooks can be done by **Name, Last Modified**, or **Last Modified By**.

Steps:

On the *Webhooks* tab, either:

- click the **Sort By** option on the *Toolbar* of the *Grid View.*

  By default, the sorting is by **Name**.

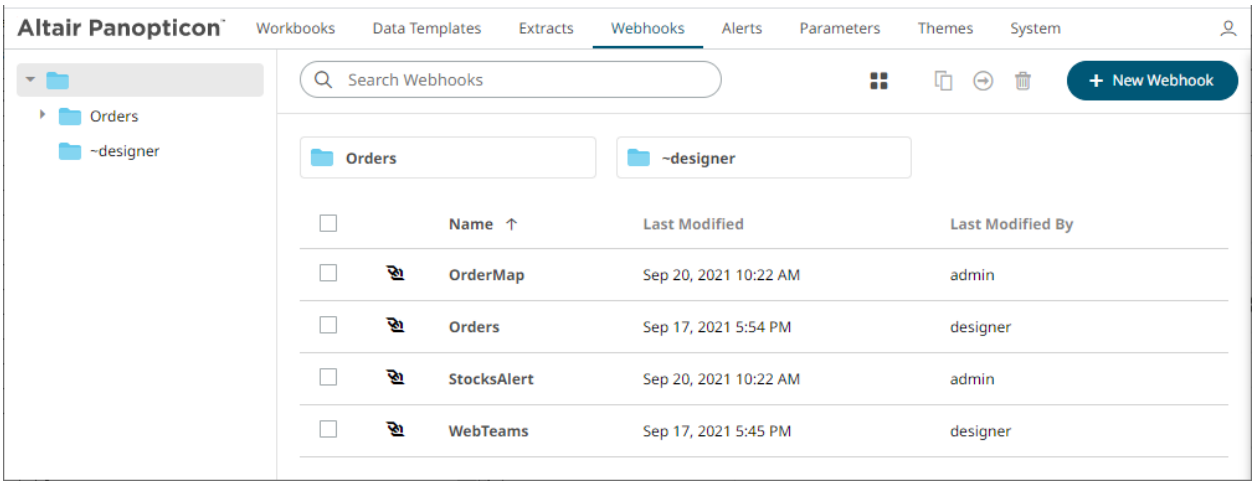
  

  - Name
  - Last Modified

  Then click the *Sort Order*:

  - Ascending
  - Descending

- click on the **Name**, **Last Modified**, or **Last Modified By** column header of the *List View.*



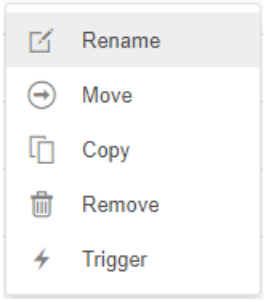
Then click the *Sort Order*:

- Ascending
- Descending

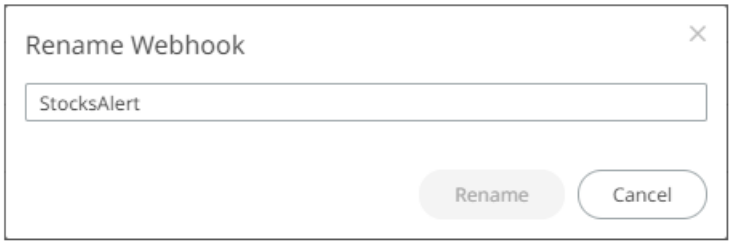
Renaming a Webhook

Steps:

1. Right-click on a webhook then select **Rename** on the context menu.



The *Rename Webhook* dialog displays.



2. Enter a new name then click 

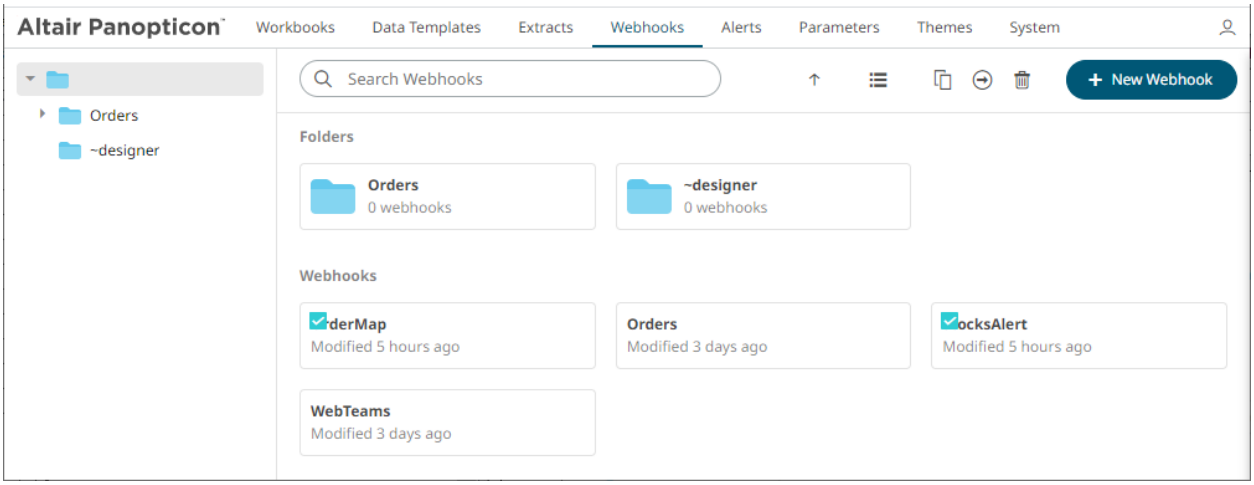
.

Moving Webhooks

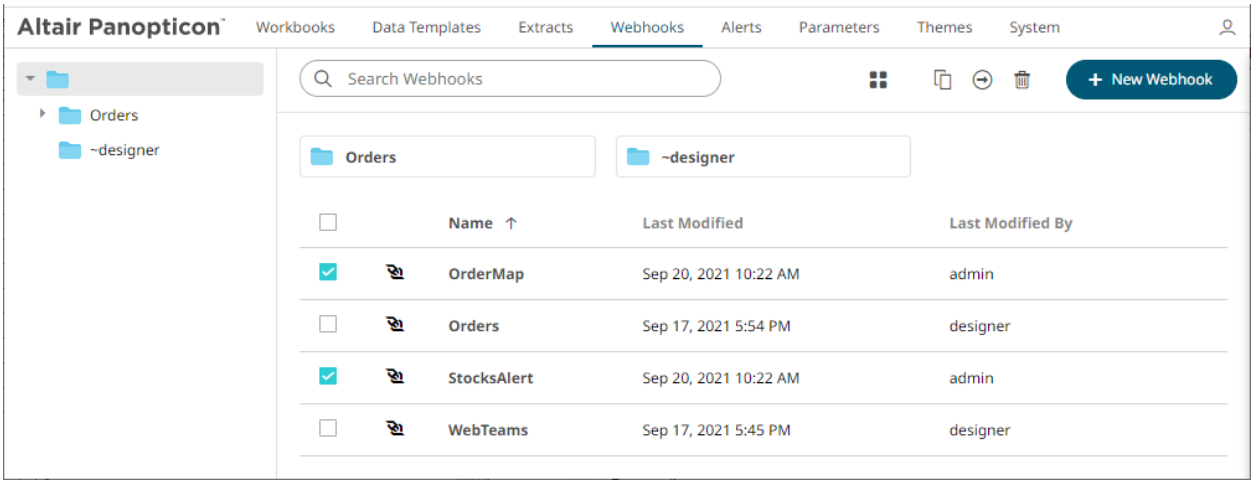
Users with an Administrator role are allowed to move webhooks to another folder or subfolder where they have permission.

Steps:

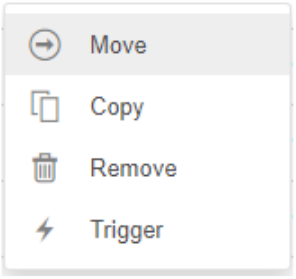
1. Check the box of one or several webhooks either:
   - on the *Grid View,* or



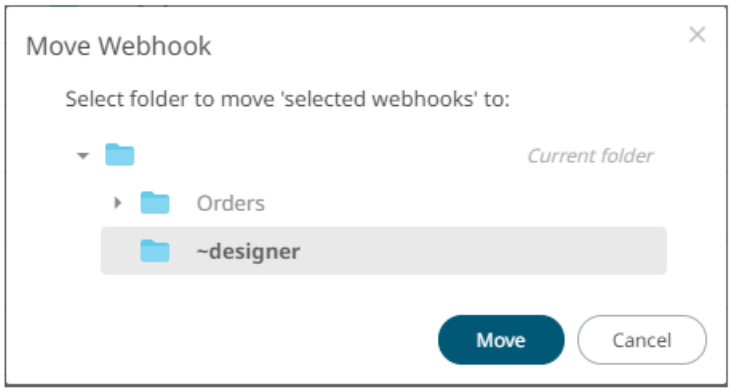
   - on the *List View*



2. Then select either:
   - the **Move**  icon on the toolbar
   - **Move** on the content menu



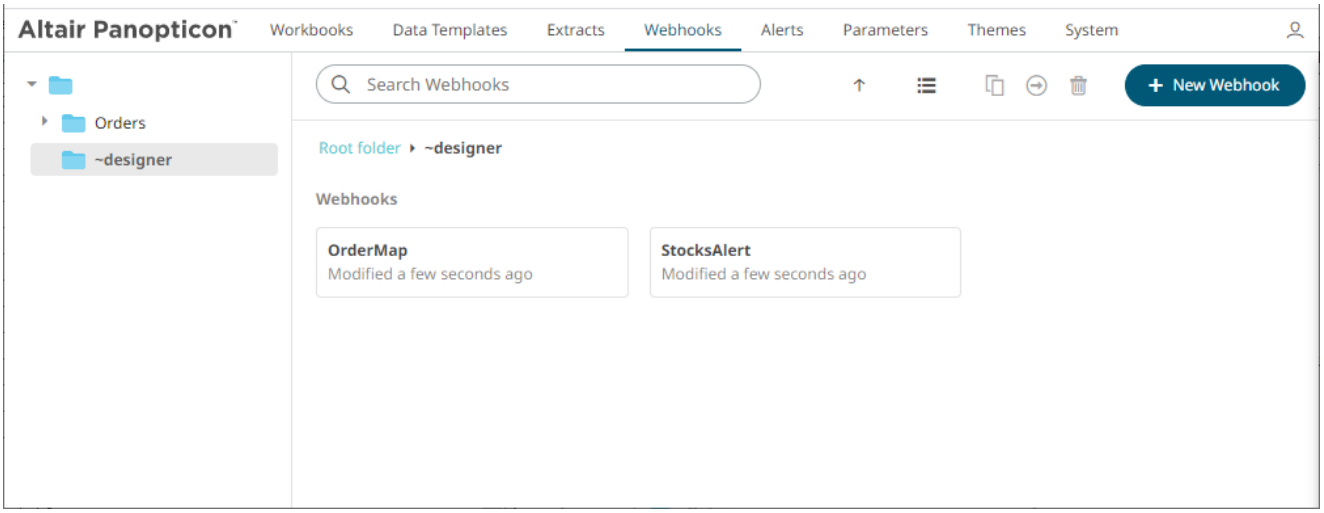
The *Move Webhook* dialog displays with the folder or subfolders that the user is allowed to move the webhooks. Select the folder or subfolder.



3. Click 

.

The webhooks are moved and displayed on the selected folder.

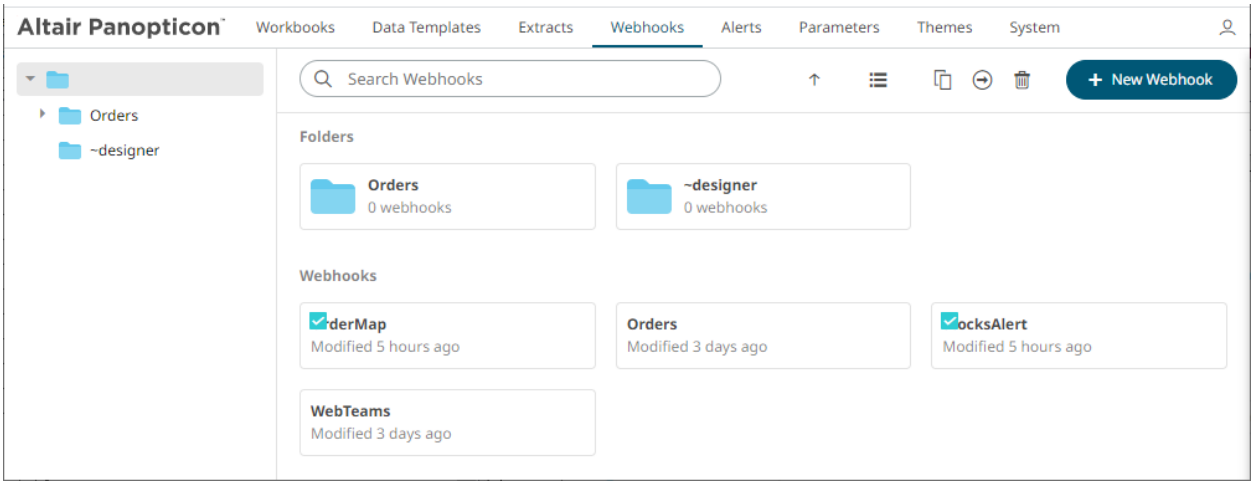


Copying Webhooks

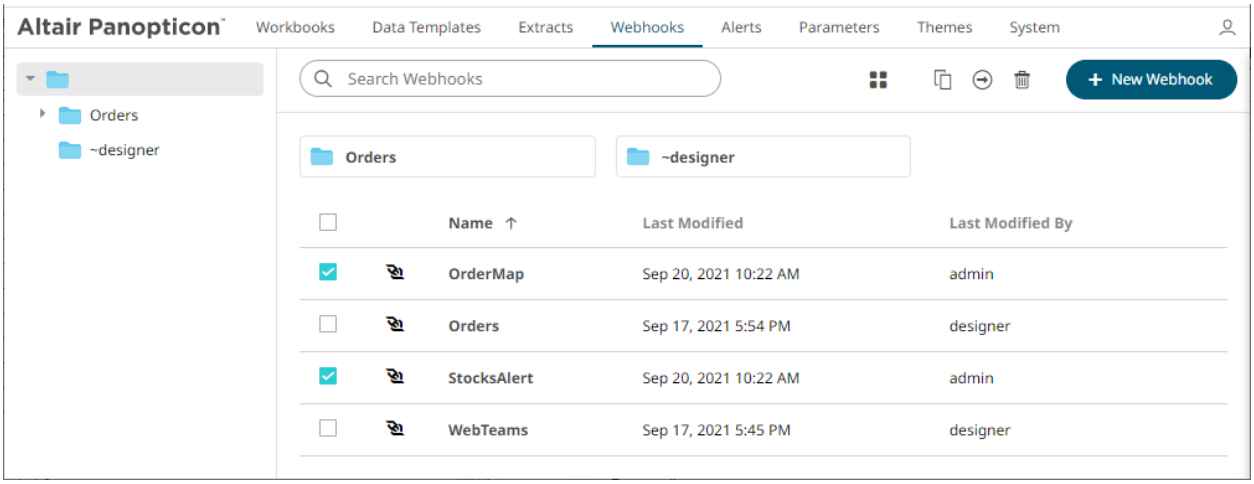
Users with an Administrator role are allowed to copy webhooks to another folder or subfolder where they have permission.

Steps:

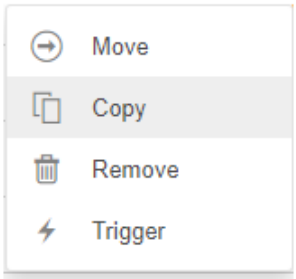
1. Check the box of one or several webhooks either:
   - on the *Grid View,* or



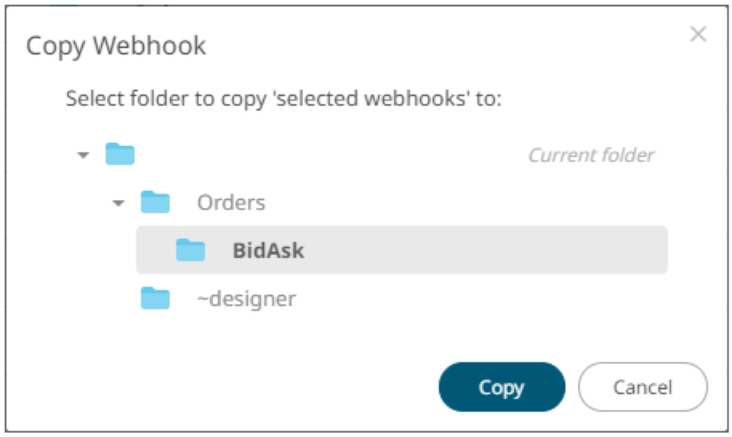
   - on the *List View*



2. Then select either:
   - the **Copy**  icon on the toolbar
   - **Copy** on the content menu

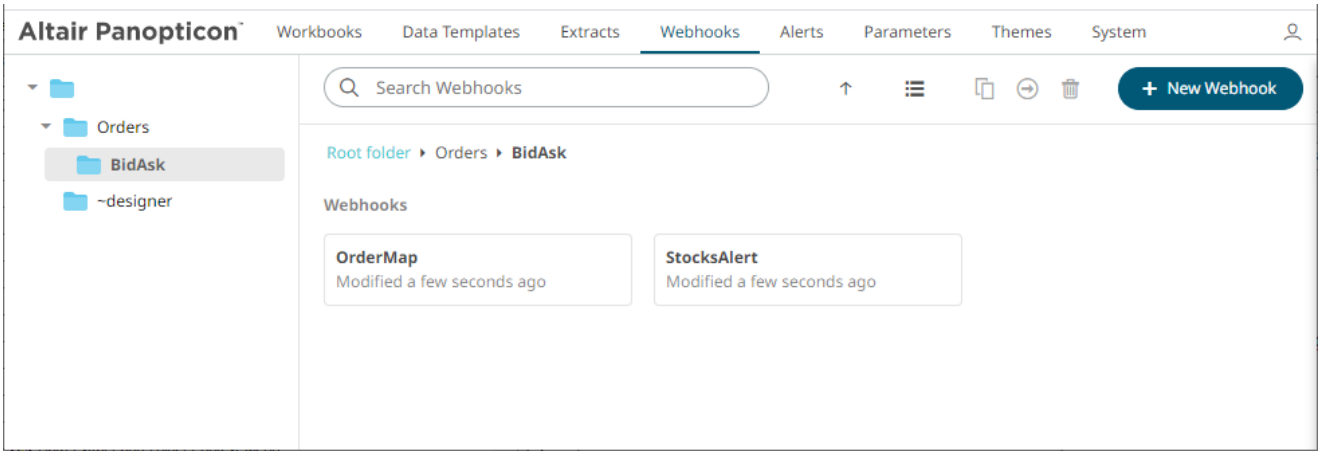


The *Copy Webhook* dialog displays with the folder or subfolders the user is allowed to copy the webhooks to. Select the folder or subfolder.



3. Click Copy .

   The webhooks are copied and displayed on the selected folder.

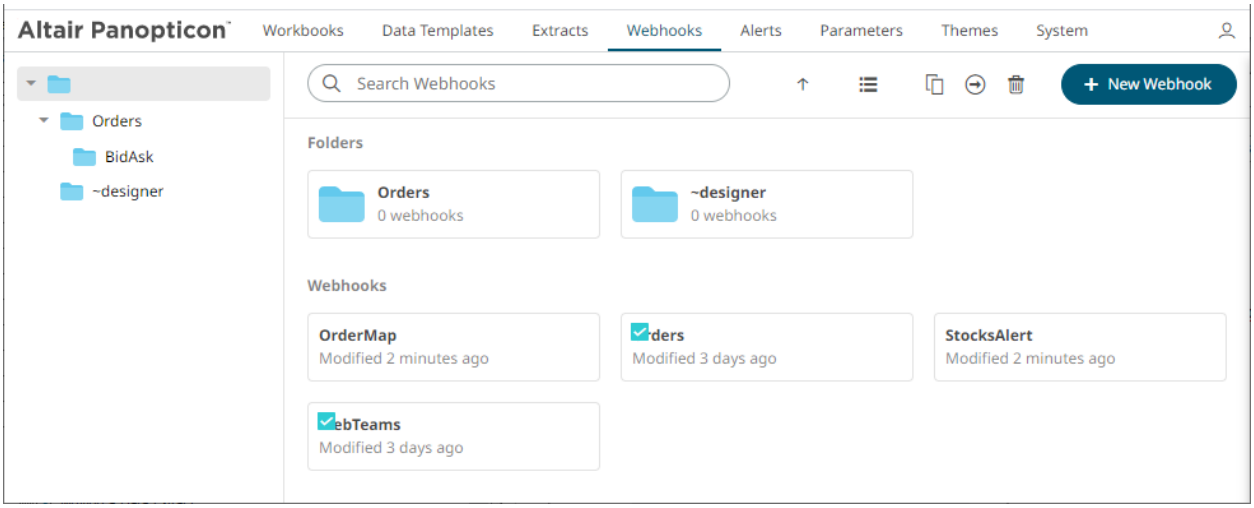


Deleting Webhooks

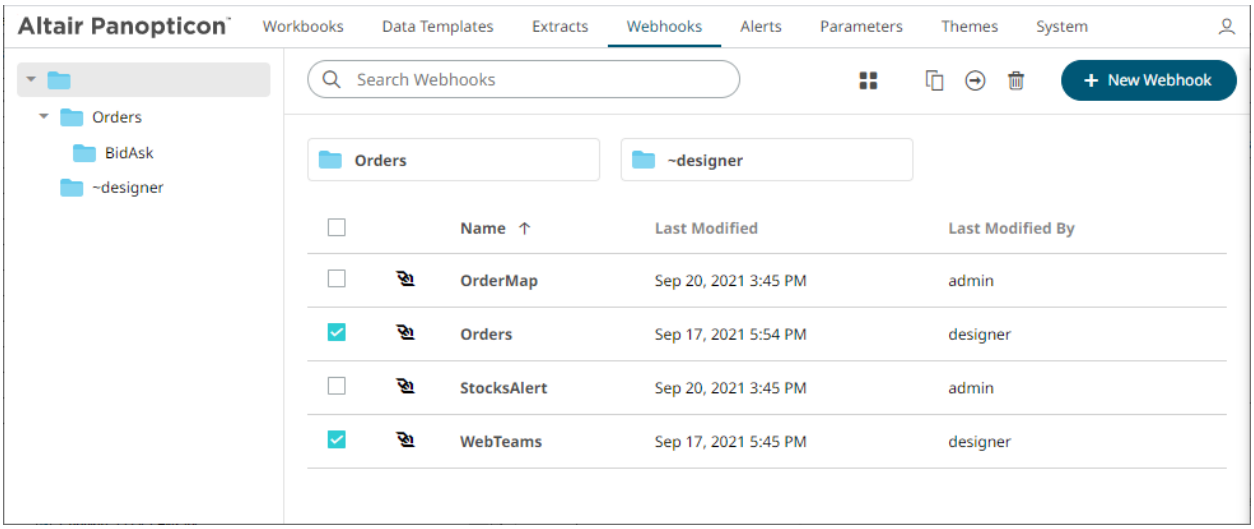
Users with an Administator role have the ability to remove webhooks.

Steps:

1. Check the box of one or several webhooks either:
   - on the *Grid View,* or

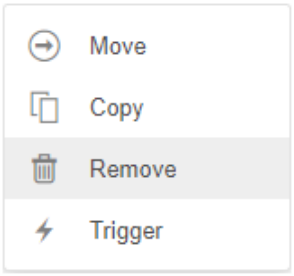


- on the *List View*

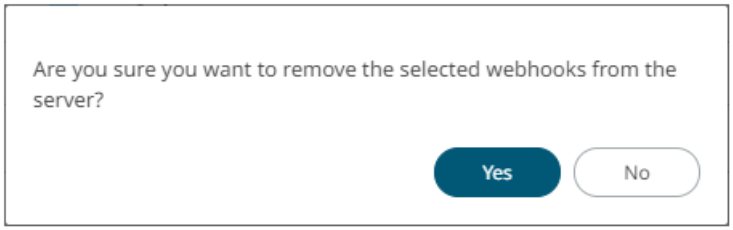


2. Then select either:

   - the **Remove** icon on the toolbar
   - **Remove** on the content menu



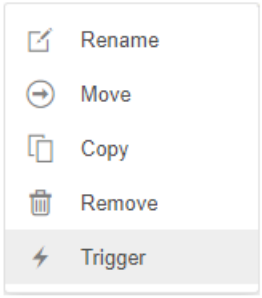
A notification message displays.



3. Click Yes to remove.

Triggering Webhooks

To trigger a webhook, right-click on it and select **Trigger** on the context menu.



Any parameter in the request body will be replaced by its value when triggering the webhook request.

For example:

```
{_current_time} - 2021-07-01T12:34:56Z
```

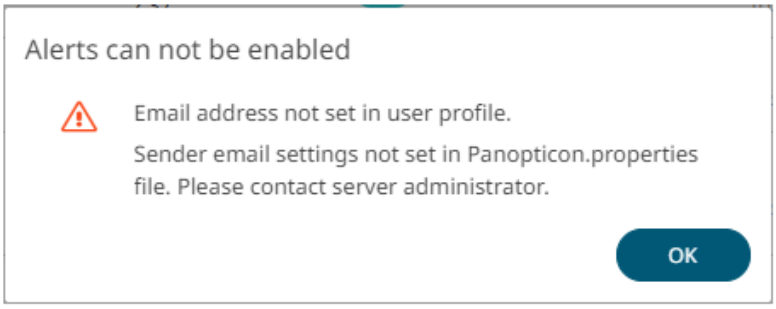
[13] ALERTING

The Alerts function allows a notification to be sent when the data in a visualization has met the predefined settings.

If alerts are required to be sent via email, Panopticon Real Time must be configured with valid email server information in the `Panopticon.properties` file located in the `AppData` folder (e.g., `c:\vizserverdata`).

See Panopticon Real Time Configurations for Email Send Outs and Alerts for instructions.

Otherwise, when trying to enable an alert, this error will be displayed:



In addition, you can also set the `alert.creation.only.by.administrators property` to **true** for only the Administrators to create alerts.

NOTE	Alerts are not supported in the Combination Graphs.

SETTING UP ALERTS

Alerts can be defined against:

- Streaming data sources (including CEP Engines and message queues)
- Periodically refreshed data sources (like Oracle, SAP Sybase, SQL Server, and so on)

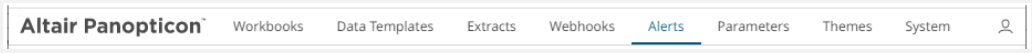
Alert definition can be done by right-clicking on a streaming numeric or text data in a visualization in the Web Client and setting the limits, duration, what will be included, how many and when an email will be sent.

NOTE

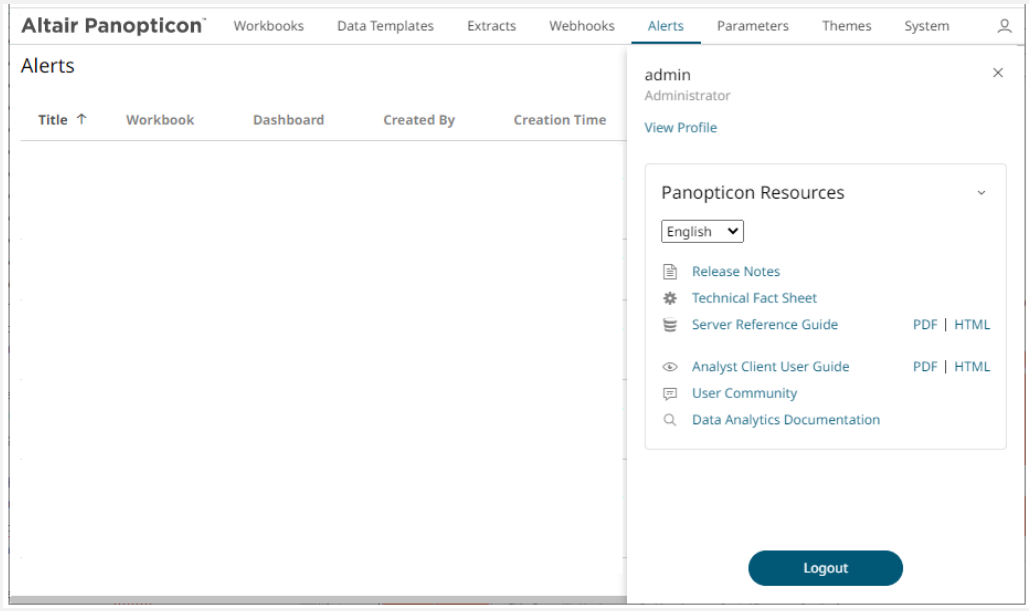
Before setting up the visualization alert, enter the email of the user or group who will receive the alert on the *User Profile*:

Steps:

1. On the toolbar, click  .

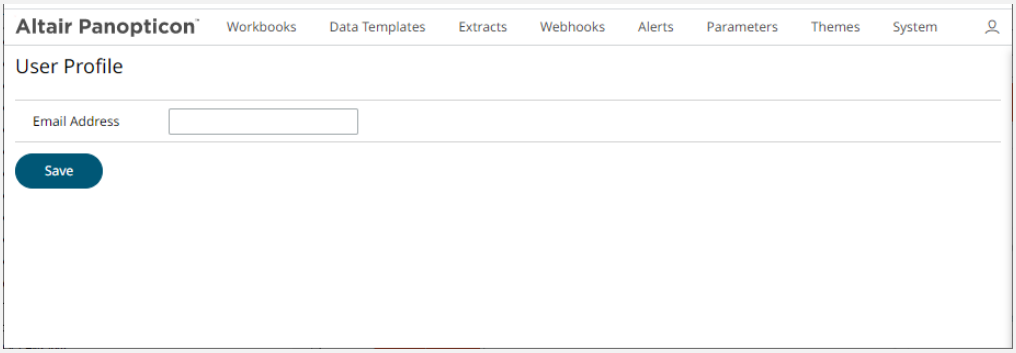


The *Profile* pane displays with the name of the user and the role.



2. Click *View Profile*.

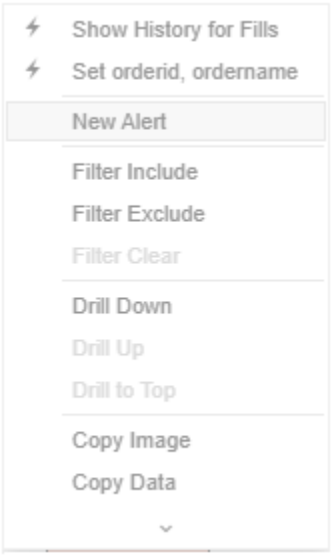
The *User Profile* page displays.



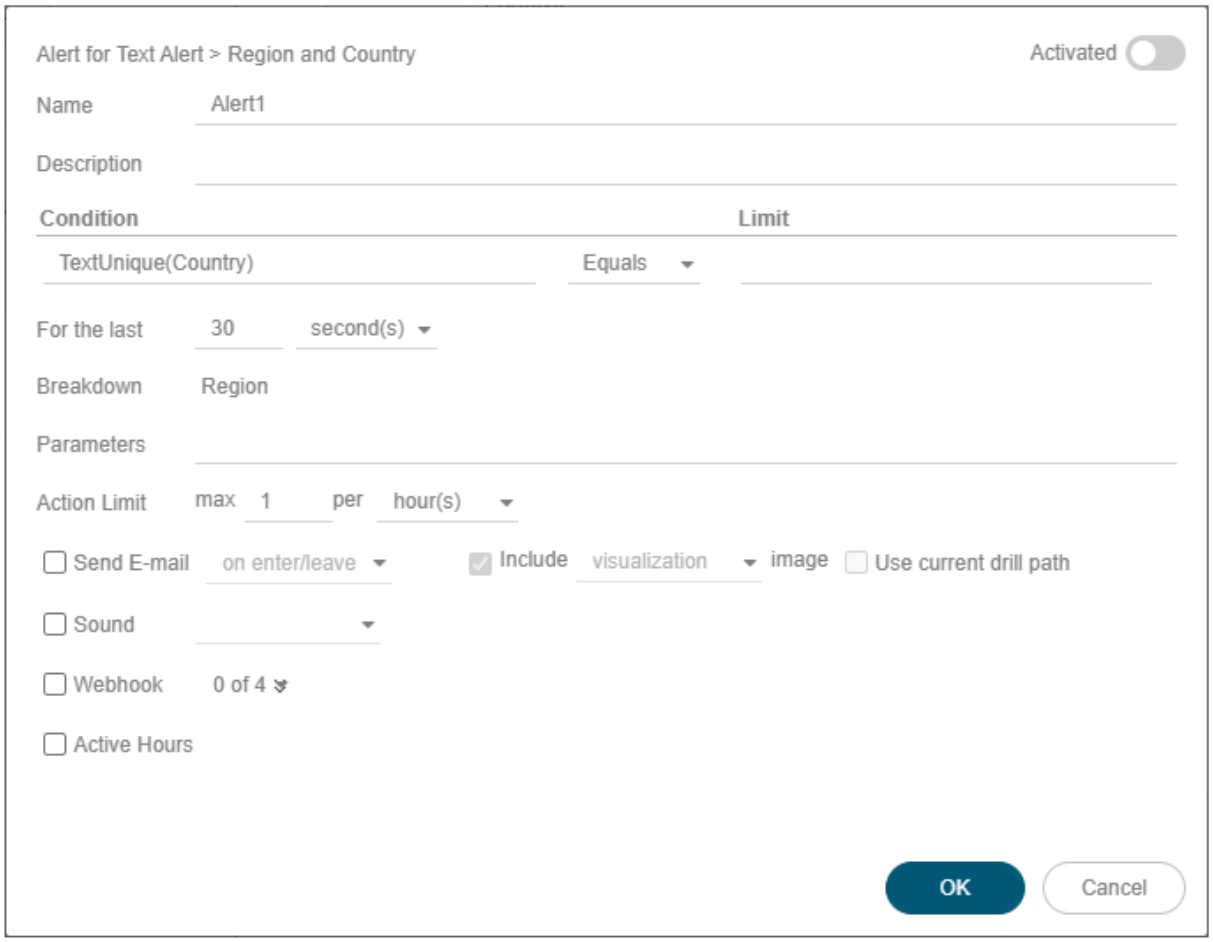
3. Enter the *Email Address*.
4. Click  .

Steps:

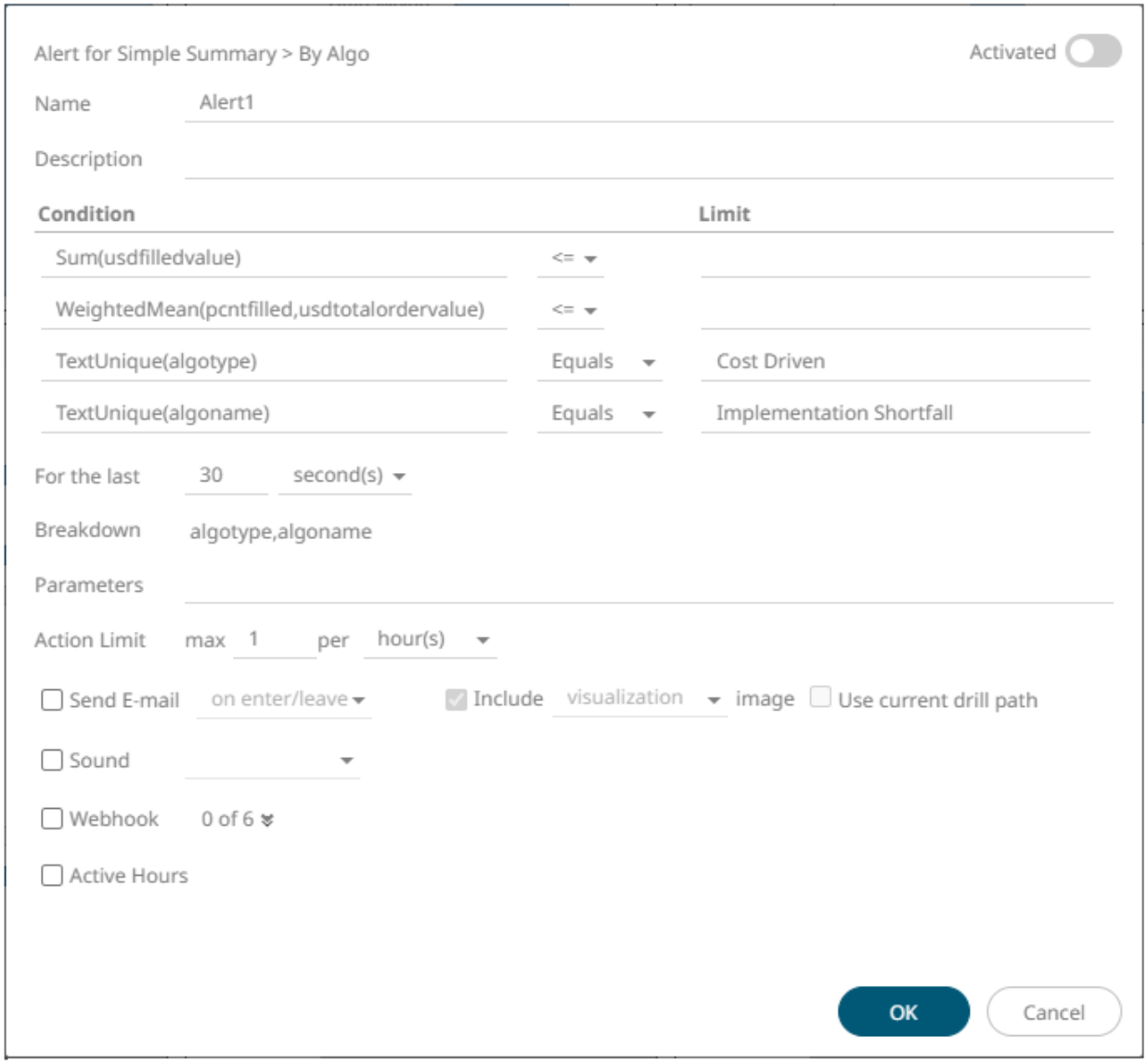
1. Open a workbook on the Web client and right-click on a streaming numeric or text data in a visualization. Select *New Alert* on the context menu.



The *Alerts* dialog displays with the name of the visualization where the alert will be set.



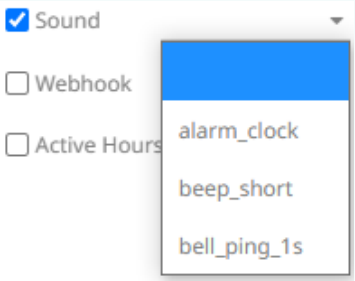
Sample Text Alerting



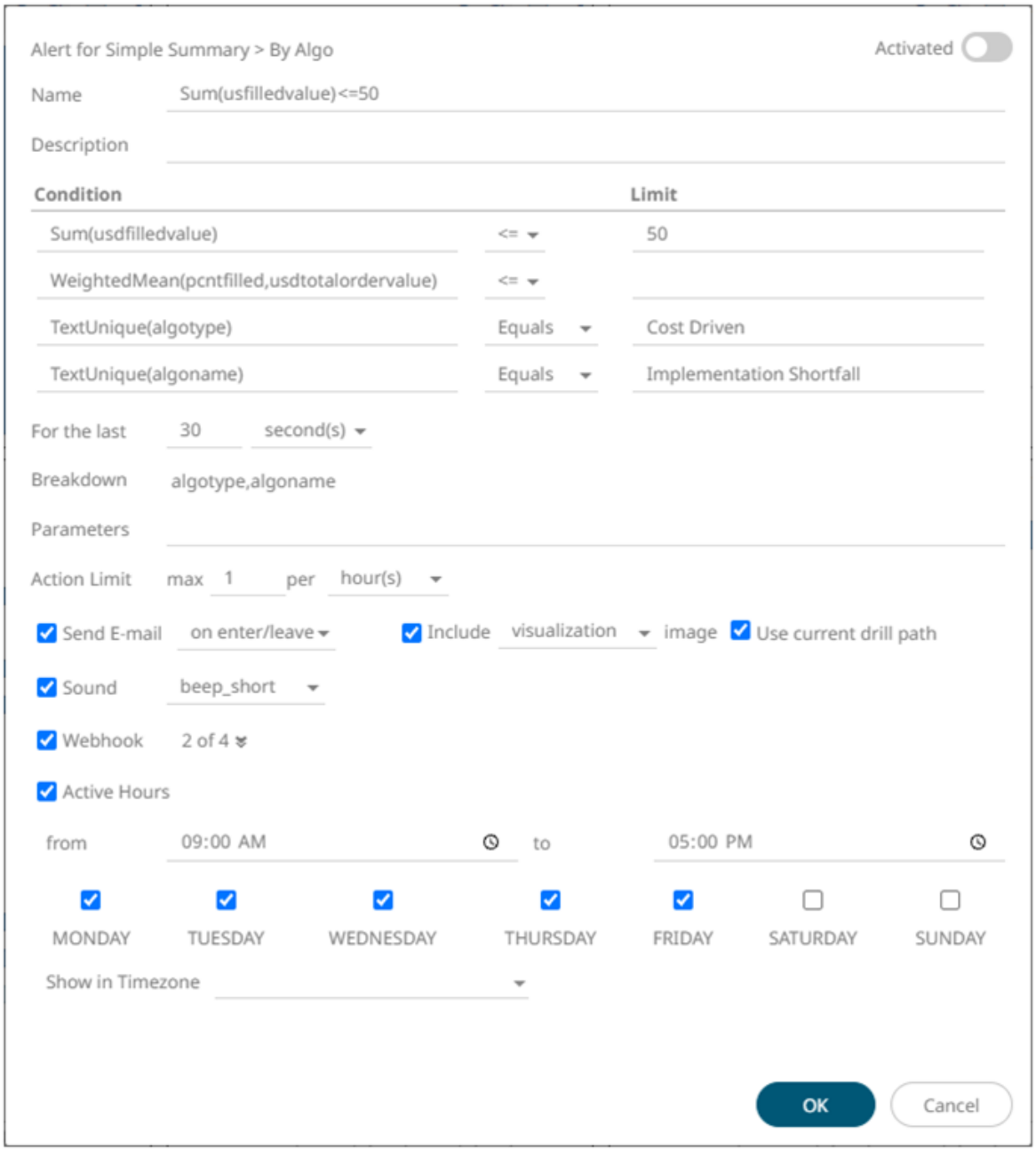
Sample Numeric Alerting

2. Enter or select the following properties:

Property	Description
Name	Name of the alert.
Description	Description of the alert.
Condition	Allows setting the Upper (<= or Lower (>=) *Limit* of all the available numeric variables in the visualization. For text variables, there are three types of conditions: • Equals - The string is equal to another string, e.g., Country=Sweden • Wildcard: The string matches a wildcard expression, e.g., Country=Norwa* would match Country=Norway

	• Regex: The string matches a regex expression, e.g., Country=I[a-zA-Z]+a would match Country=India and Country=Indonesia
For the Last	Checks if a value has reached the limit on the set Date/Time unit: • second(s) • minute(s) • hour(s) • day(s)
Breakdown	Current breakdown of the visualization.
Parameters	Available parameters in the visualization.
Action Limit	The maximum number of times an alert will be sent on the set Date/Time unit: • second(s) • minute(s) • hour(s) • day(s)
Send E-mail	Determines when an alert email will be sent: • on enter • on leave • on enter/leave If unchecked, the notification will only be displayed on the Web client.
Include	Determines whether the image of the visualization or dashboard will be included in the alert email. For the included image of the visualization, check the **Use current drill path** box to generate a drilled image in the email.
Sound	The sound that will be played for a triggered alert. The available sounds are mp3 files placed in the `AppData/Sounds` folder (i.e., `C:\vizserverdata\Sounds`). Panopticon is shipped with one sound (i.e., bell_ping_1s.mps).  Default is **None**.
Webhook	Webhooks that will executed when the alert is triggered.
Active Hours	Determines when an alert should be active. Proceed to step 3.

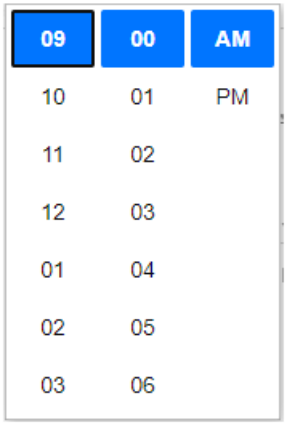
3. Check the *Alert Hours* box. The dialog changes to display:



By default, the duration is from **9:00 AM** to **5:00 AM** on **Monday**, **Tuesday**, **Wednesday**, **Thursday**, and **Friday**.

4. To modify the *Active Hours*, click .

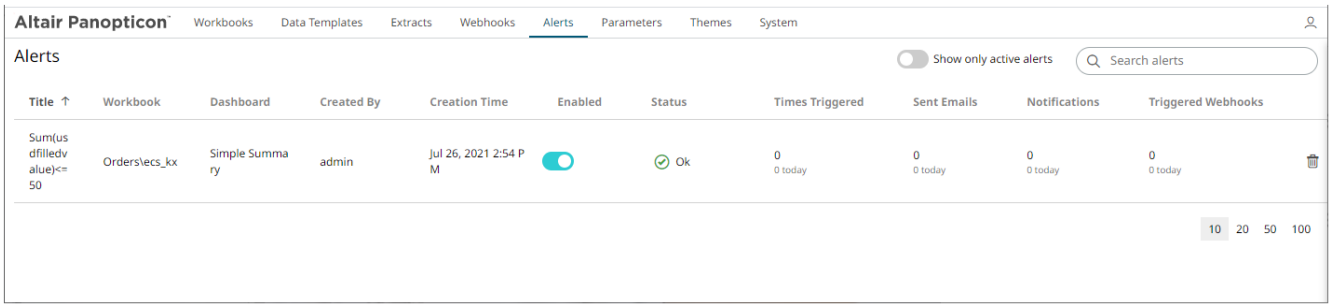
   The *Clock* settings display.



5. Select the *Hour*, *Minutes*, and *AM/PM* settings.
6. To modify the *Active Days*, check the boxes of the desired days.
7. To apply the active hours in another time zone, select the desired value from the *Show in Timezone* drop-down list box.

   Once set, the *From* and *To* limits will be applied for that time zone. If not set, the server default time zone will be used.
8. Tap the *Activated* slider to turn it on.
9. Click 

. The new alert is added on the *Alerts* page.



NOTE	When creating alerts for grand total, ensure that no breakdown is set.

An alert displays with the following properties or settings:

Property	Description
Title	Name of the alert that was entered in the *Alerts* dialog.
Workbook	The path and name of the workbook where the alert was set.
Dashboard	The dashboard name where the alert was set.
Created By	The author of the alert.
Creation Time	The Date/Time when the alert was set.
Enabled	Determines if the alert is enabled (or active).

Status	Status of the alert.
Times Triggered	The number of times the alert was triggered.
Sent Emails	The number of emails sent.
Notifications	The number of notifications sent.
Triggered Webhooks	The number of triggered webhooks.

You can then opt to perform any of the following operations:

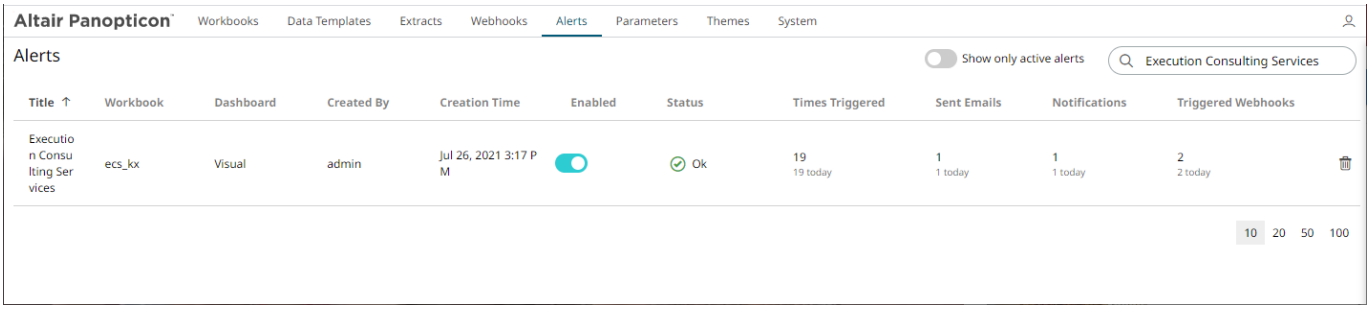
- Sort alerts
- Search for alerts
- Enable an alert
- Modify alerts
- Delete alerts
- Display active alerts

Sorting Alerts

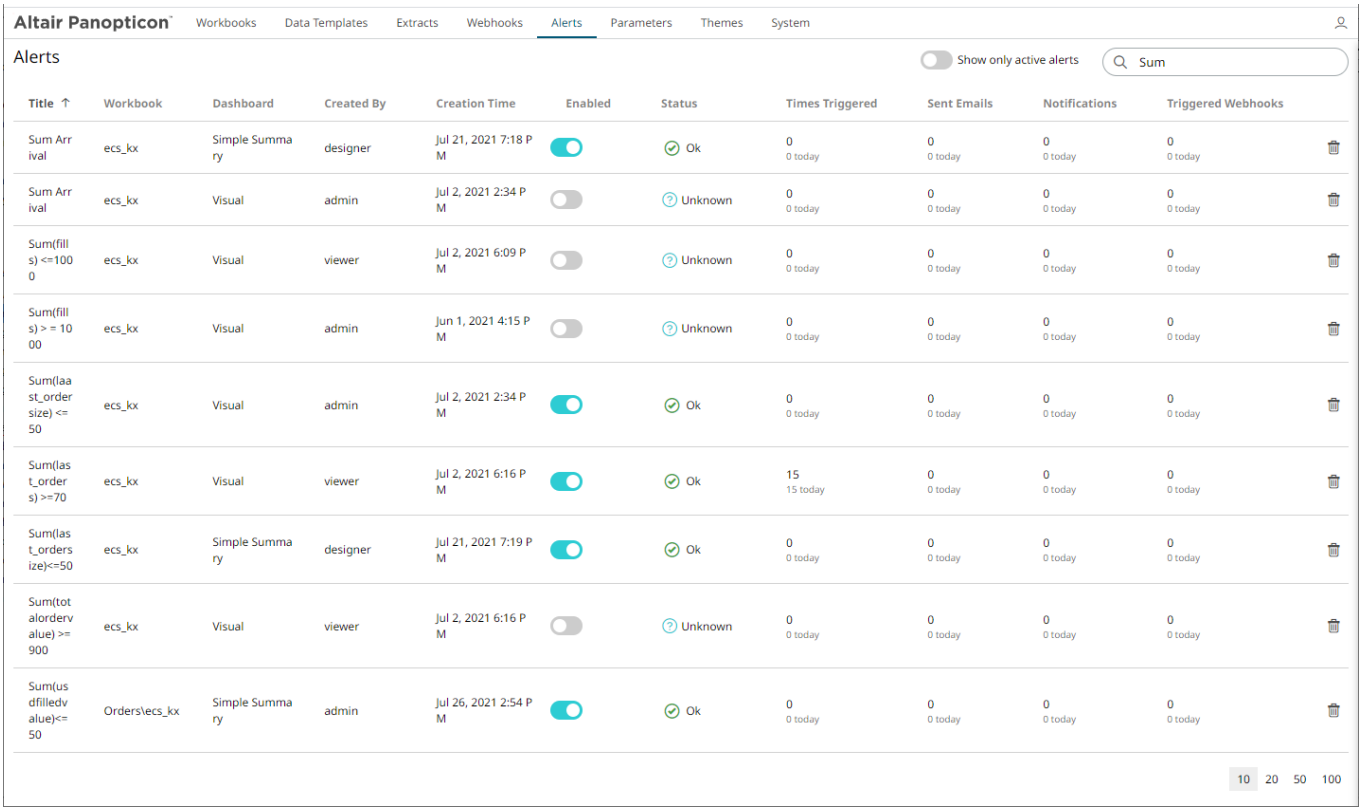
By default, the list of alerts is sorted by *Title* in an ascending order. You can modify the sorting of the list by clicking the ↓ or ↑ button of the *Title, Workbook, Dashboard, Created By, Creation Time, Enabled, Status, Times Triggered, Sent Emails, or Notifications* columns. The icon beside the column that was used for the sorting will indicate if it was in an ascending or descending order.

Searching for Alerts

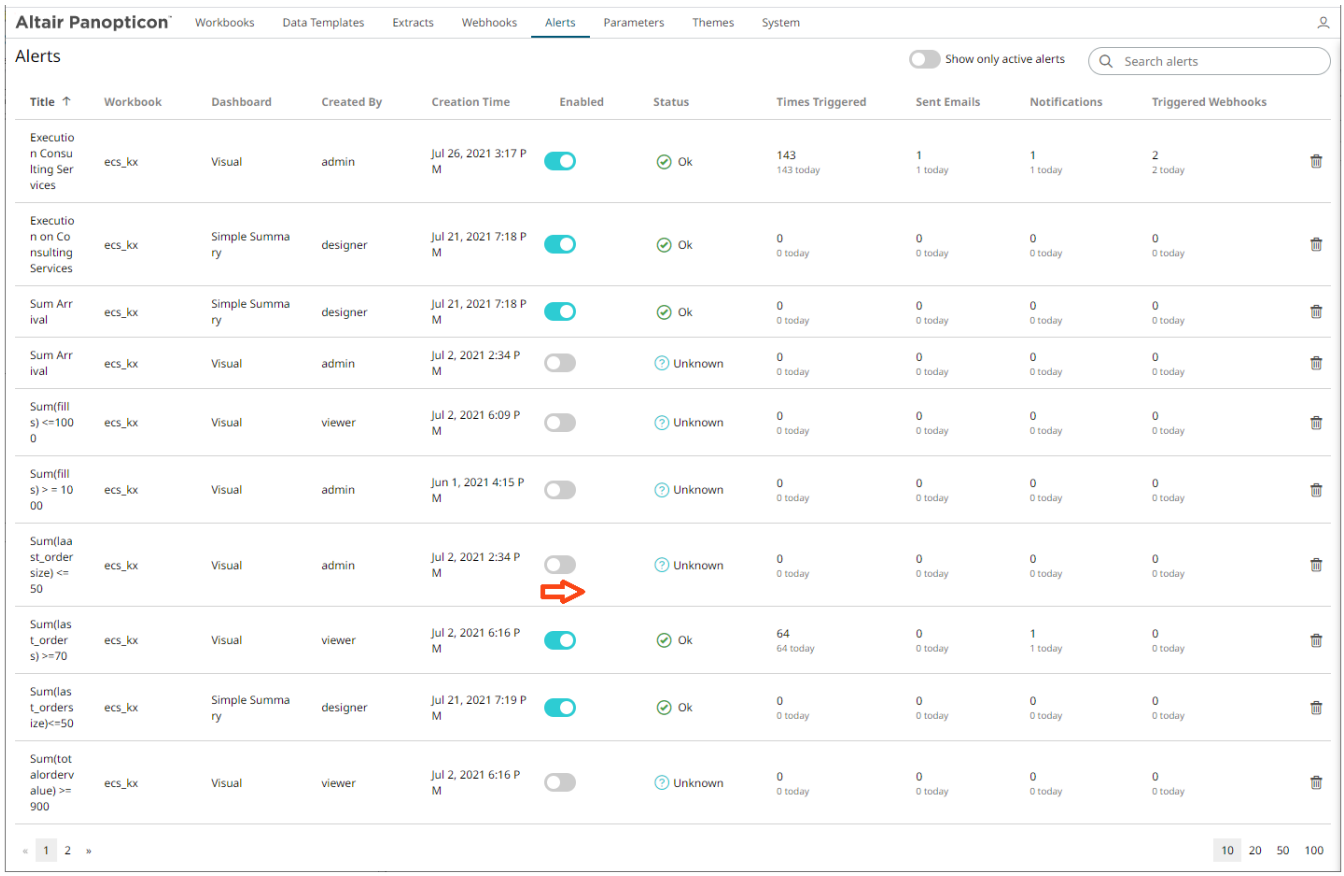
To search for a particular alert, enter it in the *Search* box.



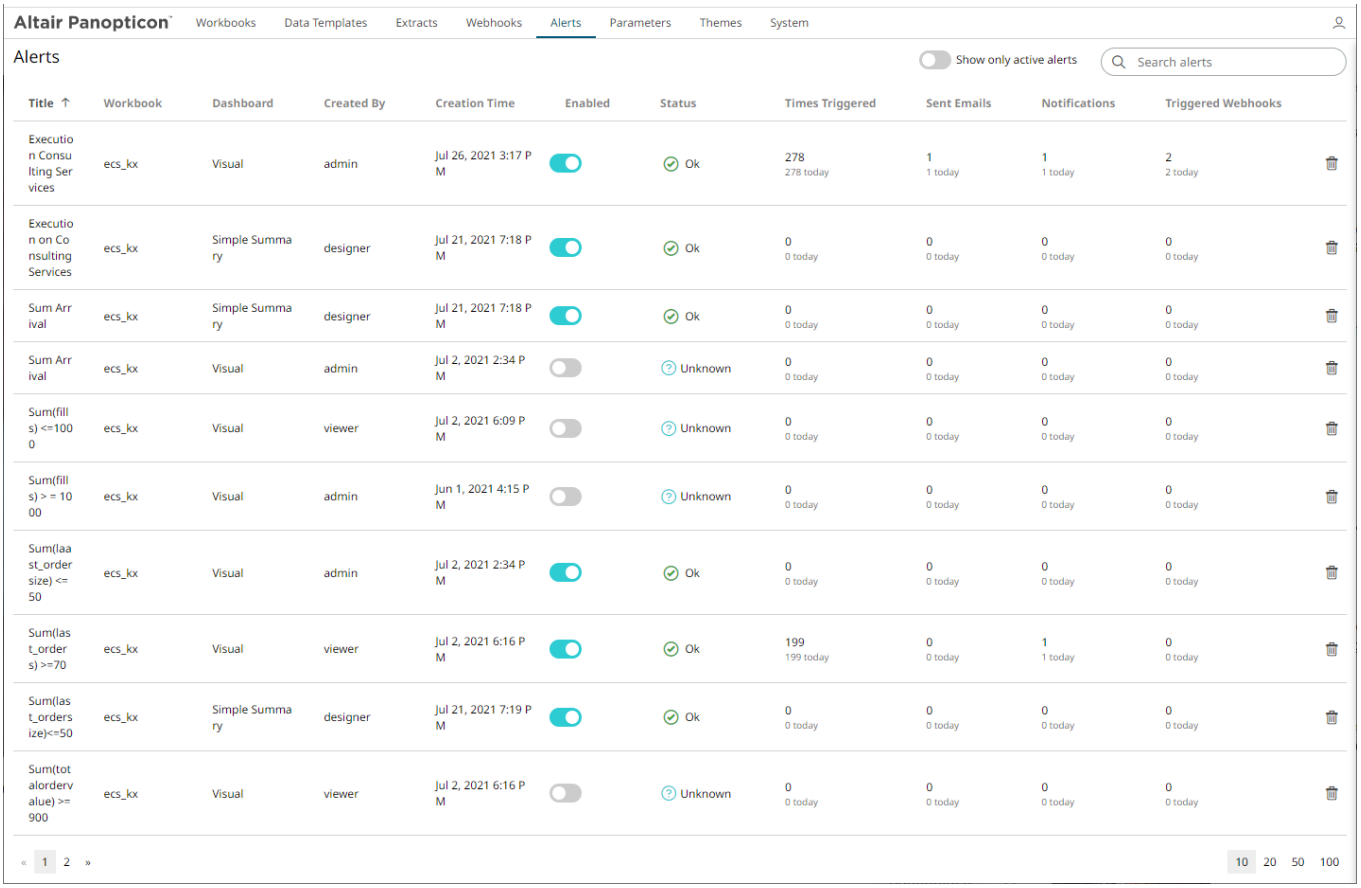
You can also enter one of more characters into the *Filter Applications* box and the suggested list of alerts that matched the entries will be displayed.


Altair Panopticon
Workbooks
Data Templates
Extracts
Webhooks
Alerts
Parameters
Themes
System
Alerts
Show only active alerts
Sum
Title
Workbook
Dashboard
Created By
Creation Time
Enabled
Status
Times Triggered
Sent Emails
Notifications
Triggered Webhooks
Sum Arrival
ecs_kx
Simple Summary
designer
Jul 21, 2021 7:18 PM
Ok
0
0 today
Sum Arrival
ecs_kx
Visual
admin
Jul 2, 2021 2:34 PM
Unknown
Sum(fills) <=1000
ecs_kx
Visual
viewer
Jul 2, 2021 6:09 PM
Sum(fills) > = 1000
ecs_kx
Visual
admin
Jun 1, 2021 4:15 PM
Sum(laast_ordersize) <= 50
ecs_kx
Visual
admin
Jul 2, 2021 2:34 PM
Sum(last_orders) >=70
ecs_kx
Visual
viewer
Jul 2, 2021 6:16 PM
15
15 today
Sum(last_ordersize)<=50
ecs_kx
Simple Summary
designer
Jul 21, 2021 7:19 PM
Sum(totalordervalue) >= 900
ecs_kx
Visual
viewer
Jul 2, 2021 6:16 PM
Sum(usdfilledvalue)<= 50
Orders\ecs_kx
Simple Summary
admin
Jul 26, 2021 2:54 PM
10
20
50
100

Enabling Alerts on the Alerts Page



Tap the *Enabled* slider to turn it on.

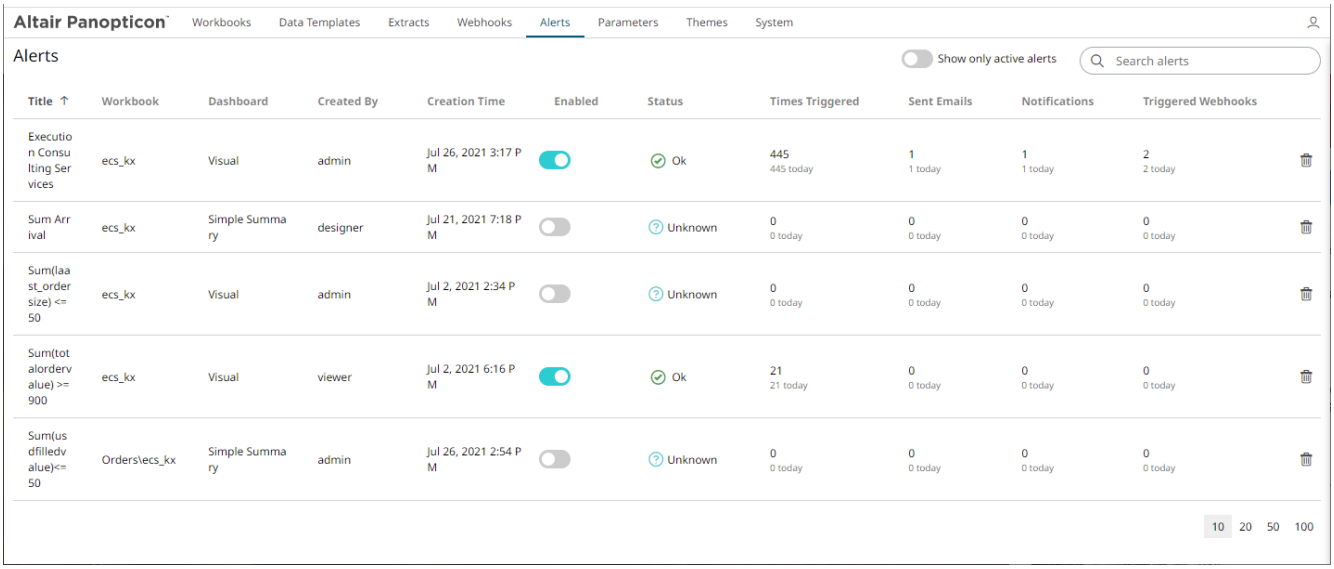


Enabling alerts can also be performed on a visualization's Alerts panel.

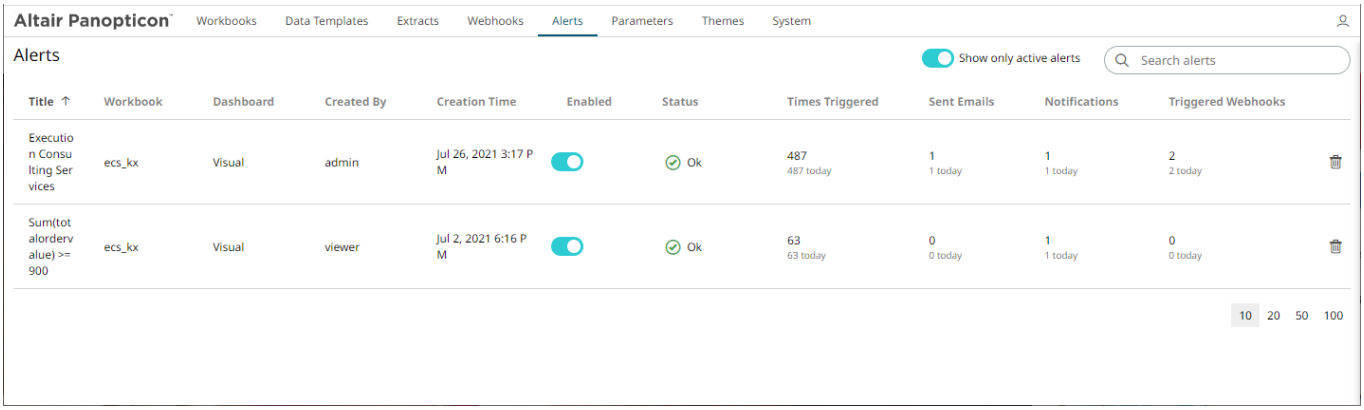
Other Alerts Operations can be modified, enabled, and deleted in the workbook where it was set.

Displaying Active Alerts

Tap the **Show only active alerts** slider to turn it on.



Only the active or enabled alerts are displayed on the *Alerts* tab.

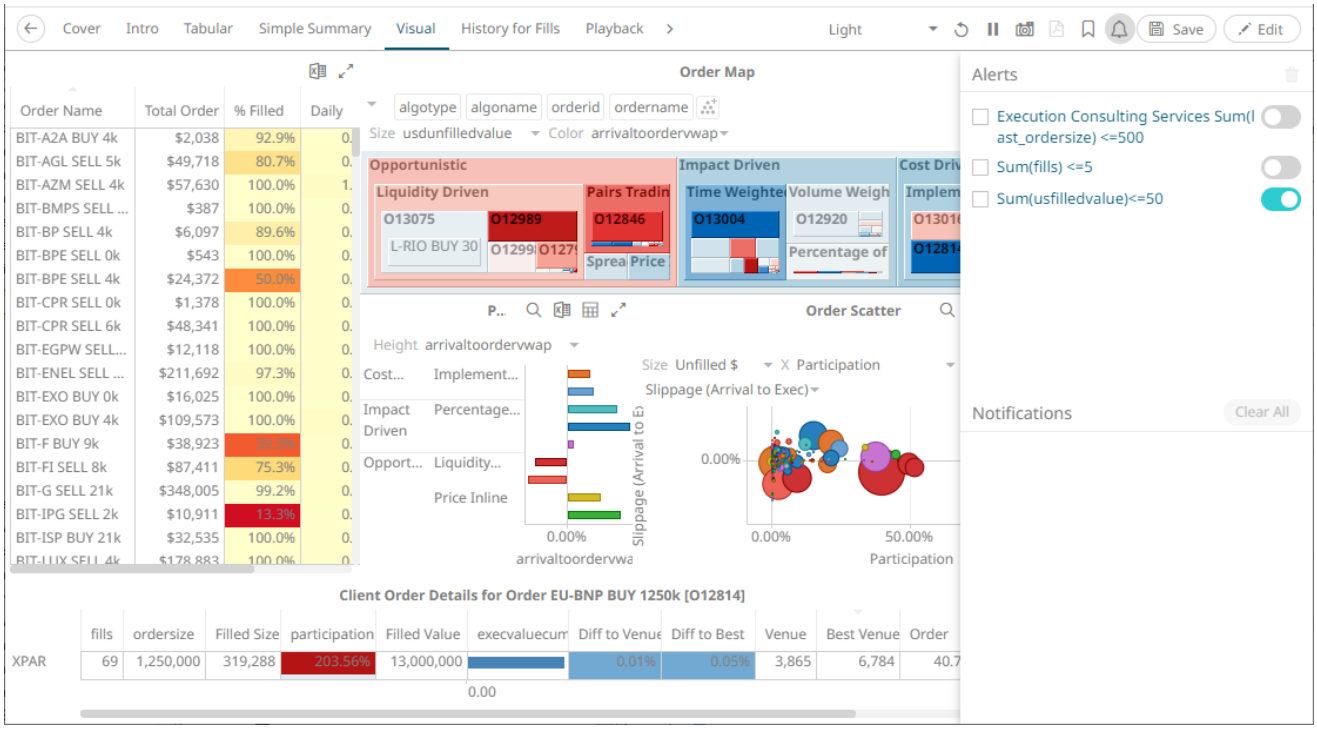


Modifying Alert Settings

Steps:

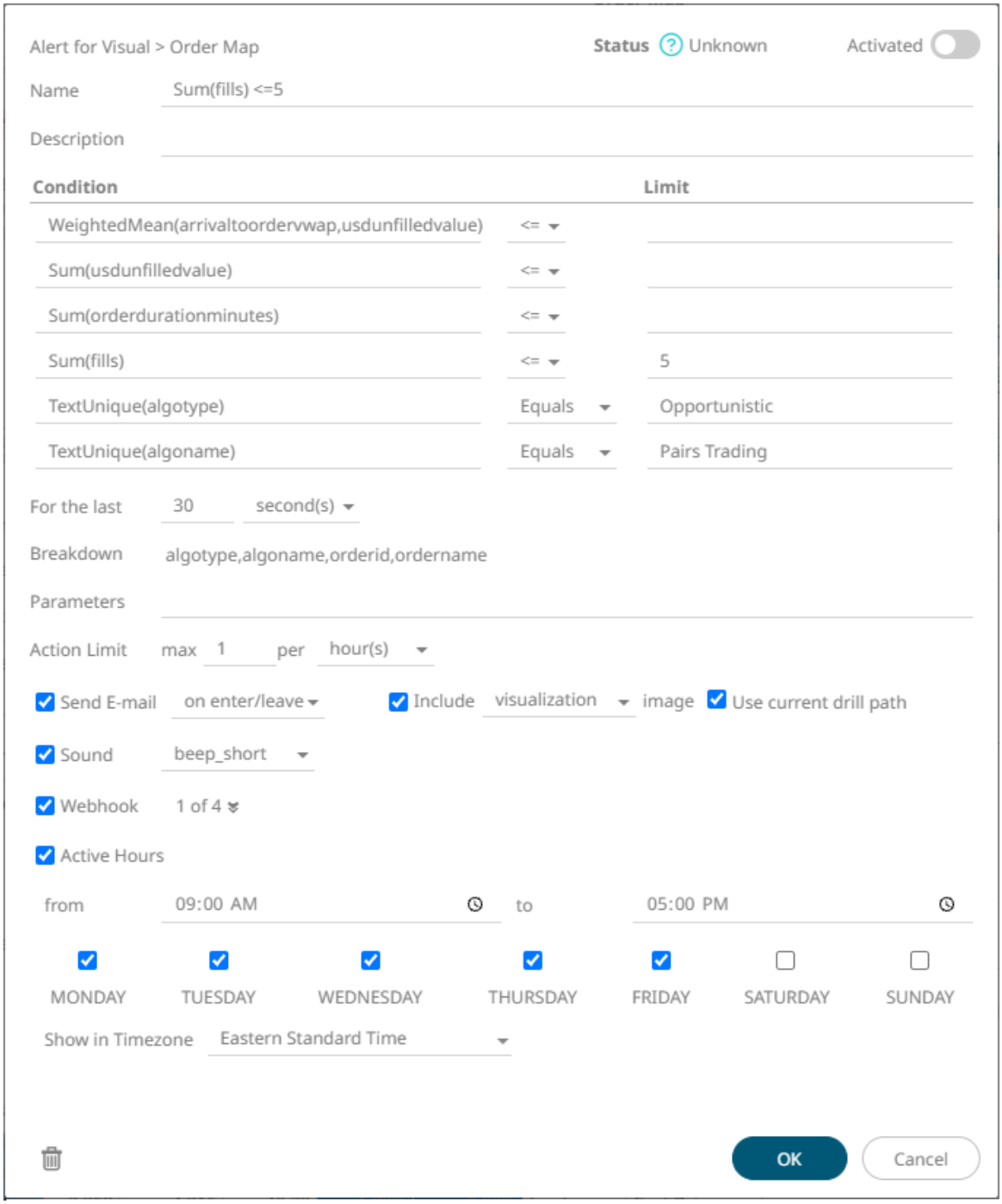
1. Open a workbook with an alert and click on the **Alerts** icon.

   The *Alerts* panel displays with the list of alerts.



2. Click an alert to modify.

   The *Alerts* dialog displays.



3. Make the necessary changes then click OK to save them.

Deleting Alerts

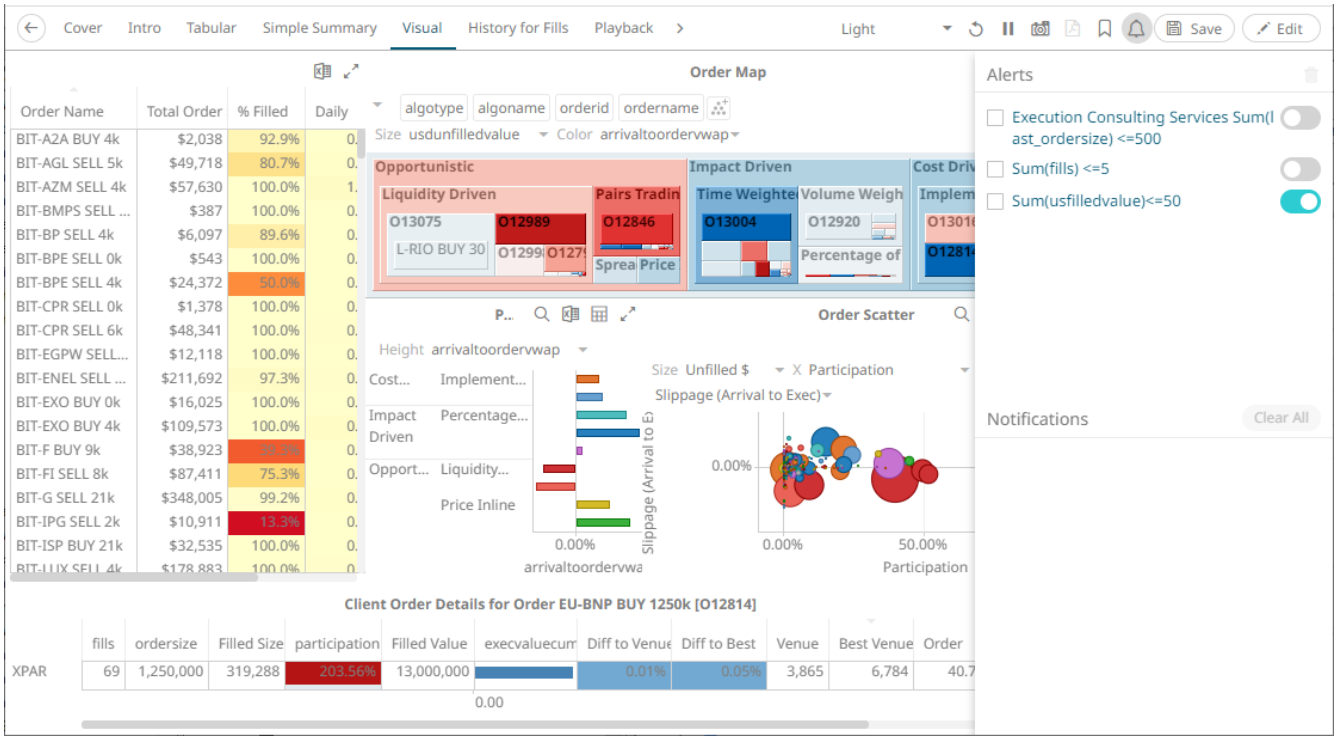
Alerts can be deleted on:

- ❑ the Alerts panel
- ❑ an Alerts dialog
- ❑ the Alerts tab

Deleting Alerts on the Alerts Panel:

1. Open a workbook with an alert and click on the **Alerts** icon.

   The *Alerts* panel displays with the list of alerts.

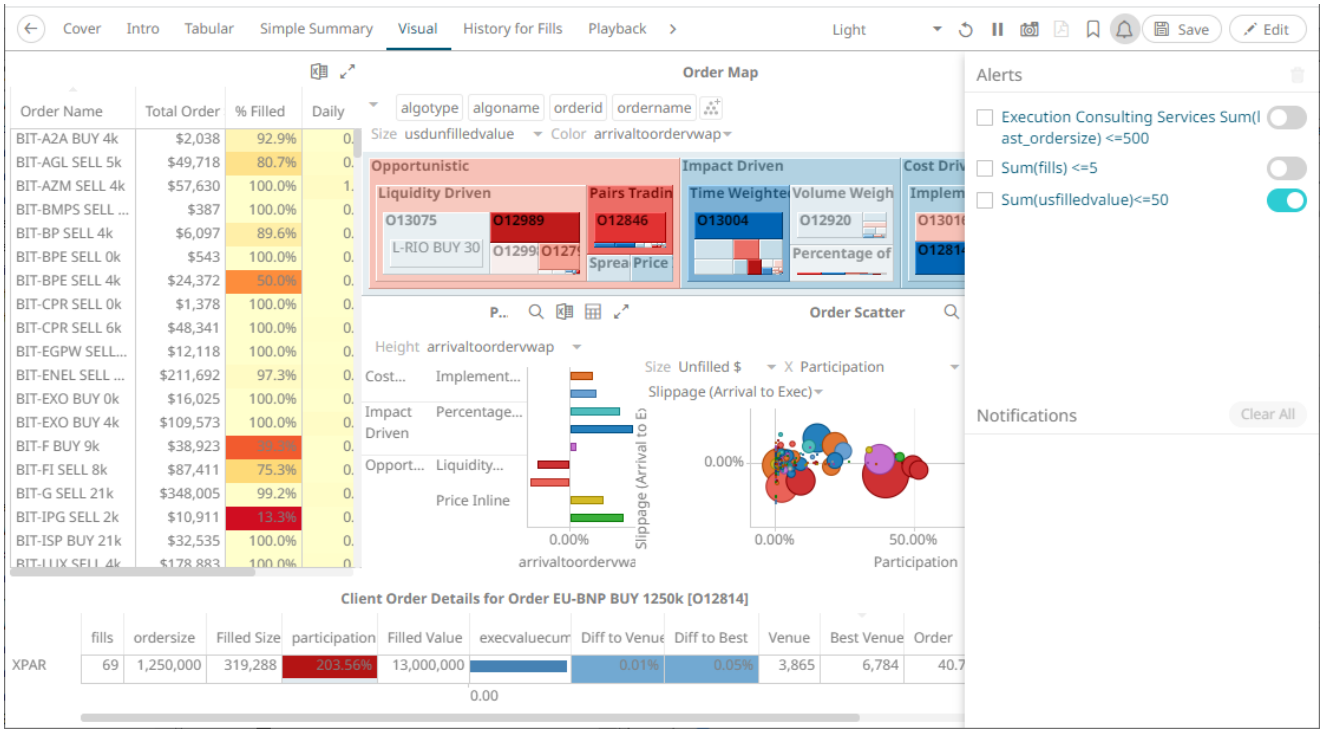


2. Check the box of an alert and click the **Delete** icon. You can also check several boxes to delete multiple alerts.

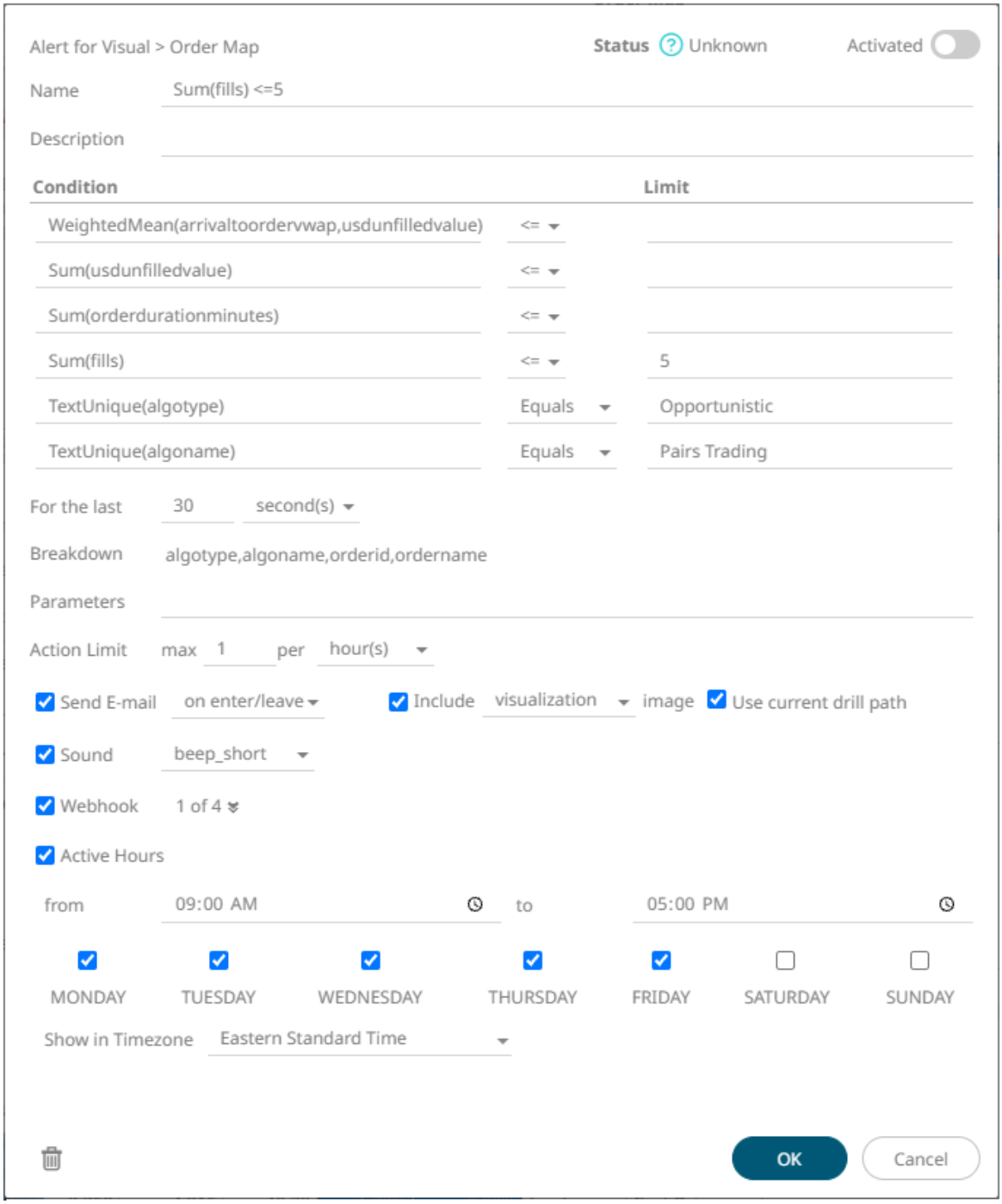
Deleting Alerts on an Alerts Dialog:

1. Open a workbook with an alert and click on the **Alerts** icon.

   The *Alerts* panel displays with the list of alerts.



2. Click an alert. The *Alerts* dialog displays.

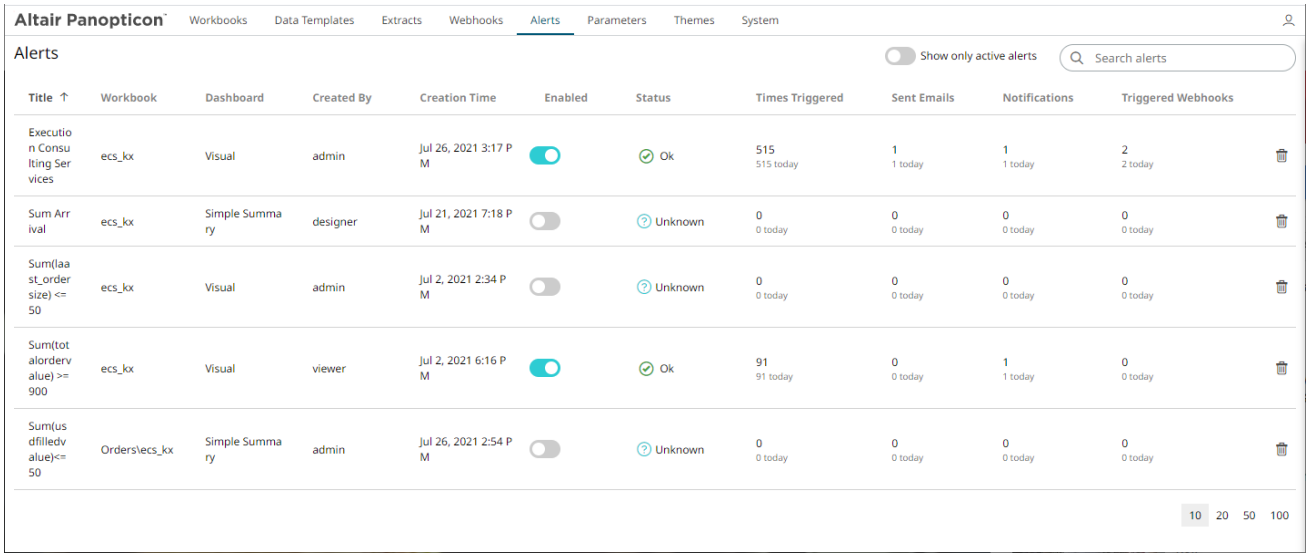


3. Click the **Delete** icon.

Deleting Alerts on the Alerts tab:

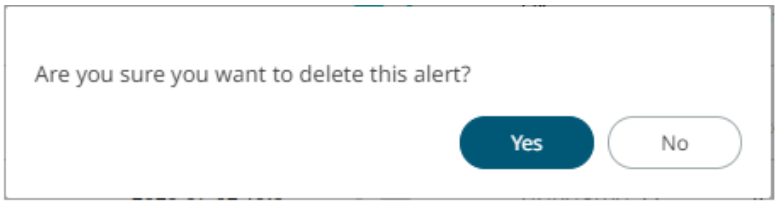
1. Go to the **Alerts** tab.

   The **Alerts** tab displays the list of alerts.



3. Click the of an alert to delete.

   A confirmation message displays.



4. Click Yes .

Enabling Alerts

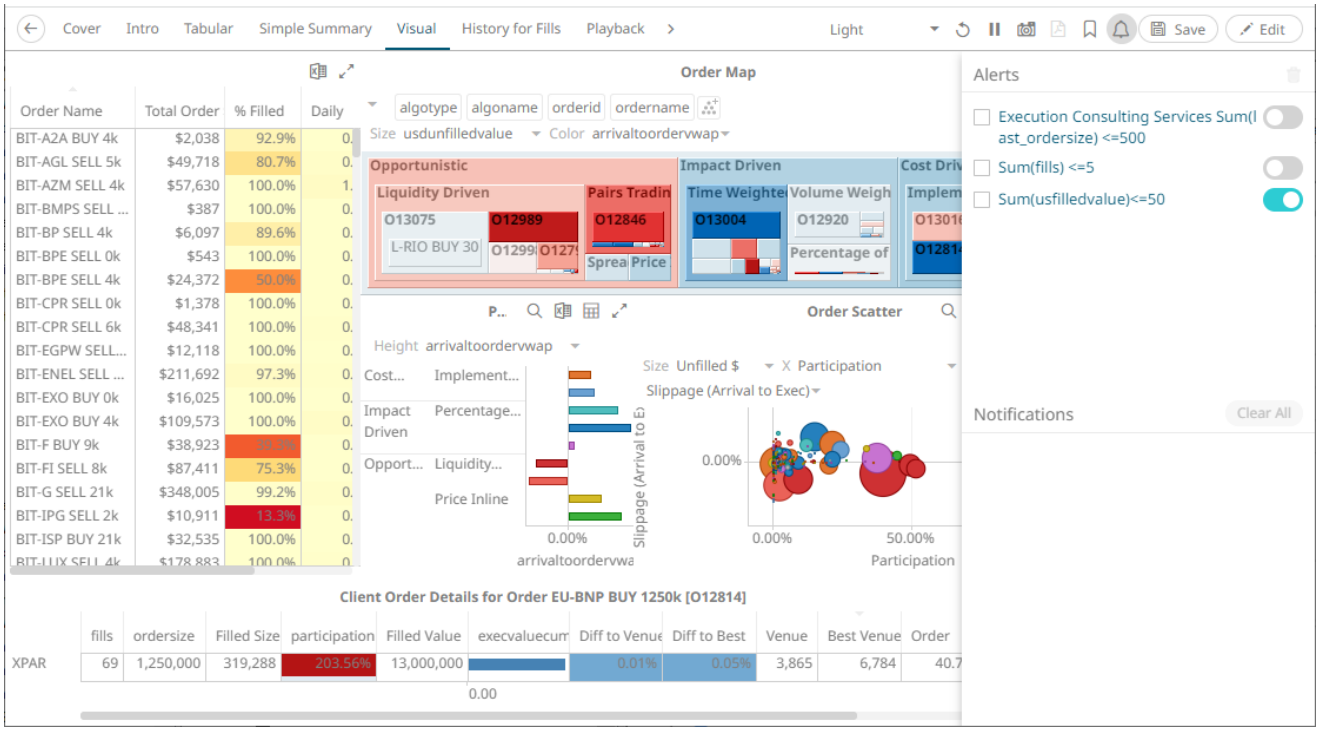
Alerts can be enabled either on:

- the Alerts panel
- an Alerts dialog

Enabling Alerts on the Alerts Panel:

1. Open a workbook with an alert and click on the **Alerts** icon.

   The *Alerts* panel displays with the list of alerts.

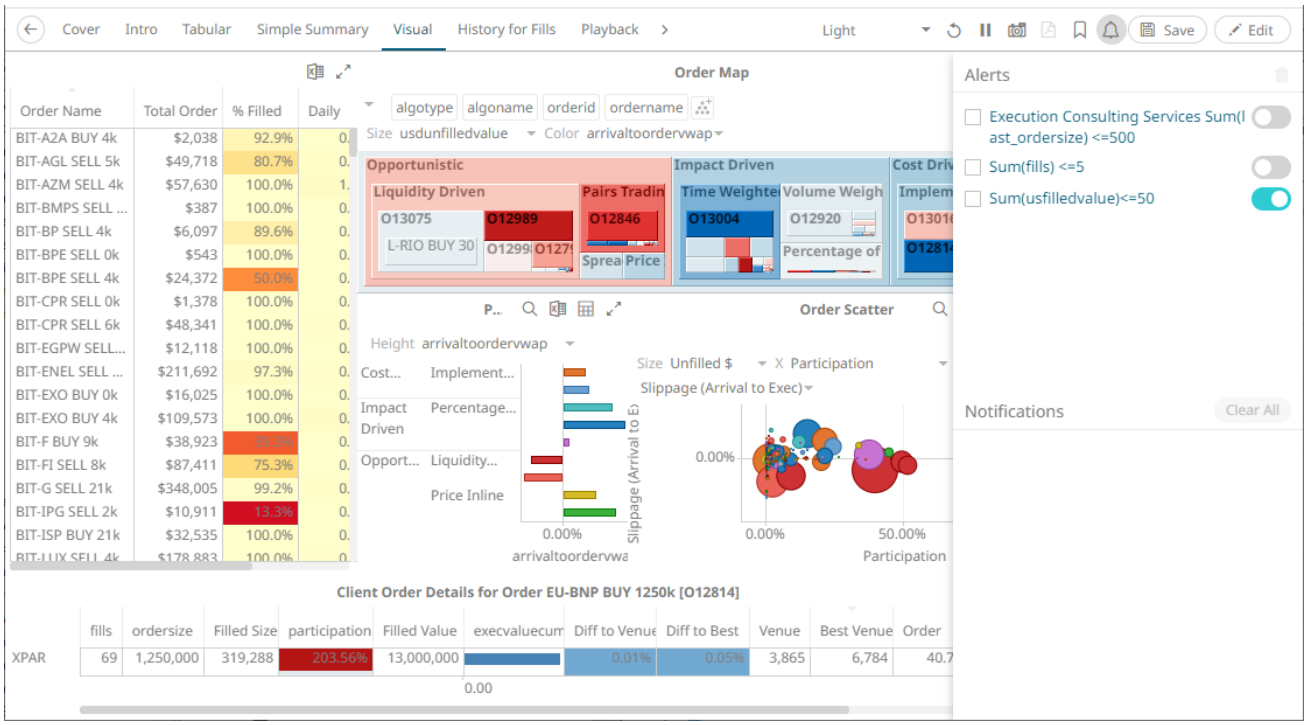


2. Tap the *Activated* slider to turn it on.

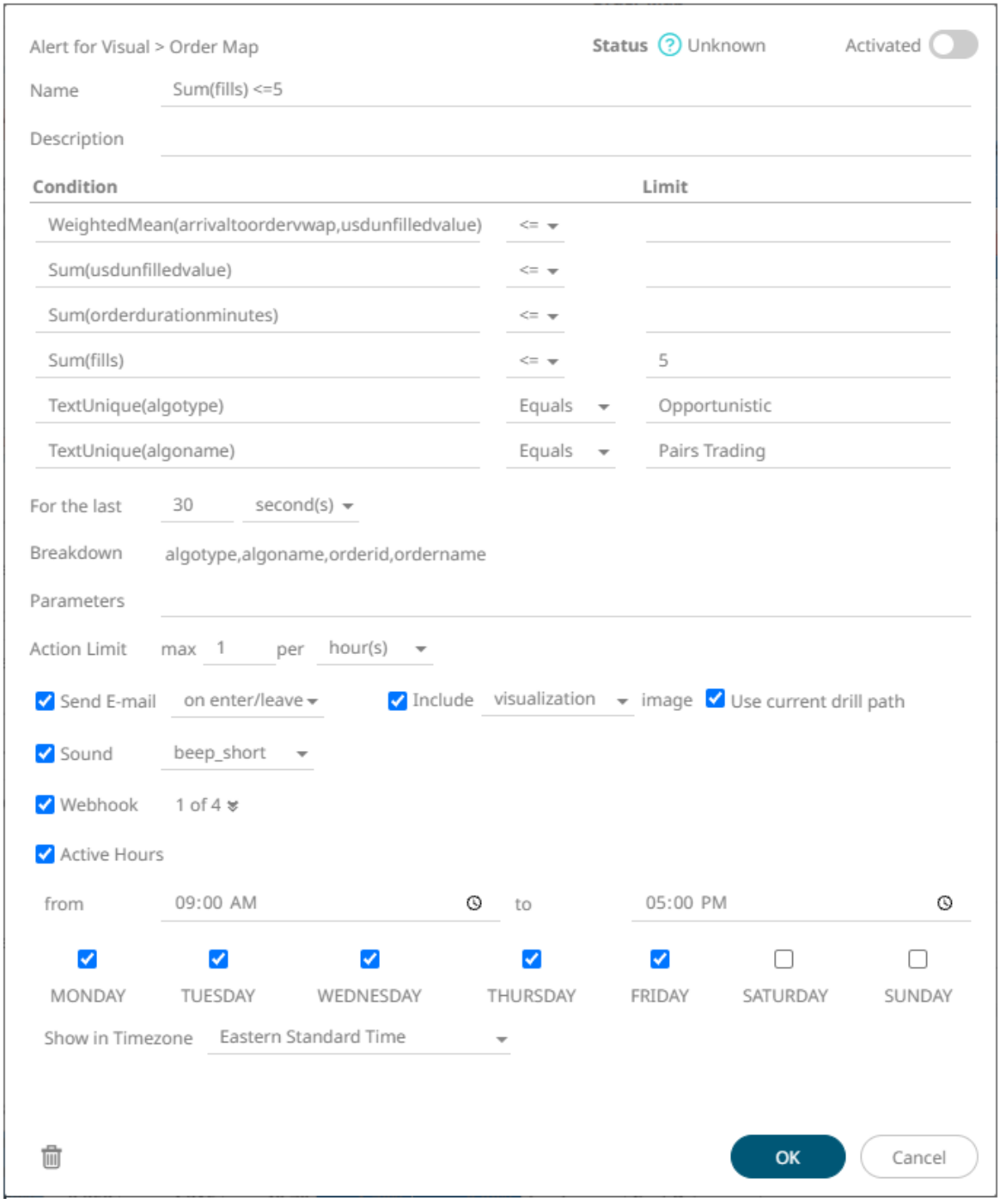
Enabling Alerts on an Alerts Dialog:

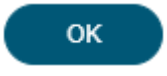
1. Open a workbook with an alert and click on the **Alerts** icon.

   The *Alerts* panel displays with the list of alerts.



3. Click an alert. The *Alerts* dialog displays.

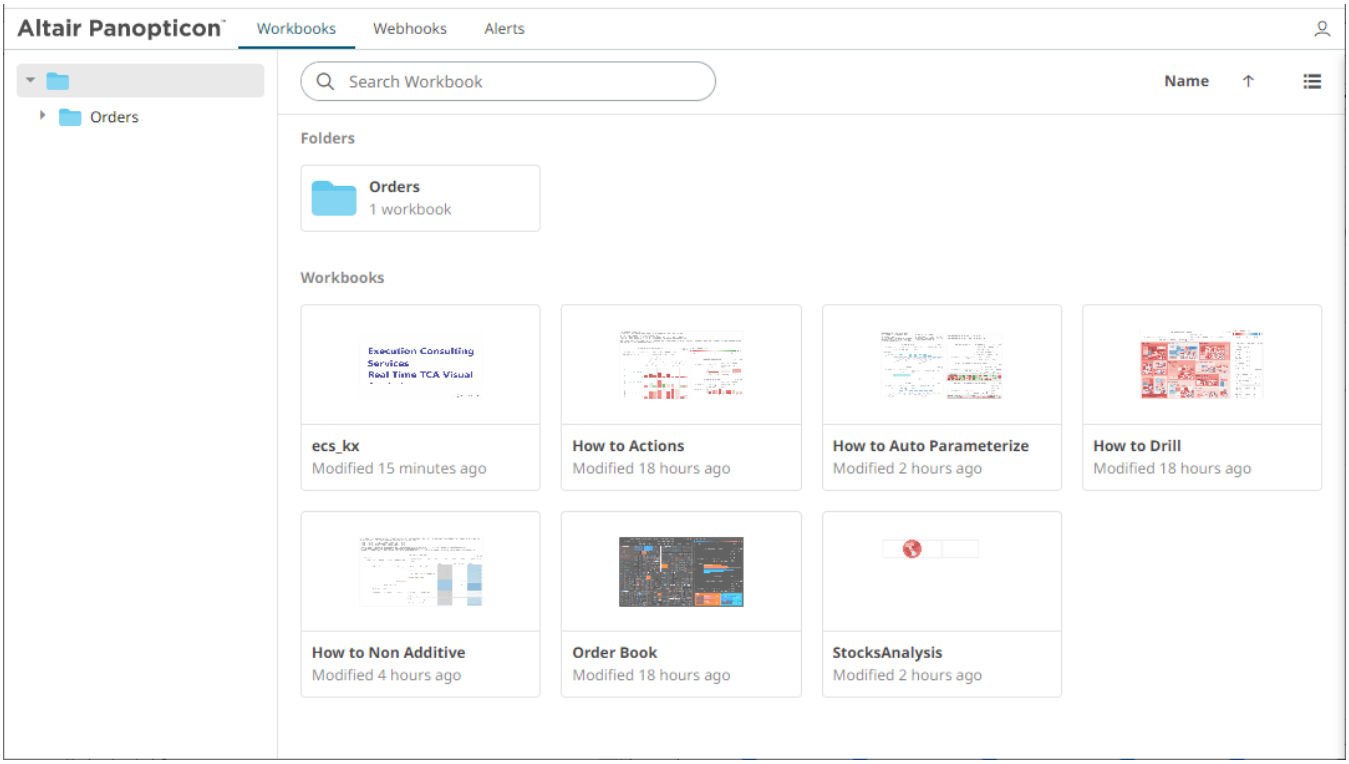


4. Tap the *Activated* slider to turn it on and click 

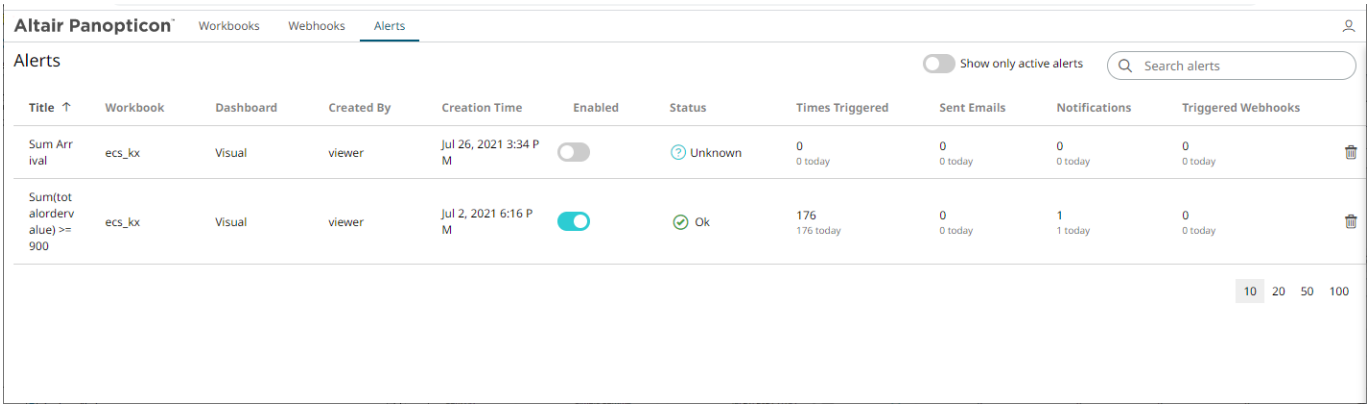
.

Viewing and Managing Alerts for Non-Administrator users

There are three tabs that are available for non-Administrator users:



Click on the **Alerts** tab to view and manage the available alerts.



Also perform any of the following operations:

- ❑ Sort alerts
- ❑ Search for alerts
- ❑ Enable an alert
- ❑ Delete alerts
- ❑ Display active alerts

On the **Profile** tab, enter the email of the user or group who will receive the alert.

SAMPLE EMAIL ALERTS

An alert is generated when the alert set state changes from **Off** to **On** and recorded in the alert history.

An alert is only issued by email if the alert has not already been sent in the last 'n' minutes as defined in the *Alerts* dialog.

When an alert is issued, an email is sent to the defined email address.

The email includes:

- Link to the workbook or dashboard
- Condition and limit value
- Breakdown
- Name of the visualization where the alert was set
- PNG image of the visualization or dashboard

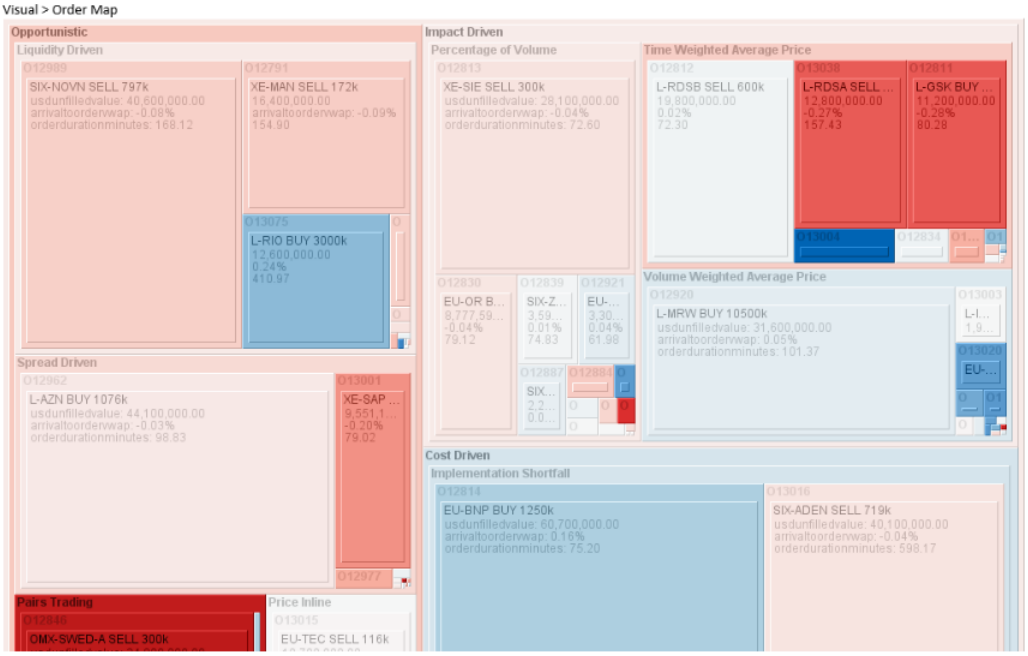
All items that do not match the criteria are excluded from the display.

Dashboard: http://localhost:8080/panopticon/workbook/#/ecs_kx/Visual

Condition: Sum(fills) >= 10.0

The alert was triggered by the following items:
algotype:Opportunistic, algoname:Liquidity Driven, sym:O12989, ordername:SIX-NOVN SELL 797k

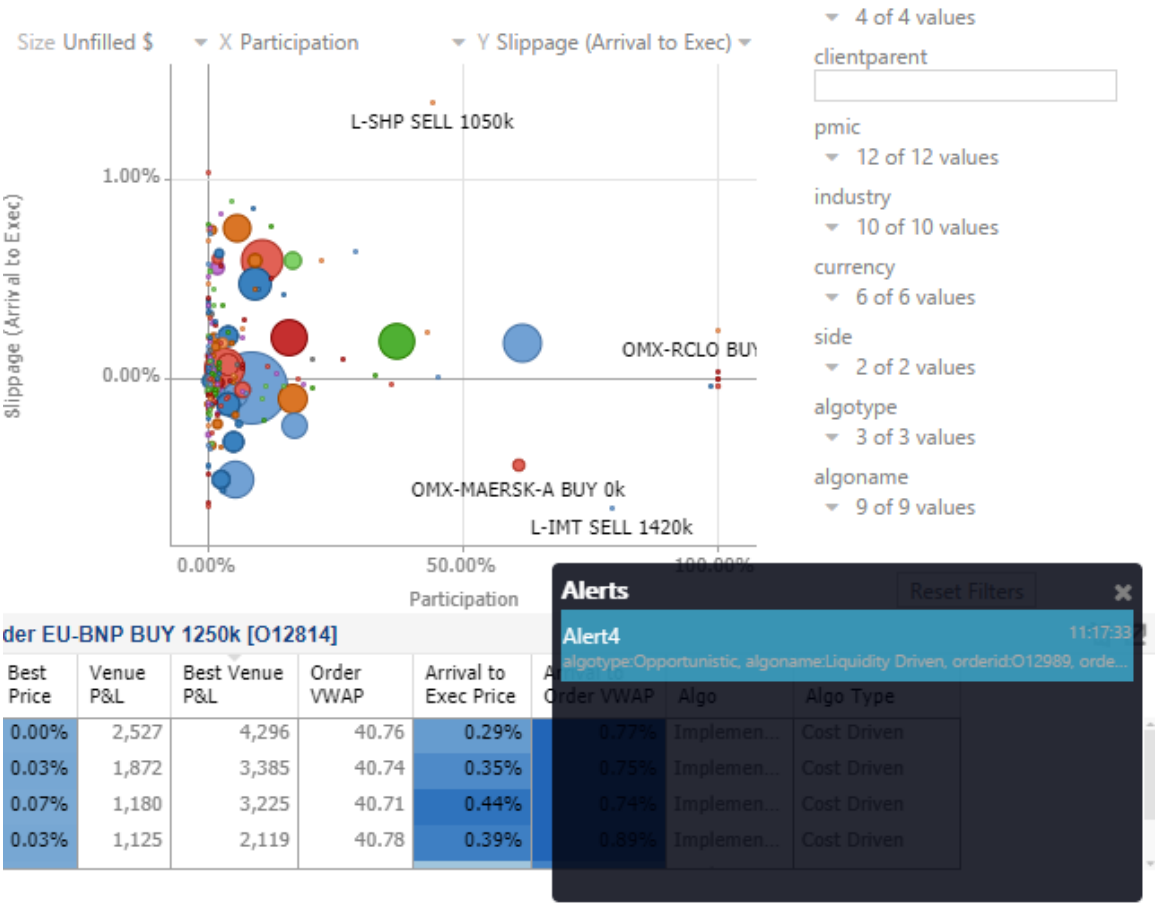
Visual > Order Map



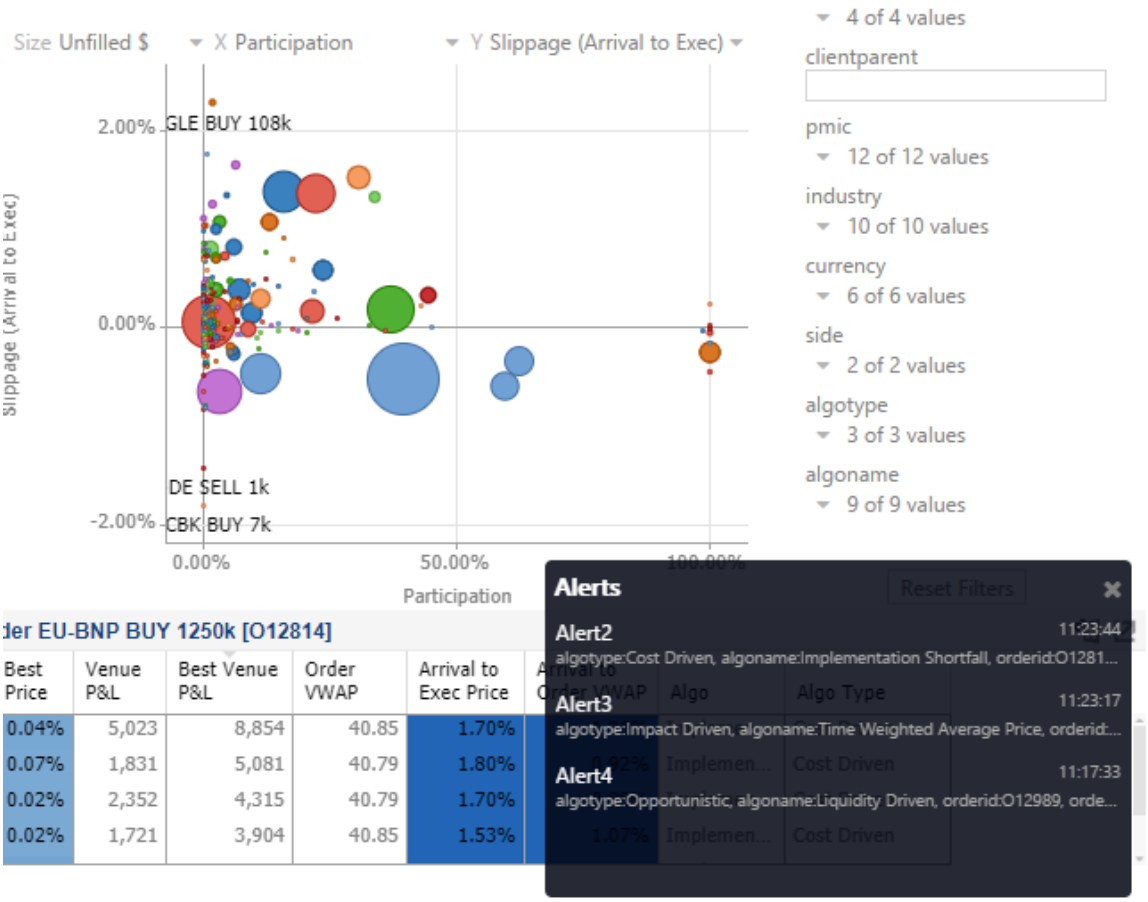
SAMPLE WEB CLIENT ALERTS

When an alert is triggered, aside from the email notifications, a visual indication or pop-up in active Web clients will draw attention to the alerting visualization or dashboard.

In the example below, an alert initially displays highlighted in blue:

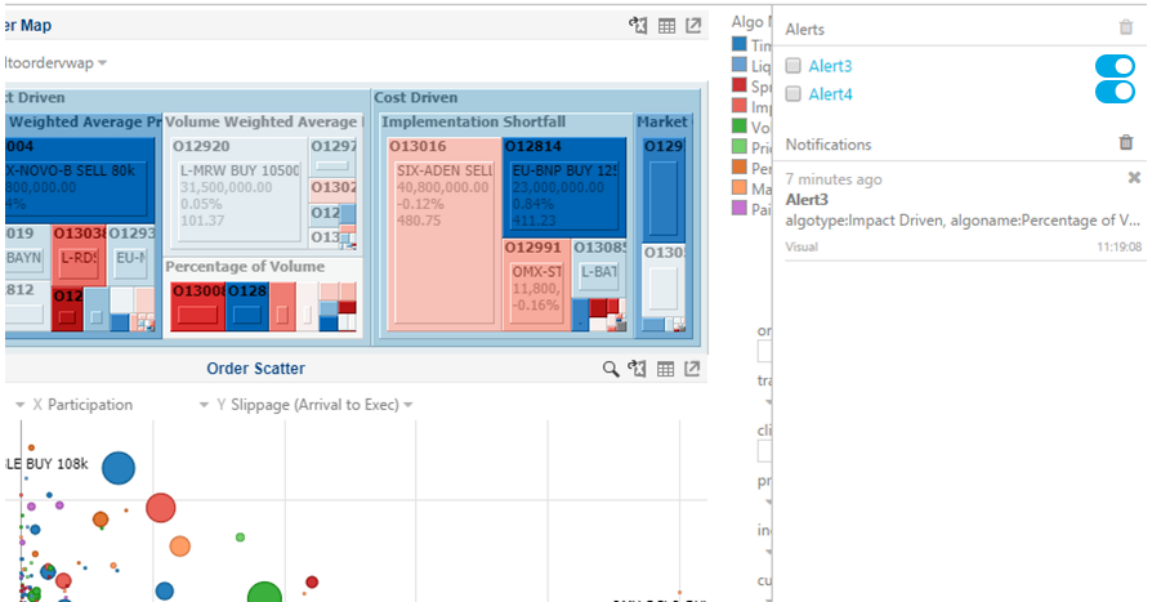


The alert eventually fades away and the pop-up screen fills up with the four latest triggered alerts.

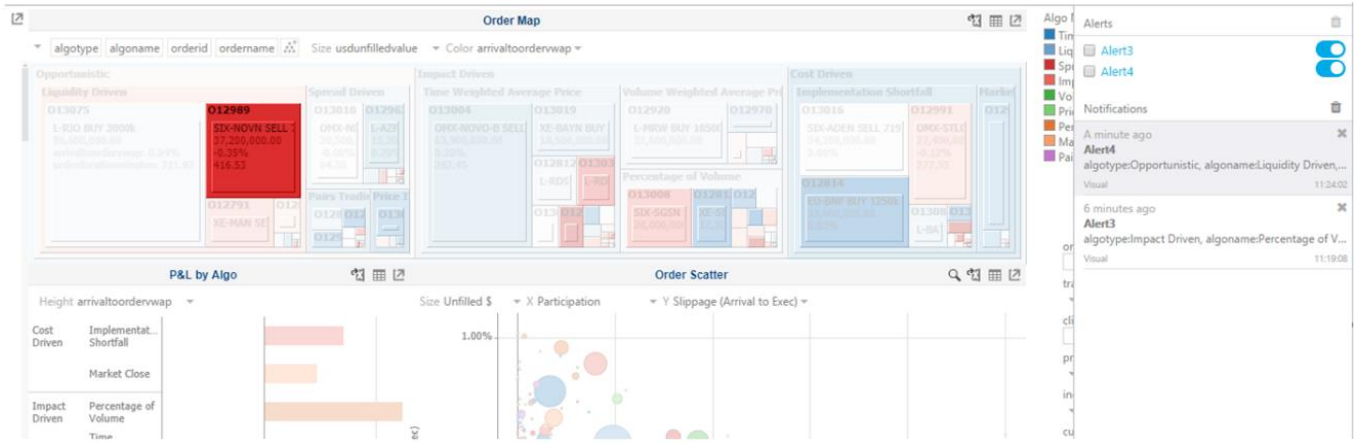


The pop-up stays on screen until it is closed by clicking the button.

Saved alert notifications can be opened on the *Notifications* panel by clicking the icon.



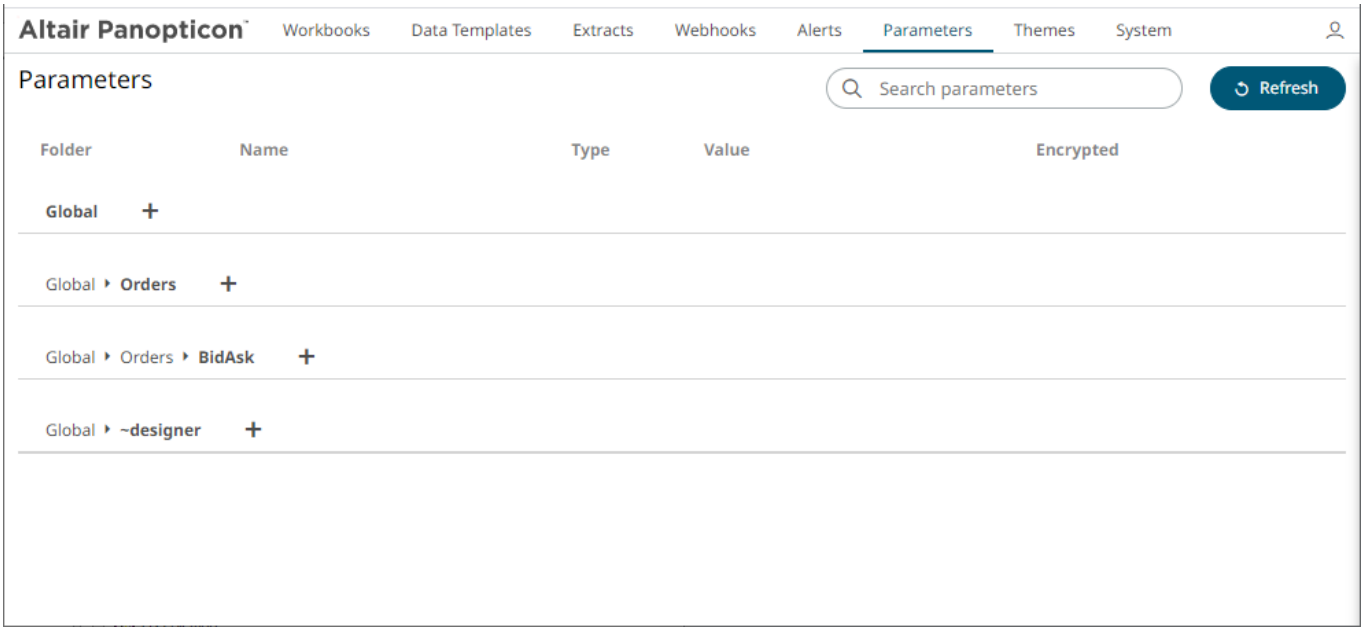
Clicking on a notification highlights the item in the workbook that triggered the alert.



Click the button to delete a notification or click to delete all of the notifications.

[14] GLOBAL PARAMETERS

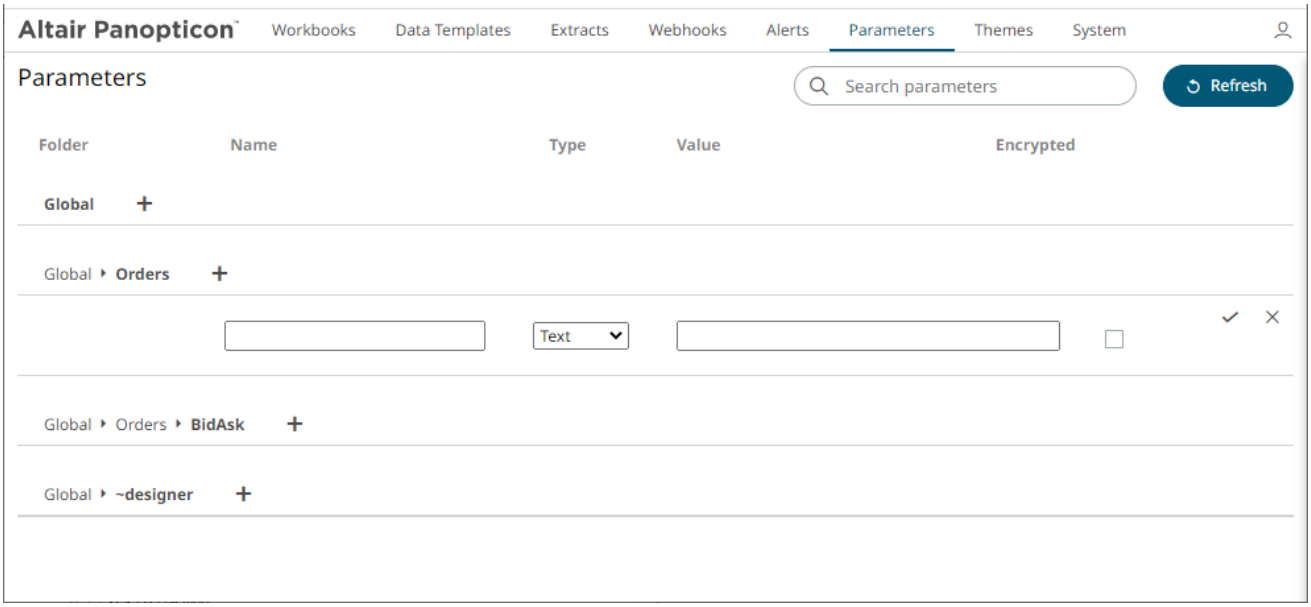
The **Parameters** tab supports adding, modifying, and deleting global parameters that will pull and enter specific data into the different sets that are assigned to workbook folders, as well as user specific folders (e.g., **Global > Orders**, **Global > Orders > BidAsk, Global > ~designer)**).



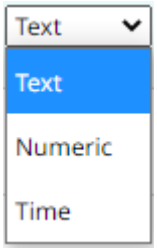
ADDING PARAMETERS

Steps:

1. On the **Parameters** tab, click the Add + icon of a global folder (parent or subfolder).
   A new parameter entry displays.



2. Enter a *Name* for the new parameter.
3. Select the *Type*: **Text**, **Numeric**, or **Time**.



4. Enter the *Default Value*.

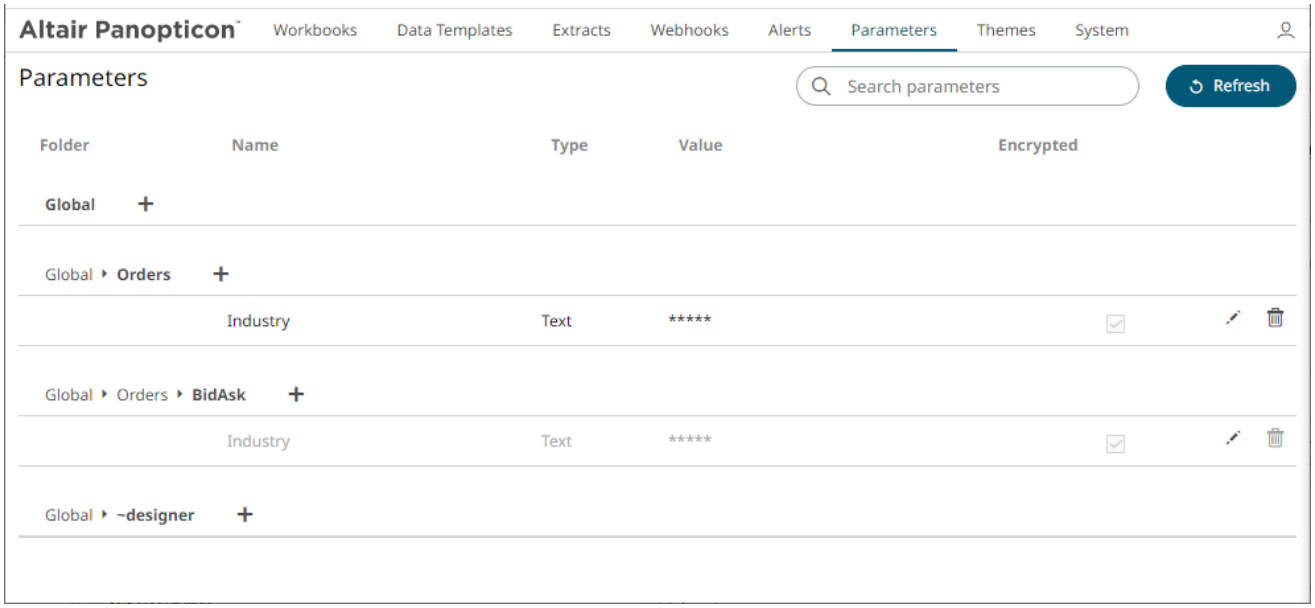
NOTE	

	• You can enter several default values, separated by a comma.
	• Single quotes on parameter value/s are removed when saving global parameters.
	• For the **Time** type, the following formats for the default value are accepted:
	  ○ "yyyy-MM-dd"
	  ○ "yyyy-MM-ddTHH:mm:ss"
	  ○ "yyyy-MM-ddTHH:mm:ss.SSS"

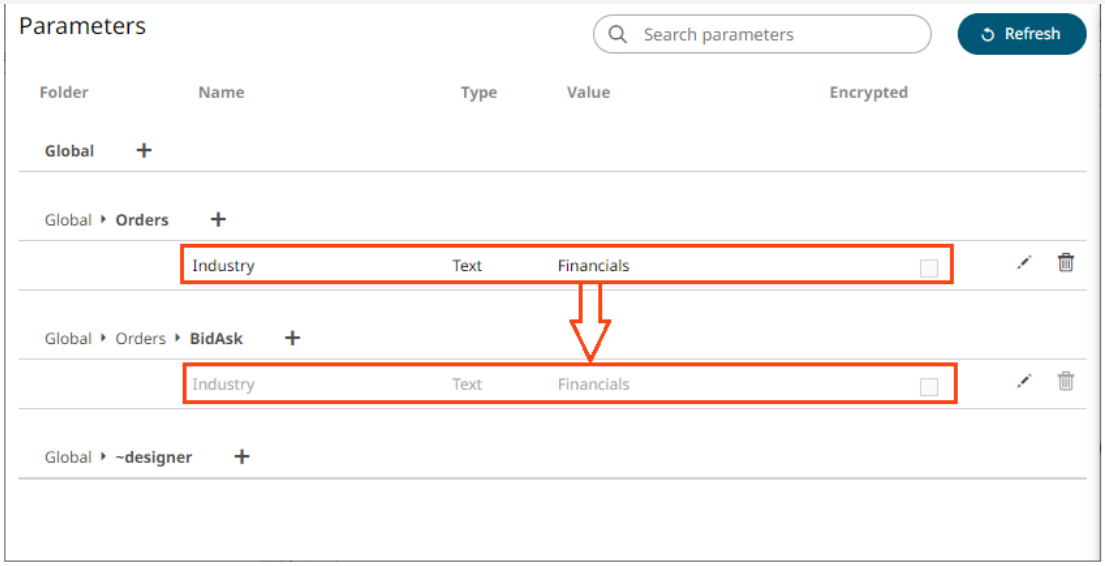
5. Check the *Encrypted* box to encrypt the value.

NOTE	
	Encryption is only supported for text parameters.

6. Click ✓. The new parameter is added in the list.



NOTE Global parameters are inherited from the corresponding parent folder.

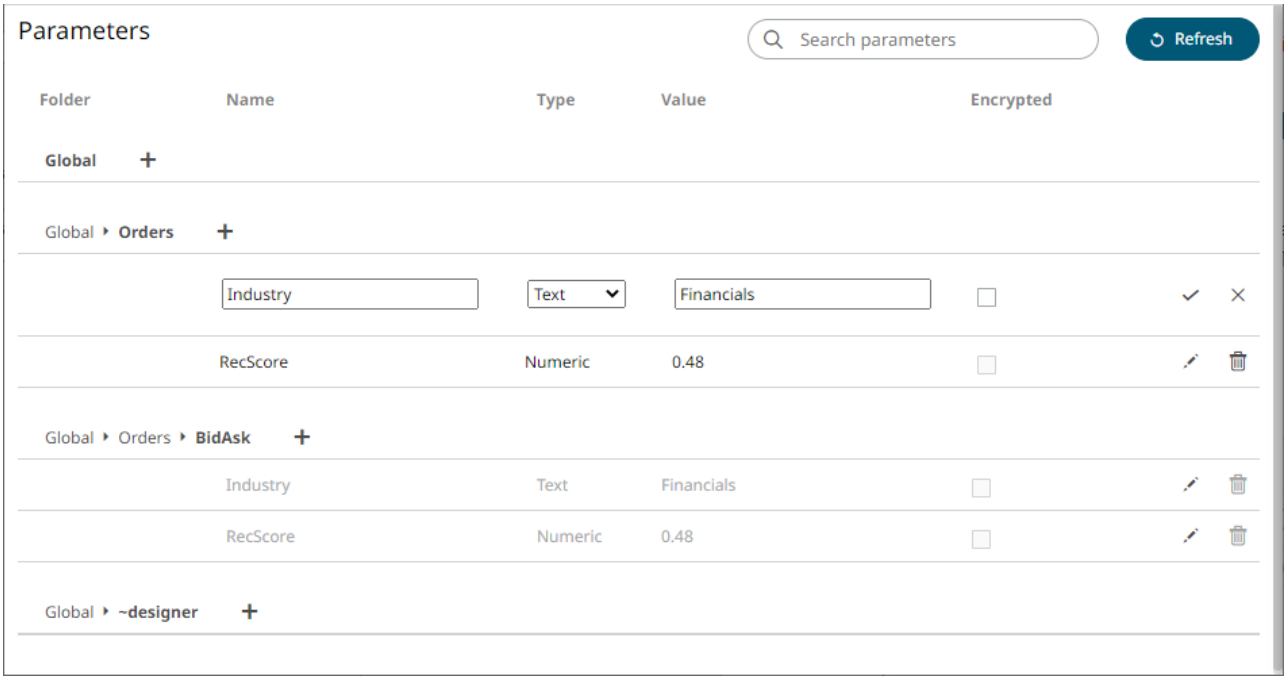


MODIFYING PARAMETERS

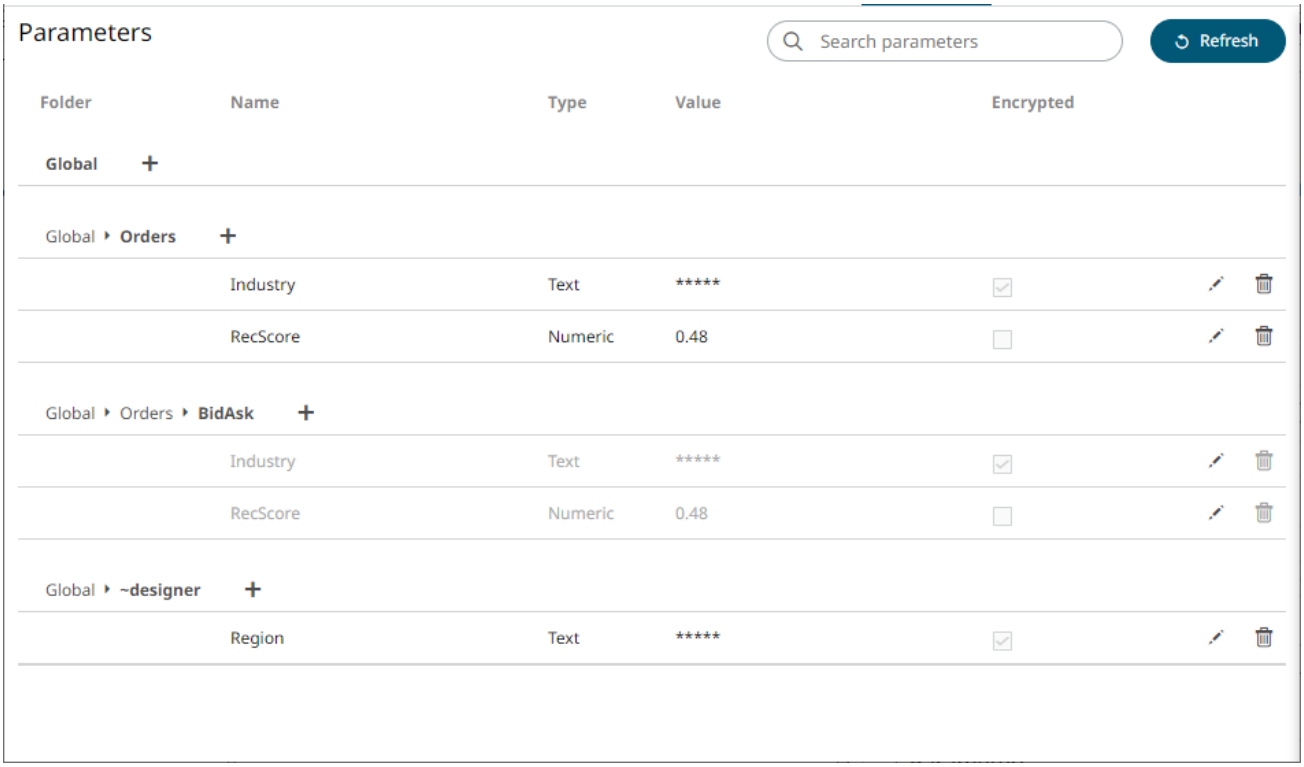
Steps:

1. On the **Parameters** tab, click the **Edit** icon of a parameter.

   The *Name, Value, and Encrypted* controls are enabled.

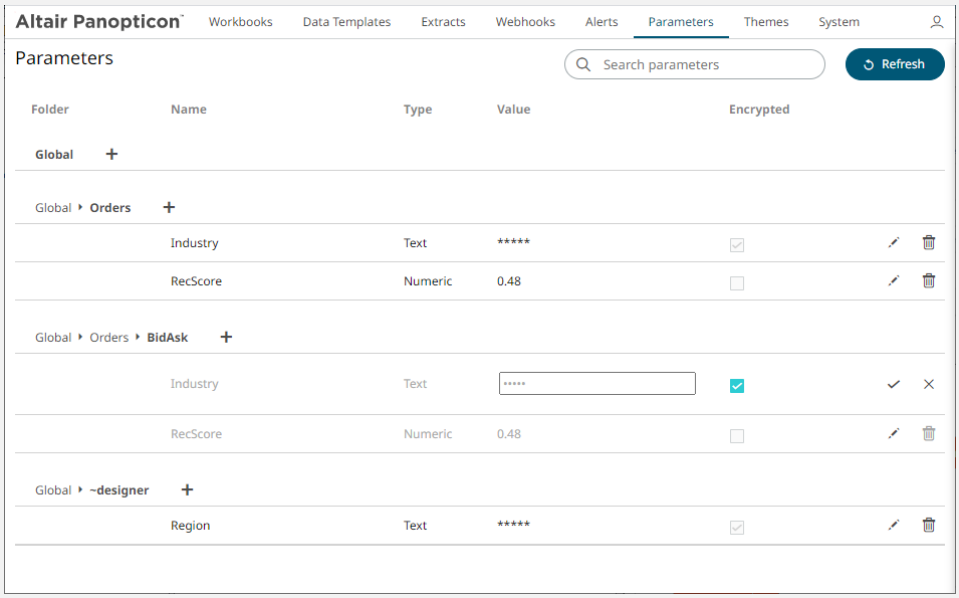


2. Make the necessary changes then click ✓.

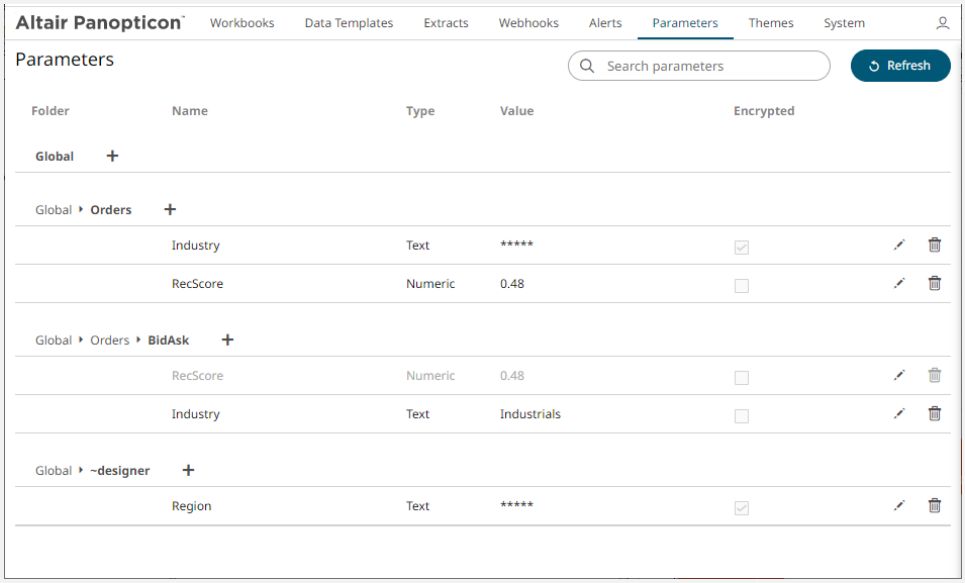


NOTE

For the inherited parameters, the *Name* and *Type* are not editable.



Once the value of the inherited parameter is changed, it is displayed as a global parameter and can also be deleted.



DELETING PARAMETERS

Steps:

1. On the **Parameters** tab, click the **Remove** icon of a parameter.

   A confirmation message displays.

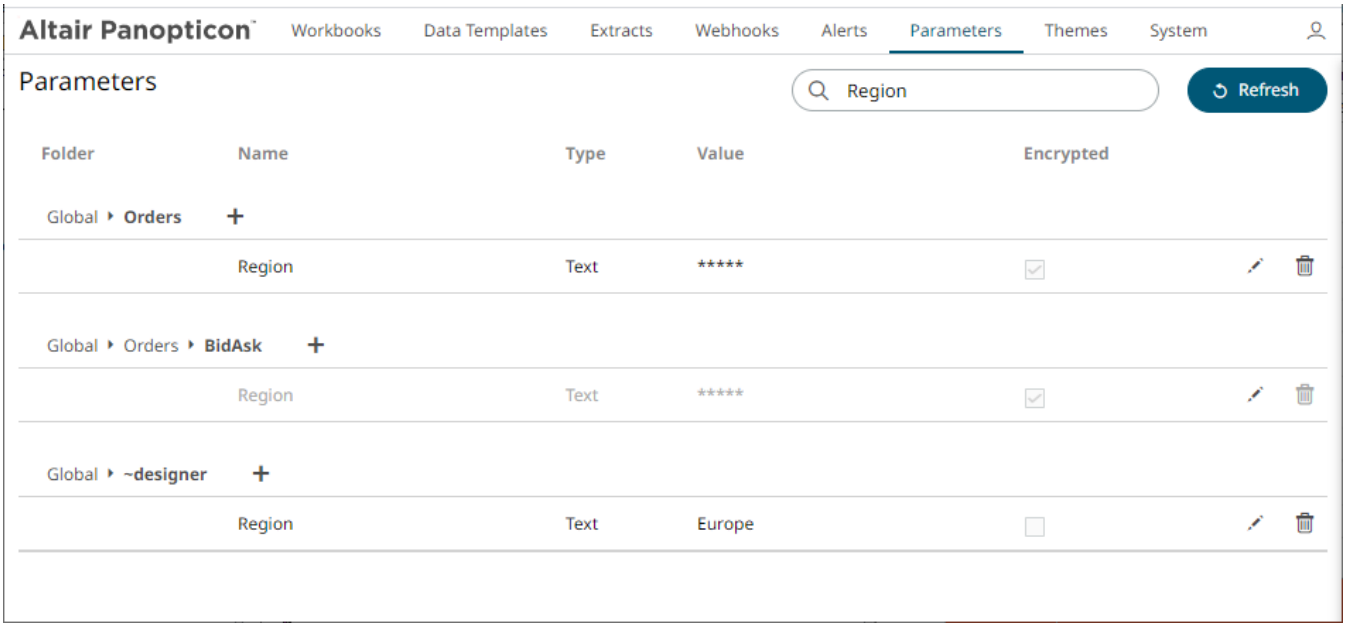
2. Click Yes to delete.

Refresh Parameters

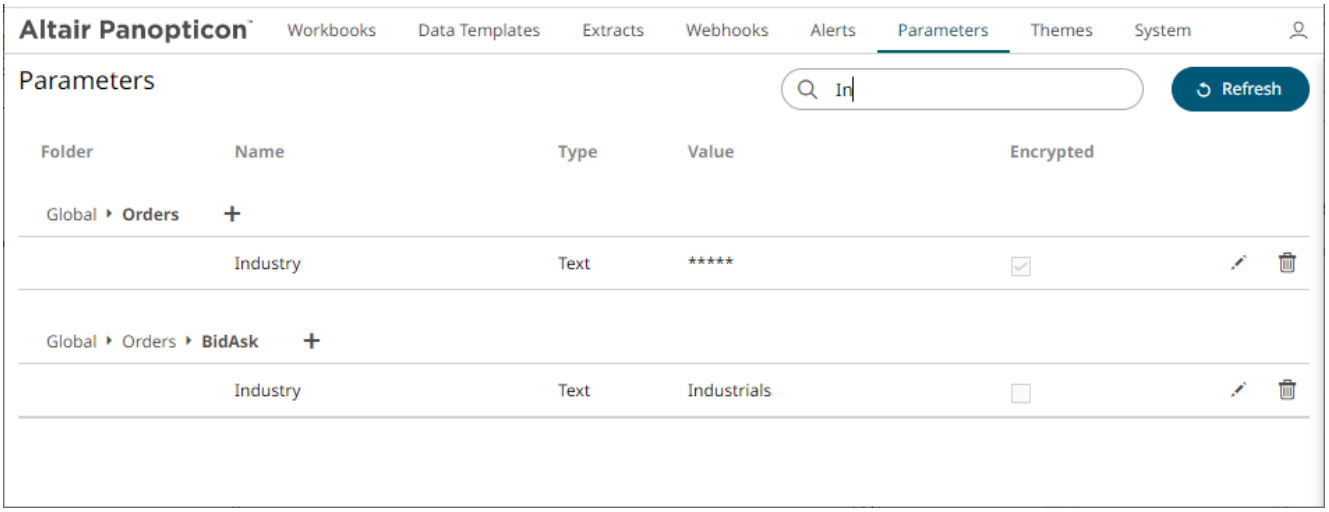
Click Refresh to refresh the values that are being pulled by the workbook models.

Searching Parameters

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



You can also enter one of more characters into the *Search* box and the suggested list of parameters that matched the entries will be displayed.

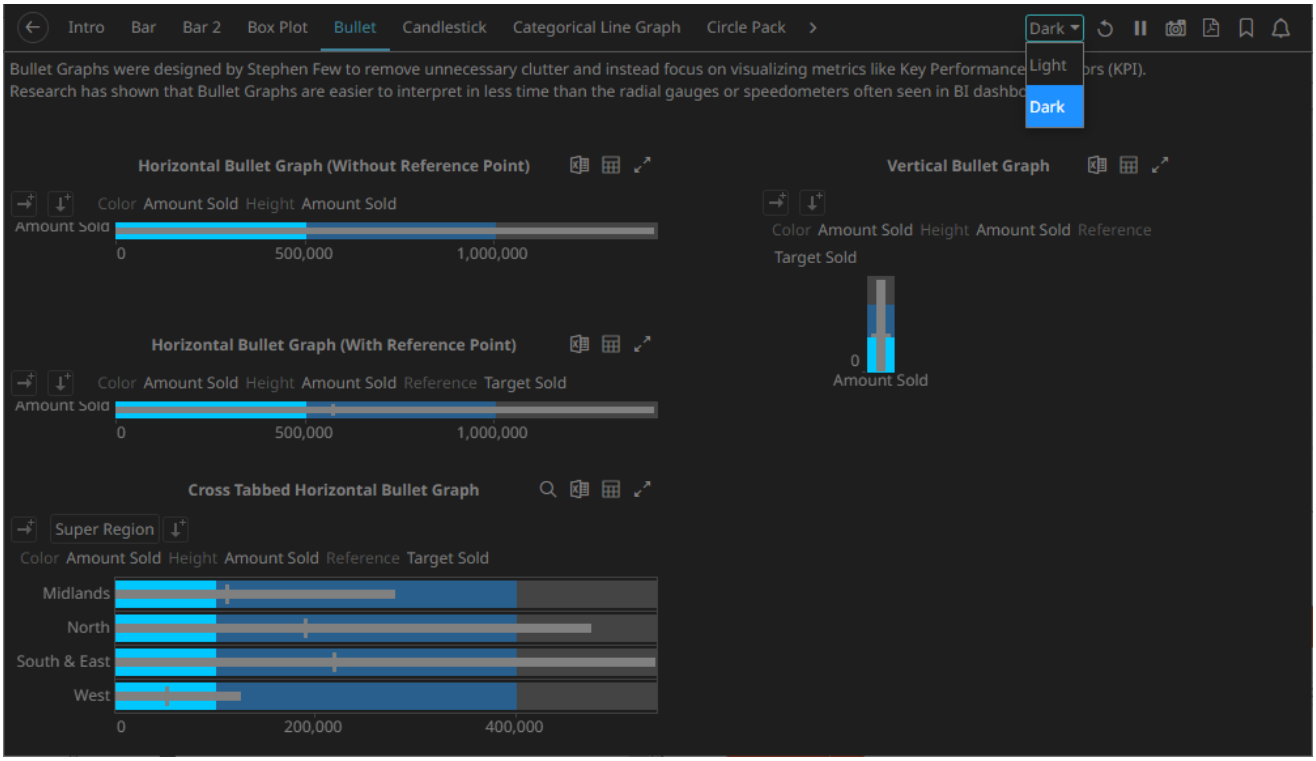

Altair Panopticon
Workbooks
Data Templates
Extracts
Webhooks
Alerts
Parameters
Themes
System
Parameters
In
Refresh
Folder
Name
Type
Value
Encrypted
Global ▸ Orders
Industry
Text
*****
Global ▸ Orders ▸ BidAsk
Industry
Text
Industrials

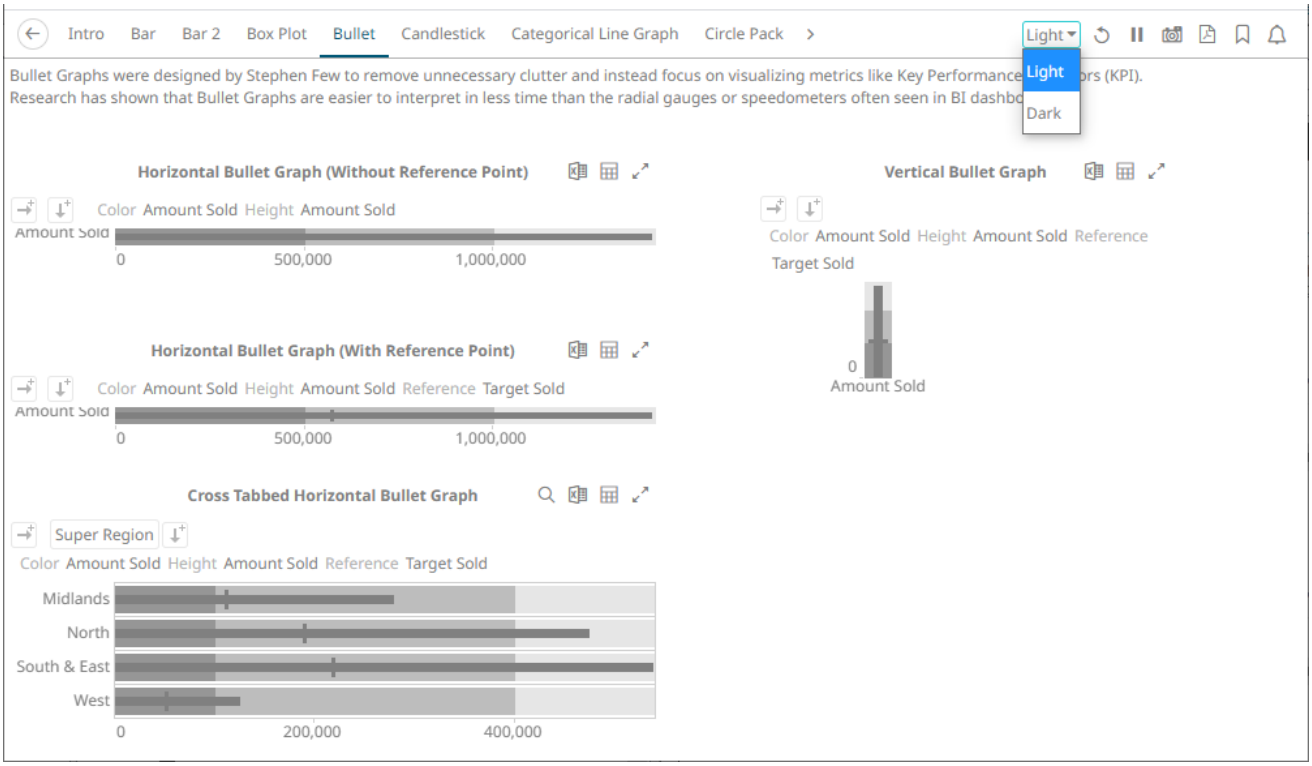
[15] MANAGING WORKBOOK THEMES

Workbook themes are set of configurable settings that affect all colors and fonts of dashboards and visualizations in a workbook. This configuration also includes setting which among the color palettes will be available for the Color variable or shape palettes for the Shape variable in the visualizations. Furthermore, the general colors to be used in visualizations such as axis, background, border, and focus colors can be defined.

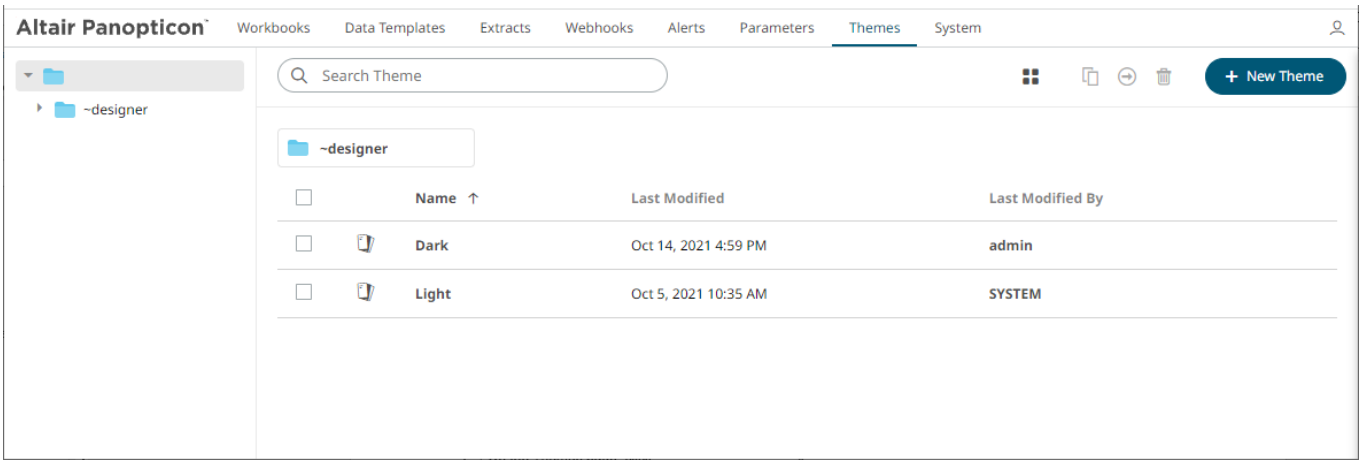
Theme files are independent of workbooks and can be uploaded to and downloaded from the server through the administration UI.

On an opened workbook, users can dynamically switch between the two provided default workbook themes: **Light** or **Dark**.





The **Theme** tab allows management of these workbook themes which are stored in the repository on the server.



NOTE	In the previous versions of Panopticon, all of theme-related settings are part of the workbook style, making it difficult to dynamically switch styles (e.g., colors, fonts etc.) When a workbook (created using versions before 17.5) is opened, all of the existing styles are extracted then saved as its inline workbook theme.

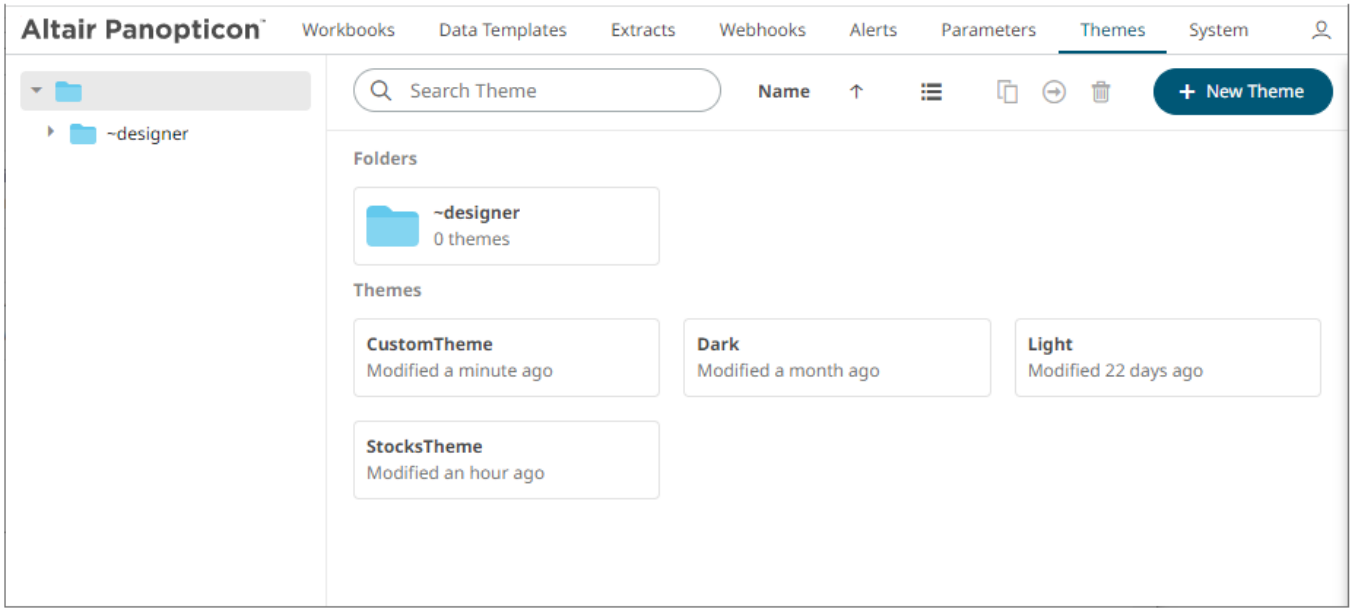
On the **Themes** tab, the following sections are available:

Property	Description
Search Theme	Entering text will filter the themes.
Toolbar	Allows copying, moving, and removing of themes. Also, to display the themes list either on *List View* or *Grid View*.
Create Theme	Allows creating new themes.
Theme Context Menu	Allows uploading, renaming, moving, copying, downloading, and deleting themes.

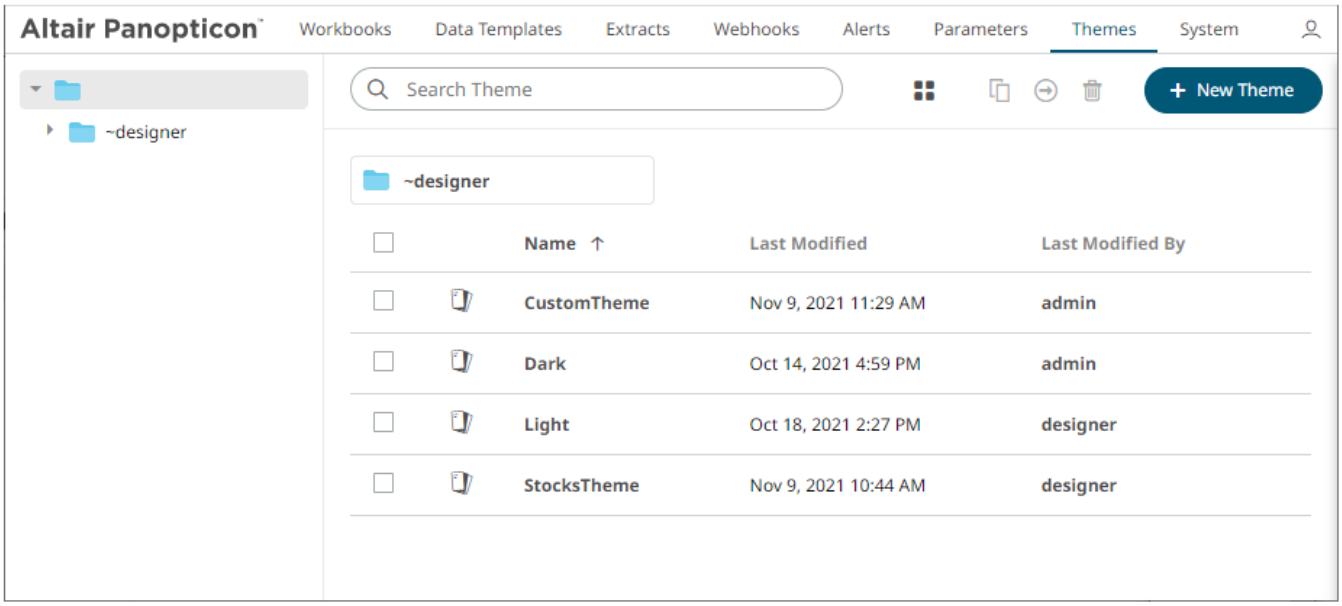
FOLDERS AND THEMES DISPLAY VIEW

Themes can be displayed either on a *List* or *Grid View*.

On the *Toolbar*, click **Grid View** . The folders and themes are displayed as thumbnails.



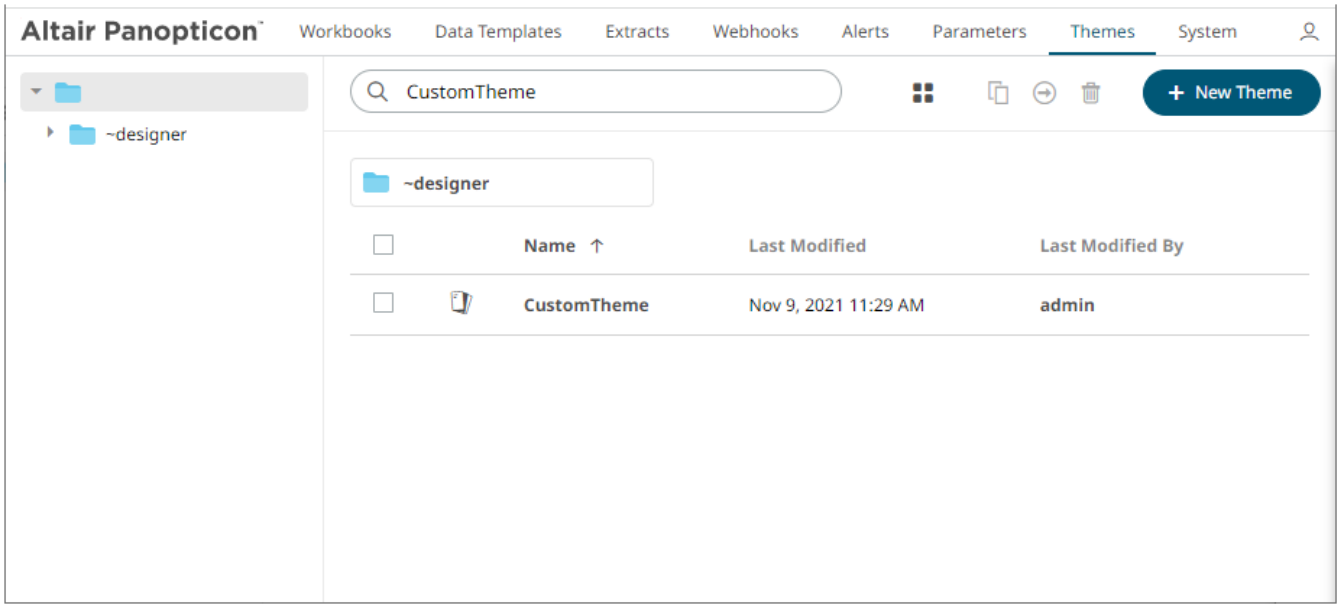
Or click **List View** , the themes are displayed in a standard listing.



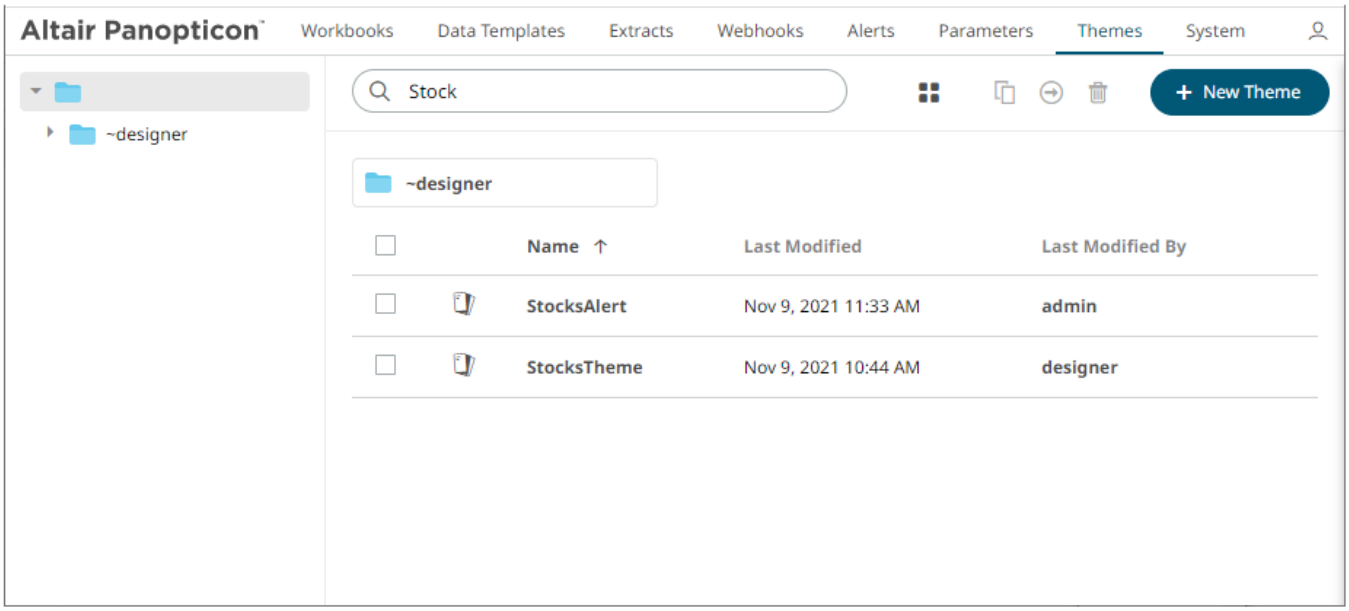
On either display view style, clicking on a themes title or thumbnail displays the *Theme* page.

SEARCHING FOR THEMES

On the *Themes* tab, to search for a particular theme, enter it in the *Search Theme* box.



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



Click on a theme to open the settings page.

To clear the filter, delete the text entry in the *Search Theme* box.

CREATING A NEW THEME

Creating a new theme allows setting the colors, fonts, color palettes, general colors, and shape palettes to be used in workbooks and visualizations.

Steps:

1. On the *Themes* page, click 

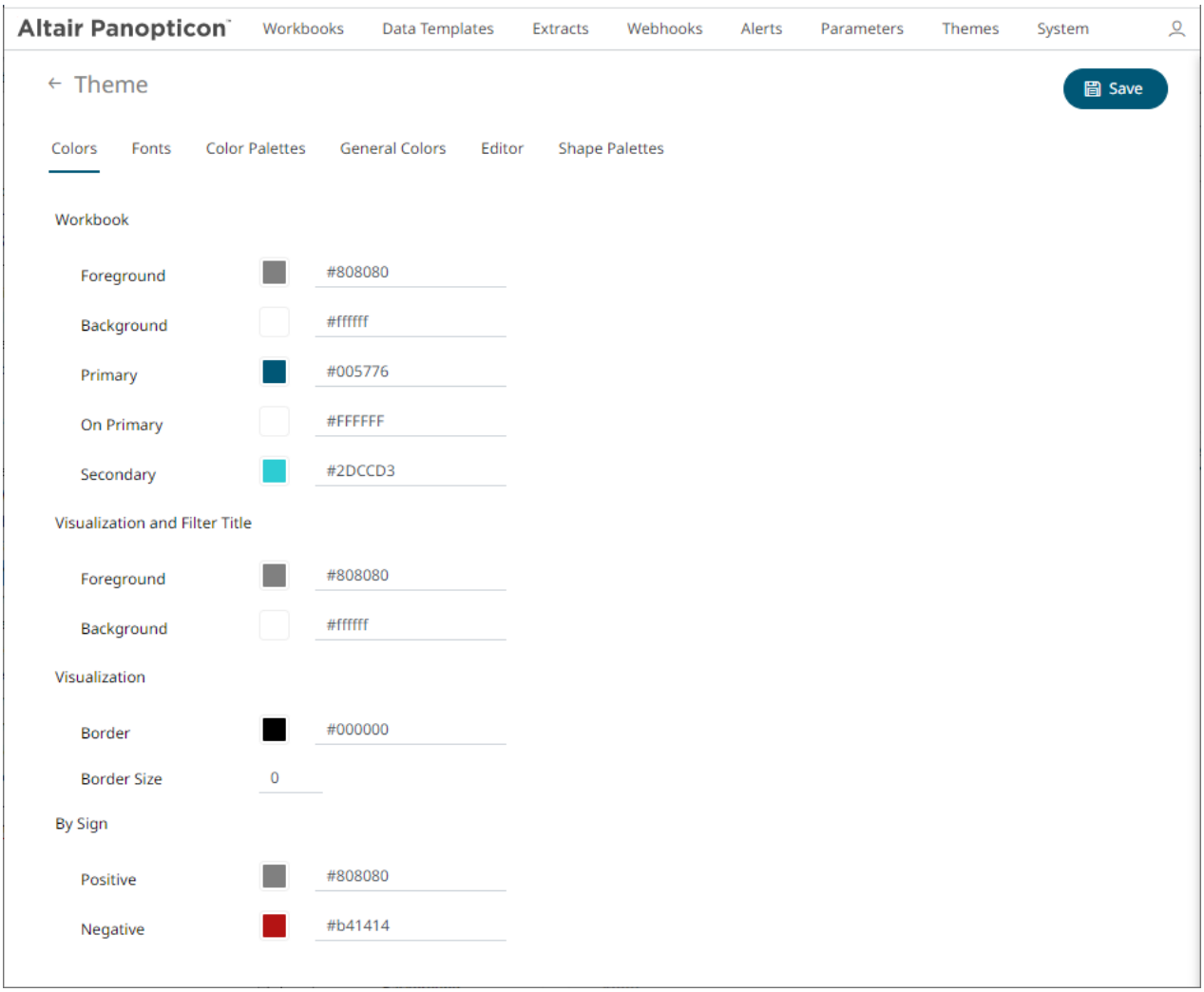
.

   The *New Theme* dialog displays.

2. Enter the name of the theme then click 

.

   The new theme is displayed on the *Themes* page.

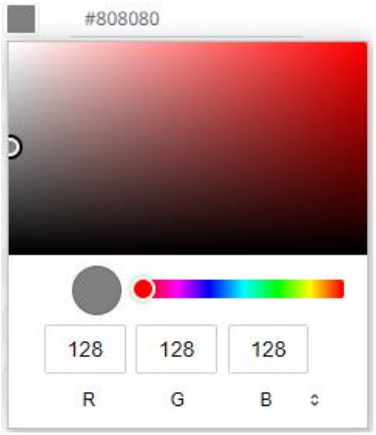


3. On the **Colors** tab, you are allowed to modify the colors of the following properties:

Property	Description
Foreground	Foreground color of the workbook.
Background	Background color of the workbook.
Primary	Primary color of the workbook.
On Primary	Foreground color within the primary color.
Secondary	Secondary color of the workbook.
Title Foreground	Foreground color of the title in visualizations and filters.
Title Background	Background color of the title in visualizations and filters.
Border	Border color of the visualizations.
Positive	Color of the positive values for the *By Sign* option used in numeric visual members.
Negative	Color of the negative values for the *By Sign* option used in numeric visual members.

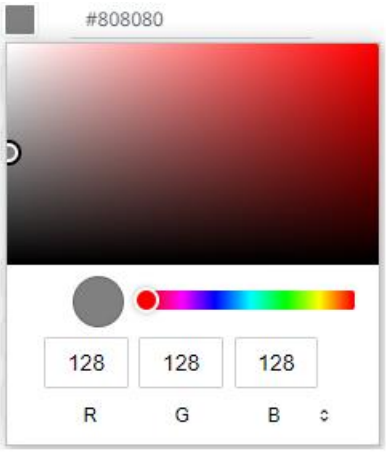
you can either:

- click the corresponding *Color* box to display the *Color* dialog to:

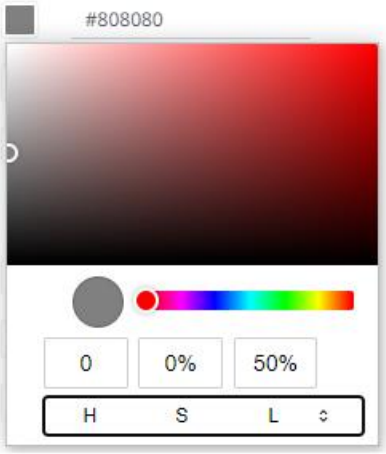


  - select the color, or
  - click to enter the values

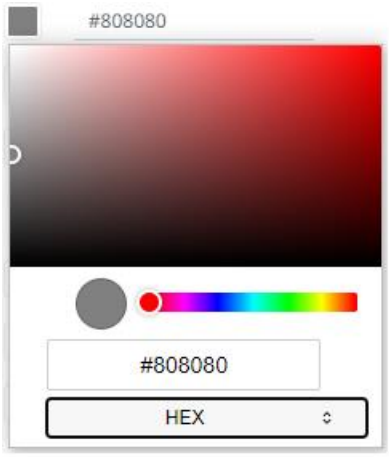
    for RGB



    for HSL

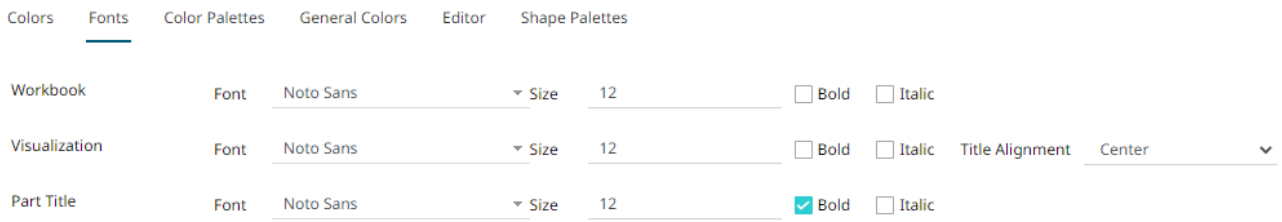


for the Hex color code



- or enter the Hex color code

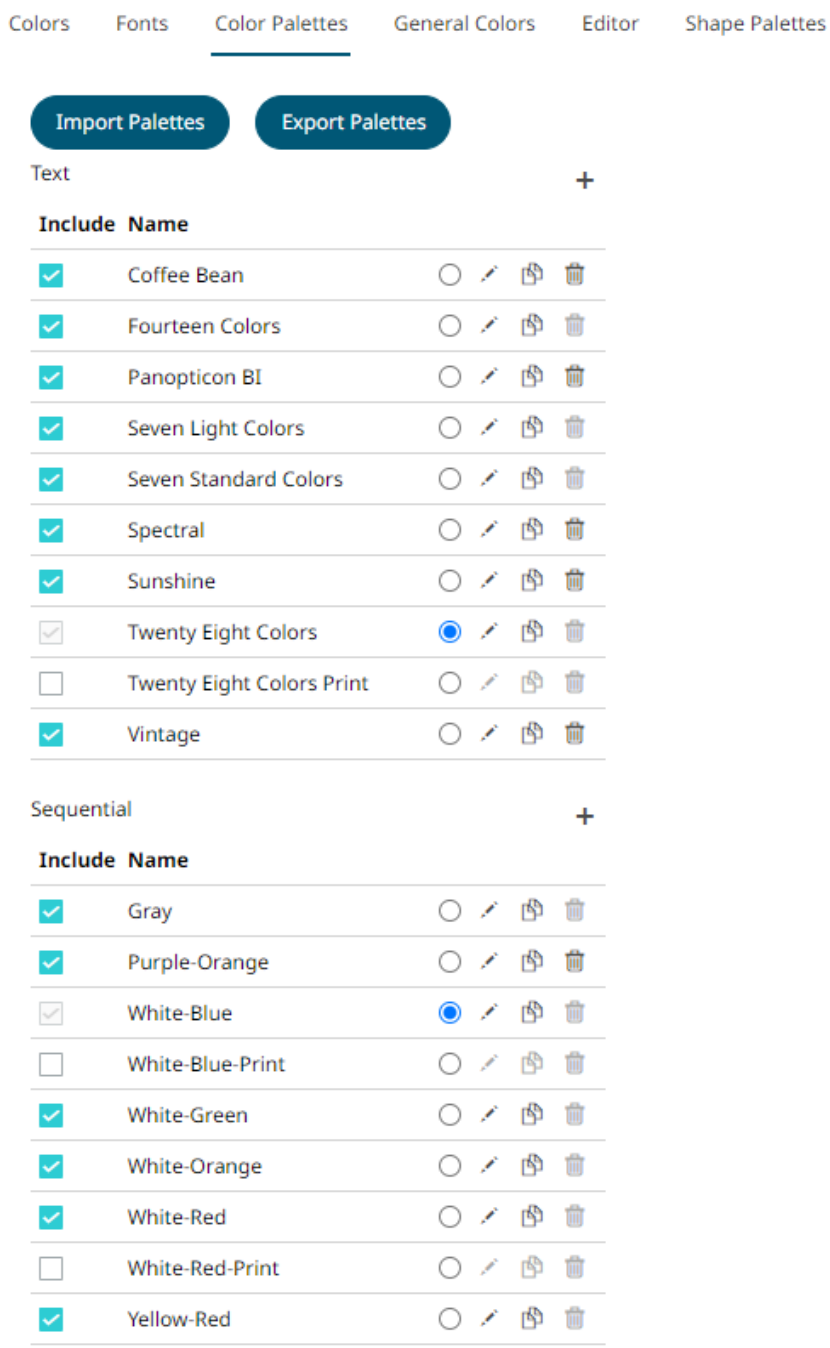
4. Enter the *Border Size* of the visualizations.
5. To set the fonts to be used, click the **Fonts** tab:

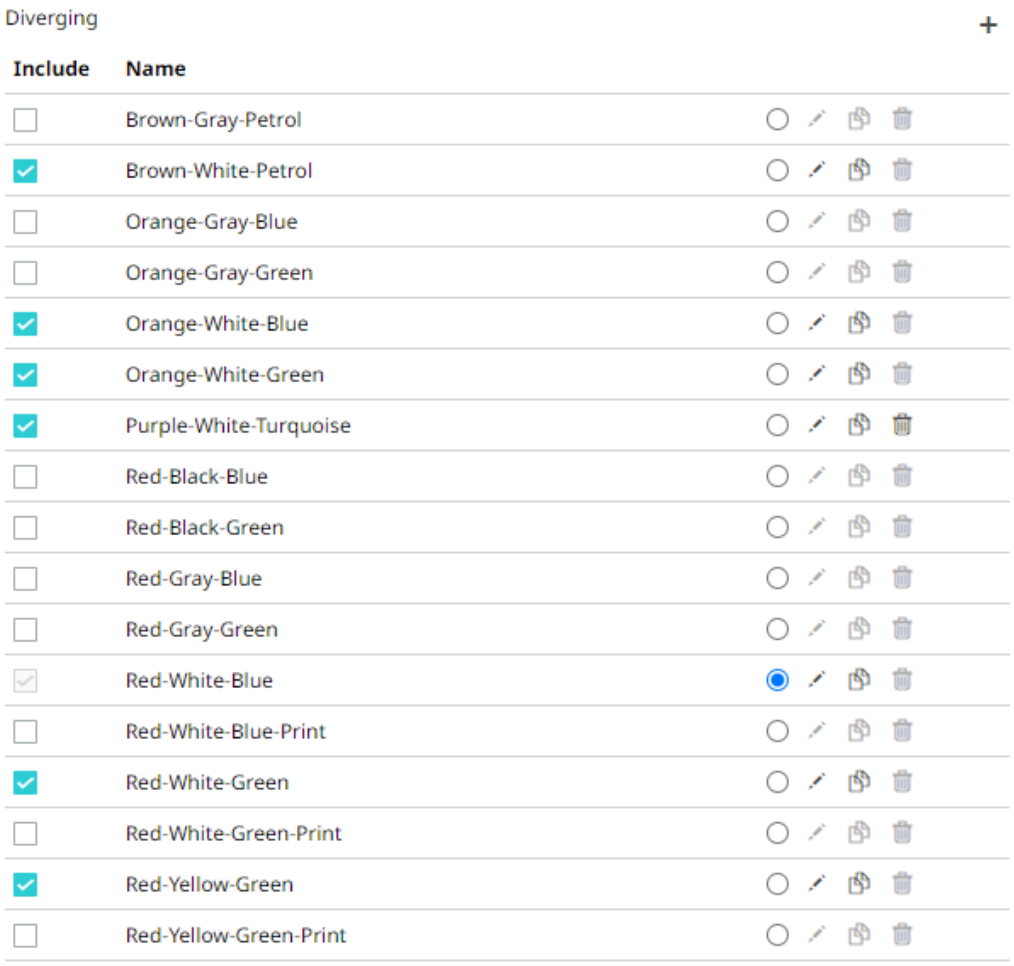


6. For the workbooks, visualization and part titles, enter or select the preferred font *Type* and *Size* and check the *style* boxes: **Bold** and **Italic**.

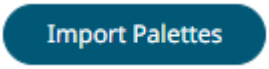
NOTE	• The available custom fonts in Panopticon Real Time can be selected on the *Font* drop-down list. • For the *Part Title*, the **Bold** check box is selected by default.

7. Select the visualization title *Alignment*: **Left** or **Center**.
8. To select the *Diverging*, *Sequential*, and *Text* color palettes to use within the workbooks, click the **Color Palettes** tab.

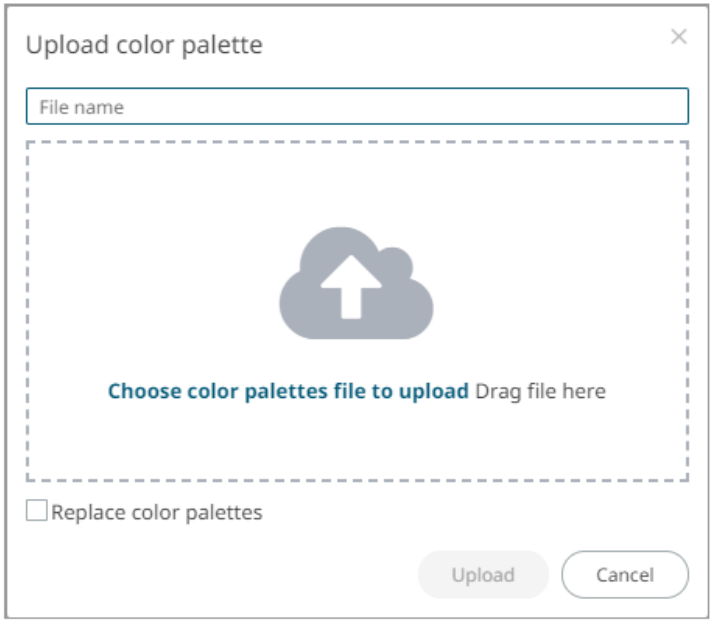

Colors
Fonts
Color Palettes
General Colors
Editor
Shape Palettes
Import Palettes
Export Palettes
Text
Include
Name
Coffee Bean
Fourteen Colors
Panopticon BI
Seven Light Colors
Seven Standard Colors
Spectral
Sunshine
Twenty Eight Colors
Twenty Eight Colors Print
Vintage
Sequential
Include
Name
Gray
Purple-Orange
White-Blue
White-Blue-Print
White-Green
White-Orange
White-Red
White-Red-Print
Yellow-Red



NOTE	For more information on how to create, modify, duplicate, or delete Text, Sequential, or Diverging Palettes, refer to the sections below.

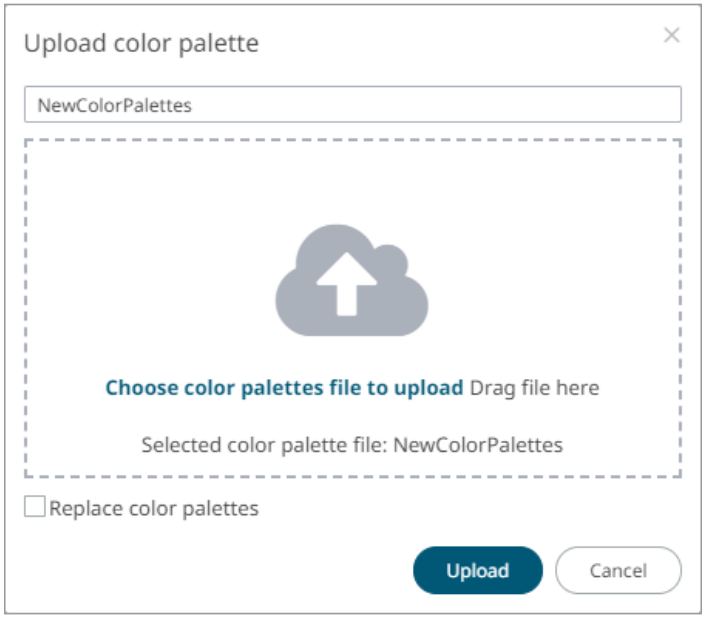
9. Check the boxes of the provided color palettes that will be included for each category.
10. Click the radio button of the preferred *Default* color palette for each category.
11. To upload color palettes, click 

. The *Upload Color Palette* dialog displays.



12. To upload a color palette, either:
    - drag the file from your desktop and drop on the dialog, or
    - click **Choose color palettes file to upload** and then browse and select one on the *Open* dialog that displays

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

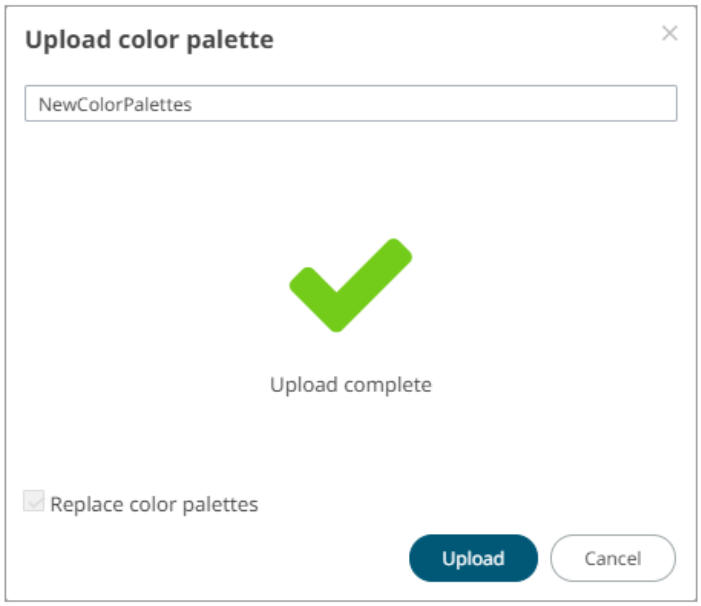


    You can opt to rename the uploaded color palette.

13. To replace the color palettes, check the *Replace Color Palettes* box.
14. Click 

.

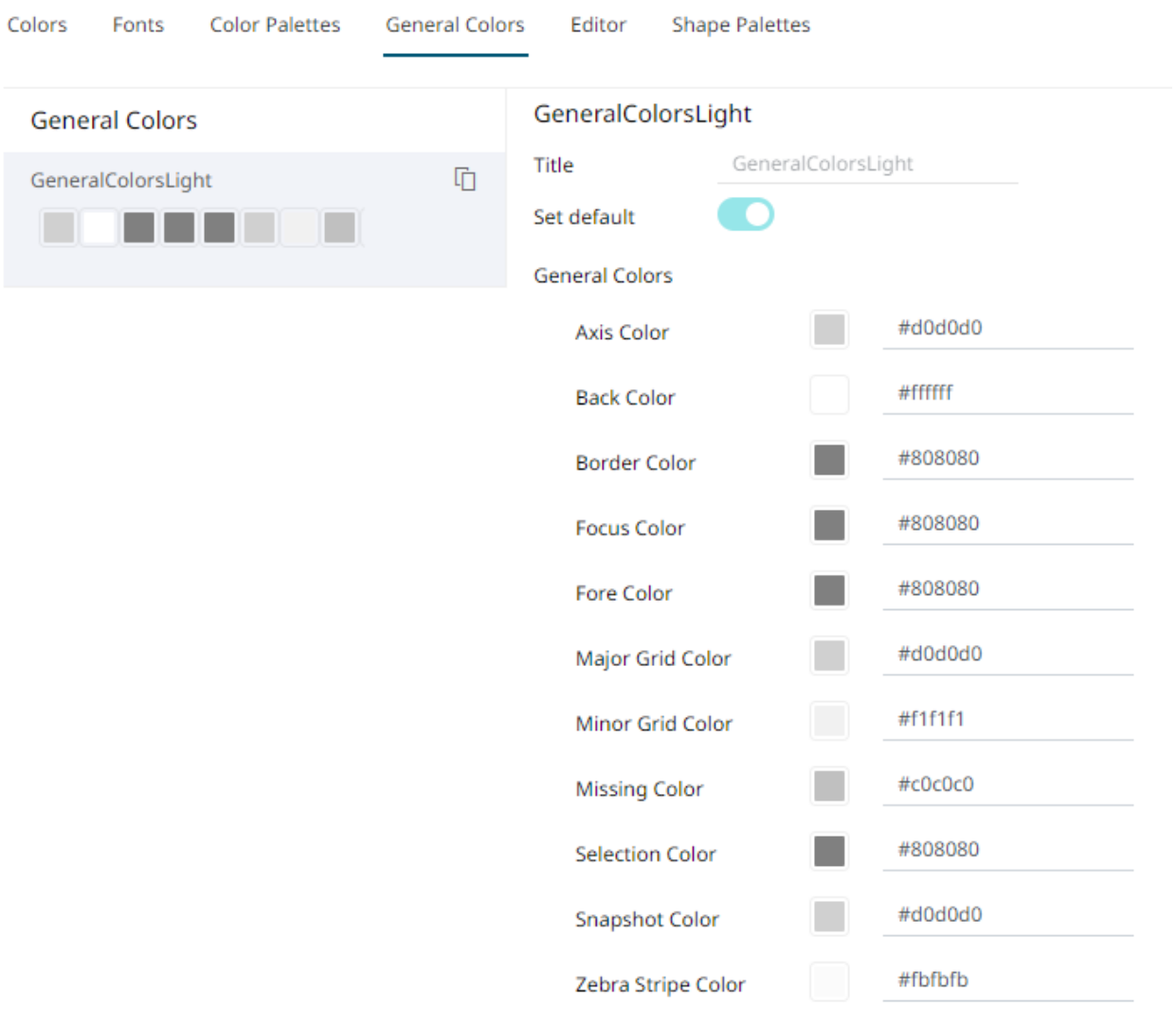
    A notification displays once the color palettes file is uploaded.



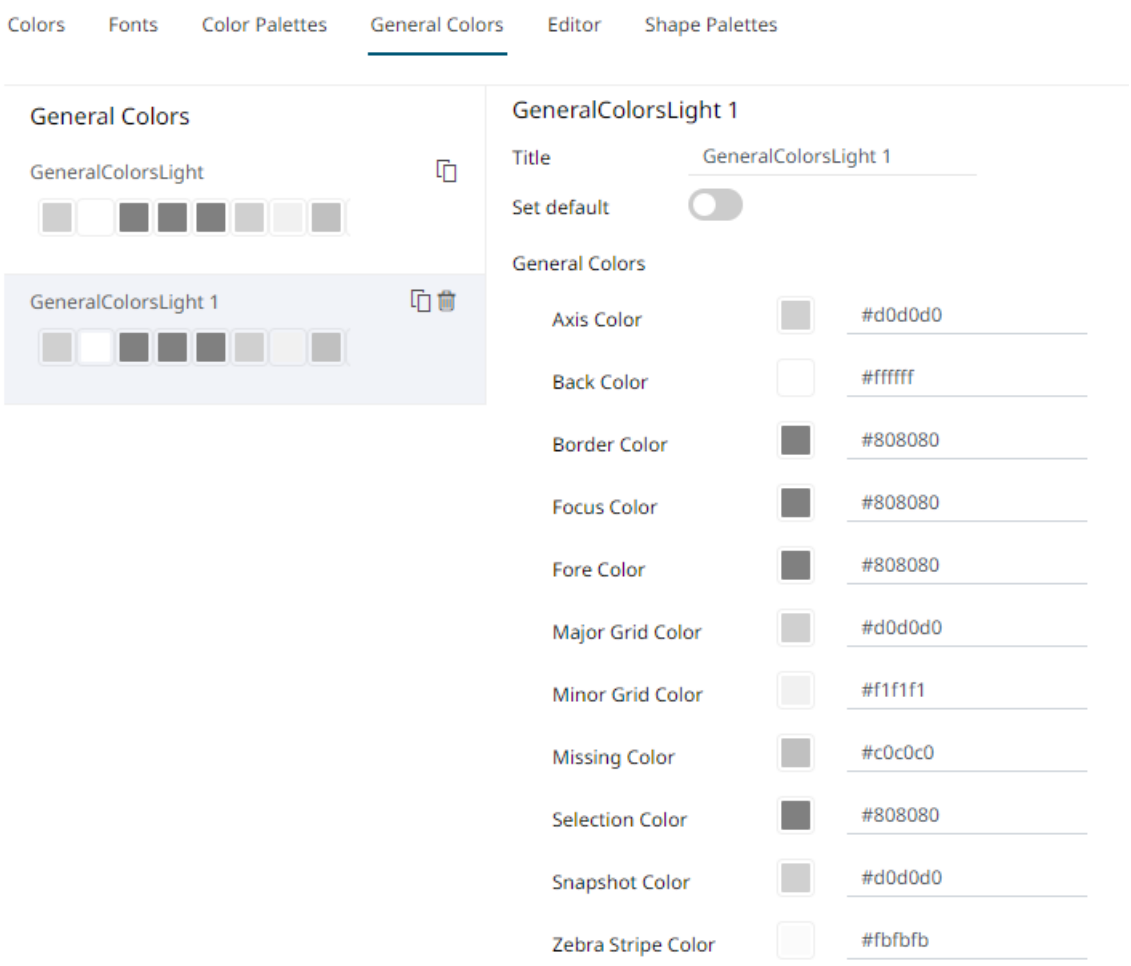
15. To export color palettes, click 

. The `.excp` file is exported. You can now move this file to the desired location.
16. To set the general colors to be used for visualizations, click the **General Colors** tab.

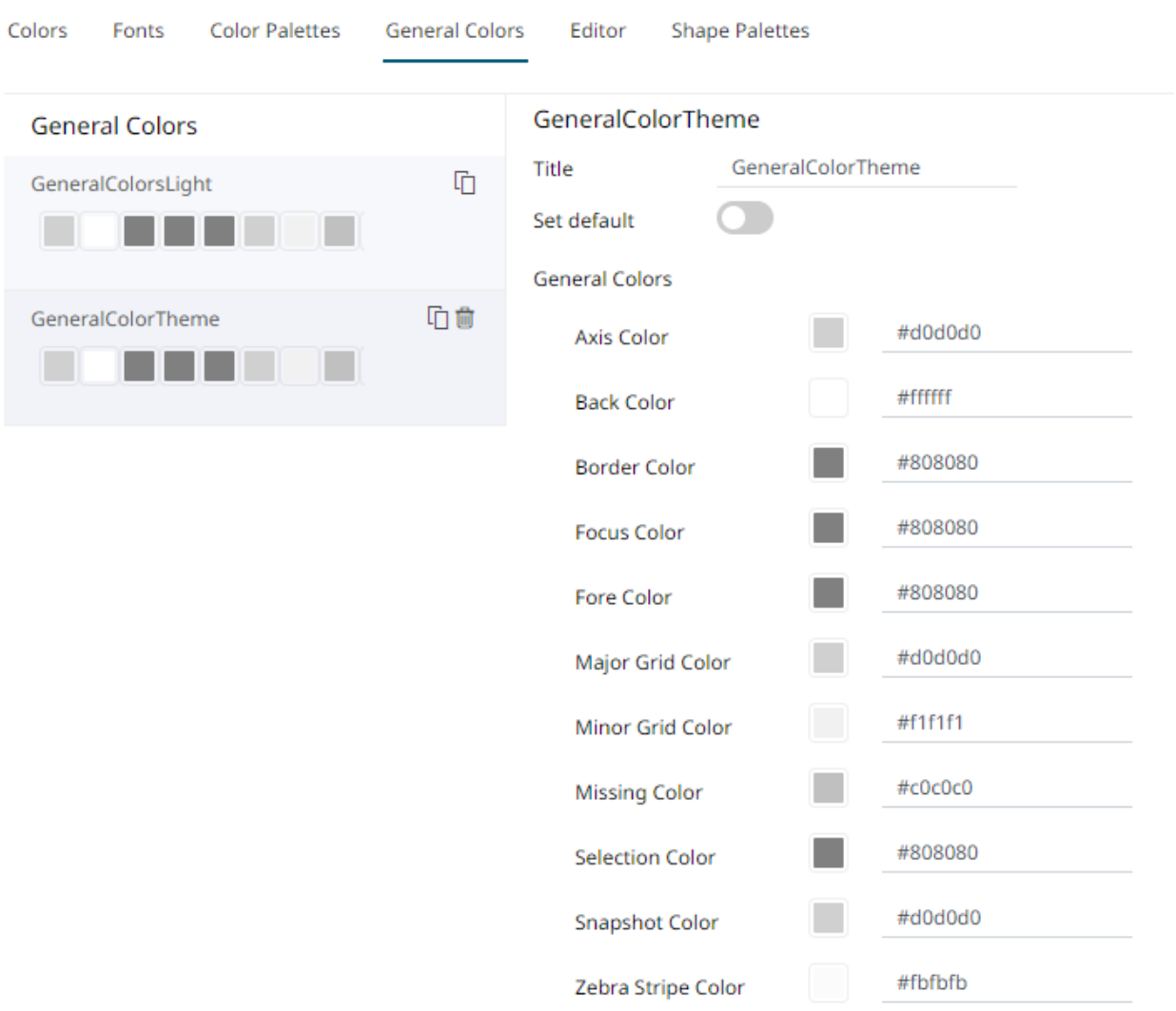
    By the default, the new *General Colors* is named **GeneralColorsLight**.



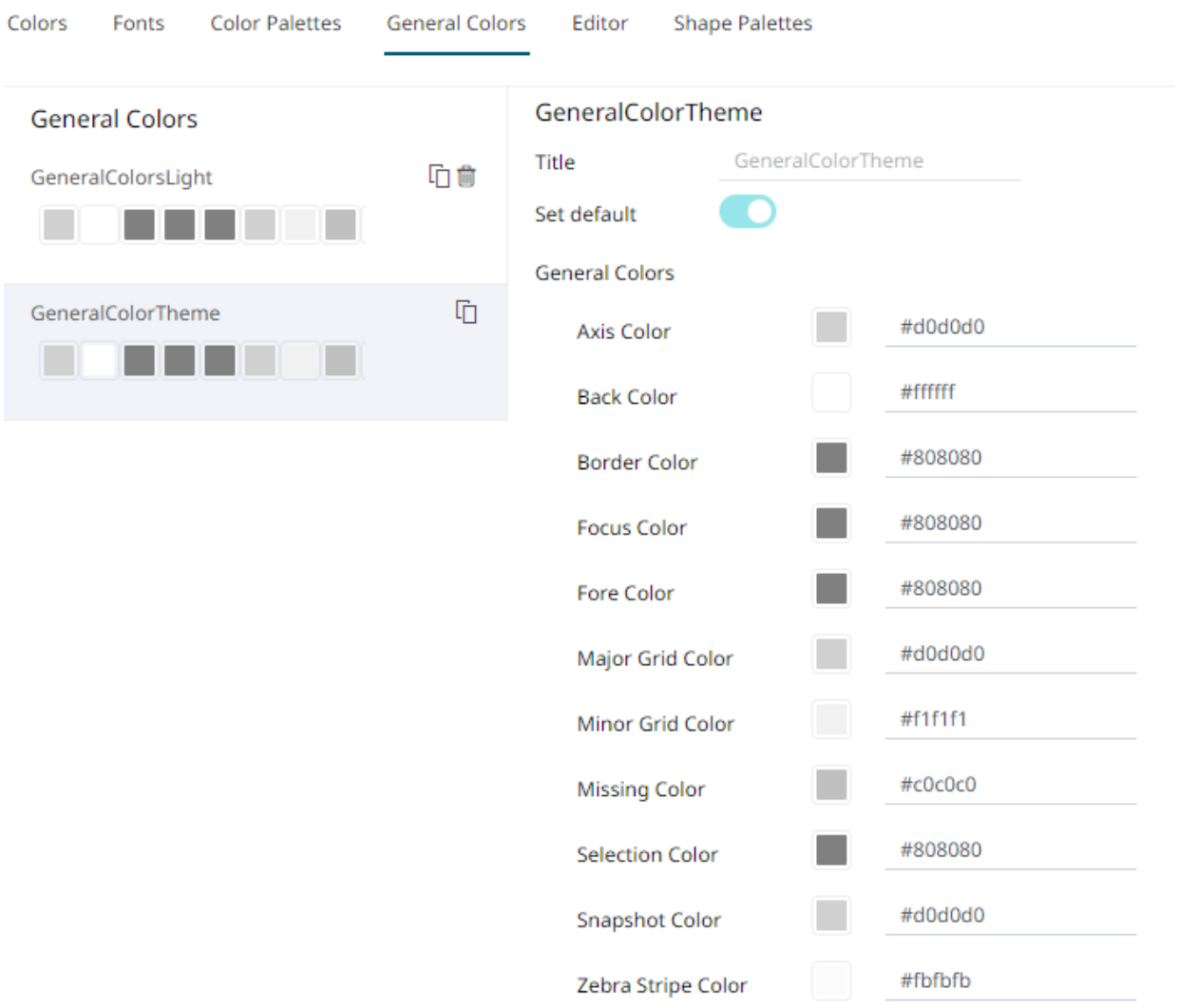
17. Click **Duplicate** to make a duplicate copy of the new general colors.



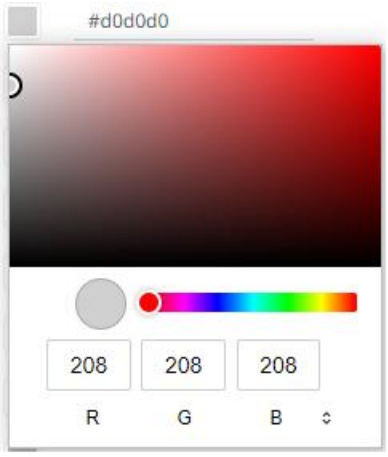
18. You can enter a new name and click ✓. **Set Default** is turned off and the **Remove** icon is now available.



Tap the **Set Default** slider to turn it on and the **Remove** icon is no longer available.



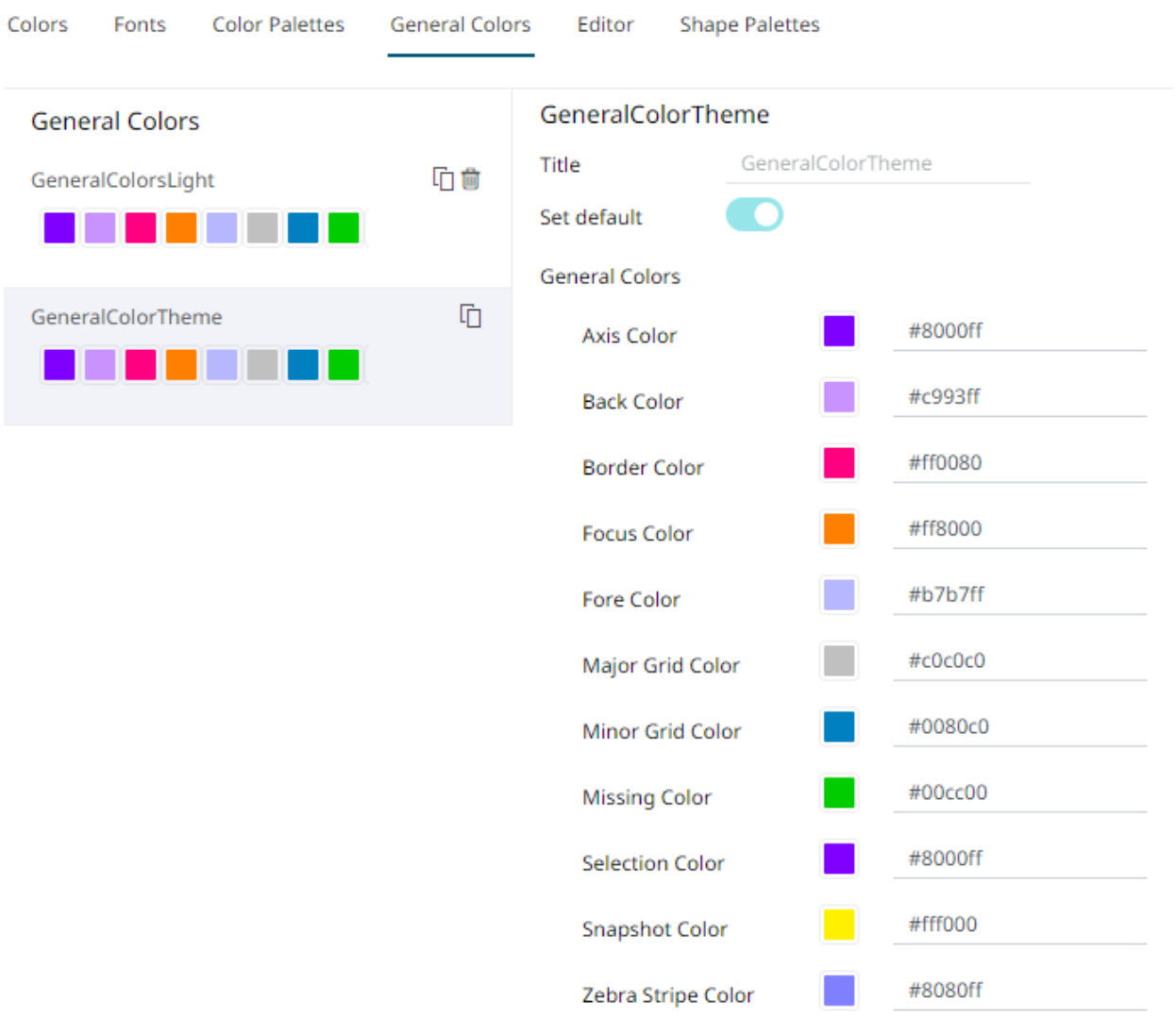
19. Click any of the color boxes to display the *Color* dialog.



Select or specify the new general colors: *AxisColor*, *BackColor*, *BorderColor*, *FocusColor*, *ForeColor*, *MajorGridColor*, *MinorGridColor*, *MissingColor*, *SelectionColor*. *SnapshotColor*,*ZebraStripeColor*.

Or enter the corresponding *Hex* color code.

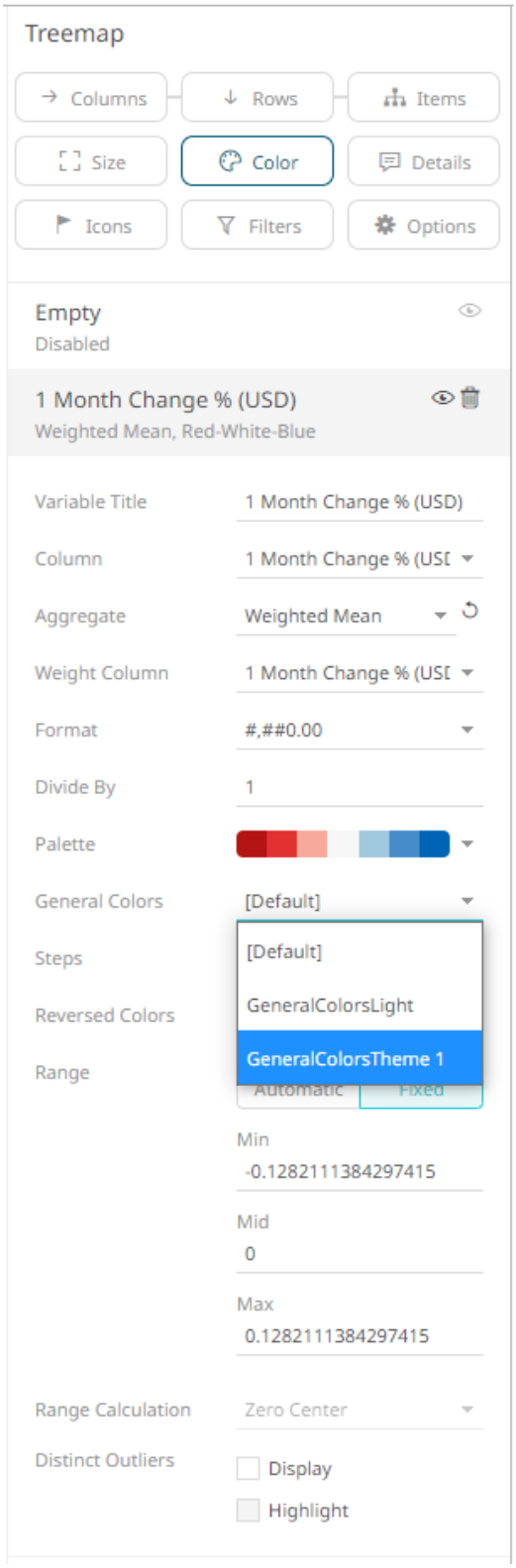
For example:



Repeat steps 17 to 19 to add more general colors.

Once the new theme is saved and selected in the opened workbook, all of the defined *General Colors* will be added as options in the *General Colors* drop-down list of a *Color* variable in a visualization.

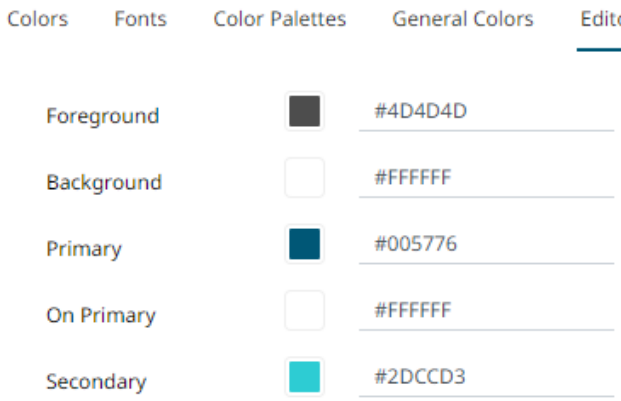
For example:



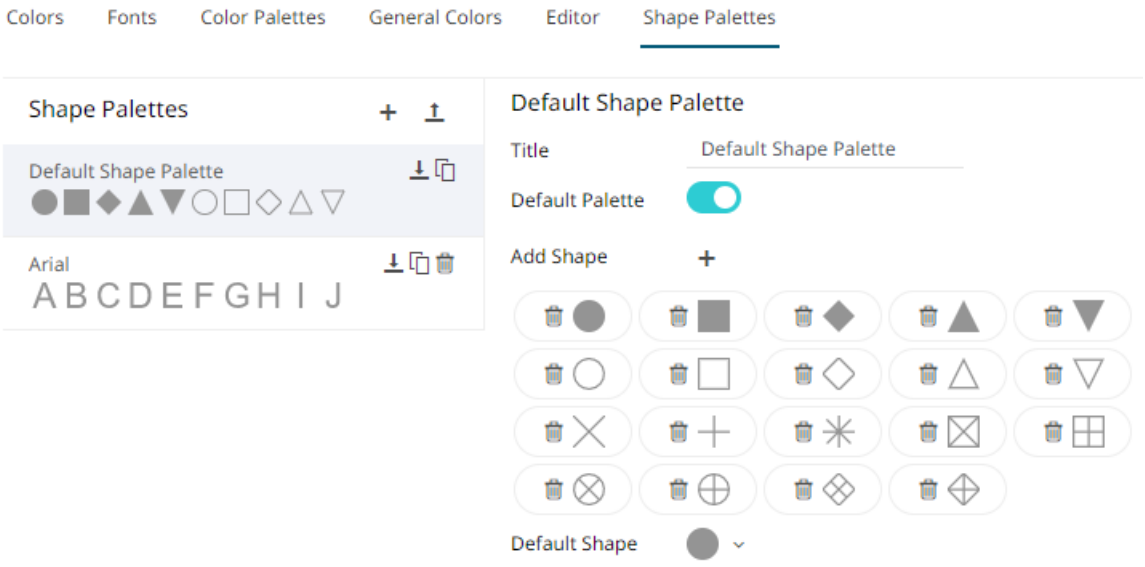
20. Select any of the general colors and tap the **Set Default** slider to make it the default.

21. Select any of the general colors that is not set as the default, and click **Delete** to remove.

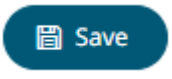
22. To set the *Foreground*, *Background*, *Primary*, *On Primary*, and *Secondary* colors for the editor style of the **Dark** theme, click the **Editor** tab.



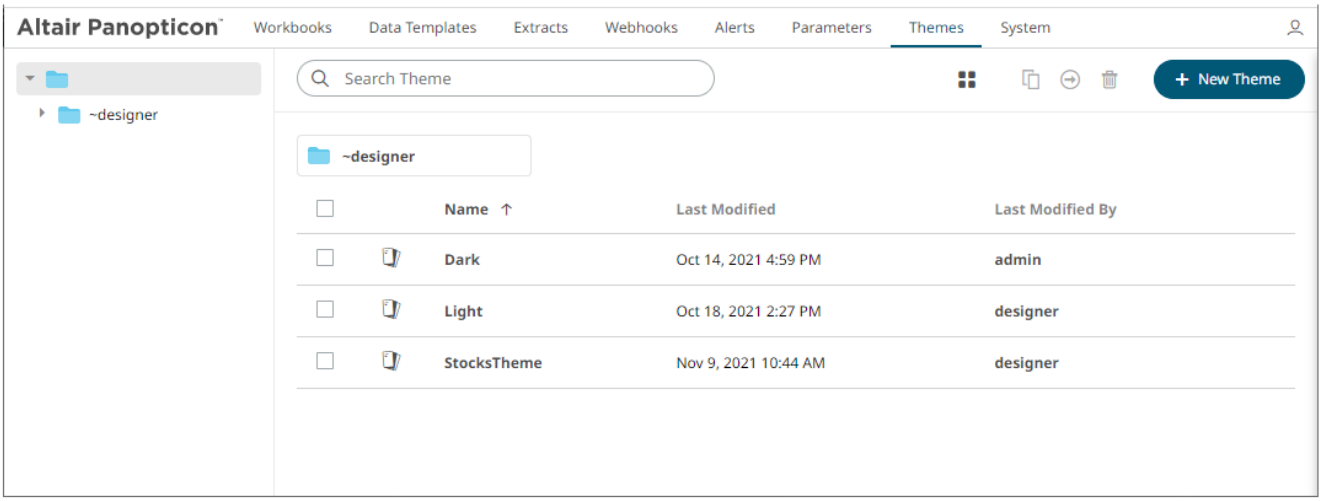
23. Click on any of the color boxes to display the *Color* dialog and select or enter the preferred color.
24. To set the shape palettes that can be used with the workbook theme, click the **Shape Palette** section to expand.



NOTE	For more information in how to create, upload, download, modify, duplicate, or delete shape palettes, refer to the sections below.

25. Click **Save**  to save the new theme.

    Clicking the ← displays the *Themes* page with the new theme added in the list.



Modifying Themes

The colors, fonts, color palettes, shape palettes, and general colors to be used in workbooks and visualizations can be modified on the *Themes* page.

Steps:

1. On the *Themes* page, click the theme to be modified.

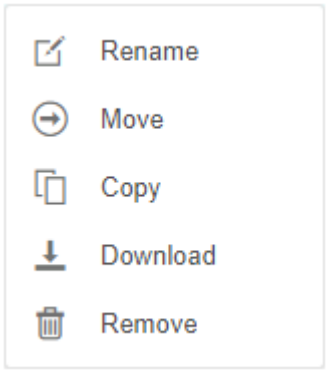
   The corresponding *Theme* page is displayed.

2. Follow steps 2 to 25 in Creating a New Theme to modify any of the properties of the theme.

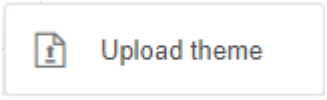
THEMES TOOLBAR AND CONTEXT MENU

Moving, copying, and removing themes can either be done using:

- Context menu

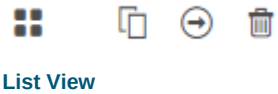


Theme Context Menu



Theme Folder Context Menu

- Toolbar



List View

Grid View

The toolbar options include:

Toolbar Option	Description
Sort By / Sort Order	Allows sorting of themes by *Name, Last Modified,* or *Last Modified By*.
Display View	Display themes either by *List View* or *Grid View*.
Copy	Copy themes to another folder or subfolder where the user has permission.
Move	Move themes to another folder or subfolder where the user has permission.
Remove	Remove themes.

The context menu options include:

Toolbar Option	Description
Upload Theme	Upload theme.
Rename	Rename the theme.
Move	Move themes to another folder or subfolder where the user has permission.
Copy	Copy themes to another folder or subfolder where the user has permission.
Remove	Remove themes.

Sorting Themes

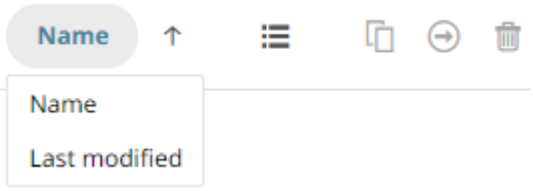
Sorting themes can be done by **Name, Last Modified**, or **Last Modified By**.

Steps:

On the *Themes* tab, either:

- click the **Sort By** option on the *Toolbar* of the *Grid View.*

  By default, the sorting is by **Name**.

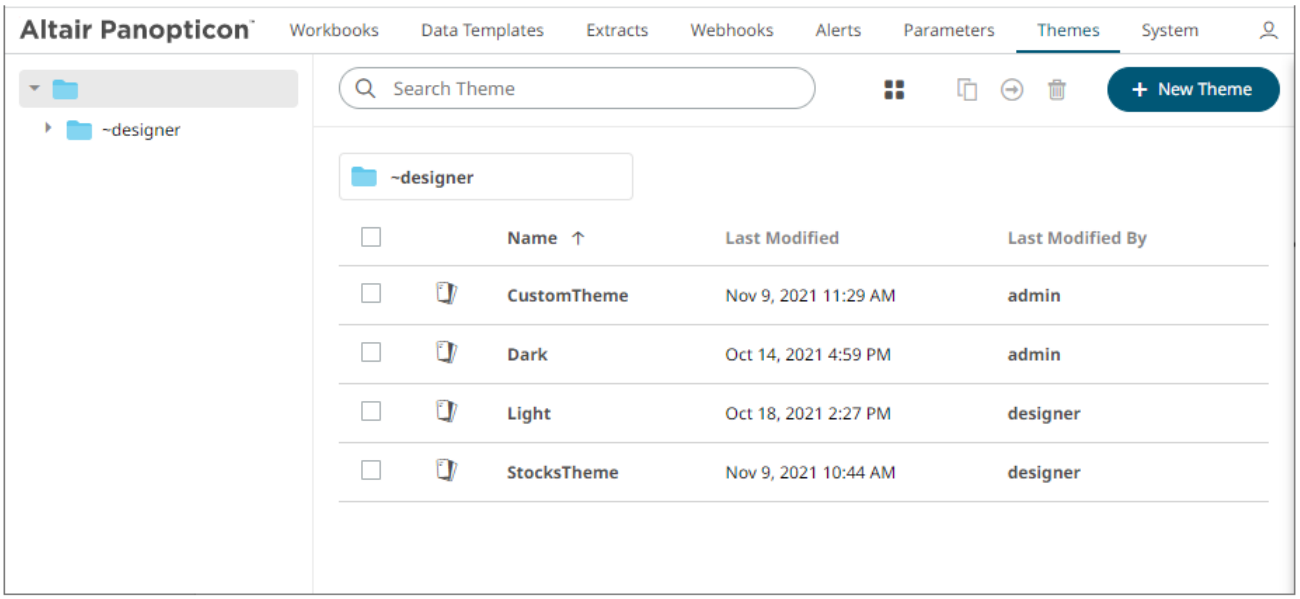
  

  - Name
  - Last Modified

  Then click the *Sort Order*:

  - Ascending
  - Descending

- click on the **Name**, **Last Modified**, or **Last Modified By** column header of the *List View.*

  Then click the *Sort Order*:

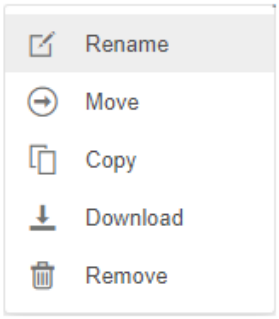
  - Ascending

- Descending

Renaming a Theme

Steps:

1. Right-click on a theme then select **Rename** on the context menu.



The *Rename Theme* dialog displays.

2. Enter a new name then click 

.

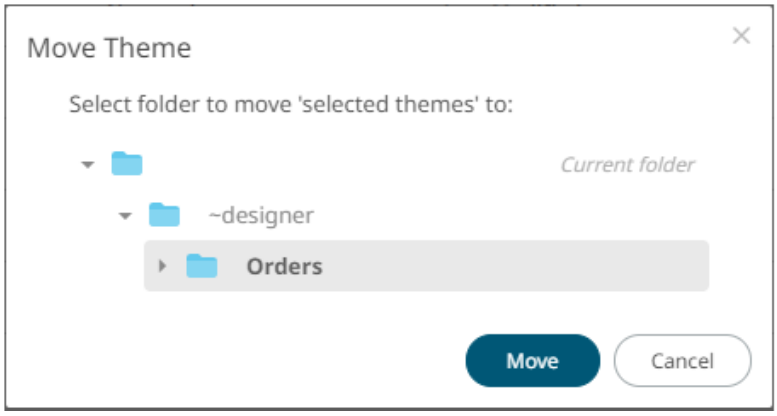
Moving Themes

Users with a Designer role are allowed to move themes to another folder or subfolder where they have permission.

Steps:

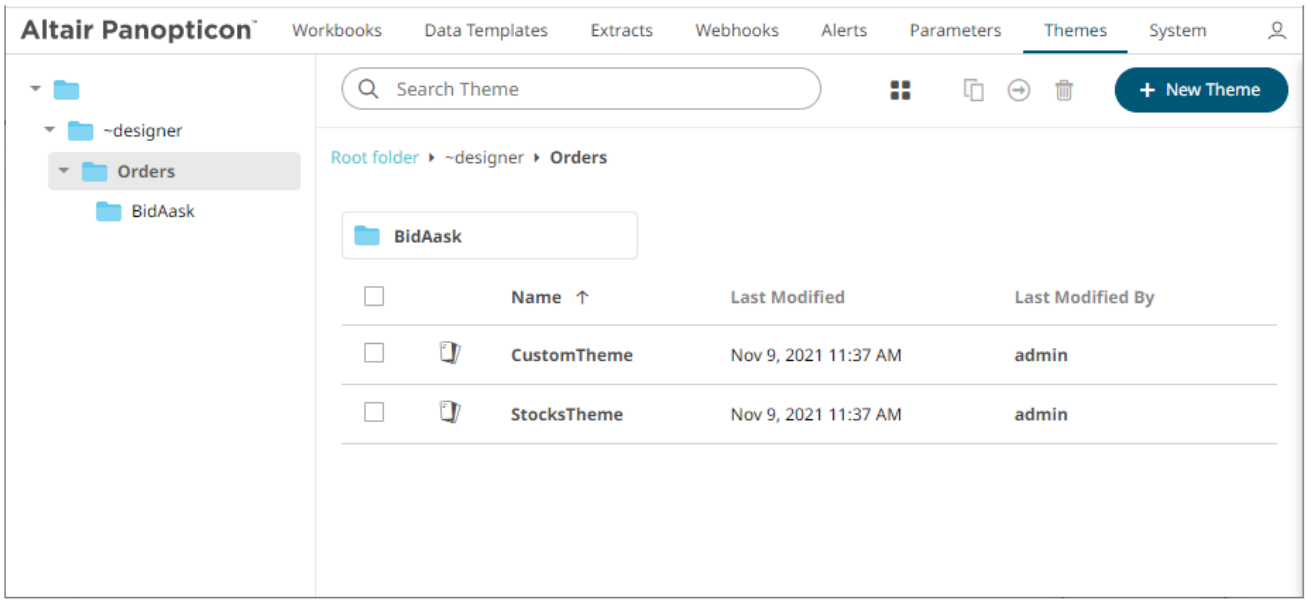
1. On the *List* or *Grid* view, select one or several themes then:
   - right-click and select **Move** on the context menu, or
   - click the **Move**  icon on the toolbar.

   The *Move Theme* dialog displays with the folder or subfolders that the user is allowed to move the themes. Select the folder or subfolder.



2. Click Move.

   The themes are moved and displayed on the selected folder.



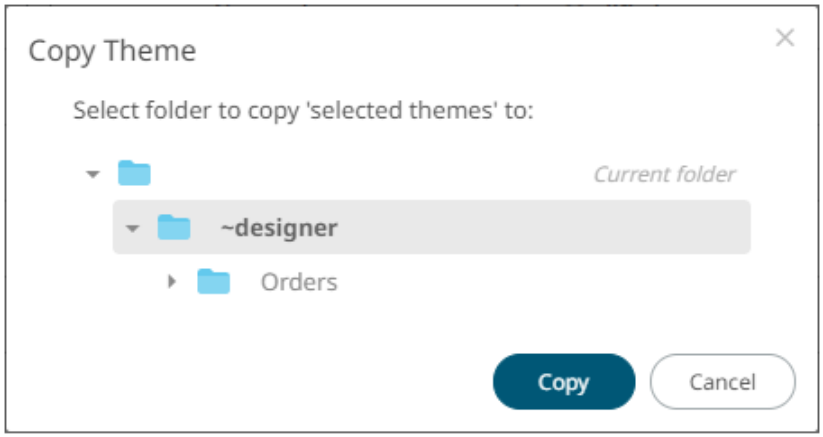
Copying Themes

Users with a Designer role are allowed to copy themes to another folder or subfolder where they have permission.

Steps:

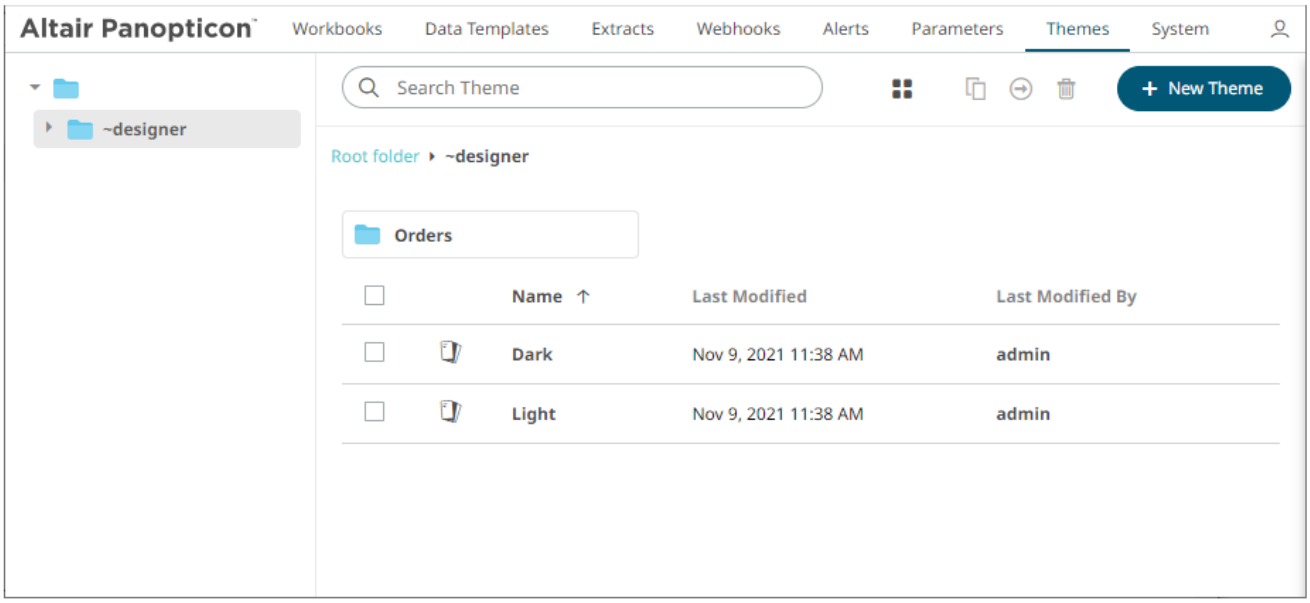
1. On the *List* or *Grid* view, select one or several themes then:
   - right-click and select **Copy** on the context menu, or
   - click the **Copy** icon on the toolbar.

   The *Copy Theme* dialog displays with the folder or subfolders the user is allowed to copy the themes to. Select the folder or subfolder.



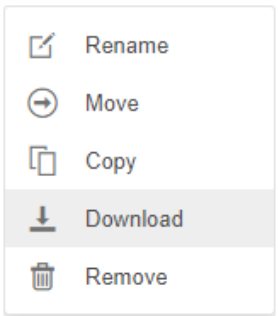
2. Click Copy.

The themes are copied and displayed on the selected folder.



Downloading Themes

On the *List* or *Grid* view, right-click on a theme and selected **Download** on the context menu to download a copy.



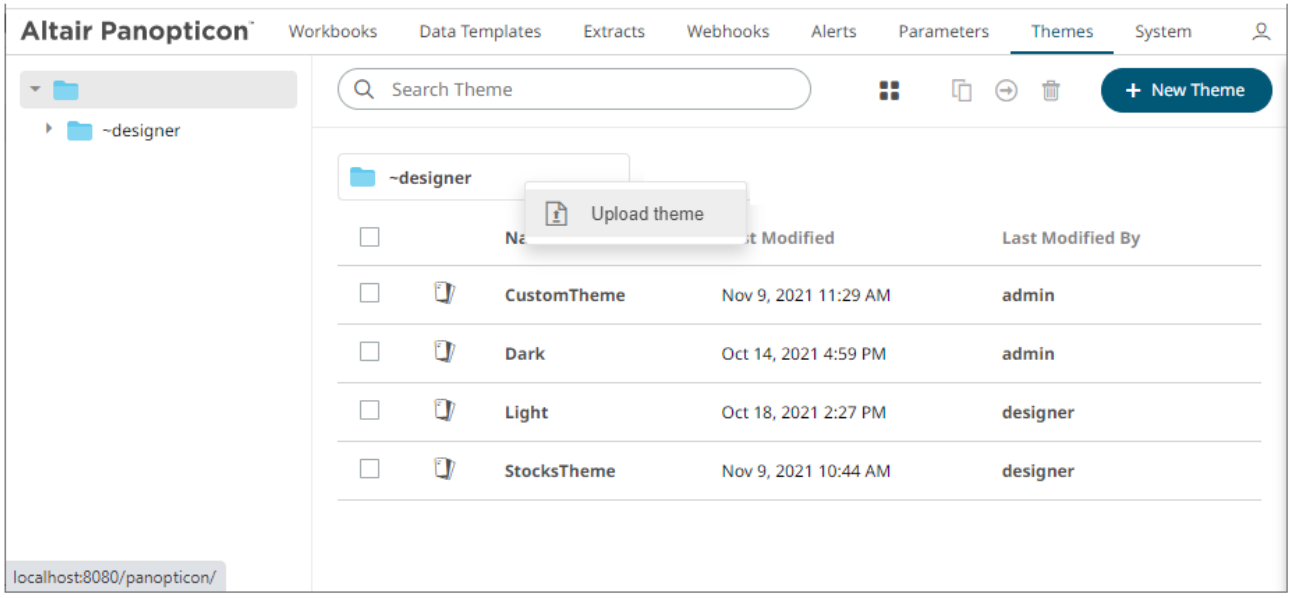
You can copy this file to the desired location.

Uploading Themes

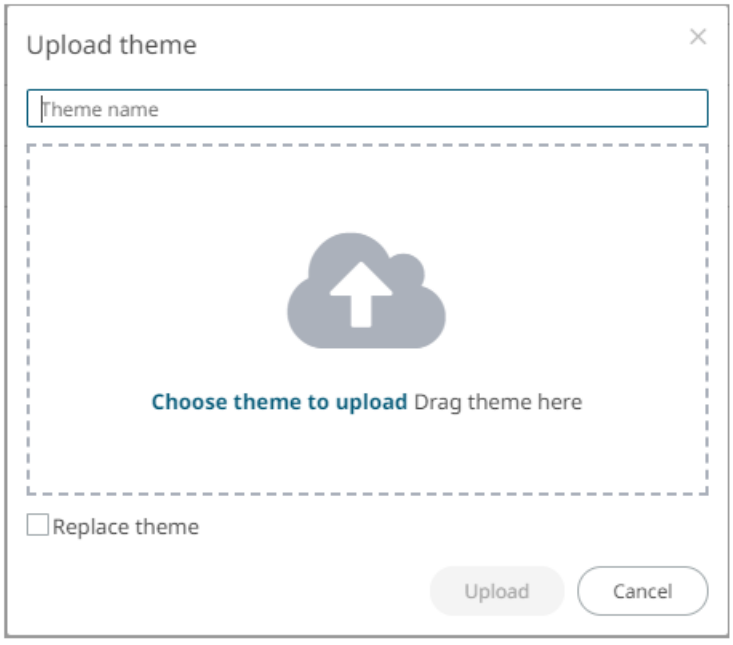
Users can upload their own workbook themes and also replace existing ones.

Steps:

1. Click on a folder of subfolder where the user has permission to upload a theme then select **Upload Theme** on the context menu.

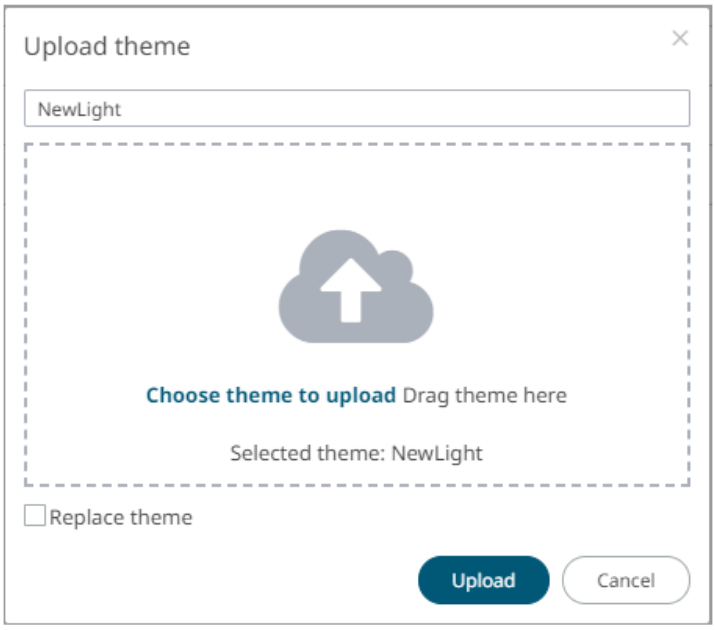


The *Upload Theme* dialog displays.



2. To upload a workbook theme, either:
   - drag the file from your desktop and drop on the dialog, or
   - click **Choose theme to upload** and then browse and select one on the *Open* dialog that displays

   The name of the workbook theme is displayed on the uploaded workbook palette area and in the *Name* box.

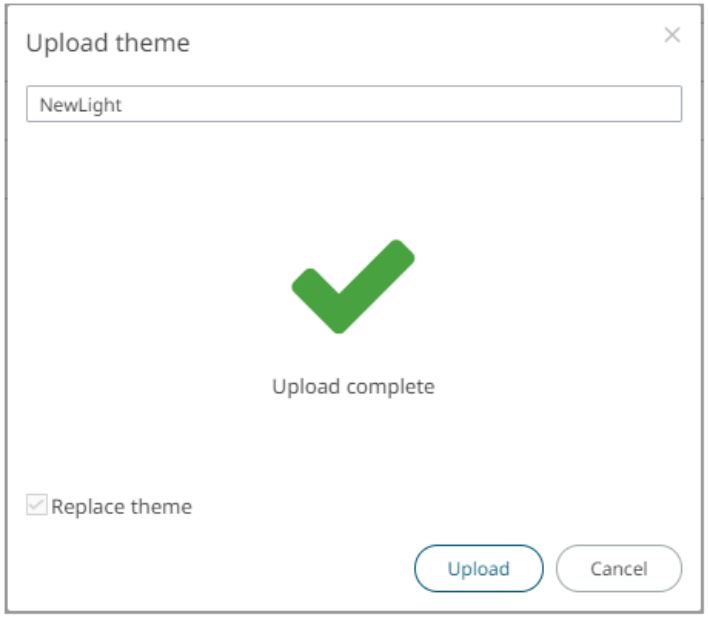


   You can opt to rename the uploaded workbook theme.

3. To replace the workbook theme, check the *Replace Theme* box.
4. Click 

.

   A notification displays once the file is uploaded.



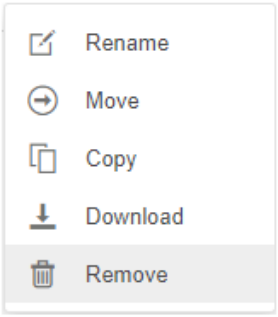
The uploaded theme is added in the *Theme* list.

Deleting Themes

The default themes (**Dark** and **Light**) cannot be removed.

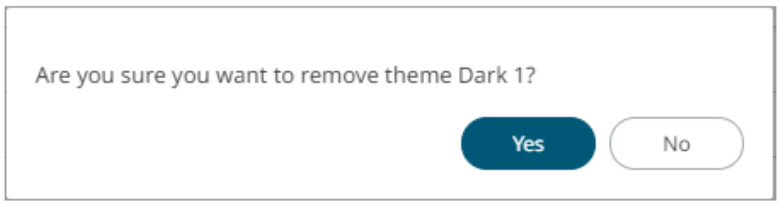
Steps:

1. Right-click on one or two themes then either:
   - select **Remove** on the context menu, or



   - click the **Remove** icon on the toolbar.

   A notification message displays.

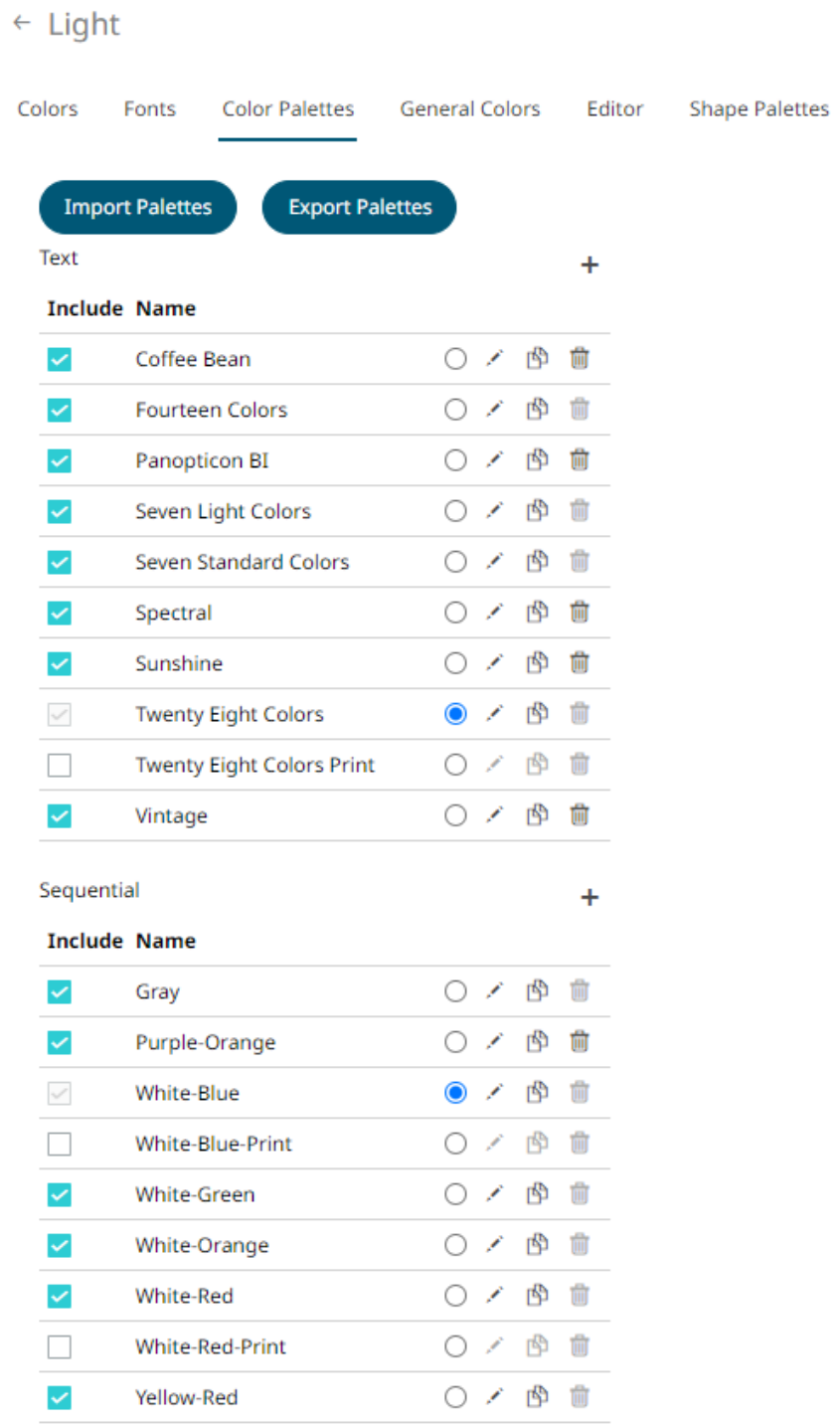


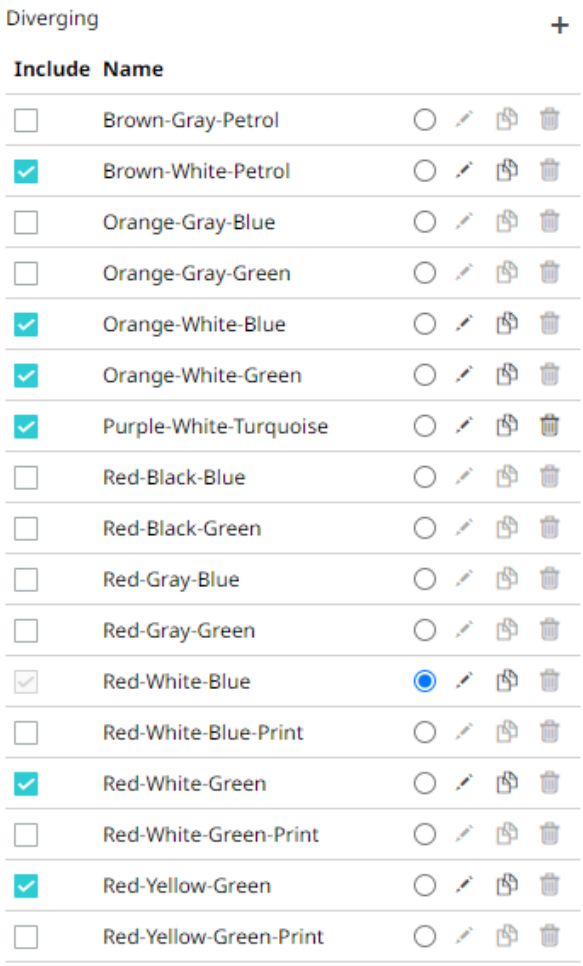
2. Click 

.

COLOR PALETTES

The text, sequential, and diverging color palettes that is used in text or numeric color variables in visualizations can be created, modified, duplicated, or deleted in the **Color Palettes** tab of a *Theme* page.





NOTE	Creating, modifying, duplicating, or deleting color palettes can also be done inside a workbook in the *Web Authoring*. However, these changes will only be associated with the inline theme of the workbook and will not be reflected in the Color Palettes tab of the *Themes* page in Panopticon Real Time.

Creating a New Text Color Palette

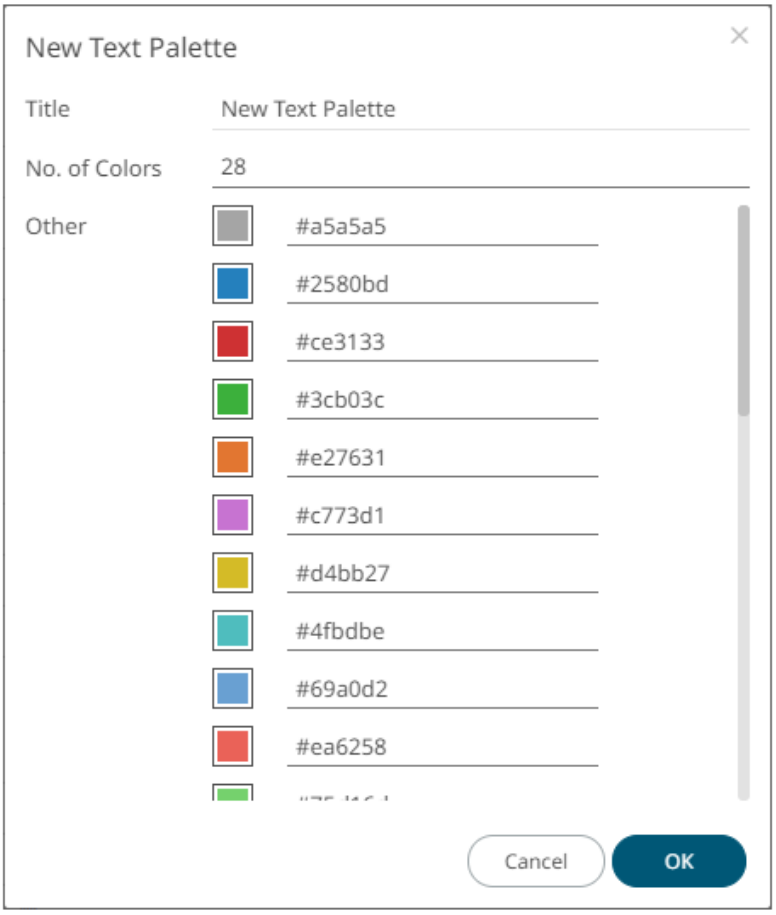
The configuration pane for the *Color* variable changes depending on the column data type.

In the Web Authoring, when a text column is added to the *Color* variable, the configuration pane displays the color associated with each categorical item, as specified with a default color palette (e.g., **Twenty Eight Colors**).

Steps:

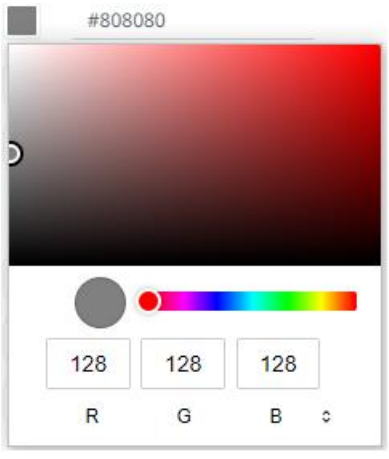
1. On the *Text* section, click the **New** + icon.

   The *Next Text Palette* dialog displays.



2. Enter the *Title* then click ✓ .
3. Select the *Number of Colors* in the drop-down list. Default is **28** colors.

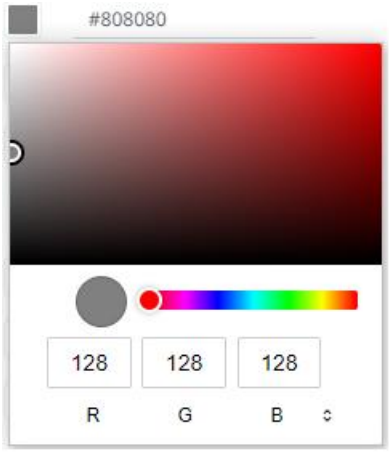
   The *Other* list is updated accordingly.
4. To set the colors:
   - click the corresponding *Color* box to display the *Color* dialog to:



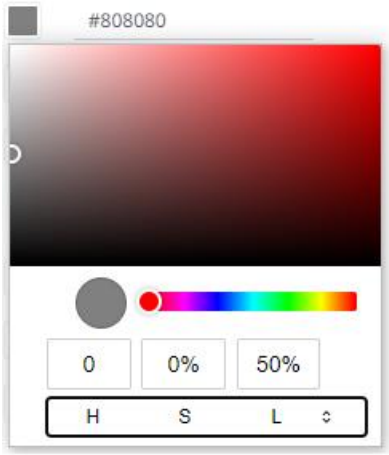
     - select the color, or

- click A to enter the values

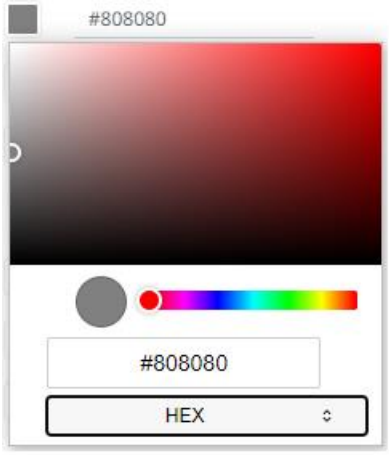
for RGB



for HSL



for the Hex color code

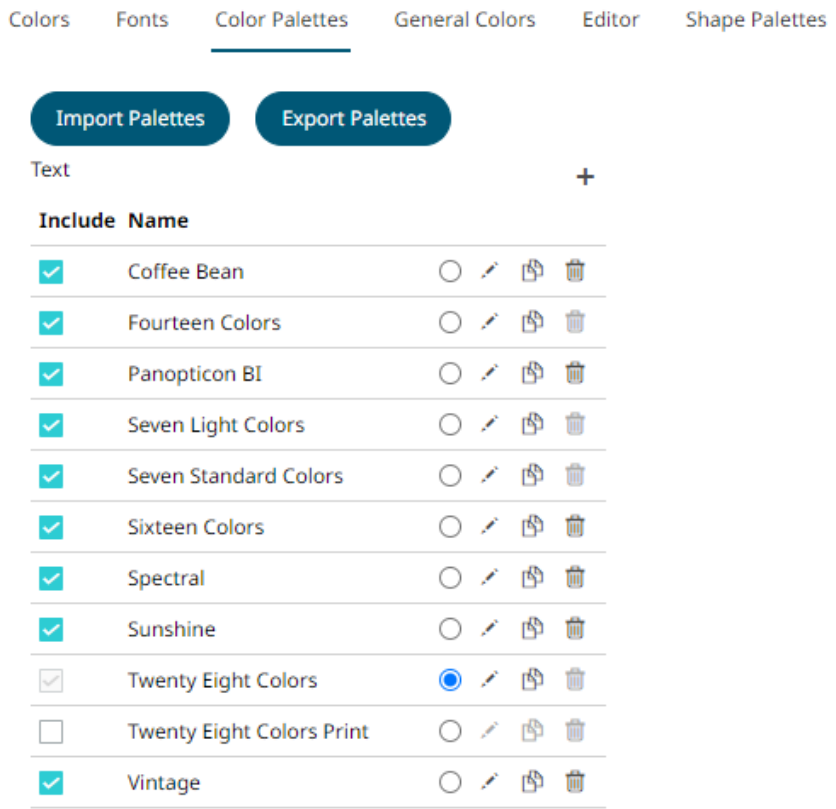


- or enter the *Hex* color code

5. Click 

.

The new text color palette is added in the list (e.g., **Sixteen Colors**). Note that it can be deleted.



Creating a Sequential or Diverging Numeric Color Palette

Panopticon visualizations support two types of Numeric Color Palettes: Sequential and Diverging.

- Sequential Color Palettes

  Sequential palettes use a two-color gradient between a minimum and a maximum value. Numeric column containing only positive values default to a Sequential Palette using the **White-Blue** color palette.

  In this case the range *Mid* point is disabled, and the *Min* and *Max* points are populated with defaults from the data set.

- Diverging Color Palettes

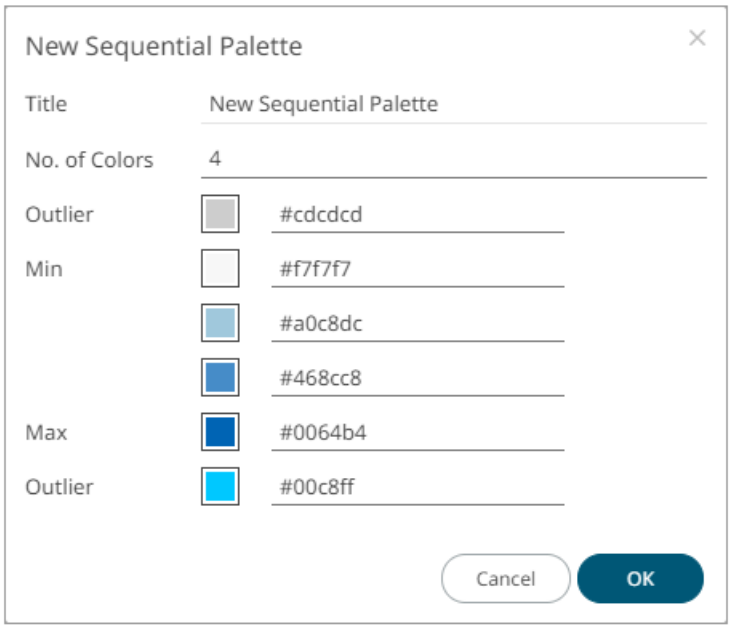
  Diverging Palettes use a three-color gradient between a minimum, middle and a maximum value. Numeric columns containing both positive and negative values default to the Diverging Palette with the **Red White Blue** color palette selected.

  Diverging Palettes use the **Range Midpoint**. The *Min*, *Mid* and *Max* points are populated with defaults from the data set.

To create a new sequential numeric color palette:

1. On the *Sequential* section, click the **New** + icon.

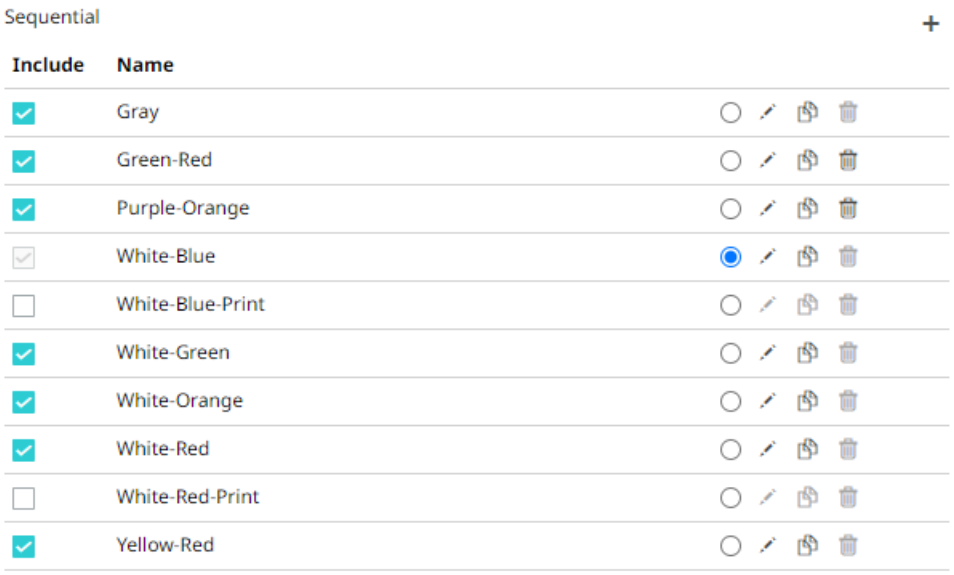
   The *New Sequential Palette* dialog displays.



2. Enter the *Title* and click ✓ .
3. Select the *Number of Colors* in the drop-down list. Default is **4** colors.

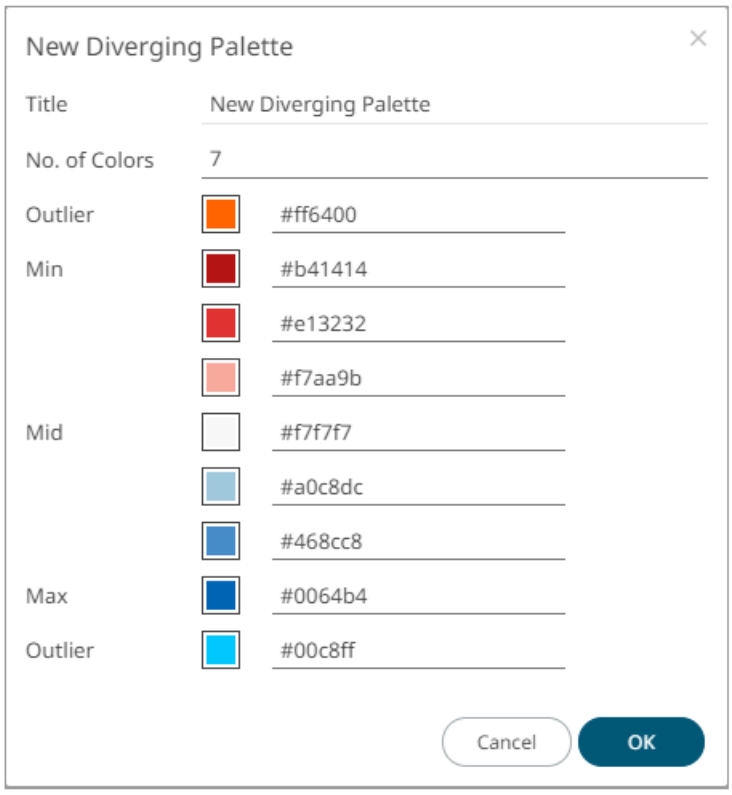
   The number of colors from *Min* to *Max* is updated accordingly.
4. Set the *Outliers*, *Min*, and *Max* colors. Refer to step 4 of Creating a New Text Color Palette for more information.
5. Click Ok .

   The new sequential numeric color palette is added in the list and can be deleted (e.g., **Green-Red**).



To create a new diverging numeric color palette:

1. On the *Diverging* section, click the **New** icon.
   The *New Diverging Palette* dialog displays.

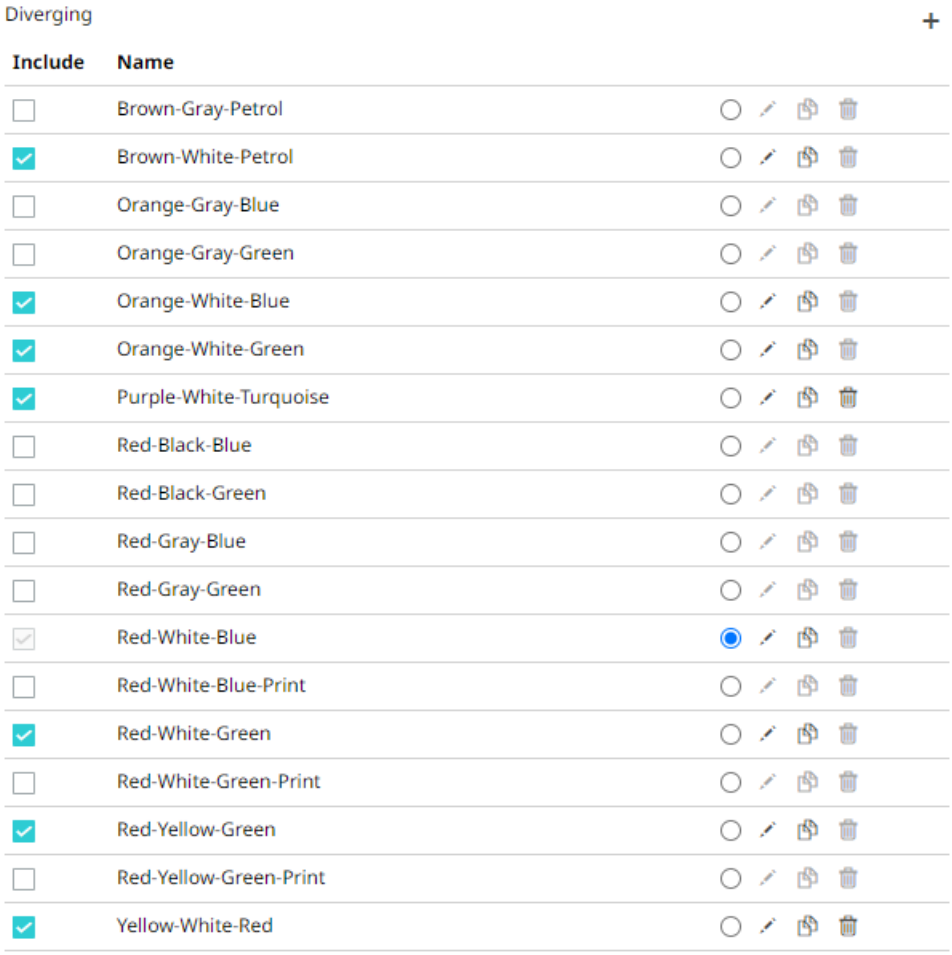


2. Enter the *Title* and click .
3. Select the *Number of Colors* in the drop-down list. Default is **7** colors.

The number of colors from *Min, Mid,* to *Max* is updated accordingly.

4. Set the *Outliers*, *Min*, *Mid*, and *Max* colors. Refer to step 4 of Creating a New Text Color Palette for more information.
5. Click Ok.

The new diverging numeric color palette is added in the list and can be deleted (e.g., **Yellow-White-Red**).



Modifying Color Palettes

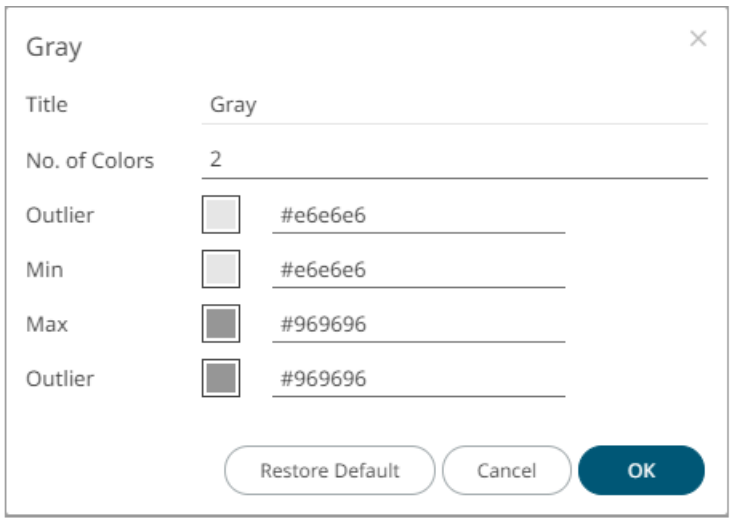
Any of the included or checked color palettes can be modified.

NOTE	• For the selected default color palette, only the *Number of Colors* and assigned colors can be modified. • Color palettes that are not selected cannot be modified.

Steps:

1. Click the **Edit** icon of an included or checked color palette.

   The corresponding dialog box displays.



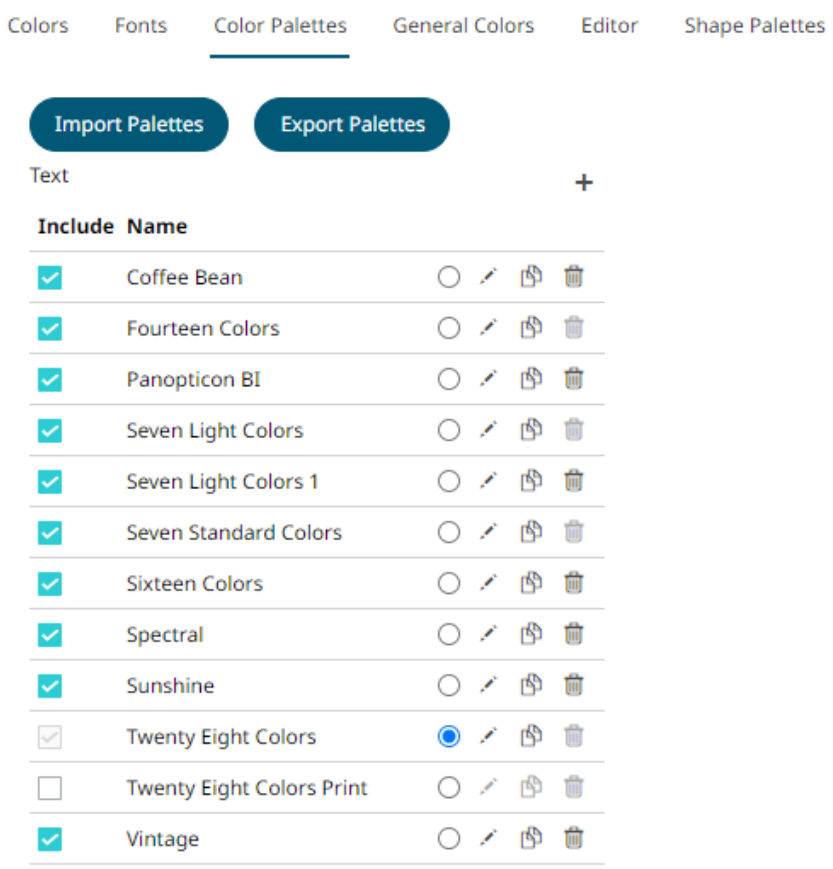
2. Modify the *Title*, *Number of Colors*, and assigned colors.
3. Click 

to commit the changes or 

to revert to the original settings.

Creating a Duplicate of a Color Palette

Click the **Duplicate** icon of a color palette. A copy of the color palette is added in the list (e.g., **Seven Light Colors 1**).



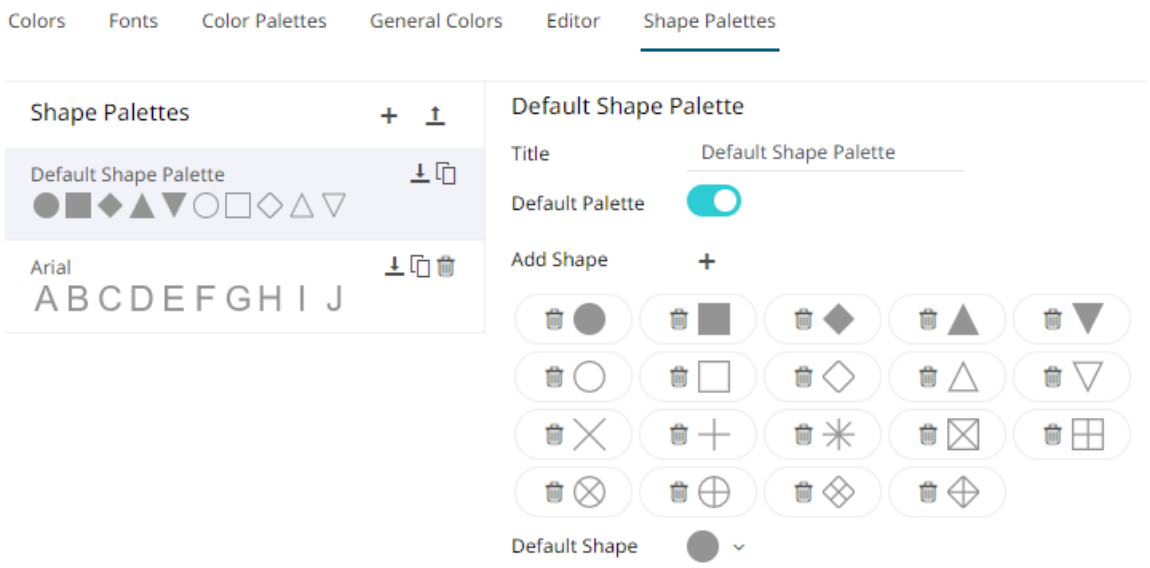
You can opt to modify the settings.

Deleting Color Palettes

New or duplicate color palettes can be deleted. Click the **Delete** icon to remove the color palette in the list.

SHAPE PALETTES

Shape palettes that can be used with the workbook theme can be created, uploaded, downloaded, modified, duplicated, rearranged, or deleted on the *Shape Palettes* page.



NOTE	Panopticon is shipped with two shape palettes (Default Shape Palette and Arial) for the Dark and Light themes.

Creating a New Shape Palette

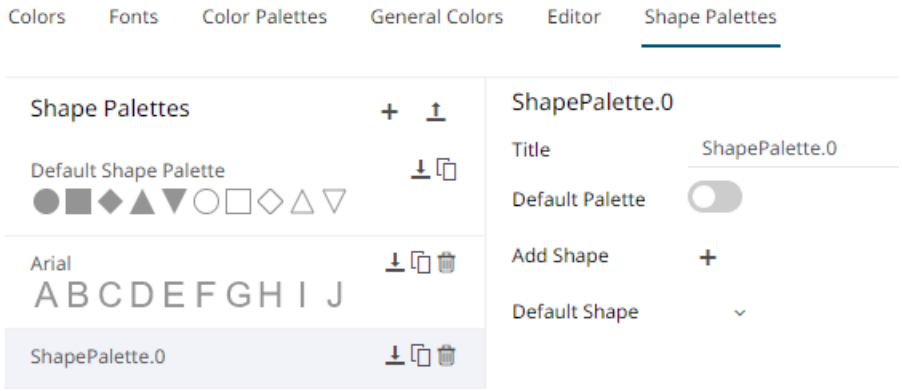
Steps:

1. Click **Add Palette** + .

   A new shape palette displays (i.e., **ShapePalette.0**).

2. Click *ShapePalette.<Number>*.

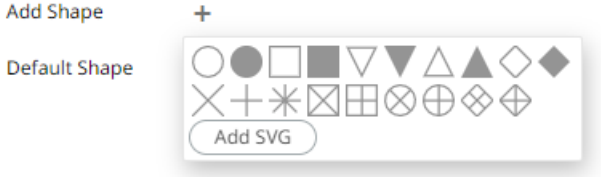
   The page changes to allow the definition of the new shape palette.



3. Enter the shape palette *Title* and click ✓.
4. To make this shape palette the default for the workbook theme, tap the **Default Palette** slider to turn it on.

NOTE	The default shape palette can not be deleted.

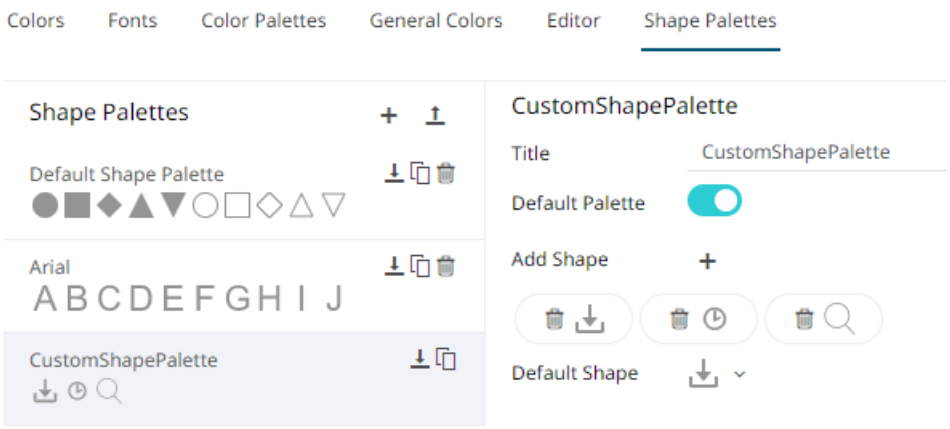
5. To add the shapes, click +.



You can either:

- click on a shape.
- click Add SVG . Select one or more SVG files in the *Open* dialog box that displays.

The added shapes are displayed.



To delete a shape, click it's corresponding **Delete** icon.

6. Select the *Defauilt Shape* in the drop-down list.
7. Click the **Save** Save .
8. When saved, the Saved! notification is displayed.

   The new shape palette is available in the *Shape Palette* drop-down list in the *Shape* variable when the workbook theme, where it is added, is used (i.e., **Light**).

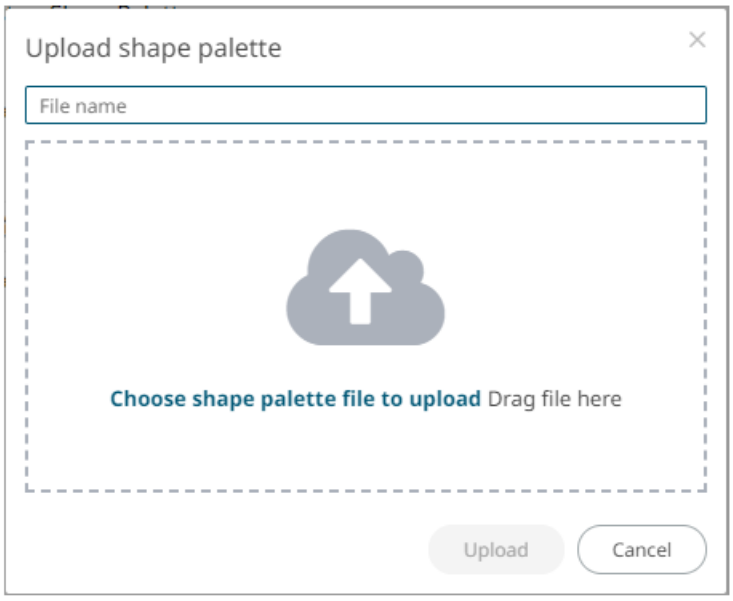
Uploading a Shape Palette

Users can upload their own shape palettes.

Steps:

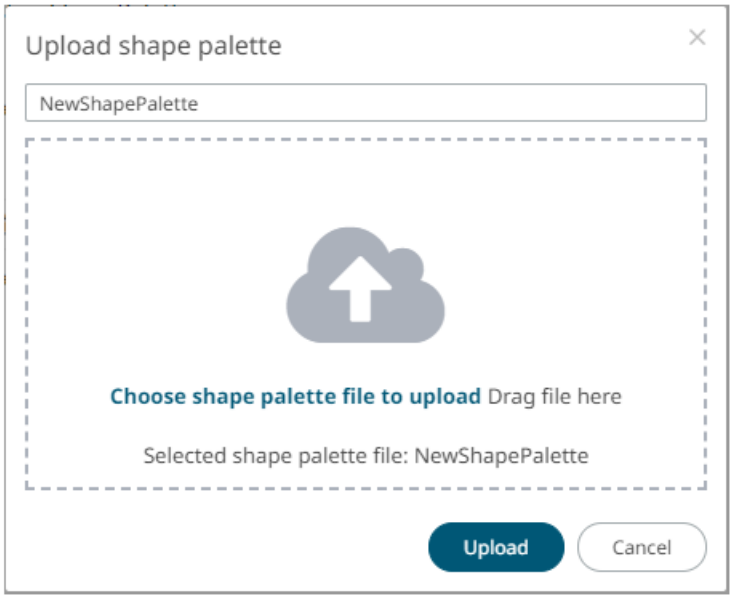
1. On the *Shape Palettes* pane, click .

   The *Upload Shape Palette* dialog displays.



2. To upload a shape palette, either:
   - drag the file from your desktop and drop on the dialog, or
   - click **Choose shape palette file to upload** and then browse and select one on the *Open* dialog that displays.

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

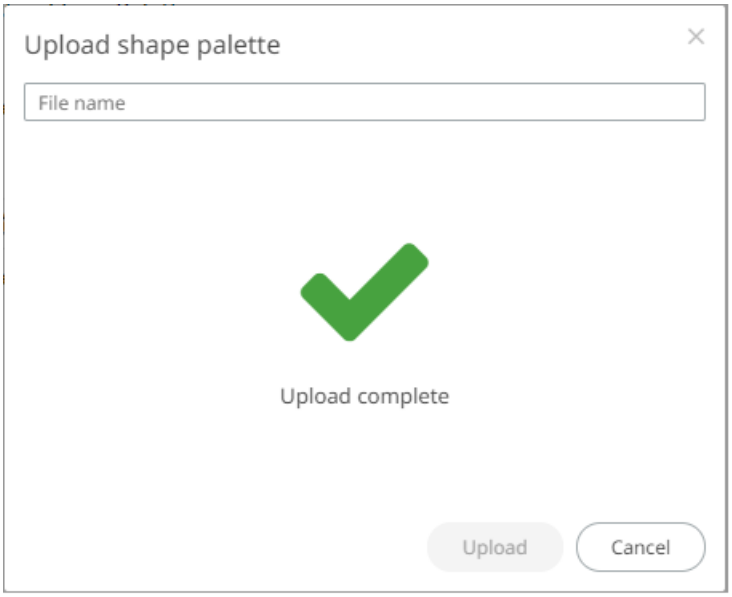


   You can opt to rename the uploaded shape palette.

3. Click 

.

   A notification displays once the file is uploaded.



The uploaded shape palette is added in the list.

Downloading a Shape Palette

You can download a copy of any of the shape palettes.

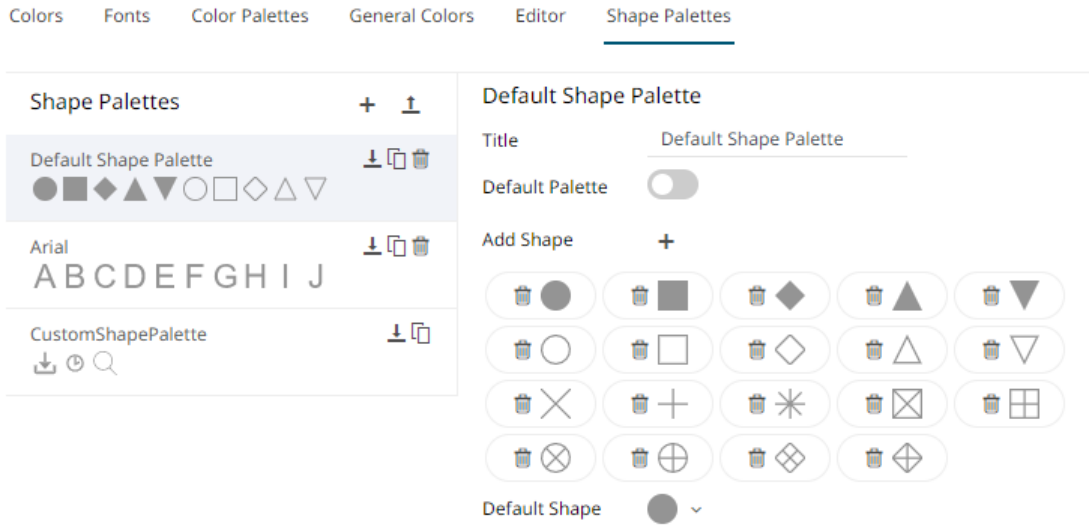
Click the **Download** icon of a shape palette.

Modifying Shape Palettes

Any of the shape palettes can be modified.

Steps:

1. Click on a shape palette to display its settings.

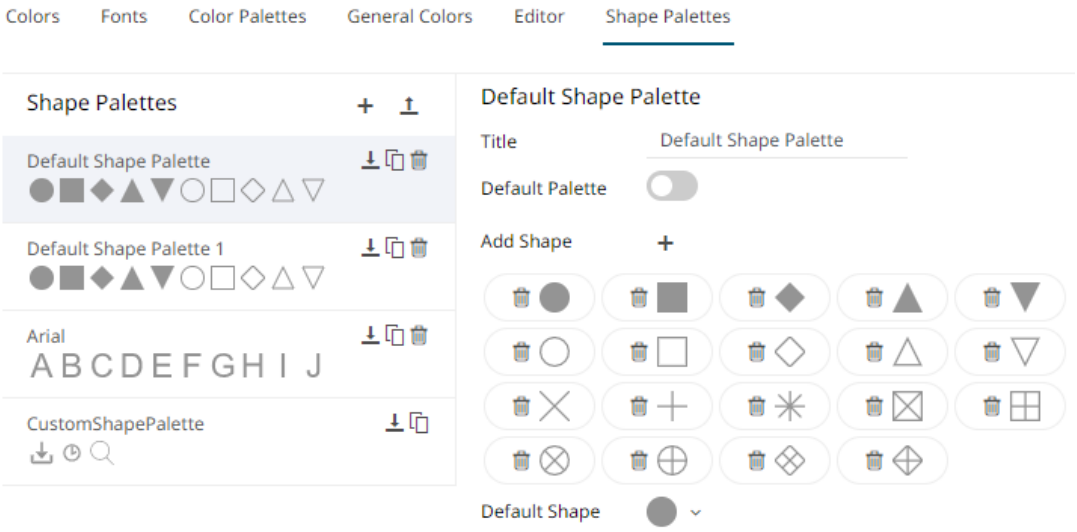


2. You can modify the following properties:
    - Title
    - Default Palette. Tap to enable or disable.
    - Add or delere shapes
    - Default Shape
3. Click the **Save** 

icon to save the changes.

Creating a Duplicate of a Shape Palette

Click the **Duplicate** icon of a shape palette. A copy of the shape palette is added in the list (e.g., **Default Shape Palette 1**).



You can opt to modify the settings.

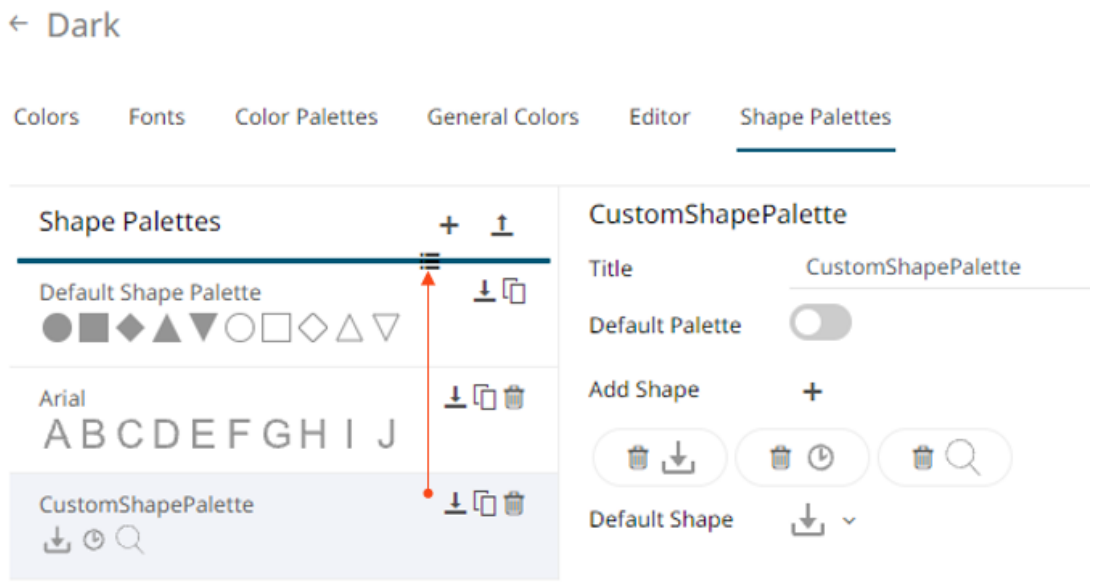
Rearranging Shape Palettes

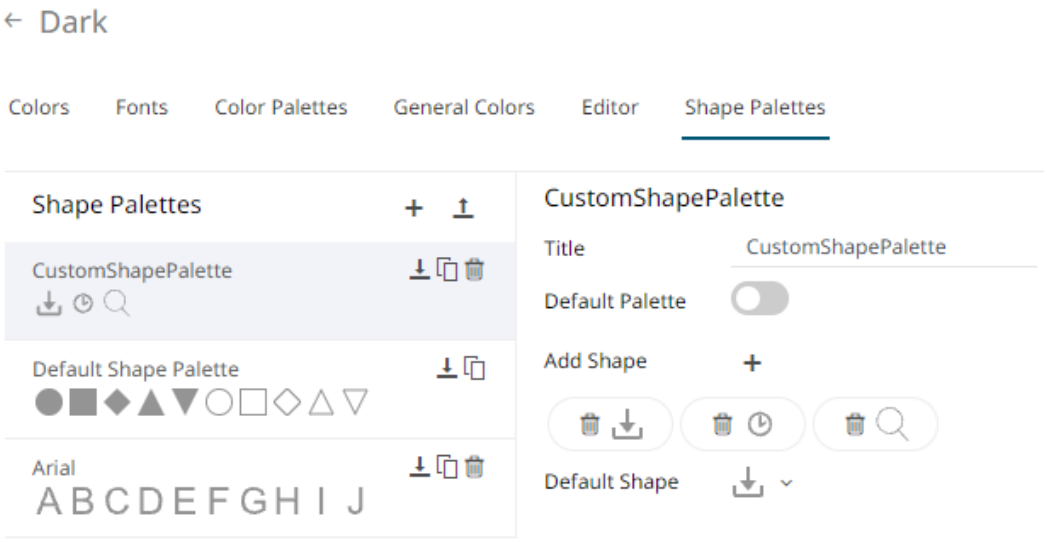
The order of the shape palettes can be rearranged.

Steps:

1. Click on a shape palette you want to move.

    The **Hand Hover** icon displays along with the blue marker before or after a shape palette where you can drop the item.
2. Drag and drop the shape palette to the desired position.





3. Click the **Save**  icon to save the changes.

Deleting Shape Palettes

Any shape palette can be deleted except the default. Click the **Delete**  icon to remove the shape palette in the list.

[16] PCLI: COMMAND UTILITIES FOR PANOPTICON REAL TIME

Panopticon Real Time is supplied with a command line utility `PCLI.jar`.

After extracting all of the contents of the pcli archive (pcli-java.zip), it is necessary to copy all of the JAR files from `…\apache-tomcat\webapps\panopticon\WEB-INF\lib\` to `…\pcli-java\lib\.`

NOTE	Ensure to overwrite any existing files when copying all of the JAR files from `…\apache-tomcat\webapps\panopticon\WEB-INF\lib\` to `…\pcli-java\lib\.`

This supports the following:

clearcache	Clears the cache on a Panopticon Real Time.
plugins	Troubleshoot the plugins that this program utilizes.
publish	Publishes a workbook to a server or folder.
version	Prints program (and optionally server) version and exits.
help	Use 'help <command>' to get help on a specific command.
upgrade	Upgrades specified workbook to the newest version.
schemify	Updates workbook data tables with missing schema information.
exportdatasource	Export workbook data sources.
convertpermissions	Converts old permission files to the new format.
migratedatabasetojdbc	Migrates all usages of the Database connector to JDBC connector in the given workbook/directory of workbooks.
mockdata	Updates workbook data tables with mock data based on the stored schema.

Summary help is displayed through: `[pcli-java folder]>java -jar pcli.jar help`

More detailed help is displayed through: `[pcli-java folder]>java -jar pcli.jar help [command]`

Clearcache

Clears the cache in Panopticon Real Time.

Option	Description
-w, --workbook	Workbook name. Syntax example: java -jar pcli.jar clearcache -w "workbook" -u "http://username:password@host:port/app_name/"
-d, --datatable	Datatable name. Syntax example: java -jar pcli.jar clearcache -w "workbook" -d "datatable" -u "http://username:password@host:port /app_name/"
-u, --url	URL to Panopticon Real Time, syntax: java -jar pcli.jar clearcache -u http://username:password@host:port/app_name/.

Command example: `java -jar pcli.jar clearcache -w "How To Actions" -d "StocksTimeSeriesFilteredTimeParameters" -u "http://username:password123@localhost:8080/panopticon"`

Plugins

Troubleshoot the plugins that this program utilizes.

Option	Description
-v, --verbose	Print all information normally traced by the plugin manager.

Publish

You can either publish a workbook to a Panopticon Real Time or to a specific folder.

Publishing a Workbook to Panopticon Real Time

Publishes a workbook to Panopticon Real Time.

Option	Description
-f, --force	Overwrite existing workbook on server. Syntax example: java -jar pcli.jar publish -w "workbook" -d -u "http://username:password@host:port/app_name/" -n "name" -f
-w, --workbook	The workbook file to publish.

	Syntax example: java -jar pcli.jar publish -w "workbook" -u "http://username:password@host:port/app_name/"
-d, --dataFiles	Find and upload data files used by workbook.
-u, --url	URL to Panopticon Real Time, syntax: java -jar pcli.jar publish -w "workbook" -d -u "http://username:password@host:port/app_name/"
-n, --name	Publish workbook with a different name. Syntax example: java -jar pcli.jar publish -w "workbook" -d -u "http://username:password@host:port/app_name/" -n "name"
--local	Publishes workbook by file copy, instead of HTTP, for use when server exists on the local system. Specifies target file location path including file name. If the server is running, the application pool must be recycled after publication.
-cp	Java classpath. Syntax example: 'java -cp pcli.jar; [plugin dir]/* com.panopticon.dashboards.pcli.Pcli publish -w "workbook" -u "http://username:password@host:port/app_name/"'

Command example: `java -cp pcli.jar;lib/* com.panopticon.dashboards.pcli.Pcli publish -w "How To Actions.exw" -d -u "http://username:password123@localhost:8080/panopticon" -n "Published by pcli" -f`

Publishing a Workbook to a Folder

Publishes a workbook to a specific folder.

Option	Description
-w, --workbook	The workbook file to publish. Syntax example: java -jar pcli.jar publish -w "workbook" -u "http://username:password@host:port/app_name/"
-u, --url	URL to Panopticon Real Time, syntax: java -jar pcli.jar publish -w "workbook" -d -u "http://**username**:password@host:port/app_name/" **NOTE:** The username in the -u command must have permission to the folder. Just being in the list of Administrators is not enough.
-n, --name	Publish workbook to a folder on the server with a different name. Syntax example: java -jar pcli.jar publish -w "workbook" -d -u "http://username:password@host:port/app_name/" -n "folder\name"
-d, --dataFiles	Find and upload data files used by workbook.

Command example: `java -jar pcli.jar publish -w "E:\Temp\How to Actons.exw" -u "http://username:password123@localhost:8080/panopticon" -n "test\How to Actions.exw"`

Publishing a Workbook Folder to Panopticon Real Time

Publishes a workbook folder to Panopticon Real Time.

Option	Description
-tf, --targetFolder	The target folder to which workbooks will be published. Use -r to publish all workbooks to the ROOT folder. This is only applicable with -wf Syntax example: java -jar pcli.jar publish -u "http://username:password@host:port/app_name/" -wf "folder containing workbooks"-tf "server folder name" -r
-r, --root	Publish workbooks to the ROOT folder. This is only applicable with -wf Syntax example: java -jar pcli.jar publish -u "http://username:password@host:port/app_name/" -wf "folder containing workbooks"-tf "server folder name" -r "default or root folder"
-u, --url	URL to Panopticon Real Time, syntax: java -jar pcli.jar publish -w "workbook" -u "http://username:password@host:port/app_name/"
-wf, --workbookFolder	The workbook folder from which workbooks will be picked to publish. Use -w to publish single workbook. Syntax example: java -jar pcli.jar publish -u "http://username:password@host:port/app_name/" -wf "folder containing workbooks"-tf "server folder name" -r

Command example: `java -cp pcli.jar publish "http://username:password123@localhost:8080/panopticon" -wf "C:\Serverdata\Data" -tf "c:\Streamsdata\Data" -r`

Version

Prints program (and optionally server) version and exits.

Option	Description
-u, --url	URL to Panopticon Real Time, syntax: java -jar pcli.jar version -u "http://username:password@host:port/app_name/"

Command example: `java -jar pcli.jar version -u "http://username:password123@localhost:8080/panopticon"`

Help

Lists all commands or options for a single command.

Command example: `java -jar pcli.jar help access`

Upgrade

Upgrades specified workbook to the newest version.

Option	Description
-w, --workbook	Workbook path to upgrade. Syntax example: java -jar pcli.jar upgrade -w workbook.exw
-o, --output	Output workbook path. Syntax example: java -jar pcli.jar upgrade -w workbook.exw -o workbook1.exw

Schemify

Updates workbook data tables with missing schema information.

Option	Description
--dd, --data-directory	Data directory path. Syntax example: java -jar pcli.jar schemify -wd "workbook directory" -od "output directory" –dd "C:\Users\Public\Documents\Datawatch Desktop\Data"
-D	Default parameter. This can be supplied either by using: -dp command to pass the path to Parameters.json which is the default parameter file Syntax example: java -jar pcli.jar schemify -w "workbook path" -o "output path" -l "license file path" -dp "default parameters file" -D switch to specify parameters Syntax example: java -jar pcli.jar schemify -w "workbook path" -o "output path" -l "license file path" -D "parameter=value"
-od, --output-directory	Output directory path. Syntax example: java -jar pcli.jar schemify -wd "workbook directory" -od "C:\Users\Public\Documents\Datawatch Desktop\NewWorkbooks" –dd "data directory"
-w, --workbook	Workbook to schemify.
-l, --license-file	License file path. Syntax example: java -jar pcli.jar schemify -wd "workbook path" -o "output path" -l "C:\vizserverdata\PanopticonLicense.xml"
-wd, --workbook-directory	Directory of the workbooks to schemify.
-o, --output	Output path.

Exportdatasource

Export workbook data source.

Option	Description
-dd, --data-directory	Data directory path.
-od, --output-directory	Output directory path.
-w, --workbook	Export data sources of workbook.
-l, --license-file	License file path.
-wd, --workbook-directory	Directory of workbooks.

Command example: `java -jar pcli.jar exportdatasource -l "E:\projects\Dashboards.NET\PanopticonLicense.xml" -w "E:\workbooks\exportdb.exw" -dd "E:\Serverdata\export" -od "E:\Streamsdata\export" -wd "E:\workbooks"`

Convertpermissions

Takes an old `Workbooks` folder and scans it for `GroupAccessPermissions.xml` files, collects them, and outputs a single file that can then be consumed by the server.

Option	Description
-wf, - workbookFolder	Path to old `Workbooks` folder, defaults to the current folder.
-o, - outputFile	Path to file where the result will be output. Default is **stdout**.
-wa, - writersAdmin	If users that had write permission on the old server should additionally get admin permission on the new server, defaults to not. The old server only had **read** and **write**, the new one has **read**, **write**, and **admin**.
-tf, - targetFolder	Path to subfolder on target server where you intend to import the workbooks, if not the root folder.

NOTE

- Special treatment of empty input folders:

  If a workbook folder did not have a `GroupAccessPermissions.xml` file, the old server would treat it as if the “Everyone” group had both read and write access to it. This is made explicit in the PCLI verb, which adds this permission to the output.

  For example:

  ```
  pcli convertpermissions -wf /appdata17/Workbooks/ -o perms.json -wa -tf /migrated/
  ```

  This creates `perms.json` which can then be used to restore the permissions from the old server on the new server if you import the old workbooks into the “migrated” workbook folder.

- See also the `Panopticon.properties` parameter `repository.startup.apply.permissions.path`.

MigrateDatabaseToJDBC

Migrates all usages of the Database connector to JDBC connector to enable editing in the Web Client.

Option	Description
-w, - workbook	Full path of workbook to migrate.
-o, --output	Output path. Can be used together with the '-w' option, when a new name to migrated workbook is needed. Output directory should exist.
-od, --output-directory	Output directory path. Output directory should exist.
-wd, --workbook-directory	Directory of workbooks to upgrade.

Command example: `java -jar pcli.jar migratedatabasetojdbc -w "E:\ \Workbooks\Database.exw" -o "E:\MigratedWorkbooks\JDBC.exw"`

Mockdata

Updates workbook data tables with mock data based on the stored schema.

Option	Description
-w, - workbook	Workbook to mock the data table data.
-o, --output	Output path.

Command example: `java -jar pcli.jar mockdata -w "z Custom Index - v4 (2).exw" -o MockData.exw Workbooks\z Custom Index - v4: saved updated workbook to MockData.exw`

[17] REST INTERFACE

DISCLAIMER

As part of the deprecation of Desktop Designer and related legacy visualization- and data pipelines, we have unfortunately had to retire a set of previously documented REST service endpoints. The endpoints below will no longer be available in the product:

- `GET media/image/dashboard`
- `GET media/image/dashboard/part`

All Panopticon APIs should be considered proprietary, internal and subject to change. Going forward, all REST endpoints will be classified into private and publicly supported APIs. Please let us know if your implementation relies on REST API, to ensure that the functionality is made available in future public API.

API

Panopticon Real Time provides a REST API that allows you to interact with the server through requests that can be scripted and automated. For example, the REST API supports downloading data, uploading workbooks, getting server usage statistics and much more. The API follows the OpenAPI Specification from Swagger.io, an API description format. Thereby, a web interface for the REST API including documentation is available, to facilitate exploration and testing of the REST API. To enable the web interface for testing, and the documentation in it, you first have to set `documentation.enabled=true` in `Panopticon.properties`.

NOTE

If you have not set `documentation.enabled=true`, there is no web interface for testing available, but the REST API is responding in any case.

Updating the property file requires Panopticon Real Time application to be restarted, which is either just the application or the entire Tomcat server. Once Panopticon Real Time has been restarted, the REST API can be accessed from the browser. Use the following address to view and interact with the documentation:
`http://[host]:[port]/[serverappname]/server/swagger-ui.html`

The OpenAPI specification can be accessed on the following address:
`http://[host]:[port]/[serverappname]/server/swagger-ui.html`

Example: `http://localhost:8080/panopticon/server/swagger-ui.html`

EXPORT DATA

CSV

Panopticon Real Time provides the functionality to export data from a visualization to a CSV file.

Use the following URL to download the CSV file from the Server:

- URL: http://[server]/[path]/server/rest/media/data/dashboard/part

Each URL has the following properties:

- Mandatory arguments
  - **Workbook** – Workbook name without an extension.
  - **Dashboard** – Dashboard name in the workbook.
  - **Part** – The visualization part ID

The following examples show how to export the data of a visualization from a local server. For these examples, we have used the example workbook **How To Actions**.

- Export data as a CSV file
  - **Syntax**: http://[server]/[path]/server/rest/media/data/dashboard/part?workbook={Workbook name}&dashboard={Dashboard name}&part={Visualization part id}
  - **Example**: http://localhost:8080/panopticon/server/rest/media/data/dashboard/part?workbook=How+To+Actions&dashboard=Data+Entry&part=visualization.Treemap1

Dashboard Parameters

The CSV file can be generated based on the workbook data table parameters. The parameter and its values can be specified to determine the context of the exported data.

Syntax:

```
http://[server]/[path]/server/rest/media/data/dashboard/part?workbook={Workbook name}&dashboard={Dashboard name}&part={Visualization part id}&{dashboardParameterName1=value1}&{dashboardParameterName2=value2}
```

Adding `Region=Europe` and `Industry=Financials` parameters

Example:

```
http://localhost:8080/panopticon/server/rest/media/data/dashboard/part?workbook=How+To+Actions&dashboard=Scatter+of+Filtered+Universe&part=visualization.ScatterPlot1&Region=Europe&Industry=Financials
```

Adding `Region=Asia Pacific`, or `Region=Europe` and `Industry= Financials` parameters produces a CSV file that is focused on Asia Pacific & European Financials. In this case the Region parameter is repeated for each of the supplied regions.

Example:

```
http://localhost:8080/panopticon/server/rest/media/data/dashboard/part?workbook=How+To+Actions&dashboard=Scatter+of+Filtered+Universe&part=visualization.ScatterPlot1&Region=Asia+Pacific&Region=Europe&Industry=Financials
```

PDF

Panopticon Real Time provides the functionality to generate and download PDFs. Use the following URL to download PDFs from the server:

- URL: http://[server]/panopticon/server/rest/media/pdf

The URL can be accessed through scheduled batch tasks to retrieve and process generated PDFs. (e.g., email to predefined mailing list).

Each URL has the following properties:

- Mandatory arguments
  - **Workbook** – Workbook name without an extension.
- Optional arguments
  - **Dashboard** – Dashboard name in the workbook.
  - **HideScrollbars** – Show/Hide the visualization scrollbar in the PDF. Possible values are true/false. The default value is true.
  - **EnablePagination** – Enable pagination in the PDF. Possible values are true/false. The default value is **true**.

The following examples show how to export a PDF from a local server. For these examples, we have used the example workbook **How To Actions**.

- Generate PDF report of the entire workbook
  - **Syntax**: http://[server]/[path]/server/rest/media/pdf?workbook={Workbook name}
  - **Example**: http://localhost:8080/panopticon/server/rest/media/pdf?workbook=How+To+Actions
- Generate PDF report of the entire workbook in a folder
  - **Syntax**: http://[server]/[path]/server/rest/media/pdf?workbook={Folder name%5CWorkbook name}
  - **Example**: `http://localhost:8080/panopticon/server/rest/media/pdf?workbook=my+folder%5CHow+To+Actions`

NOTE	When the workbook name specifies any folder or subfolders, the path delimiter must be backslash (URL-encoded as %5C) and not forward slash (URL-encoded as %2F).

- Generate PDF report of a single dashboard in the workbook
  - **Syntax**: http://[server]/[path]/server/rest/media/pdf?workbook={Workbook name}&dashboard={Dashboard name}
  - **Example**: http://localhost:8080/panopticon/server/rest/media/pdf?workbook=How+To+Actions&dashboard=How+To+Actions
  - **Example** (Multiple dashboards): http://localhost:8080/panopticon/server/rest/media/pdf?workbook=How+To+Actions&dashboard=How+To+Actions&dashboard=Data+Entry
- Hide scrollbars from visualizations in the PDF
  - **Syntax**: http://[server]/[path]/server/rest/media/pdf?workbook={Workbook name}&hideScrollbars={true/false}

- **Example**: http://localhost:8080/panopticon/server/rest/media/pdf?workbook=How+To+Actions&hideScrollbars=true

- ❑ Enable or disable pagination of visualizations with vertical scrollbars in the PDF report
  - **Syntax**: http://[server]/[path]/server/rest/media/pdf?workbook={Workbook name}&enablePagination={true/false}
  - **Example**: http://localhost:8080/panopticon/server/rest/media/pdf?workbook=How+To+Actions&enablePagination=true

Dashboard Parameters

The PDF report can be generated based on the workbook data table parameters. The parameter and its values can be specified to determine the context of the generated PDF report.

Syntax: `http://[server]/[path]/server/rest/media/pdf?workbook={Workbook name}&{dashboardParameterName1=value1}&{dashboardParameterName2=value2}`

Adding `Region=Europe` and `Industry=Financials` parameters

Example:
`http://localhost:8080/panopticon/server/rest/media/pdf?workbook=How+To+Actions&dashboard=Scatter+of+Filtered+Universe&Region=Europe&Industry=Financials`

Adding `Region=Asia Pacific`, or `Region=Europe` and `Industry= Financials` parameters produces an output PDF that is focused on Asia Pacific & European Financials. In this case the Region parameter is repeated for each of the supplied regions.

Example:
`http://localhost:8080/panopticon/server/rest/media/pdf?workbook=How+To+Actions&dashboard=Scatter+of+Filtered+Universe&Region=Asia+Pacific&Region=Europe&Industry=Financials`

Authentication

In order to generate certain workbooks, the user might need to be authenticated. The user will be prompted with a login window if the user tries to export a PDF from a web browser. The user can also send the credentials via a header to be authenticated. This could be necessary if the user is using commands like `wget` to invoke the server to generate PDFs.

The credentials are sent as basic authorization. The user provides the credentials in the **Authorization** header. The value is formatted in the following way: `Basic username:password`. Please note that the username and password must be Base64 encoded. Example: `MyUsername:MyPassword = TXlVc2VybmFtZTpNeVBhc3N3b3Jk`

Wget example: `wget -O "Output.pdf" --header="Authorization: Basic TXlVc2VybmFtZTpNeVBhc3N3b3Jk" "http://localhost:8080/panopticon/server/rest/media/pdf?workbook=How+To+Actions"`

The PDF generator supports the following authentication mechanisms:

- ❑ BASIC
- ❑ LDAP
- ❑ Filter authentication
- ❑ Header authentication

- Windows authentication

Excel Workbook

Panopticon Real Time provides the functionality to export a Panopticon workbook as an Excel workbook. All of the dashboards in the Panopticon workbook will be inserted into their own corresponding Excel sheet. In addition, all of the visualizations in the dashboard will be exported as a PNG image and inserted into an Excel sheet.

The images will be laid out as visualizations on the dashboard. However, the table visualizations will not be exported as images. The visualization tables will instead be exported as Excel tables. The Excel table will always be laid out under all of the exported visualization images.

Please note that only one table will be exported for each dashboard.

Use the following URL to download the Excel workbook from Panopticon Real Time:

- URL: http://[server]/[path]/server/rest/media/excel

Each URL has the following properties:

- Mandatory arguments
  - **Workbook** – Workbook name without an extension.
- Optional arguments
  - **Dashboard** – Dashboard name(s) in the Panopticon workbook. All of the dashboards will be exported if no dashboard names are provided. The dashboard argument can be used multiple times depending on how many dashboards should be exported.
  - **Width** – The width of the exported dashboards. The default value is **1024px**.
  - **Height** – The height of the exported dashboards. The default value is **768px**.
  - **Style** – The Excel table style of an exported table. The default value is **TableStyleMedium7**.

The following examples show how to export an Excel workbook from a local server. For these examples, we have used the example workbook **How To Actions**.

- Generate and export Excel workbook
  - **Syntax**: `http://[server]/[path]/server/rest/media/excel?workbook={Workbook name}`
  - **Example**: `http://localhost:8080/panopticon/server/rest/media/excel?workbook=How+To+Actions`
- Set dashboards
  - **Syntax**: `http://[server]/[path]/server/rest/media/excel?workbook={Workbook name}&dashboard={Dashboard name1}&dashboard={Dashboard name2}`
  - **Example**: `http://localhost:8080/panopticon/server/rest/media/excel?workbook=How+To+Actions&dashboard=Data+Entry&Dashboard=Time+Parameters`
- Set height and width for Dashboard
  - **Syntax:** `http://[server]/[path]/server/rest/media/excel?workbook={Workbook name}&width={value}&height={value}`

- **Example**:
  `http://localhost:8080/panopticon/server/rest/media/excel?workbook=How+To+Actions&width=512&height=384`

- Set Excel table style
  - **Syntax**: `http://[server]/[path]/server/rest/media/excel?workbook={Workbook name}&style={Style}`
  - **Example**:
    `http://localhost:8080/panopticon/server/rest/media/excel?workbook=How+To+Actions&style=TableStyleMedium6`

Possible Excel Table Styles

- TableStyleLight1– TableStyleLight21
- TableStyleMedium1 – TableStyleMedium28
- TableStyleDark1 – TableStyleDark11

Dashboard Parameters

The Excel workbook can be generated based on the workbook data table parameters. The parameter and its values can be specified to determine the context of the generated Excel workbook.

Syntax: `http://[server]/[path]/server/rest/media/excel?workbook={Workbook name}&{dashboardParameterName1=value1}&{dashboardParameterName2=value2}`

Adding `Region=Europe` and `Industry=Financials` parameters

Example:
`http://localhost:8080/panopticon/server/rest/media/excel?workbook=How+To+Actions&Region=Europe&Industry=Financials`

Adding `Region=Asia Pacific`, or `Region=Europe` and `Industry= Financials` parameters produces an Excel workbook that is focused on Asia Pacific & European Financials. In this case the Region parameter is repeated for each of the supplied regions.

Example:
`http://localhost:8080/panopticon/server/rest/media/excel?workbook=How+To+Actions&Region=Asia+Pacific&Region=Europe&Industry=Financials`

EMAIL DATA

NOTE	To allow the triggering of the email send out via the REST API, Panopticon Real Time must be configured with valid email server information in the `Panopticon.properties` file located in the `AppData` folder (e.g., `c:\vizserverdata`). See Panopticon Real Time Configurations for Email Send Outs and Alerts for instructions.

PDF

Panopticon Real Time provides the functionality to generate and email PDFs.

This feature works exactly as the URL PDF generation and uses the same URL parameters. The main difference between the two features is that this feature sends the PDF in an email rather than downloading it as a file. Another difference is this feature requires a POST request to the following URL: `http://[server]/[path]/server/rest/media/pdf/email`.

Usage

The following properties can be configured:

- URL: http://[server]/[path]/server/rest/media/pdf/email
- Method: POST
- Content-Type: application/json
- Request body:
  - **bodyText** – The text will appear in the message body. The text can be formatted in HTML. Special characters, such as double quotation marks (“) should have a backslash preceding them in order for the Server to regard them as special characters.
  - **to** - One or more email recipients. Comma is used as a delimiter to separate the email recipients.
  - **cc** – One or more email recipients. Comma is used as a delimiter to separate the email recipients.
  - **bcc** – One or more email recipients. Comma is used as a delimiter to separate the email recipients.
  - **sender** – The sender’s email address. This value will also be used as a username.
  - **senderpassword** – The password to the sender’s email account.
  - **subject** – the subject of the email.

Example

For example, an On-Demand PDF will be emailed based on the following information:

Property	Description
Workbook	How to Actions
Dashboard Name	Scatter of Filtered Universe
Recipients (To)	to-mail1@mail.com to-mail2@mail.com
Sender	from-mail@mail.com
Password	password
Subject	Altair PDF Generator
Body Message	Hello. This is an auto-generated PDF.

As an example:

Panopticon configuration (`Panopticon.properties`):

```
email.host=smtp.server.com
email.port=587
email.security.mode=TLS
```

URL:

```
http://localhost:8080/panopticon/server/rest/media/pdf/email?workbook=How+To+Actions&dashboard=Scatter+of+Filtered+Universe
```

Body:

```
{
    "bodyText": "<h1>Hello.</h1><p>This is an auto-generated PDF.</p>",
    "to": "to-mail1@mail.com, to-mail2@mail.com",
    "sender": "from-mail@mail.com",
    "senderPassword": "password",
    "subject": "Altair PDF generator"
}
```

Image

Panopticon Real Time provides the functionality to generate and email dashboard images.

This feature is similar with Email PDF discussed above and uses the same URL parameters. However, this feature sends dashboard images as part of the email body and not as a PDF attachment. In addition, it does not support pagination.

In addition, hyperlinks can also be used in email dashboard images. Hyperlinks can redirect to a workbook and a dashboard in the server.

NOTE	In cases when you schedule the emailing of dashboard images or when you are behind a proxy or load balancer, it is recommended to specify the server address in the `Panopticon.properties` file. For example: `server.host=http://www.company.com/dashboards/`

The email contains the following *Body* components:

- Body message: The email starts with the provided body message in the request.
- Dashboard Title: The title displays before the dashboard image and uses a h2 heading tag.
- Dashboard image: The image (.png) of the dashboard.

Usage

- URL: `http://[server]/[path]/server/rest/media/image/dashboard/email`
- Method: POST
- Content-Type: application/json
- Request body:

- **bodyText** – The text will appear in the message body. The text can be formatted in HTML. Special characters, such as double quotation marks (“) should have a backslash preceding them in order for the Server to regard them as special characters.
- **to** - One or more email recipients. Comma is used as a delimiter to separate the email recipients.
- **cc** – One or more email recipients. Comma is used as a delimiter to separate the email recipients.
- **bcc** – One or more email recipients. Comma is used as a delimiter to separate the email recipients.
- **sender** – The sender’s email address. This value will also be used as a username.
- **senderpassword** – The password to the sender’s email account.
- **subject** – The subject of the email.
- **useHyperlink** – The property that determines whether the images should be hyperlinks. The hyperlink then opens the dashboard in the Thin Client. Hyperlinks will be used when set to true (default value). The images will be regular images and not a hyperlink when the property is set to **false**.

Example

Property	Value
Workbook	How to Actions
Dashboard Name	Scatter of Filtered Universe
Recipients (To)	to-mail1@mail.com to-mail2@mail.com
Sender	from-mail@mail.com
Password	password
Subject	Altair Image Generator
Body Message	Hello. This email contains dashboard images.
Use hyperlink	true

As an example:

Panopticon configuration (`panoption.properties`):

```
email.host=smtp.server.com
email.port=587
email.security.mode=TLS
```

URL:

```
http://localhost:8080/panopticon/server/rest/media/image/dashboard/email?workbook=How+To+Actions&dashboard=Scatter+of+Filtered+Universe
```

Body:

```
{
    "bodyText": "<h1>Hello.</h1><p>This email contains dashboard images.</p>",
    "to": "to-mail1@mail.com, to-mail2@mail.com",
    "sender": "from-mail@mail.com",
```

```
    "senderPassword": "password",
    "subject": "Altair Image generator",
    "useHyperlink": "true"
}
```

[18] LOGGING/MONITORING

SERVER LOGGING

Logging occurs:

- Within the platform
- In the underlying web / application server
- In the underlying OS

Panopticon Real Time logs are written to the `Tomcat` logs folder.

The logging level can be set from:

- Error – Only Errors are logged (the Default)
- Info – Operational logging is enabled including logging of data queries.
- Finest – All possible debugging logging is enabled.

Typically, when support issues are raised, the user is requested to change the logging level to **INFO**, which additionally records:

- Data Plugin (Visualization and Data Connector) Initialization
- Data Cache Initialization
- Data Subscriptions
- Data Queries including:
  - Database connection settings
  - Database SQL query
  - Number of rows & columns retrieved, and response time

Data query logging capabilities are specific to each data connector, with the most detailed logging available for the Database and kdb+ connectors.

Panopticon Real Time logging and auditing capabilities include Java JMX counters for usage and load monitoring, and additional logging around secured access to workbooks.

Configuring Server Logs

Panopticon Real Time is preconfigured with recommended logging settings for performance. All of the logging will be directed to a file prefixed by `panopticon` in the Tomcat `logs` folder. The Panopticon-specific logging configuration file is located inside the `.war` file at `WEB-INF/classes/logging.properties`. This configuration takes precedence over the general Tomcat logging configuration. If the logging is to be configured in Tomcat, the file `WEB-INF/classes/logging.properties` must be removed from the `.war` file.

Configuring Apache Tomcat Logs[1]

The internal logging for Apache Tomcat uses **JULI**, a packaged renamed fork of Apache Commons Logging that is hard-coded to use the `java.util.logging` framework. This ensures that Tomcat's internal logging and any web application logging will remain independent, even if a web application uses Apache Commons Logging.

To configure Tomcat to use an alternative logging framework for its internal logging, follow the instructions provided by the alternative logging framework for redirecting logging for applications that use `java.util.logging`. Keep in mind that the alternative logging framework will need to be capable of working in an environment where different loggers with the same name may exist in different class loaders.

A web application running on Apache Tomcat can:

- Use any logging framework of its choice
- Use system logging API, `java.util.logging`
- Use the logging API provided by the Java Servlets specification: `javax.servlet.ServletContext.log(...)`

The logging frameworks used by different web applications are independent. See class loading for more details. The exception to this rule is `java.util.logging`. If it is used directly or indirectly by your logging library, then the elements of it will be shared across web applications because it is loaded by the system class loader.

Java Logging API (java.util.logging)

Apache Tomcat has its own implementation of several key elements of `java.util.logging` API. This implementation is called **JULI**. The key component there is a custom LogManager implementation, that is aware of different web applications running on Tomcat (and their different class loaders). It supports private per-application logging configurations. It is also notified by Tomcat when a web application is unloaded from memory, so that the references to its classes can be cleared, preventing memory leaks.

This `java.util.logging` implementation is enabled by providing certain system properties when starting Java. The Apache Tomcat startup scripts do this for you, but if you are using different tools to run Tomcat (such as jsvc, or running Tomcat from within an IDE), you should take care of them by yourself.

Servlets Logging APICalls to `javax.servlet.ServletContext.log(...)` to write log messages are handled by internal Tomcat logging. Such messages are logged to the category named

`org.apache.catalina.core.ContainerBase.[${engine}].[${host}].[${context}]`

This logging is performed according to the Tomcat logging configuration. You cannot overwrite it in a web application.

The Servlets logging API predates the `java.util.logging` API that is now provided by Java. As such, it does not offer you much options. e.g., you cannot control the log levels. It can be noted, though, that in Apache Tomcat implementation the calls to `ServletContext.log(String)` or `GenericServlet.log(String)` are logged at the **INFO** level. The calls to `ServletContext.log(String, Throwable)` or `GenericServlet.log(String, Throwable)` are logged at the **SEVERE** level.

Console

When running Tomcat on unixes, the console output is usually redirected to the file named `catalina.out`. The name is configurable using an environment variable. Whatever is written to `System.err/out` will be caught into that file. That may include:

- Uncaught exceptions printed by java.lang.ThreadGroup.uncaughtException(..)

[1] http://tomcat.apache.org/tomcat-9.0-doc/logging.html

- Thread dumps, if you requested them via a system signal

When running as a service on Windows, the console output is also caught and redirected, but the file names are different.

The default logging configuration in Apache Tomcat writes the same messages to the console and to a log file. This is great when using Tomcat for development, but usually is not needed in production.

Old applications that still use `System.out` or `System.err` can be tricked by setting **swallowOutput** attribute on a Context. If the attribute is set to **true**, the calls to `System.out/err` during request processing will be intercepted, and their output will be fed to the logging subsystem using the `javax.servlet.ServletContext.log(...)` calls.

Note, that the **swallowOutput** feature is actually a trick, and it has its limitations. It works only with direct calls to `System.out/err`, and only during request processing cycle. It may not work in other threads that might be created by the application. It cannot be used to intercept logging frameworks that themselves write to the system streams, as those start early and may obtain a direct reference to the streams before the redirection takes place.

Access Logging

Access logging is a related but different feature, which is implemented as a **Valve**. It uses self-contained logic to write its log files. The essential requirement for access logging is to handle a large continuous stream of data with low overhead, so it only uses Apache Commons Logging for its own debug messages. This implementation approach avoids additional overhead and potentially complex configuration. Please refer to the Valves documentation for more details on its configuration, including the various report formats.

Using java.util.logging (Default)[2]

The default implementation of `java.util.logging` provided in the JDK is too limited to be useful. The key limitation is the inability to have per-web application logging, as the configuration is per-VM. As a result, Tomcat will, in the default configuration, replace the default LogManager implementation with a container friendly implementation called **JULI**, which addresses these shortcomings.

JULI supports the same configuration mechanisms as the standard JDK `java.util.logging`, using either a programmatic approach, or properties files. The main difference is that per-`classloader` properties files can be set (which enables easy redeployment friendly webapp configuration), and the properties files support extended constructs which allows more freedom for defining handlers and assigning them to loggers.

JULI is enabled by default, and supports per `classloader` configuration, in addition to the regular global `java.util.logging` configuration. This means that logging can be configured at the following layers:

- Globally

  That is usually done in the `${catalina.base}/conf/logging.properties` file. The file is specified by the `java.util.logging.config.file` System property which is set by the startup scripts. If it is not readable or is not configured, the default is to use the `${java.home}/lib/logging.properties` file in the JRE.

- In the web application

  The file will be `WEB-INF/classes/logging.properties`

The default `logging.properties` in the JRE specifies a `ConsoleHandler` that routes logging to **System.err**. The default `conf/logging.properties` in Apache Tomcat also adds several `FileHandlers` that write to files.

---

[2] http://tomcat.apache.org/tomcat-9.0-doc/logging.html

A handler's log level threshold is **INFO** by default and can be set using **SEVERE**, **WARNING, INFO, CONFIG, FINE, FINER, FINEST** or **ALL**. You can also target specific packages to collect logging from and specify a level.

To enable debug logging for part of Tomcat's internals, you should configure both the appropriate logger(s) and the appropriate handler(s) to use the FINEST or ALL level. e.g.:

```
org.apache.catalina.session.level=ALL
java.util.logging.ConsoleHandler.level=ALL
```

When enabling debug logging it is recommended that it is enabled for the narrowest possible scope as debug logging can generate large amounts of information.

The configuration used by JULI is the same as the one supported by plain `java.util.logging`, but uses a few extensions to allow better flexibility in configuring loggers and handlers. The main differences are:

- A prefix may be added to handler names, so that multiple handlers of a single class may be instantiated. A prefix is a String which starts with a digit and ends with '.'. For example, **22foobar**. is a valid prefix.
- System property replacement is performed for property values which contain `${systemPropertyName}`.
- If using a class loader that implements the `org.apache.juli.WebappProperties` interface (Tomcat's web application class loader does) then property replacement is also performed for `${classloader.webappName}`, `${classloader.hostName}` and `${classloader.serviceName}` which are replaced with the web application name, the host name and the service name respectively.
- By default, loggers will not delegate to their parent if they have associated handlers. This may be changed per logger using the `loggerName.useParentHandlers` property, which accepts a Boolean value.

The root logger can define its set of handlers using `the .handlers` property.

By default, the log files will be kept on the file system forever. This may be changed per handler using the `handlerName.maxDays` property. If the specified value for the property is <=0 then the log files will be kept on the file system forever, otherwise they will be kept the specified maximum days.

There are several additional implementation classes, that can be used together with the ones provided by Java. The notable one is `org.apache.juli.FileHandler`.

`org.apache.juli.FileHandler` supports buffering of the logs. The buffering is not enabled by default. To configure it, use the `bufferSize` property of a handler. The value of 0 uses system default buffering (typically an 8K buffer will be used). A value of **<0** forces a writer flush upon each log write. A value **>0** uses a `BufferedOutputStream` with the defined value but note that the system default buffering will also be applied.

Example `logging.properties` file to be placed in `$CATALINA_BASE/conf`:

```
handlers = 1catalina.org.apache.juli.FileHandler, \
           2localhost.org.apache.juli.FileHandler, \
           3manager.org.apache.juli.FileHandler, \
           java.util.logging.ConsoleHandler

.handlers = 1catalina.org.apache.juli.FileHandler, java.util.logging.ConsoleHandler

############################################################
# Handler specific properties.
# Describes specific configuration info for Handlers.
############################################################

1catalina.org.apache.juli.FileHandler.level = FINE
1catalina.org.apache.juli.FileHandler.directory = ${catalina.base}/logs
1catalina.org.apache.juli.FileHandler.prefix = catalina.

2localhost.org.apache.juli.FileHandler.level = FINE
2localhost.org.apache.juli.FileHandler.directory = ${catalina.base}/logs
2localhost.org.apache.juli.FileHandler.prefix = localhost.

3manager.org.apache.juli.FileHandler.level = FINE
3manager.org.apache.juli.FileHandler.directory = ${catalina.base}/logs
3manager.org.apache.juli.FileHandler.prefix = manager.
3manager.org.apache.juli.FileHandler.bufferSize = 16384

java.util.logging.ConsoleHandler.level = FINE
java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter


############################################################
# Facility specific properties.
# Provides extra control for each logger.
############################################################

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].handlers = \
   2localhost.org.apache.juli.FileHandler

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].handlers = \
   3manager.org.apache.juli.FileHandler

# For example, set the org.apache.catalina.util.LifecycleBase logger to log
# each component that extends LifecycleBase changing state:
#org.apache.catalina.util.LifecycleBase.level = FINE
```

Example `logging.properties` for the servlet-examples web application to be placed in `WEB-INF/classes` inside the web application:

```
handlers = org.apache.juli.FileHandler, java.util.logging.ConsoleHandler

############################################################
# Handler specific properties.
# Describes specific configuration info for Handlers.
############################################################

org.apache.juli.FileHandler.level = FINE
org.apache.juli.FileHandler.directory = ${catalina.base}/logs
org.apache.juli.FileHandler.prefix = ${classloader.webappName}.

java.util.logging.ConsoleHandler.level = FINE
java.util.logging.ConsoleHandler.formatter =
java.util.logging.SimpleFormatter

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].leve
l = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].hand
lers = \
   3manager.org.apache.juli.FileHandler

# For example, set the org.apache.catalina.util.LifecycleBase logger to log
# each component that extends LifecycleBase changing state:
#org.apache.catalina.util.LifecycleBase.level = FINE
```

AUDIT LOGGING

Panopticon Real Time can also produce audit logs. All of the audit logging will be directed to a file prefixed 'panopticon-audit' in the Tomcat `log` folder. The audit logs can be configured just like the regular logs produced by Panopticon Real Time. Refer to Configuring Panopticon Real Time Logs for more information on how to configure logs.

Panopticon Real Time is pre-configured to generate audit logs on an **INFO** level. Most of the messages are logged with **INFO** level. However, there are certain actions that are logged at different levels, such as **FINE**.

The audit logs contain the following information:

Attribute	Description
Timestamp	Timestamp for when the executed action occurred. The format of the timestamp is YYYY-mm-ddTHH:MM:SS (e.g., 2015-12-24T15:30:40).
Log Level	The severity of the log level.
Username	The username of the user that executed the action. The username will be **ANONYMOUS** if the user is not authenticated.
IP-address	The user's IP address.
Action	Detailed message about the executed action.

Audit logs use comma (,) as a delimiter to separate these values.

SERVER MONITORING

Panopticon Real Time publishes the following JMX counters:

- ServerDataRequestCount
- ActiveDataRequestCount
- InfoMessageCount
- ErrorMessageCount
- ActiveRealtimeSubscriptionCount
- LoadedWorkbooksCount
- MemoryStoreObjectCount
- ObjectCount

These can be accessed through any JMX monitoring toolset, such as Jconsole from the Java Development Kit (JDK).

As a basic configuration:

3. Download and install Java Development Kit (JDK) http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.htm
4. Add the following parameters to your Tomcat:
    - -Dcom.sun.management.jmxremote.port=8855
    - -Dcom.sun.management.jmxremote.authenticate=false
    - -Dcom.sun.management.jmxremote.ssl=false
5. Open Jconsole. The jconsole executable can be found in `JDK_HOME/bin`, where `JDK_HOME` is the directory in which the Java Development Kit (JDK) is installed.
6. When the connection dialog opens, you are also given the option of connecting to a remote process.
    - -Host name: name of the machine on which the Java VM is running.
    - -Port number: the JMX agent port number you specified when you started the Java VM (e.g., **8855**)

WEB PORTAL INTEGRATION

Panopticon workbooks can be embedded into existing portals with minimal effort. An iframe folder example in `webapps\panopticon\api` is included in the release, which details how to include the HTML client inside an iframe.

This folder includes the following files:

- CSS for styling and animations
- JavaScript for logic and control
- Help page with the post message example and the list of actions that HTML5 Client supports
    - getWorkbooks

- getDashboards
- getSelectedWorkbook
- getSelectedDashboardParameters
- getBookmarks
- addDashboardChangedListener
- addParametersChangedListener
- setWorkbook
- setDashboard
- setBookmark
- setParemeters

The custom page simply needs to implement the embedded Web client as the source of the iframe tag. For example:

```
<div class=”right-column”>
   <iframe id=”datawatch” src=”[url_to_web_client]”
name=”datawatch”></iframe>
</div>
```

You can then access the JavaScript API through the iframe ID reference, where you can for instance, navigate to another dashboard or workbook.

[19] TROUBLESHOOTING

RESOLVING INSTALLATION ISSUES

Issues are investigated and resolved through investigation and controlled reproduction. A number of known issues are included in the next section and predominately relate to problematic installations of Panopticon Real Time.

If you experience an unknown issue, send complete details to: dasupport@altair.com

Be sure to send this important information to Altair Support in the event of a problem.

Server Log

Panopticon Real Time log files are located in the `[tomcat_home]\logs` folder.

The level of detail for these log files are configured at the ”level” sections of `logging.properties` file in `[tomcat_home]\conf` folder.

By default, it is set to **Error**, while the most verbose is **Info**.

Steps:

1. Edit the value of ”level” in the `logging.properties` file:

   From:

   ```
   org.apache.catalina.core.ContainerBase.[Catalina].[localhost].level = ERROR
   ```

   To:

   ```
   org.apache.catalina.core.ContainerBase.[Catalina].[localhost].level = INFO
   ```

NOTE	Modifying the `level` setting will consume more disk space, so make sure to only do this while troubleshooting.

2. Restart Tomcat after making these changes.

NOTE	Refer to *Configuring Server Logs* for more information. When sending your issue, include your workbook and associated data sources if the issue is specific to a particular workbook.

NO APPROPRIATE PROTOCOL ERROR WHEN PUBLISHING SPLUNK DATA ON PANOPTICON REAL TIME

The Altair log written into Panopticon Real Time log can report errors similar to the following:

Caused by: `javax.net.ssl.SSLHandshakeException`: No appropriate protocol (protocol is disabled or cipher suites are inappropriate)

This is caused by having the SSLv3 disabled by default in the updated versions of JDK.

Steps:

1. Open the /lib/security/java.security file.
2. Comment the following line:

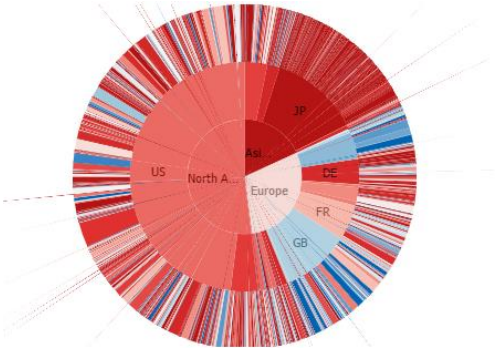
   ```
   #jdk.tls.disabledAlgorithms=SSLv3
   ```

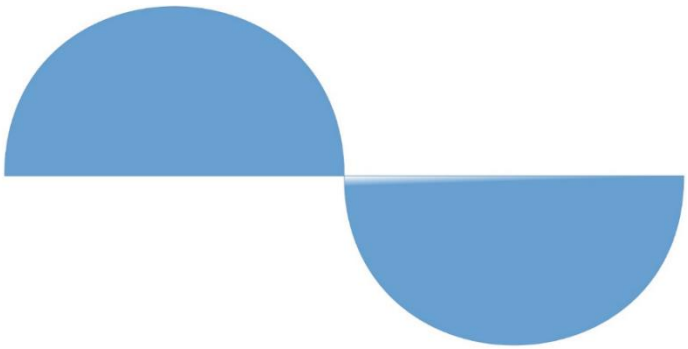
3. Save the updated file.

PIE CHARTS AND SHAPES NOT DISPLAYING CORRECTLY IN CHROME

When Hardware Acceleration is enabled in Chrome, Pie Chart and Shape visualization may not display as expected.

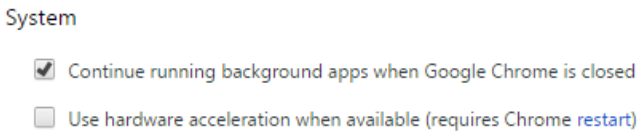
For example:





To resolve this issue, follow the steps below to disable Hardware Acceleration in Chrome:

1. Open the Chrome web browser.
2. You can either:
   - click to the right of the *Address* box and select **Settings**
   - Or enter `chrome://settings` in the *Address* box.
3. Scroll to the bottom of the page and click **Show Advanced Settings…**
4. Uncheck Use Hardware Acceleration when Available box.

5. Restart Chrome.

SESSION TOKENS NOT WORKING IN CHROME

Setting the `authentication.token.persistence` property to **SESSION** in `Panopticon.properties` removes the token from the browser if it is shutdown.

In Google Chrome, you can override the session functionality if you select **Continue where you left off** option in the *On startup* section. However, if you opt to use session cookies, select **Open the New Tab page** option.

Steps

1. Open the Chrome web browser.
2. You can either:
   - click to the right of the *Address* box and select **Settings**
   - Or enter `chrome://settings` in the *Address* box.
3. Scroll to the bottom of the page and on the *On startup* section, you can either select:
   - Open the New Tab page

     To use the session cookies.
   - Continue where you left off

To override the session functionality.

4. Restart Chrome.

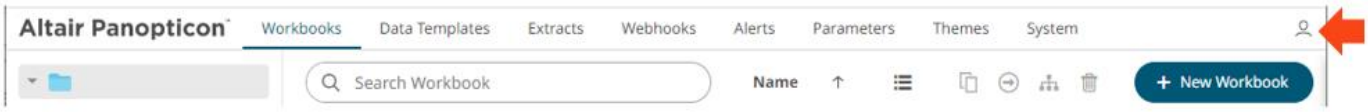
[20] KNOWN ISSUES

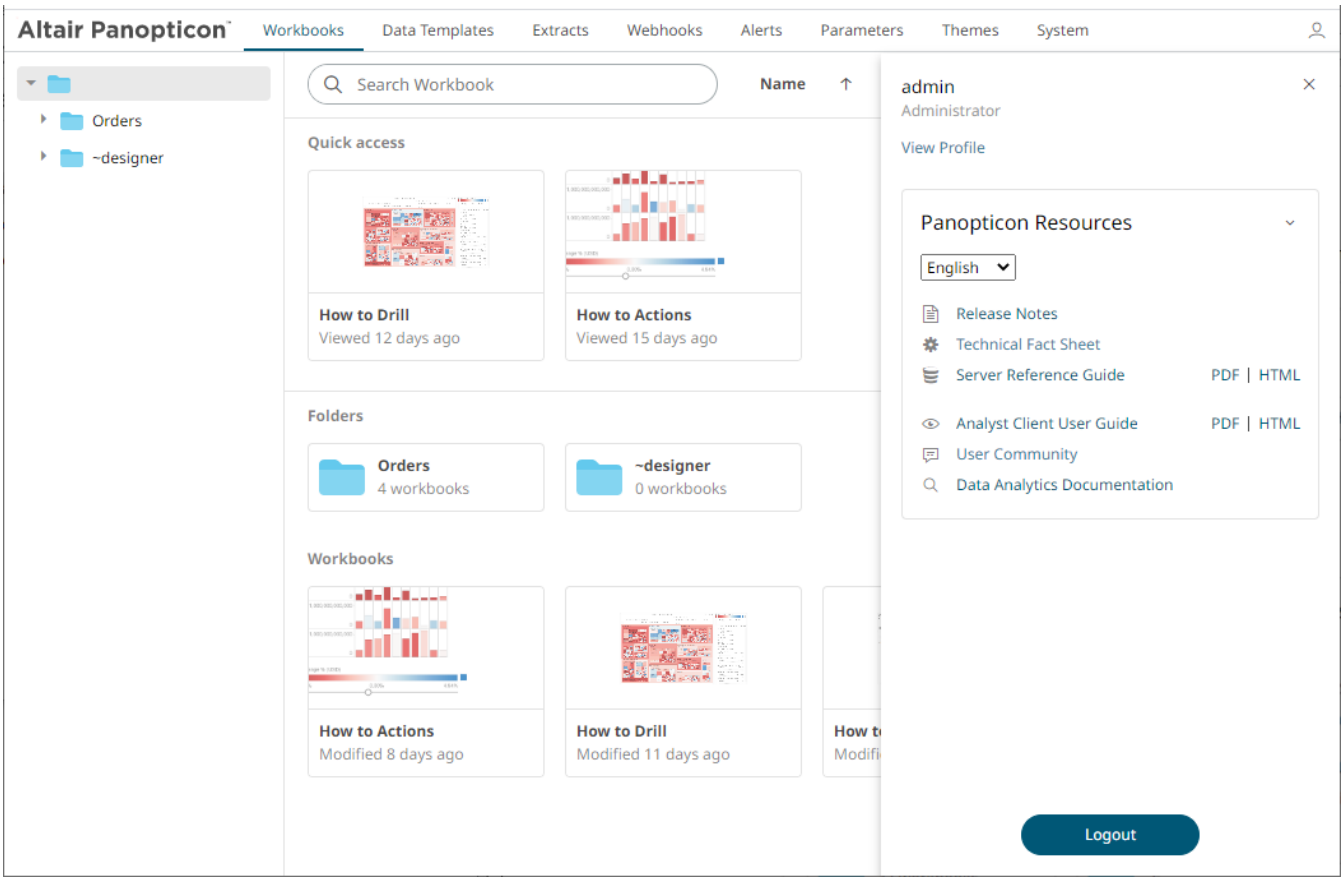
OUT OF MEMORY EXCEPTION

If the data is too big, an out of memory exception may occur. To increase the memory of Panopticon Real Time in Tomcat for Linux, refer to Tomcat Memory Configuration for Linux for more information.

[21] PANOPTICON RESOURCES

Clicking on the top right section of the toolbar displays the other Panopticon online resources that users with an Administrator role can access.





Select the *Language* on the drop-down list: **English** or **Japanese**.

Resource	Description
Release Notes	List of new features and fixed issues in the release.
Technical Fact Sheet	Overview of the Panopticon components which consists of: • system requirements • features • supported data connectivity and visualizations • custom code data connections, transforms, and ML model scoring • filtering and data capabilities
Real Time Reference Guide	Panopticon Real Time documentation for users with an Administrator role which consists of: • installation, system requirements, and licensing options • supported authentication mechanisms • advanced and optional steps or deployments • system administration of the server including the logs, subscriptions, caches, scheduled tasks, and logged in users • viewing and managing of workbooks and data templates • creating and managing of data templates, global parameters, alerts, workbook themes • PCLI command utilities • REST API examples • troubleshooting guide • `Panopticon.properties` discussion Available upon installation.
Analyst Client User Guide	Panopticon Real Time documentation for users with a Viewer role which consists of: • viewing and analysing of workbooks • creating, monitoring, and deleting of alerts Available upon installation.
User Community	Link to the Panopticon User Community page.
Data Analytics Documentation	Link to the Altair Data Analytics Documentation page.

[APPENDIX]

PROPERTIES: PANOPTICON

The `Panopticon.properties` file located in the `PanopticonAppdata` folder (i.e., `c:\vizserverdata`), contains majority of properties for controlling the configuration of Panopticon Real Time. The following properties can be overridden by updating the file.

Property	Access
Attribute	`access.administrator.groups`
Description	The role that is mapped to the administrator group.
Default Value	**admin**
Property	Access
Attribute	`access.default.roles`
Description	The default roles applied to all users of the server. For example, if `access.default.roles`=DESIGNER,ADMINISTRATOR and a user with a VIEWER role logs on to the server, then the user will simultaneously have a VIEWER, DESIGNER, and ADMINISTRATOR 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 ADMINISTRATOR, VIEWER, ANONYMOUS, and/or DESIGNER. This property is case sensitive.
Default Value	**VIEWER**
Property	Access
Attribute	`access.designer.groups`
Description	The role that is mapped to the designer group.
Default Value	**designer**
Property	Access
Attribute	`access.list.delimiter`
Description	The value delimiter to use when parsing access groups. Examples: access.list.delimiter=, access.administrator.groups=group1,group2 The groups are mapped to {'group1', 'group2'} access.list.delimiter=, access.administrator.groups=group1;group2,group3

	The groups are mapped to {'group1;group2', 'group3'} access.list.delimiter=; access.administrator.groups=group1;group2,group3 The groups are mapped to {'group1', 'group2,group3'}
Default Value	**',' (comma)**
Property	Access
Attribute	`access.viewer.groups`
Description	The role that is assigned to the viewer group.
Default Value	
Property	Alert
Attribute	`alert.creation.only.by.administrators`
Description	Enable or disable whether only the Administrators can create alerts.
Default Value	**false**
Property	Alert
Attribute	`alert.detailed.logging`
Description	Enables or disables extra alert logging.
Default Value	**false**
Property	Service authentication level
Attribute	`authentication.domain`
Description	The default domain information for user authentication.
Default Value	
Property	Authentication: Filter Token
Attribute	`authentication.filter.authenticate.token`
Description	Applies only if `authentication.type` is set to FILTER or is blank. If this property is set to true, the server will validate the token on incoming requests. If set to false, it ignores the token and authenticates based on the rest of the request instead.
Default Value	**false**
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`

<table>
<tr><td>Description</td><td>The name of the header that contains all the roles.</td></tr>
<tr><td>Default Value</td><td></td></tr>
<tr><td>Property</td><td>Authentication: Header</td></tr>
<tr><td>Attribute</td><td><code>authentication.header.rolesdynamic</code></td></tr>
<tr><td>Description</td><td>Supports the ability to create dynamic roles using free form patterns or string replacement.
To create dynamic roles, use '{header value to be used}'.
Example:
authentication.header.rolesdynamic={HEADER_ROLES},financials,role_for_company_{HEADER_COMPANY}

Given this table:
<table><tr><th>KEY</th><th>VALUE</th></tr><tr><td>HEADER_ROLES</td><td>designer, watcher</td></tr><tr><td>HEADER_COMPANY</td><td>industrials, consumers</td></tr></table>
Then the roles to create the authentication token will be the following:
designer
watcher
financials
role_for_company_industrials
role_for_company_consumers</td></tr>
<tr><td>Default Value</td><td></td></tr>
<tr><td>Property</td><td>Authentication: Header</td></tr>
<tr><td>Attribute</td><td><code>authentication.header.username</code></td></tr>
<tr><td>Description</td><td>The name of the header that contains the username</td></tr>
<tr><td>Default Value</td><td></td></tr>
<tr><td>Property</td><td>Authentication: Header</td></tr>
<tr><td>Attribute</td><td><code>authentication.header.validate.token</code></td></tr>
<tr><td>Description</td><td>If set to true, the authentication will validate the token. If set to false, the authentication of every request will be based on headers.</td></tr>
<tr><td>Default Value</td><td></td></tr>
<tr><td>Property</td><td>Authentication: Logout</td></tr>
<tr><td>Attribute</td><td><code>authentication.logout.redirect.url</code></td></tr>
<tr><td>Description</td><td>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.</td></tr>
<tr><td>Default Value</td><td></td></tr>
</table>

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.
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: `cn=admin,ou=groups,dc=openam,dc=openidentityplatform,dc=org,cn=designer,ou=groups,dc=openam,dc=openidentityplatform,dc=org` 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	`authentication.oauth2.identity.url`
Description	The URL to the REST service that provides details about the authenticated user.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	`authentication.oauth2.login.callback.url`
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 Real Time.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	`authentication.oauth2.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: OAuth 2.0
Attribute	`authentication.oauth2.login.response.type`
Description	The response type. The only response type that is currently supported is **code**.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	`authentication.oauth2.login.scope`
Description	The requested scope.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	`authentication.oauth2.login.state`
Description	The requested state.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	`authentication.oauth2.login.url`
Description	The URL to the OAuth 2.0 login resource.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	`authentication.oauth2.logout.redirect.url`
Description	Logging out revokes the token from the authentication server if the property authentication.oauth2.logout.url 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 the 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.
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.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	Service authentication level
Attribute	`authentication.role`
Description	The authentication role.
Default Value	
Property	Authentication: SAML
Attribute	`authentication.saml.assertion.roles`
Description	User attribute for roles configured in the IdP.
Default Value	
Example	`authentication.saml.assertion.roles=roles`
Property	Authentication: SAML
Attribute	`authentication.saml.assertion.username`
Description	User attribute for username configured in the IdP.
Default Value	
Example	`authentication.saml.assertion.username=name`
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 Example: `http://localhost:8080/panopticon/server/rest/auth/login`
Default Value	
Example	`authentication.saml.assertionconsumerservice.url=http://localhost:8080/panopticon/server/rest/auth/login`
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	
Example	`authentication.saml.certificate.name=saml-cert`
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.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.logout.url`
Description	The URL to the IdP logout service.
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.identityprovider.url`
Description	The URL to the IdP login service.
Default Value	
Example	`authentication.saml.identityprovider.url=https://192.168.99.100:443/simplesaml/saml2/idp/SSOService.php`
Property	Authentication: SAML
Attribute	`authentication.saml.keystore.file`
Description	The location of the Keystore file that contains the certificate.
Default Value	

Example	`authentication.saml.keystore.file=D:/SAML/mykeystore.jks`
Property	Authentication: SAML
Attribute	`authentication.saml.keystore.password`
Description	The password to the Keystore file.
Default Value	
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	Authentication: SAML
Attribute	`authentication.saml.provider`
Description	The IdP provider. Possible values are **OPENSAML**, **OPENAM**.
Default Value	**OPENSAML**
Property	Authentication: SAML
Attribute	`authentication.saml.serviceprovider.id`

Description	The ID of the service provider configured in the IdP.
Default Value	
Example	`authentication.saml.serviceprovider.id=DwchFrontLocal8080`
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	The name of the cookie used to store the authentication cookie. Must be unique for each server instance on the host.
Default Value	**ptoken**
Property	Authentication: Token
Attribute	`authentication.token.cookie.httponly`
Description	This property determines how the browser will treat the cookie. If set to **true**, the cookie will be stored in the browser as a HttpOnly cookie and will not be available to the JavaScript. If set to **false** (default), the cookie will be stored in the browser as https and will be accessible to the JavaScript.
Default Value	**false**
Property	Authentication: Token
Attribute	`authentication.token.cookie.samesite`
Description	Used by browsers to control the behavior of same or cross origin requests. There are three possible values. **Lax**, **Strict**, and **None**. Please refer to browser specific documentation for its usage.
Default Value	**Lax**
Property	Authentication: Token
Attribute	`authentication.token.cookie.secure`
Description	The property determines how the browser will treat the cookie depending on the security of the connection. If set to **true**, when the browser receives a secure cookie (HttpOnly cookie), you will not be able to transmit it unless the connection is secure.
Default Value	**false**
Property	Authentication: Token
Attribute	`authentication.token.domain`
Description	Specifies the token cookie domain.
Default Value	
Property	Authentication: Token
Attribute	`authentication.token.in.login.response.body`

Description	This property determines if the REST login response body should contain a token info. **NOTE:** Does not affect the SOAP login response body.
Default Value	**false**
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 `AppData/Token` 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	Service authentication level
Attribute	`authentication.type`
Description	The type of the authentication mechanism that will be used on the Server.
Default Value	**BASIC**
Property	Bookmark Administration
Attribute	`bookmark.administration.only.by.administrators`
Description	Set to **true** if only Administrators should be able to manage bookmarks.
Default Value	**false**

Property	Cache
Attribute	`cache.data.datasource.enabled`
Description	Enable or disable the caching of the data source.
Default Value	**true**
Property	Cache
Attribute	`cache.data.datasource.size`
Description	The data source cache size.
Default Value	**100**
Property	Cache
Attribute	`cache.data.datasource.type`
Description	The data source cache type.Allowed values: **MEMORY, NONE**
Default Value	**MEMORY**
Property	Cache
Attribute	`cache.data.datatable.enabled`
Description	Enable or disable the caching of the data table.
Default Value	**true**
Property	Cache
Attribute	`cache.data.datatable.size`
Description	The data table cache size.
Default Value	**100**
Property	Cache
Attribute	`cache.data.datatable.type`
Description	The data table cache type.Allowed values: **MEMORY, NONE**
Default Value	**MEMORY**
Property	Cache
Attribute	`cache.data.query.enabled`
Description	Enable or disable the caching of data query.
Default Value	**true**
Property	Cache
Attribute	`cache.data.query.size`
Description	The data query cache size.
Default Value	**100**
Property	Cache

Attribute	`cache.data.query.type`
Description	The data query cache type.Allowed values: **MEMORY, NONE**
Default Value	**MEMORY**
Property	Cache
Attribute	`cache.plugin.id`
Description	The ID of the plugin that will be used to store data extracts. Possible values: **BinaryTableFile-Cache**.
Default Value	**BinaryTableFile-Cache**
Property	Cache
Attribute	`cache.purge.condition`
Description	Defines the condition for when the cache will be purged. Allowed values: NONE, MEMORY
Default Value	**MEMORY**
Property	Cache
Attribute	`cache.purge.condition.memory.threshold`
Description	Defines a percentual memory threshold for cache purging, when the cache.purge.condition = MEMORY.
Default Value	**80**
Property	Cache
Attribute	`cache.purge.enabled`
Description	Enables scheduled cache purging.
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	Cache
Attribute	`cache.service.enabled`
Description	Enables or disables the service cache.
Default Value	**true**
Property	Cache
Attribute	`cache.service.type`
Description	The service cache mechanism being used.
Default Value	**IN_MEMORY**

Property	Client Cache
Attribute	`client.cache.control.age.max`
Description	Controls the cache-control max-age header for static content.
Default Value	**31536000**
Property	Client Data
Attribute	`client.data.load.transport`
Description	Configure the transportation protocol for loading data from the Web client. Possible values: WEBSOCKET, LONG_POLLING. **NOTE:** This property has been deprecated. Refer to Setting the Transportation Protocol for more information.
Default Value	**WEBSOCKET**
Property	Server Cluster
Attribute	`cluster.bully.bind`
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	`cluster.bully.boot`
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 cluster.bully.bind value on each boot server.
Default Value	
Property	Server Cluster
Attribute	`cluster.bully.id`
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	`cluster.fixed.leader`
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	`cluster.kubernetes.container_name`

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	`cluster.kubernetes.id`
Description	Set to the name of the pod that runs the container.
Default Value	(blank)
Property	Server Cluster
Attribute	`cluster.kubernetes.label_selector`
Description	Standard Kubernetes label selector that should only match the pods that are running the server.
Default Value	
Property	Server Cluster
Attribute	`cluster.kubernetes.peer_path`
Description	Path to the web application on each server. For example, "panopticon/", or "/" if you have deployed to Tomcat's root.
Default Value	
Property	Server Cluster
Attribute	`cluster.mode`
Description	**NONE** (default), **FIXED, BULLY,** or **KUBERNETES**
Default Value	
Property	Server Cluster
Attribute	`cluster.shared.secret`
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	`cluster.shared.store.shared_directory.path`
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	`cluster.shared.store.type`
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	Host Lookup
Attribute	`connector.kdb.host.lookup.script`

Description	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.
	`{ "host": "localhost", "port": 5001, "username": "", "password": "" }`
	
	NOTE: Starting with the 21.2 release, the the kdb+ connection pool feature of Panopticon (`kdb.connection.pool.xx`) can be used together wth the host lookup. So any new connection request from the pool, will first execute the script set here, to get the host info before the pool is looked up for available connections.
	Examples:
	• For Windows
	`connector.kdb.host.lookup.script=E://Data/host.bat`
	• For Linux
	`connector.kdb.host.lookup.script= /etc/panopticon/appdata/host.sh`

Default Value	
Property	Host Lookup
Attribute	`connector.kdb.host.lookup.script.arguments`
Description	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. 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.
Default Value	**{host},{port},{userid},{password}**
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	`connector.kinesis.datastreams.accesskeyid`
Description	The Access Key ID from the AWS account.
Default Value	
Property	Amazon Kinesis – Data Streams connector
Attribute	`connector.kinesis.datastreams.secretaccesskey`
Description	The Secret Access Key ID from the AWS account.
Default Value	
Property	OAuth Token URL
Attribute	`connector.oauth.tokenurl`
Description	Sets the server-wide token URL.
Default Value	**http\://localhost\:5000/oauth/token**
Property	Python connector
Attribute	`connector.python.host`
Description	The default Python Pyro instance host address. **NOTES:** For `connector.python.host`, `connector.python.password`, `connector.python.port`, and `connector.python.serializertype` properties : • If set in the `Panopticon.properties` file, these fields will be hidden in the Python connector and will be applied to the Python transform as well. • These default Panopticon Real Time connection properties will be applied at runtime. These default Panopticon Real Time connection properties will override old Python connection settings.
Default Value	
Property	Python connector
Attribute	`connector.python.password`
Description	The default HMAC Key.
Default Value	
Property	Python connector
Attribute	`connector.python.port`
Description	The default Python Pyro host port.

Default Value	
Property	Python connector
Attribute	`connector.python.serializertype`
Description	The default Python serialization type. Possible values are **serpent** or **pickle**.
Default Value	
Property	Rserve connector
Attribute	`connector.rserve.host`
Description	The default Rserve host address. **NOTES:** For `connector.rserve.host`, `connector.rserve.password`, `connector.rserve.port`, and `connector.rserve.userid` properties: • If set in the `Panopticon.properties` file, these fields will be hidden in the Rserve connector and will be applied to the R transform as well. • These default Panopticon Real Time connection properties will be applied at runtime. These default Panopticon Real Time 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	Alert
Attribute	`email.address`
Description	The email address where the alert will be sent from.

Default Value	
Property	Email
Attribute	`email.host`
Description	The host name used by the email server.
Default Value	
Property	Alert
Attribute	`email.password`
Description	The email password, if available.
Default Value	
Property	Email
Attribute	`email.port`
Description	The port number used by the email server.
Default Value	
Property	Email
Attribute	`email.security.mode`
Description	The security mode used when sending emails. Possible values: **NONE**, **SSL**, **TLS**.
Default Value	**NONE**
Property	Email
Attribute	`email.username`
Description	Email account username.
Default Value	
Property	Error Message
Attribute	`error.default.message`
Description	Defines a generic error message override.
Default Value	
Property	Image export
Attribute	`export.image.height`
Description	The default height for an exported image.
Default Value	**768**
Property	Image export
Attribute	`export.image.width`
Description	The default width for an exported image.
Default Value	**1024**

Property	File Upload
Attribute	`file.upload.size.max.bytes`
Description	Limit for files size (in bytes) to be uploaded through the web browser (i.e., workbooks, streams applications, streams data sources).
Default Value	**30000000**
Property	Copy Image
Attribute	`image.client.timeout`
Description	Specifies a timeout (in milliseconds) when producing an image or PDF. If it takes longer than the timeout, the process will be interrupted, and the image/PDF will not be produced.
Default Value	**600000**
Property	kdb+ Connection Pooling
Attribute	`kdb.connection.pool.max.size`
Description	The maximum number of connections that will be kept open for reuse for each kdb+ server (among kdb+ servers that use the same username, password, TLS flag, and timeout), so that established connections can be reused when subsequent queries come in for the same server. A benefit of the connection pool is that it can reduce latency. Setting this property to **0** disables the connection pool.
Default Value	**10**
Property	kdb+ Connection Pooling
Attribute	`kdb.connection.pool.ttl`
Description	Time to live in milliseconds for each connection instance created.
Default Value	**30000**
Property	Licensing
Attribute	`license.hwu.hosted`
Description	Boolean stating if you wish to use Hosted or Local Altair Units licensing. Set to **true** if you wish to use hosted licensing.
Default Value	**false**
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 Hosted Altair Units system.
Default Value	
Property	Licensing

Attribute	`license.hwu.hosted.authorization.username`
Description	Username to the Altair One account.
Default Value	
Property	Licensing
Attribute	`license.hwu.operating.system`
Description	The operating system where Panopticon Real Time is installed. Possible values are: **WIN_X86**, **WIN_X64**, **MAC**, **LINUX_X64**, or **LINUX_ARM64** **NOTE:** If the Java bitness (e.g., **32-bit**) is different from the operating system (e.g., **64-bit**), it is recommended to add the Java bitness in this property (e.g., **WIN_X86**).
Default Value	
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 `Panopticon.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.hwu.version`
Description	Value must match the license version found in the Altair Units license file.
Default Value	**19.0**
Property	Licensing
Attribute	`license.mode`
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	Log level
Attribute	`logger.level.file`
Description	Controls the level that is logged to file.
Default Value	**WARNING**
Property	Server Metrics

Attribute	`metrics.authorization.level`
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	`metrics.collection.rate`
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 specifies 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**
Property	Bookmarks repository
Attribute	`repository.import.bookmarks.paths`
Description	Will import bookmarks from the old format into the repository. Will override any existing bookmarks inside the repository. Must be set to an absolute path. Only bookmarks for workbooks that exists inside the repository will be imported.
Default Value	
Property	Workbook repository
Attribute	`repository.migrate.archive.path`

Description	Use this property if you have an older (pre 2020) server and wish to start the new server with the same workbook content as the old one, and also to import the workbooks' change history from the old server. Set the property to the absolute path to the old server's `<AppData>/Archive/` directory, delete the new server's `<AppData>/.repository/` directory, and start the new server. You typically use this property with the `repository.migrate.workbooks.path` property. See also the section on content migration.
Example	**C\:/vizserverdata/Archive**
Default Value	
Property	Bookmarks repository
Attribute	`repository.migrate.bookmarks.path`
Description	Will migrate bookmarks from the old format into the repository if there are no bookmarks inside the repository yet. Set to an absolute path or to the default `Bookmarks` folder. Only bookmarks for workbooks that exists inside the repository will be migrated. **NOTE:** If you do not wish to migrate bookmarks or already have bookmarks in the repository, set this property to blank to avoid a warning on startup.
Default Value	**Bookmarks**
Property	Workbook repository
Attribute	`repository.migrate.data.extracts.path`
Description	Starting with version 21.0, data extracts are stored inside the repository. If this property is set to **GlobalCaches** (default value), or to an absolute path, the server will migrate data extracts into the repository on startup as long as the repository does not contain any previous data extracts. **NOTE:** If you do not wish to migrate data extracts or already have data extracts in the repository, set this property to blank to avoid a warning on startup.
Default Value	**GlobalCaches**
Property	Data Templates Repository
Attribute	`repository.migrate.datatable.templates.path`
Description	Will migrate data table templates from the old format into the repository if there are no data table templates inside the repository yet. Set to an absolute path or to the default `Datatables` folder. **NOTE:** If you do not wish to migrate data table templates or already have data table templates in the repository, set this property to blank to avoid a warning on startup.
Default Value	**Datatables**
Property	Themes repository
Attribute	`repository.migrate.themes.path`
Description	Will migrate themes from the old format into the repository if there are no themes inside the repository yet. Set to an absolute path or to the default Themes folder. **NOTE:** If you do not wish to migrate themes or already have themes in the repository, set this property to blank to avoid a warning on startup.
Default Value	**Themes**
Property	Workbook repository

Attribute	`repository.migrate.workbooks.path`
Description	Use this property if you have an older (pre 2020) server and wish to start the new server with the same workbook content as the old one. Set the property to the absolute path to the old server's `<AppData>/Workbooks/` directory, delete the new server's `<AppData>/.repository/` directory, and start the new server. See also the section on content migration.
Example	**C\:/vizserverdata/Workbooks**
Default Value	
Property	Workbook repository
Attribute	`repository.pack.enabled`
Description	The repository tracks all changes to all workbooks. If you have a very large number of workbooks, or have kept the repository for a very long time, the sheer number of files inside the `.repository` subdirectory could cause the repository to become slower. Set this property to true to have the repository pack all the files into fewer larger ones for faster access.
Default Value	**false**
Property	Repository
Attribute	`repository.startup.apply.permissions.clean`
Description	Use this property with the `repository.startup.apply.permissions.path` to reset all existing workbook permissions on the server before applying the template. If you set it to **true**, the server will remove all permissions, then give users full permissions to their private folders, and the "Everyone" group full permissions to public folders.
Default Value	**false**
Property	Repository
Attribute	`repository.startup.apply.permissions.create`
Description	Use this property with the `repository.startup.apply.permissions.path` to create empty workbook folders for any folders that are in the template file but do not yet exist on the server. If you don't set it to **true**, these folders from the template will be ignored.
Default Value	**true**
Property	Repository
Attribute	`repository.startup.apply.permissions.path`
Description	Use this property to make the server apply workbook folder permissions from a template JSON file on startup. Workbook folder permissions in the template will overwrite any existing permissions on the server. This property will not migrate permissions from an older (pre 2020) server, you need to use the PCLI `convertpermissions` to generate a template file from the old permissions first. See also `repository.startup.apply.permissions.clean` and `repository.startup.apply.permissions.create`.
Default Value	
Property	Repository
Attribute	`repository.startup.filesystemcheck`

Description	If set to **true,** server runs on startup to verify the repository integrity and reports any of the following issues: • 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/`. **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.
Default Value	**false**
Property	Repository Import
Attribute	`repository.startup.import.paths`
Description	**NOTE:** Use this property to make the server import content at startup. This is imported on top of the existing content and will always overwrite anything that is already there. This property can be useful for example, if you have multiple servers with different content but you want the latest version of a standard set of workbooks to be deployed on all of them. This property only has effect on a stand-alone or leader server. This property is the list of paths to directories and files, separated by the system specific path separator ";" on Windows and ":" on Linux. Each directory is scanned and imported keeping its local tree structure. For example, workbooks to be imported are placed in a folder and in this property, the absolute path to that folder is specified. **ADDITIONAL NOTES:** • User-specific folders (e.g. "~john/") can be targeted this way, but only if they already exist on the server. • Bundles (exz files) directly listed in the property or found in directories listed are also imported, but always to the root, with their internal structure preserved. • Files that are not legacy workbooks or bundles are ignored. • The same set of workbooks will get imported over and over (startup, user edit, restart) and for bundles (nothing changes in the history the second time), but legacy workbooks change their meta data. • The import always overwrites local changes (it resets the workbooks in the repository). Permissions are not supported, and any folders created will have "SYSTEM" as owner.
Default Value	
Property	Request parameter mapping
Attribute	`request.cookie.parameters.mapping.entry.delimiter`
Description	The delimiter that separates the configuration entries. This property will only affect incoming parameters.
Default Value	**, (Comma)**
Property	Request parameter mapping
Attribute	`request.cookie.parameters.mapping.optional`

Description	The parameters that could be updated with certain cookie values. This property will only affect incoming parameters. The operation will not fail if the cookie values are not present in the request. The parameters will keep their default value instead of the configured cookie value if the cookie is not present. The property should be formatted as follows: Parameter name (Value delimiter) Cookie name.
Default Value	
Property	Request parameter mapping
Attribute	`request.cookie.parameters.mapping.required`
Description	The parameters that are required to be updated with certain cookie values. This property will only affect incoming parameters. The operation will fail if configured cookie values are not present in the request. The property should be formatted as follows: Parameter name (Value delimiter) Cookie name.
Default Value	
Property	Request parameter mapping
Attribute	`request.cookie.parameters.mapping.value.delimiter`
Description	The delimiter that separates the parameter name and the cookie name. This property will only affect incoming parameters.
Default Value	**: (Colon)**
Property	Request parameter mapping
Attribute	`request.header.parameters.mapping.entry.delimiter`
Description	The delimiter that separates the configuration entries. This property will only affect incoming parameters.
Default Value	**, (Comma)**
Property	Request parameter mapping
Attribute	`request.header.parameters.mapping.optional`
Description	The parameters that could be updated with certain header values. This property will only affect incoming parameters. The operation will not fail if the header values are not present in the request. The parameters will keep their default value instead of the configured header value if the header is not present. The property should be formatted as follows: Parameter name (Value delimiter) Header name.
Default Value	
Property	Request parameter mapping
Attribute	`request.header.parameters.mapping.required`
Description	The parameters that are required to be updated with certain header values. This property will only affect incoming parameters. The operation will fail if a configured header values are not present in the request. The property should be formatted as follows: Parameter name (Value delimiter) Header name.
Default Value	
Property	Request parameter mapping
Attribute	`request.header.parameters.mapping.value.delimiter`

Description	The delimiter that separates the parameter name and the header name. This property will only affect incoming parameters.
Default Value	**: (Colon)**
Property	Response parameter mapping
Attribute	`response.operation.parameters.mapping.entry.delimiter`
Description	The delimiter that separates the configuration entries. This property will only affect outgoing parameters.
Default Value	**, (Comma)**
Property	Response parameter mapping
Attribute	`response.operation.parameters.mapping.optional`
Description	The parameters that could be updated with certain Header values. This property will only affect outgoing parameters. The operation will not fail if the Header values are not present in the request. The parameters will keep their default value instead of the configured Header value if the Header is not present. The property should be formatted as follows: Parameter name (Value delimiter) Header name.
Default Value	
Property	Response parameter mapping
Attribute	`response.operation.parameters.mapping.required`
Description	The parameters that are required to be updated with certain Header values. This property will only affect outgoing parameters. The operation will fail if configured Header values are not present in the request. The property should be formatted as follows: Parameter name (Value delimiter) Header name.
Default Value	
Property	Response parameter mapping
Attribute	`response.operation.parameters.mapping.value.delimiter`
Description	The delimiter that separates the parameter name and the Header name. This property will only affect incoming parameters.
Default Value	**: (Colon)**
Property	REST
Attribute	`rest.response.error.stacktrace.included`
Description	Include the error stack trace in REST responses.
Default Value	**false**
Property	Compatibility
Attribute	`server.force_downgrade`
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 irreversable 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	Email
Attribute	`server.host`
Description	The server endpoint address. This will be used to generate links in emails sent by the server, so it should be the server's or load balancer's public URL and needs to be resolvable from the email recipient's machine. For example: `server.host=http://www.company.com/dashboards/`
Default Value	
Property	PDF and Image generation
Attribute	`server.host.internal`
Description	The local server endpoint address. To generate PDFs and images, the server fires up an external process which then makes HTTP calls to the server itself. This URL needs to be resolvable on the server itself. For example: `server.host.internal=http://127.0.0.1:8080/panopticon/`
Default Value	
Property	Server
Attribute	`server.id`
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	SOAP
Attribute	`soap.enabled`
Description	Enable or disable the SOAP interface
Default Value	**true**
Property	Data table regression testing
Attribute	`startup.regression.datatable.exclude.folders`
Description	Comma-separated list of folders that will be excluded in the testing. Use this property in combination with the `startup.regression.datatable.include.folders` property to control which workbooks to include in the testing. For example, you can set `startup.regression.datatable.include.folders` to **"pub\\"** and `startup.regression.datatable.exclude.folders` to **"pub\\examples\\,pub\\temp\\"**.
Default Value	
Property	Data table regression testing
Attribute	`startup.regression.datatable.include.folders`
Description	Comma-separated list of folders to test. The default is blank, which means the root folder and all workbooks will be tested. If you list folders here, then only the data tables in workbooks in these folders will be tested, unless also excluded. Folder paths should include a trailing backslash, and you need to

	use double backslashes since this is the escape character in Java property files. For example, to only include **prod** and **qa\final**, you should set the property to "**prod\\,qa\\final**" (without quotes).
Default Value	
Property	Data table regression testing
Attribute	`startup.regression.datatable.runonce`
Description	If set to **true**, the server will run a data table regression test during the next startup. The property is immediately reset to **false**, so you need to set it to **true** again to run another test. **NOTE:** You can set the property through an environment variable if you want to force the server to run it on every startup.
Default Value	**false**
Property	Authorization
Attribute	`statistics.authorization.level`
Description	Allows users to set the authorization level for the statistics and diagnostic REST services. Possible values include: **ANONYMOUS**, **VIEWER**, **DESIGNER**, **ADMINISTRATOR**. **NOTE:** This property is case sensitive.
Default Value	**ADMINISTRATOR**
Property	Subscription
Attribute	`subscription.broadcasting.pool.max.size`
Description	The maximum number of threads for the broadcasting thread pools of refresh events. The default value is empty, which means that there is no limit. Any value less than 1 also means that there is no limit. When setting a max value for the thread pools, it means that the pool cannot create more than that number of threads. If there are more concurrent events handled by the thread pools than there are threads, they are queued until a thread becomes available. The thread pools are also configured to only increase the pool size if all threads are busy and a new event needs to be processed. If a thread is idle more than 1 minute, it will be removed from the pool and the size of the pool thereby decreases. Any subscription for a static data source are scheduled to refresh each X seconds (based of the refresh period of the datatable) using the TaskScheduled built in to Spring. If multiple subscriptions with the same data query tries to load data at the same time, only one thread will actually load the data. The rest of the subscriptions are queued. When the data is loaded all waiting subscriptions will be given the same data set that are then broadcasted to their respective client.
Default Value	
Property	Subscription
Attribute	`subscription.compression.delta.enabled`
Description	With delta compression, the server only sends the difference from the last data result on each refresh. For data where only a fraction changes on each refresh, this means much smaller response messages. The trade-offs are that both client and server needs to keep the last result to calculate the difference and apply it, and that this operation takes some additional time both on the server and the client.

	In rare cases, delta compression may worsen performance, e.g., if you have a large data set with very high refresh rate and a large portion of the data changes on each refresh. You can then disable delta processing completely by setting this property to **false**.
Default Value	**true**
Property	Subscription
Attribute	`subscription.compression.enabled`
Description	Enable or disable compression and encoding of subscription broadcast messages.
Default Value	**true**
Property	Subscription
Attribute	`subscription.congestion.control.enabled`
Description	When the server loads data for a subscription, it checks that the previous data load for it has completed. If not, it might be a sign that the refresh rate is set too high on the data table. If this happens subscription.maximum.failure times in a row, the server will cancel the subscription. Set this property to **false** to disable this behavior.
Default Value	**true**
Property	Subscription
Attribute	`subscription.data.loading.pool.max.size`
Description	The maximum number of threads for loading thread pools of refresh events. The default value is empty, which means that there is no limit. Any value less than 1 also means that there is no limit. When setting a max value for the thread pools, it means that the pool cannot create more than that number of threads. If there are more concurrent events handled by the thread pools than there are threads, they are queued until a thread becomes available. The thread pools are also configured to only increase the pool size if all threads are busy and a new event needs to be processed. If a thread is idle more than 1 minute, it will be removed from the pool and the size of the pool thereby decreases. Any subscription for a static data source are scheduled to refresh each X seconds (based of the refresh period of the datatable) using the TaskScheduled built in to Spring. If multiple subscriptions with the same data query tries to load data at the same time, only one thread will actually load the data. The rest of the subscriptions are queued. When the data is loaded all waiting subscriptions will be given the same data set that are then broadcasted to their respective client.
Default Value	
Property	Subscription
Attribute	`subscription.limitation.action`
Description	Controls the behavior when the subscription.limitation.limit is reached. Allowed values: EXCEPTION, PURGE
Default Value	**EXCEPTION**
Property	Subscription
Attribute	`subscription.limitation.enabled`
Description	Enables limitation of subscriptions.
Default Value	**false**

Property	Subscription
Attribute	`subscription.limitation.limit`
Description	Defines a subscription limit.
Default Value	**100**
Property	Subscription
Attribute	`subscription.log.slow.data.loads.seconds`
Description	Logs a subscription that has been loading data for more than X seconds at a WARNING level. **NOTES:** • Any integer less than 1 (or an empty value) will disable the logging. • If a slow data load has been logged and then returns data, a log message at INFO level will be printed stating that a previously logged slow data load has returned data.
Default Value	**60**
Property	Subscription
Attribute	`subscription.maximum.failure`
Description	The amount of time a subscription is allowed to fail in a row before it should be cancelled. The number will be reset to zero if data loading is successful. The maximum failure limit is used so that invalid subscription will not loop forever and fill the logs with error messages. The value -1 will disable the fail mechanism. This means that a subscription can fail endless of times and not be cancelled.
Default Value	**5**
Property	Subscription
Attribute	`subscription.purge.condition`
Description	Defines the condition for when subscriptions will be purged. Allowed values: NONE, MEMORY
Default Value	**NONE**
Property	Subscription
Attribute	`subscription.purge.condition.memory.threshold`
Description	Defines a percentual memory threshold for subscription purging, when the `subscription.purge.condition = MEMORY`.
Default Value	**80**
Property	Subscription
Attribute	`subscription.purge.enabled`
Description	Enables subscription purging.
Default Value	**true**
Property	Subscription
Attribute	`subscription.purge.post.restart`

Description	Option to re-start active subscriptions after purge. Only valid when `subscription.purge.scope = ALL`
Default Value	**false**
Property	Subscription
Attribute	`subscription.purge.rate`
Description	Defines a fixed rate, in milliseconds. for subscription purging.
Default Value	**10000**
Property	Subscription
Attribute	`subscription.purge.scope`
Description	Defines the scope of subscriptions to purge. Allowed values: **NON_PERSISTENT_ORPHANS**, **ALL**.
Default Value	**NON_PERSISTENT_ORPHANS**
Property	Timeout Session
Attribute	`timeout.session.enabled`
Description	Boolean value stating if timeout functionality should be used or not.
Default Value	**false**
Property	Timeout Session
Attribute	`timeout.session.exception.delimiter`
Description	The delimiter to use for the usernames stated in the timeout.session.exception.usernames property.
Default Value	**, (comma)**
Property	Timeout Session
Attribute	`timeout.session.exception.usernames`
Description	Usernames that should be excluded from the timeout functionality. Separated by the delimiter stated in the `timeout.session.exception.delimiter` property.
Default Value	
Property	Timeout Session
Attribute	`timeout.session.minutes`
Description	Minutes of inactivity before a user session is terminated by logging out the user.
Default Value	**480**
Property	Timeout Session
Attribute	`timeout.session.notification.minutes`
Description	Minutes before a timeout that a notification about session timeout is sent to the user.
Default Value	**1**
Property	WebSocket Connection
Attribute	`transport.buffer.size.max.bytes`

Description	Maximum size of message buffer for the WebSocket connections.
Default Value	**10000000**
Property	WebSocket Connection
Attribute	`transport.message.size.max.bytes`
Description	Maximum size of messages for the WebSocket connections.
Default Value	**1000000**

CONTACT US

GET IN TOUCH

We'd love to hear from you. Here's how you can reach us.

SALES CONTACT INFORMATION

US: + 1.800.445.3311
International: + 1.978.441.2200

Sales Email

US: sales@datawatch.com
Europe: sales_euro@datawatch.com
Asia Pacific: sales_apac@datawatch.com

SUPPORT CONTACT INFORMATION

Customer Portal: https://www.altair.com/data-analytics-support/
Email: mailto:dasupport@altair.com
US: +1 800.988.4739
Canada: +1 978.275.8350
Europe, Middle East, Africa: +44 (0) 8081 892481