Altair Panopticon™

**PANOPTICON REAL TIME 2024.0 -
INSTALLATION AND REFERENCE GUIDE**

TABLE OF CONTENTS

[16] PCLI: COMMAND UTILITIES FOR PANOPTICON REAL TIME 291

[17] REST INTERFACE 299

[1] INTRODUCTION

Altair® Panopticon™ is a product for data visualization, data analysis, data monitoring, and reporting. Users create dashboards, alerts, and reports by working in a browser-based environment with a point-and-click, drag-and-drop experience. A multitude of different data connectors are included in the product, enabling connectivity to local files, web URLs, databases, and streaming data sources. Panopticon supports blending data from separate data sources on-the-fly, in-memory. Tens of thousands of individual data points can be graphically visualized at once. The dashboard applications developed by users in a Designer role can subsequently benefit a larger group of users in Viewer role.

The product has access control settings for collections of dashboards, called workbooks, and data tables. Allow and Deny settings are available for capabilities **Read**, **Write**, and **Modify**, for individual users and for groups of users. In addition, Data Policies can be applied to data tables, which controls what parts of the data are accessible to different users, provided that the user is Allowed to Read the data table at all.

A Panopticon server will act as an automated data monitoring server when users create Alerts, which is a collection of data evaluation rules for a specific dashboard and data table. An alert that is triggered as a result of detecting data that matches the criteria, can send out a notification on screen, via email and as collaboration platform webhook message to for example Teams or Slack.

Panopticon has built-in task scheduling capabilities, which can be used for automatically distributing PDF report versions of dashboards via email, or sending data exports via email, or refreshing Data Store cached data tables.

SYSTEM ARCHITECTURE
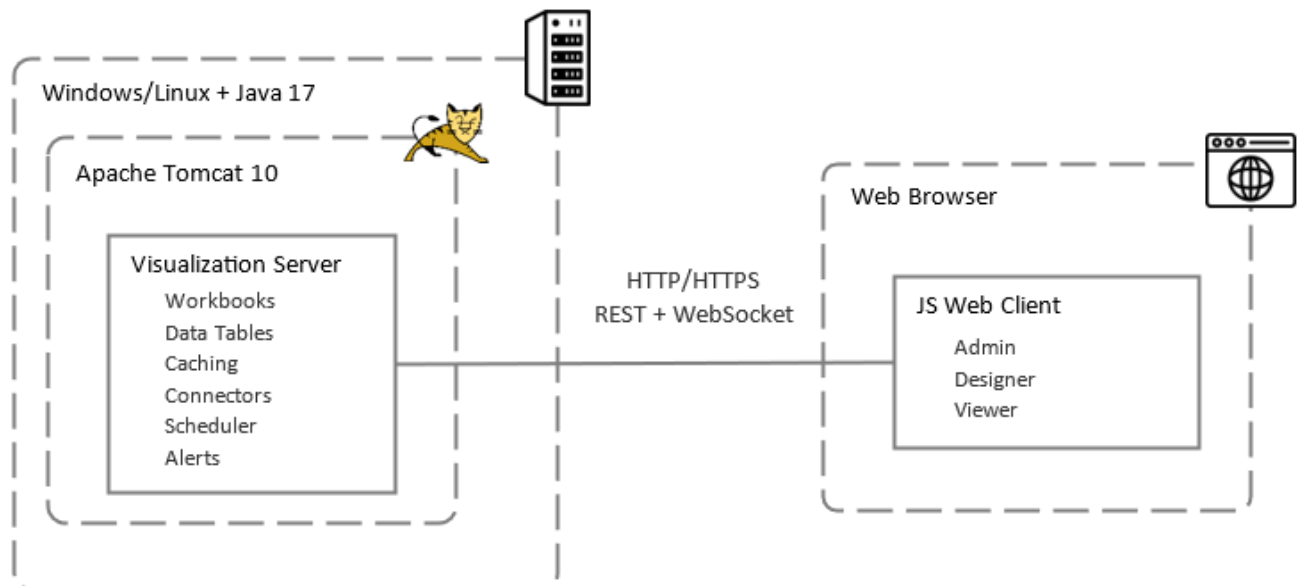
Panopticon Visualization Server Architecture
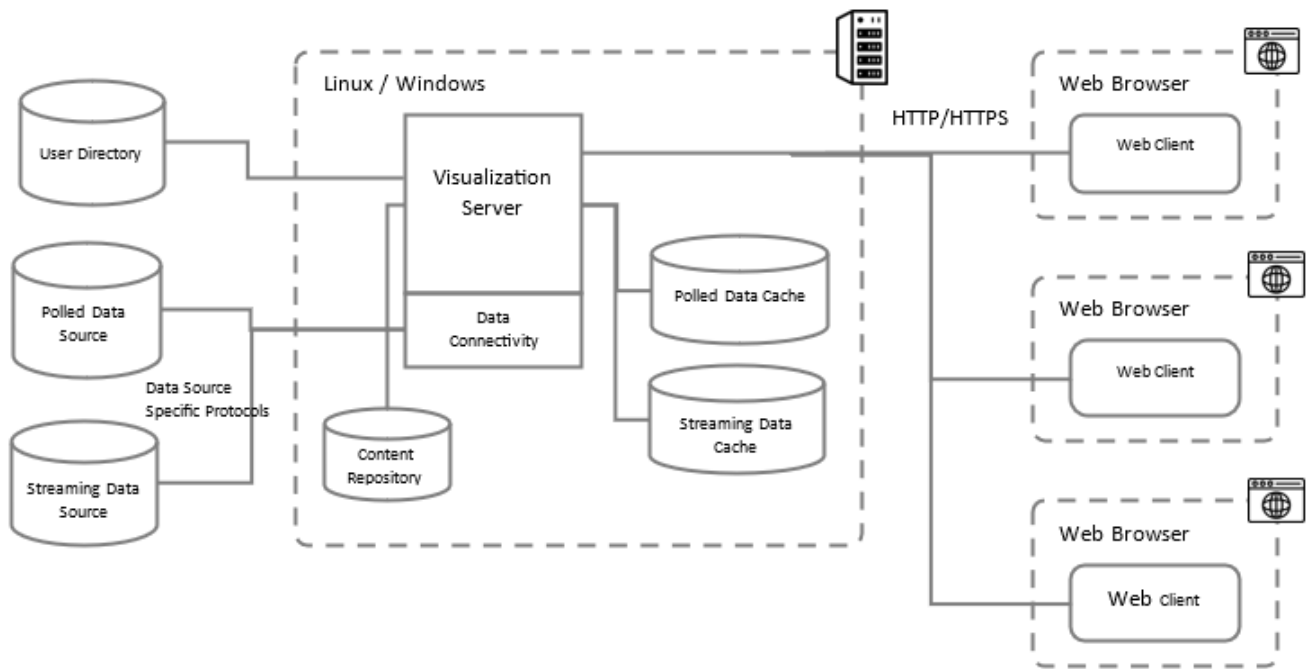
Single Node Deployment (Non-containerized)

Panopticon consists of [server](#) and [client](#) parts. The server part is written in Java and runs as a Java Application on Apache Tomcat. The client part is written in HTML, CSS, JavaScript, and Angular.

Operating system requirements are listed [here](#) and in the [Technical Fact Sheet](#) documentation.

Hardware requirements, particularly for server-side hardware, are highly dependent on the deployment scale and the data complexity and data volumes. See [System Hardware Requirements](#) for more information.

Visualization Server

The server requires Java version 17 and Apache Tomcat 10.

User-created content such as workbooks with dashboards, data tables, and style themes are stored in repository directory in the form of small binary files which maintain a change log, like Git.

The server maintains an in-memory cache of data sets recently loaded from data sources, which avoids unnecessary repeated data loading from the data sources. The in-memory cache also assures scalability in the number of concurrent users since the same data set is cached in only one copy and a single data load from a data source will deliver data to a large number of users.

Panopticon supports the use of a third-party database as a Data Store, where data from any data connectors can be persistently cached, for example for the purpose of avoiding slow data loading due to badly performing data source. MonetDB is the recommended database for this purpose, but other JDBC compliant databases can also be used.

Panopticon has full support for containerized deployment with Docker, orchestration with Kubernetes and support for all major cloud platforms (AWS, Google, Azure, Oracle).

Panopticon has support for high-availability deployment of multiple Panopticon servers running in a cluster with a leader-followers principle, where automatic synchronization between cluster member servers assures the availability of identical content on each server. A sticky load balancer will be responsible for distributing users to the different cluster member servers.

Authentication and Authorization

Panopticon is designed for single sign-on (SSO) integration with existing identity providers in your organization. User credentials and groups of users are not managed inside Panopticon, instead, Panopticon makes use of the existing organization users and groups. Panopticon has out of the box support for LDAP, SAML, OAuth2, and Basic authentication.

Panopticon has three functional roles defined in the product: **Administrator**, **Designer**, and **Viewer**. Users and Groups defined in the Identity Provider (IdP) of the organization are mapped to these three functional roles as part of configuring the deployment. Any additional user groups/roles delivered by the IdP can also be used to define folder access permissions in the Panopticon content repository.

Administrators of Panopticon server will manage settings and configurations partly via a server administration web GUI, and partly in server-side configuration files. In addition, some operations can be effectuated through a command line tool (PCLI) and some operations can be effectuated by calling a REST API.

Web Client

The browser-based client part of Panopticon is built on the AngularJS web development framework. Officially supported browsers are Safari and Google Chrome, while up-to-date versions of Firefox and MS Edge can also be expected to be compatible with Panopticon.

The client software can be customized in terms of style and look and feel on two levels: The client can be modified by making server-side changes to achieve custom branding or white-labelling. The client look-and-feel can be changed by creating and applying a custom Style Theme, which includes settings for background and foreground colors, margins, padding, border widths, font settings, color palettes, and more. Custom fonts can also be added to the server.

CONTAINERIZED DEPLOYMENT
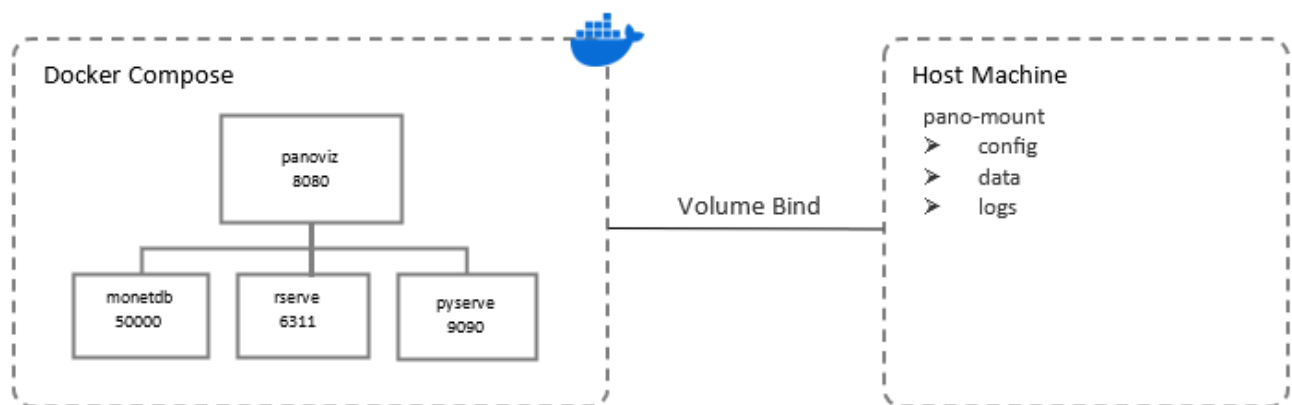
Containerized Deployment of Docker Compose

Panopticon can be obtained as a Docker Compose package consisting of four containers where the images will be pulled from Altair's image repository. See the detailed documentation to set up the Docker Compose package.

Panopticon with Docker Compose is meant for quick and easy product evaluation and testing, and not for production grade implementation.

ENTERPRISE DEPLOYMENT
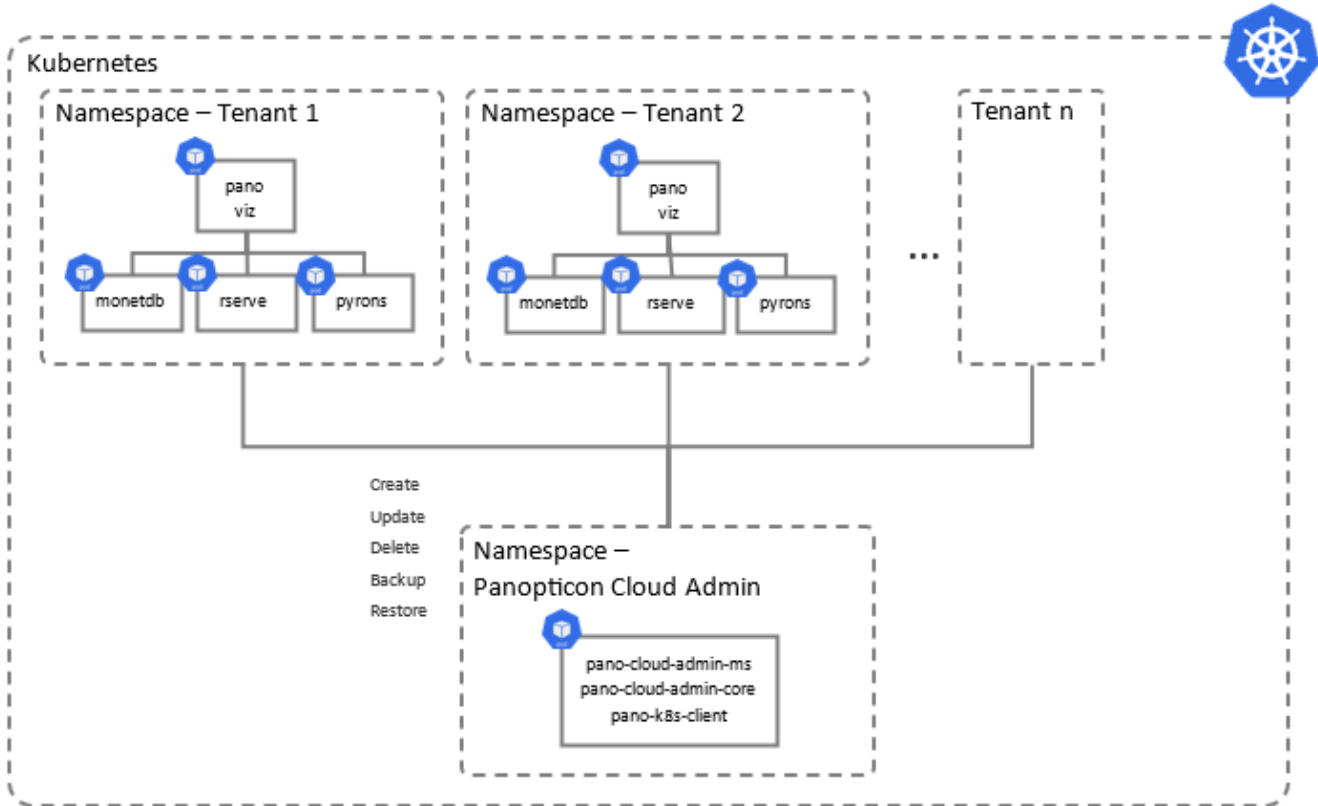
Enterprise Deployment of Kubernetes

The Kubernetes deployment of Panopticon is similar in structure to the Docker Compose deployment, with an added Namespace abstraction layer.

Each namespace represents a completely isolated Panopticon tenant deployment and can be configured to meet specific requirements.

The configuration of the pods running in each namespace is controlled by a Helm Charts package.

Cloud Admin

The Cloud Admin app allows self-service creation, updating, deletion, backup, and restoration of Panopticon tenant instances. It is deployed as a namespace alongside the other Panopticon tenants. However, while Panopticon tenants can be configured for any type of authentication (LDAP, SAML, OAuth, Basic), the Cloud Admin app has fixed "superuser" authentication to keep it separate from application users.

Cloud Admin can be configured to operate in two modes:

Mode	Description
Cloud Admin	• Manage multiple tenants. • Creation of new tenants is allowed.
Tenant Admin	• Manage one single tenant. • Creation of new tenants is not allowed.

Kubernetes Architecture for a Single Tenant / Namespace
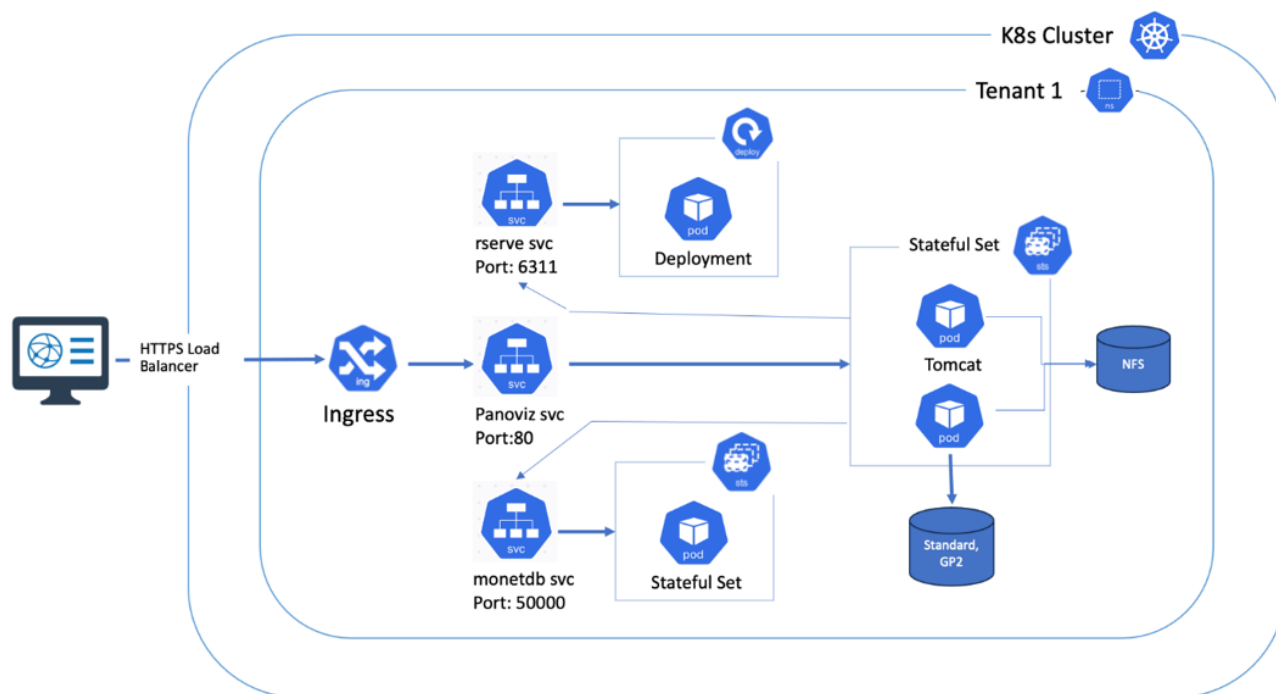
Kubernetes Architecture in Detail for a Single Tenant /Namespace

The diagram above illustrates the various elements making up a single tenant deployment of Panopticon. It is designed to be scalable, secure, and performant using the industry-standard horizontal and vertical scaling approaches.

❑ Ingress performs the functions of Load balancing and SSL termination and allows external traffic to reach the Panopticon server

❑ The **panoviz** service directs all incoming traffic to the pods running Tomcat and Panopticon application deployed to them

❑ The Rserve, Python, and MonetDB workloads are not exposed to the external world and are only accessible to the Panopticon application

❑ All of the stateful sets support local storage like GP2, Standard, Azure-disk, etc. to store local data and nfs storage for shared content amongst the replicas

❑ All of the configurations that can be modified are modeled using Config maps and can be edited or viewed at any time for better maintenance

SUPPORTED PORTS, PROTOCOLS, AND SERVICES

Services	Description	Protocol	Port
Panoviz	Panopticon Visualization Server instance.	HTTP, HTTPS, WebSocket	8080, 80, 8443, 443
Panoviz	Email server communication for alert notifications, scheduled email reports	SMTP /SMTPs	25, 465, 587
Panoviz (optional)	SSO integration	LDAP /LDAPs	389, 636
Panoviz	Data connections (e.g., databases, message buses)	Various	Various ports depending on data source
MonetDB (optional)	MonetDB database instance backing the Data Store feature	JDBC	50000 (configurable)
Python FastAPI (optional)	Python script execution backend.	REST	9090 (configurable)
Rserve (optional)	R script execution backend	Rserve	6311 (configurable)

Panopticon Online Documentation

Here are the links to the product's documentation:

❑ [Panopticon Online Documentation](#)

❑ [Technical Fact Sheet](#)

❑ [Web Authoring Quick Start Guide](#)

❑ Server Installation and Reference Guide ([PDF](#), [HTML](#))

❑ Docker Compose Package Installation Guide ([PDF](#), [HTML](#))

❑ Web Authoring Guide ([PDF](#), [HTML](#))

❑ Client User Guide ([PDF](#), [HTML](#))

❑ [White Labeling Guide](#)

Supported Connectors

Panopticon supports the following data connectors:

❑ **General Connectivity**: MS Excel, Text, XML, JSON, Restful Web services, JDBC Databases

❑ **Big Data**: Cassandra, 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 1.x

- ❑ 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 includes the following distributions and versions:

- ❑ RHEL/CentOS 7 or higher

- ❑ Debian 8 or higher

- ❑ Ubuntu 14 or higher

- ❑ Fedora 21 or higher

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

- ❑ Windows 10 or higher (64-bit)

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

Panopticon Real Time also requires:

- ❑ Oracle Java SE 17 and Open JDK 17

NOTE	- See Java documentation about setting up the **JAVA_HOME** environment variable in your system. - See Required Java Option Settings with Java 17 for the required additional configuration to load Apache Arrow files and to be able to use Arrow serialization with Python.

❏ Apache Tomcat 10

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/3ca5cbdc2f970470341926354f210dff032fc5f3

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:

o systemctl daemon-reload

o 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.

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] SETTING UP PANOPTICON REAL TIME

PANOPTICON REAL TIME DEPLOYMENT MODEL
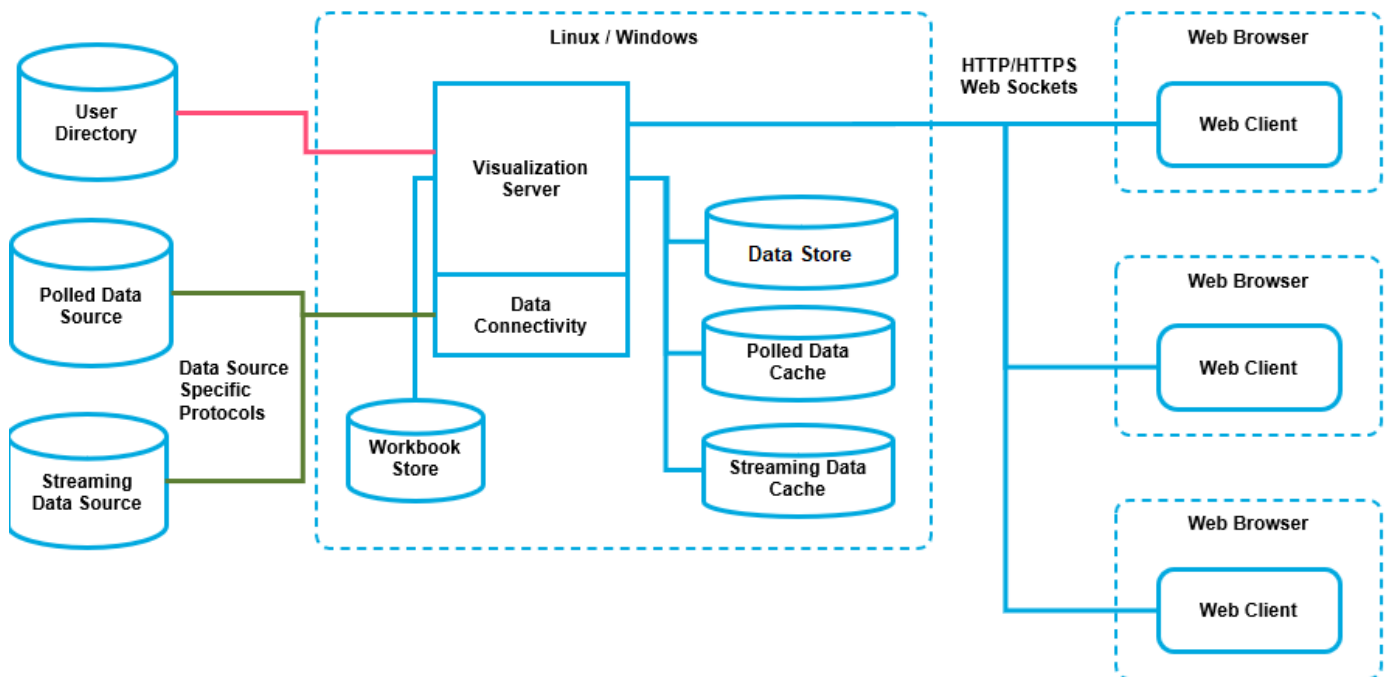
Full scale Panopticon Real Time deployment

Panopticon Real Time is deployed and hosted on an internal network. The server can be accessed 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:

❑ 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	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 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.

BEFORE YOU BEGIN

Before you begin the installation of Panopticon Real Time, you may need to:

❑ Verify that your system meets the system requirements

❑ Configure setenv.sh (for Linux) or setenv.bat (for Windows)

IMPORTANT	**Before installing Panopticon on Tomcat, you must start Tomcat at least once. Make sure Tomcat is stopped before starting the installation of Panopticon.**

Configuring setenv.sh or setenv.bat

Configuration is done by adding a setting in the `setenv.sh` (for Linux) or `setenv.bat` (for Windows) in the `<CATALINA_HOME>/bin/` folder. If the `setenv.sh` or `setenv.bat` file does not exist, you must create it.

Windows

Point out the Java installation to use.

Add the following lines:

```
set JAVA_HOME=JDK_home\[JDK folder]
set JRE_HOME= JDK_home\[JDK folder]
```

Set the UTF-8 character encoding.

Add the following line:

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

Enable Arrow Serialization.

With Java 17, this setting is necessary for loading Apache Arrow files and for using Arrow serialization with Python.
Add the following line:

```
set JAVA_OPTS=%JAVA_OPTS% --add-opens java.base/java.nio=ALL-UNNAMED
```

Example

```
set JAVA_HOME=C:\Program Files\RedHat\java-17-openjdk-17.0.10.0.7-2
set JRE_HOME=C:\Program Files\RedHat\java-17-openjdk-17.0.10.0.7-2
set JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=UTF-8
set JAVA_OPTS=%JAVA_OPTS% --add-opens java.base/java.nio=ALL-UNNAMED
```

Finally, save the file `setenv.bat`.

For further details, see the Apache Tomcat 10 [documentation](#).

Linux

Point out the Java installation to use.

Add the following lines:

```
JAVA_HOME=/path/to/jvm/version
JRE_HOME=/path/to/jvm/version
```

Set the UTF-8 character encoding.

Add the following line:

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

Enable Arrow Serialization.

With Java 17, this setting is necessary for loading Apache Arrow files and for using Arrow serialization with Python.
Add the following line:

```
JAVA_OPTS="$JAVA_OPTS --add-opens java.base/java.nio=ALL-UNNAMED"
```

Example

```
JAVA_HOME=/usr/lib/jvm/java-1.17.0-openjdk-amd64
JRE_HOME=/usr/lib/jvm/java-1.17.0-openjdk-amd64
JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=UTF-8"
JAVA_OPTS="$JAVA_OPTS --add-opens java.base/java.nio=ALL-UNNAMED"
```

Finally, save the file `setenv.sh`.

For further details, see the Apache Tomcat 10 [documentation](#).

NOTE	If you don't have the correct Java Options setting for Arrow, you will see the following error message when trying to load an Arrow file:"

```
Handler dispatch failed:
java.lang.ExceptionInInitializerError
```

The Tomcat console will log:

```
java.lang.RuntimeException: Failed to initialize
MemoryUtil. Was Java started with --add-
opens=java.base/java.nio=ALL-UNNAMED? (See
https://arrow.apache.org/docs/java/install.html)

<etc..>

Caused by:
java.lang.reflect.InaccessibleObjectException: Unable
to make field long java.nio.Buffer.address accessible:
module java.base does not "opens java.nio" to unnamed
module @2322f814
```

INSTALLING PANOPTICON REAL TIME

This section 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

   - panopticon_fastapi_py folder

   - tomcat-users_example.xml

- start_Python_connectivity.sh
- start_Python_connectivity.bat
- pyro.py
- panopticon.xml
- panopticon.war
- PanopticonJNDIRealm.jar
- PanopticonJNDIDataSourceFactory.jar
- 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

2. Make sure that your Tomcat server is stopped.

3. 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.

4. Copy the file `panopticon.xml` from the .ZIP file and place it in the Tomcat config folder (`/tomcat_home/conf/Catalina/localhost`).

5. Edit the Environment **value** property to the path where you want to keep Panopticon server content, for example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Context>
      <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.

6. Copy the `panopticon.war` file into the Tomcat `webapps` folder (`/tomcat_home/webapps`).

7. 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:

```xml
<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`

8. 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.

9. Specify the license type that will be used. Use any of the following license types:

- Volume License file (**PanopticonLicense.xml)** that must be copied to the designated `AppData` folder.

- Altair Units license. Refer to [Using Altair Units License in Altair's License Server](#) for more information.

- Managed Altair Units license. Refer to [Using Managed Altair Units License Via Altair One](#) for more information.

10. Increase the [Java heap size of Tomcat](#).

11. You can also opt to install Java data connector's dependencies.

12. 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	Python integration for data connectivity via Python and transforms using Python is achieved in either of these two ways: • With FastAPI - Refer to [Python Integration with FastAPI](#) for more information. • With Pyro4 - Refer to [Python Integration with Pyro4](#) for more information.

Setting Up Panopticon Real Time on Windows

NOTE	You should use the Windows zip distribution of Apache Tomcat for Windows rather than the Windows Service Installer.

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

   - panopticon_fastapi_py folder

   - tomcat-users_example.xml

   - start_Python_connectivity.sh

   - start_Python_connectivity.bat

   - pyro.py

   - panopticon.xml

- panopticon.war

- PanopticonJNDIRealm.jar

- PanopticonJNDIDataSourceFactory.jar

- [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

2. Make sure that your Tomcat server is stopped.

3. 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`

4. Copy the extracted `panopticon.xml` file into the Tomcat config folder. This file contains the following information:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Context>
       <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.

5. Copy the `panopticon.war` file into the Tomcat `webapps` folder (`\Apache Software Foundation\Tomcat 10.0\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 10.0\conf`).

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

```xml
<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 10.0\temp)` is available.
	• You can opt to choose a different `temp` folder with the CATALINA_TMPDIR environment variable. For example:

Variable	Value
CATALINA_TMPDIR	`C:\tomcat\dev\temp`

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. Specify the license type that will be used. Use any of the following license types:

   • Volume License file (**PanopticonLicense.xml**) that must be copied to the designated `AppData` folder.

   • Altair Units license. Refer to Using Altair Units License in Altair's License Server for more information.

   • Managed Altair Units license. Refer to Using Managed Altair Units License Via Altair One for more information.

10. Increase the Java heap size of Tomcat.

11. 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`
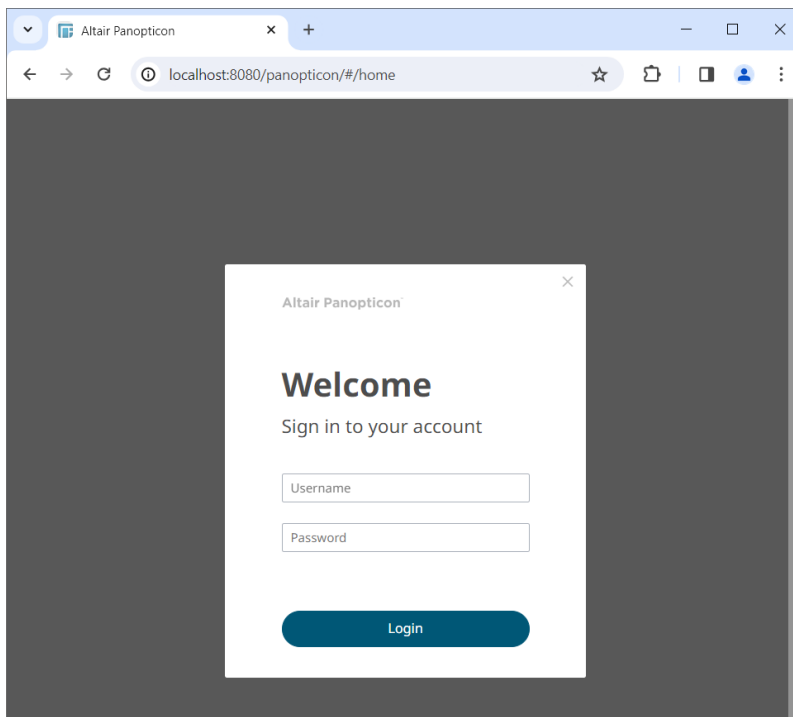
The more advanced configuration options are also discussed in this document.

NOTE	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 with Pyro4](#) for more information.

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

For example, logging on using the ADMINISTRATOR role added in step 6 (i.e., admin/admin), [will display](#):
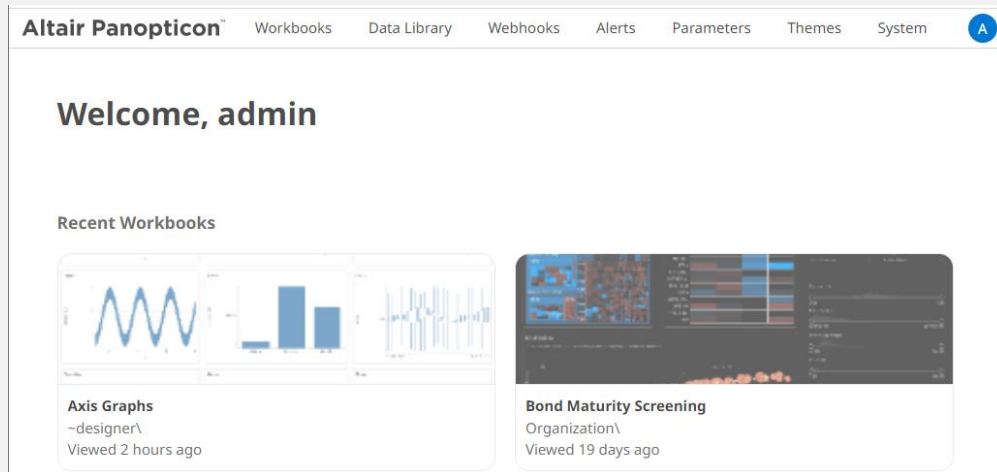
All of the available user specific folders in the [authentication](#) method used are displayed.

However, logging on with a DESIGNER role (i.e., designer/designer) will only display:
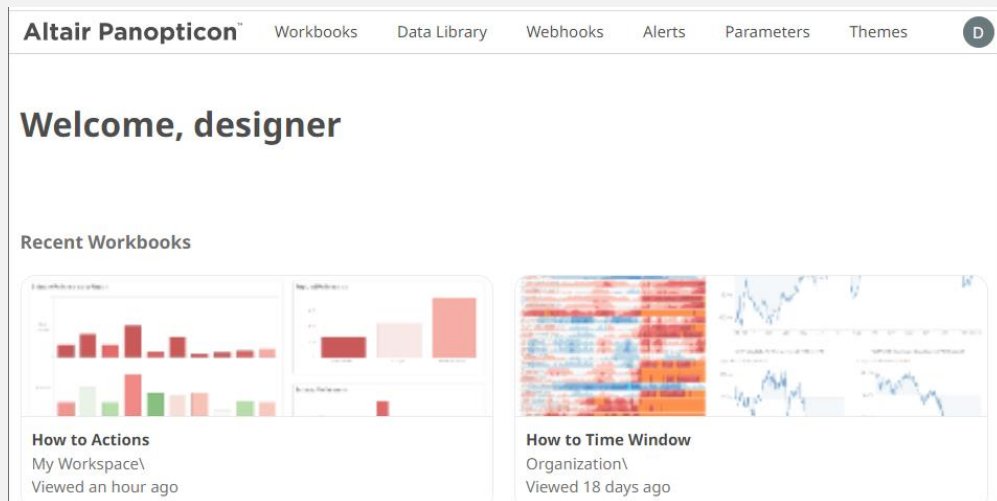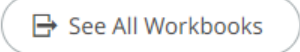
The allowed features available for the DESIGNER role is extensively discussed in the [Web Authoring Guide](#).

The Welcome Page

The *Welcome* page is the first screen that displays when you log on to Panopticon Real Time. This page can also be accessed by clicking the **Altair Panopticon** logo on the header.

From this page you can:

❑ Click [See All Workbooks] to go to the *Workbooks* page

❑ Open recently viewed workbooks

❑ [Explore workbooks available in your organization](#)

❑ [Open the Analyst User Guide](#)

❑ [View online documentation and help](#)

If there are no recently viewed workbooks, [See All Workbooks] is displayed parallel to the *Getting Started* header.
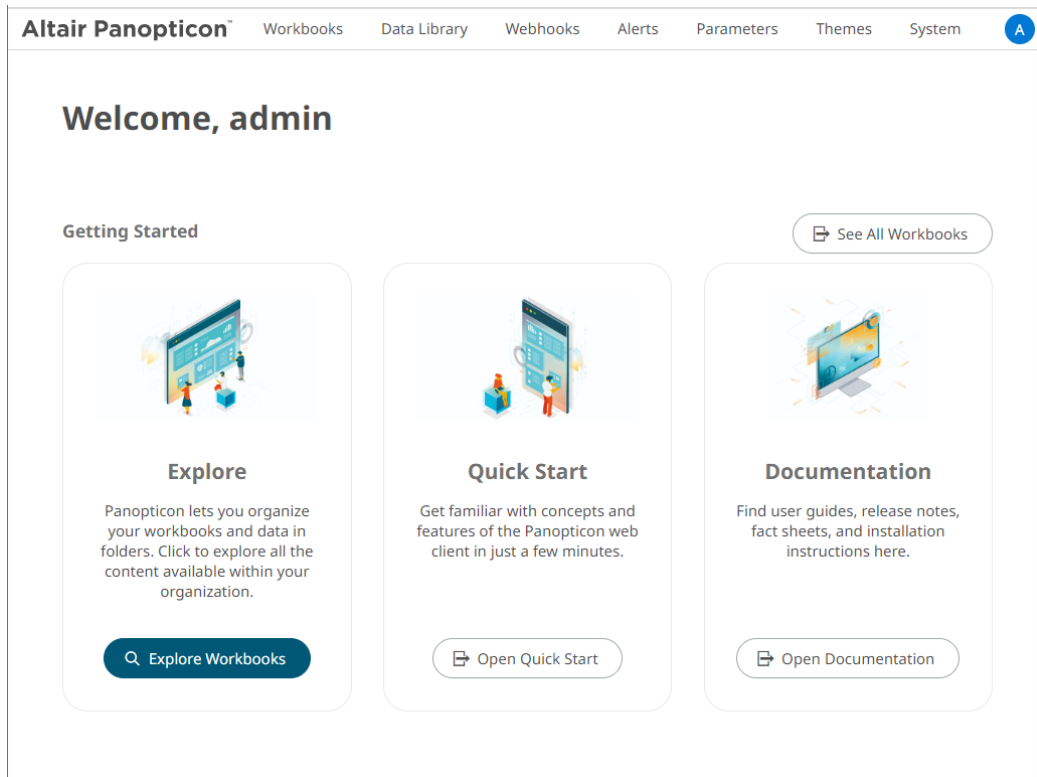
Importing the Bundle of Example Workbooks

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.
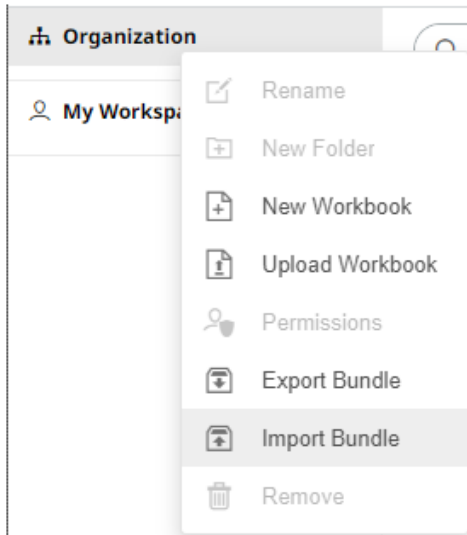
The *Import Bundle* dialog displays.

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.
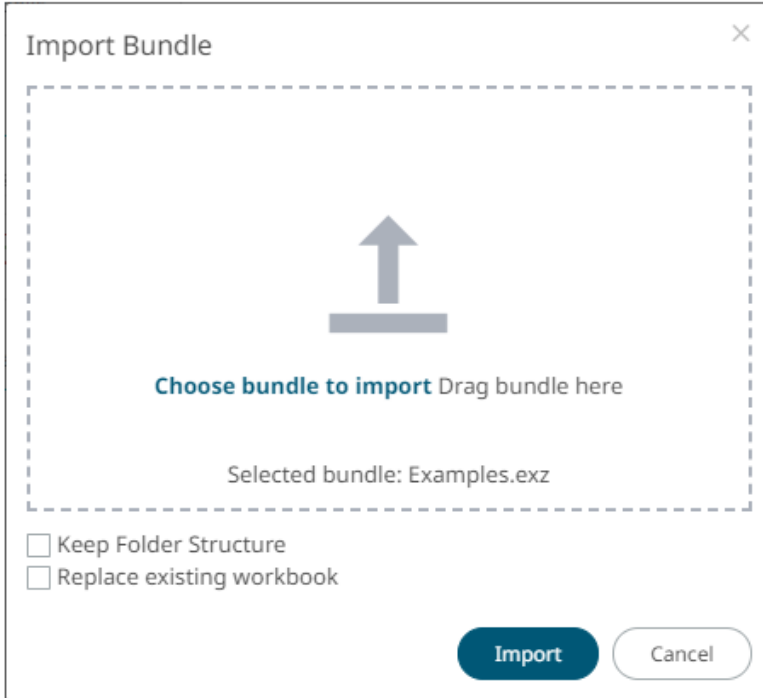
3. Select the **Keep Folder Structure** checkbox.

    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.
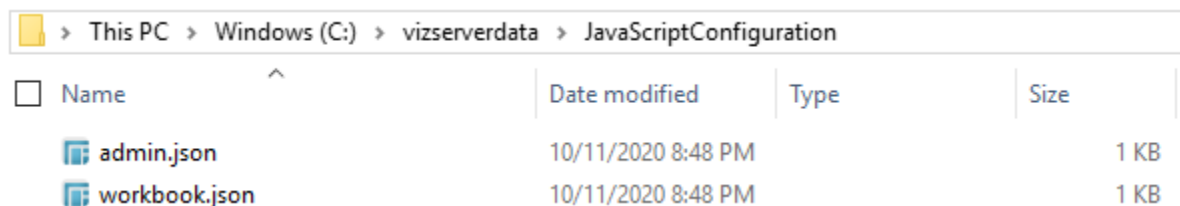
4. To replace an existing workbook, select the **Replace existing workbook** checkbox.

5. Click .

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

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**).

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
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	`showFileUploadUI`
Description	Controls the visibility of UI that is used to upload a file to the server.

Default Value	**true**
Required	No
Object/Name	`showLinkToFileUI`
Description	Controls the visibility of UI that is used to specify a data file on the server local file system as a data source. Per default, this flag is automatically inserted into the client configuration at runtime.
Default Value	**true**
Required	No
Object/Name	`visibleViewerTabs.datatables`
Description	Controls only visibility of **Data Library** tab for viewer role.
Default Value	**true**
Required	No
Object/Name	`visibleViewerTabs.webhooks`
Description	Controls only visibility of **Webhooks** tab for viewer role.
Default Value	**true**
Required	No
Object/Name	`visibleViewerTabs.alerts`
Description	Controls only visibility of **Alerts** tab for viewer role.
Default Value	**false**
Required	No
Object/Name	`visibleViewerTabs.parameters`
Description	Controls only visibility of **Parameters** tab for viewer role.
Default Value	**false**
Required	No
Object/Name	`visibleViewerTabs.themes`
Description	Controls only visibility of **Themes** tab for viewer role.
Default Value	**False**
Required	No
Object/Name	`visibleViewerTabs.system`
Description	Controls only visibility of **System** tab for viewer role.
Default Value	**false**
Required	No
Object/Name	`visibleDesignerTabs.datatables`
Description	Controls only visibility of **Data Library** tab for designer role.

Default Value	**true**
Required	No
Object/Name	`visibleDesignerTabs.webhooks`
Description	Controls only visibility of **Webhooks** tab for designer role.
Default Value	**true**
Required	No
Object/Name	`visibleDesignerTabs.alerts`
Description	Controls only visibility of **Alerts** tab for designer role.
Default Value	**true**
Required	No
Object/Name	`visibleDesignerTabs.parameters`
Description	Controls only visibility of **Parameters** tab for designer role.
Default Value	**true**
Required	No
Object/Name	`visibleDesignerTabs.themes`
Description	Controls only visibility of **Themes** tab for designer role.
Default Value	**true**
Required	No
Object/Name	`visibleDesignerTabs.system`
Description	Controls only visibility of **System** tab for designer role.
Default Value	**false**
Required	No

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 don'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 **Home** and **Back** buttons, and by the dialog for unexpected errors.
Default Value	
Required	No
Object/Name	`useBrowserHistoryToNavigateBack`
Description	If set to **true**, the **Back** button will execute `window.history.back()` when pressed. This setting takes precedence over having a configured `startUrl`.
Default Value	**false**
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 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	showBackButton
Description	Controls the visibility of the **Back** 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	`showHomeButton`
Description	Controls the visibility of the **Home** 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
Object/Name	`showFileUploadUI`
Description	Controls the visibility of UI that is used to upload a file to the server.
Default Value	**true**
Required	No
Object/Name	`showLinkToFileUI`
Description	Controls the visibility of UI that is used to specify a data file on the server local file system as a data source. Per default, this flag is automatically inserted into the client configuration at runtime.
Default Value	**true**

Required	No
Object/Name	`partCreator.showLegacyButton`
Description	Controls the visibility of the **Show Legacy** switch button in the Select part panel. Used to show/hide legacy visualization parts.
Default Value	**true**
Required	No
Object/Name	`partCreator.legacyVisible`
Description	Default value of **Show Legacy** switch button in select part panel.
Default Value	**false**
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.

MAP TILE PROVIDER CUSTOMIZATION

When creating a Map Plot visualization, the map images are retrieved from a map tile provider on web URL. Panopticon provides a default map tile provider preconfigured within the software. In addition, custom map tile providers can be added to the server configuration by editing the following file:

`/vizserverdata/default-settings/tileproviders.json`

The default map tile provider that is shipped with the product can be overridden by specifying a different one and use the `providerId` **Default**.

In case the `tileproviders.json` file is deleted or moved and hence not found on server startup, then a new default `tileproviders.json` file will be created in the `default-settings` folder.

> **NOTE** You will not be able to keep the original default map tile provider with a different name. The original default will become unavailable for as long as the new map tile provider service with the name **Default** exists in the configuration.

Each provider is a name-value pair object with the following parameters:

Parameter	Description
providerId	The name label that will be visible in the selection dropdown in the settings of the Map Plot visualization. This value must be unique across all map tile providers saved in the file.

minZoom	The default value is **0**. This is the minimum zoom level that should be available in the Panopticon Map Plot when using this map tile provider. However, the minimum zoom level available from the tile service may be a higher value (e.g., **5**). Refer to the documentation of the map tile service provider.
maxZoom	The default value is **18**. This is the maximum zoom level that should be available in the Panopticon Map Plot when using this map tile provider. However, the maximum zoom level available from the tile service may be a higher value (e.g., **20**) or a lower value (e.g., **15**). Refer to the documentation of the map tile service provider.
tileSize	The default value is **256**. This is the size of each tile image coming from the map tile service. It describes a raster size, **256 by 256**. Some services also use **512 by 512**. **NOTE:** It is required to have the correct `tileSize` setting for the service you are using. Otherwise, the tiles received will not be positioned correctly in the Map Plot and you will not see a correct map. Refer to the documentation of the map tile service provider. If the tile size is not specified by the map tile provider, assume **256**.
urlFormat	This is the URL of the map tile provider, and it will be different for each map tile provider. For map tile providers that use the OSM tile coordinates encoding, the URL will end with `/{z}/{x}/{y}.png` or `/{z}/{x}/{y}.jpg`, which for Panopticon must be changed into `/{0}/{1}/{2}.png` or `/{0}/{1}/{2}.jpg`. Some tile providers will require an API key, which is appended as a URL request parameter, on this form: `?apikey=abc123`. For map tile providers that use the QUAD_KEY tile coordinates encoding, the URL will contain a `{quadkey}` reference which for Panopticon must be changed into **{0}**.
tileCoordEncoding	The default value is **OSM**. This value means that the map tile service uses the same URL principle as OpenStreetMap, where each tile URL request contains `/zoomlevel/x-coord/y-coord/`, which corresponds to `/{0}/{1}/{2}/` in the urlFormat. Value **QUAD_KEY** means that the map tile service uses the QuadKey principle, for example, like what Bing maps do. QuadKey is short for Quadtree Key, which is a one-dimensional string of digits, created from the two-dimensional X-Y coordinates. A Quadkey uniquely identifies a map tile at a specific zoomlevel.

Additional Information About the URL Format

In the URL of map tile providers using the OSM tile coordinates encoding, **{0}** represents the zoom level value **z**, **{1}** represents the x-coordinate value **x**, and **{2}** represents the y-coordinate value **y**. For any actual map tile URL, these will have been replaced by coordinate numbers (integers) by Panopticon before the request for a tile is sent to the map tile provider server.

The Map Plot in Panopticon takes care of inserting those numeric values as required, instead of the **{0}** and **{1}** and **{2}** values. When you read the service documentation from map tile service providers, you will notice that instead of referencing **{0}**, **{1}**, **{2}**, they will be referencing **{z}**, **{x}**, **{y}**.

NOTE	• You must replace **z,x,y** with **0,1,2** when using the URL in Panopticon. Also, some services will present a URL which includes a dollar sign, like **/${z}/${x}/${y}**. These dollar signs must be removed.
	• Raster Tiles are supported.
	• Vector tiles are not supported.

Notes on using Google Maps or Bing Maps

In Panopticon, the configuration of a custom map tile provider is static. The configuration is saved in a JSON file which is read by Panopticon on start up, and any change of the information will require a server restart. For this reason, it is not practical to use either one of Google Maps map tile servers or Bing Maps map tile servers, because:

❑   The Google Maps API requires that an API key is first used to retrieve a session token, with a POST request. Map tiles are subsequently loaded with GET requests from a URL where both the API key and the session token must be appended as URL request parameters. This means that the map tile URL will no longer work when the session token expires.

❑   The Bing Maps API has frequently changing map tile URLs, and to get currently valid map tile URLs, you must start by making a request to a special API endpoint, which returns currently valid map tile URLs as well as various metadata about the map tiles. This means that the map tile URL will work for some time, but eventually will become invalid.

Implications of Moving a Workbook to Another Server

If a workbook using a custom map tile provider is exported from one Panopticon server and imported into another Panopticon server which does not have a custom map tile provider by that `providerId` name, then the Map Plot will display as blank--only marks/dots will show, but no map images. This can be corrected by changing the configuration of the Map Plot to use the **Default** map tile provider, or any other custom map tile provider added to that server.

Consequences of File Editing Mistakes

If the file `tileproviders.json` is not properly configured, for example if a quote character or comma or curly bracket is missing anywhere, then only the Panopticon default map tile provider will be available.

The server log will display any of the following warnings:

❑   Unexpected end-of-input: expected close marker for Object

❑   Illegal unquoted character

❑   Cannot construct instance of `com.panopticon.server.core.web.repository.file.SettingFileRepository$TileProviders`

❑   Unexpected character ('{' (code 123)): was expecting double-quote to start field name

These are some examples of map tile providers:

https://wiki.openstreetmap.org/wiki/Raster_tile_providers

https://stadiamaps.com/

https://www.maptiler.com/

https://www.thunderforest.com/

https://www.maptilesapi.com/

Here's a useful website for comparing different tile maps side by side:

https://mc.bbbike.org/mc/

LICENSING

Licensing within Panopticon Real Time supports the following license types:

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

❑ [Altair Units license which is available in Altair's License server](#) you are connected to (local or over the network)

❑ [Managed Altair Units license via Altair One](#)

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

NOTE	In the Panopticon documentation, HyperWorks Units (HWU) and Hosted HyperWorks Units (HHWU) are now named Altair Units.
	In the Panopticon product, these license types are still named HyperWorks Units and Hosted HyperWorks Units.
	For more information on Altair Units, visit https://www.altair.com/altair-units/.

Using Altair Units License in Altair's License Server

If your license source is Altair's License server, it is required to configure the following 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 to use any of the services provided by the server. Must be set to **true**.
Default Value	**true**
Property	Licensing
Attribute	`license.hwu.uri`
Description	The path where the License Server is running e.g., **6200@191.255.255.0** where the syntax is `PORTNUMBER@HOST`. If multiple servers are specified, use the ';' semicolon separator sign for Windows and the ':' colon separator sign for Linux. **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.mode`
Description	The license mode. Possible values are FILE or HWU. Must be set to **HWU**.
Default Value	**FILE**

For example:

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

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

Role	Altair Units License Draw
Viewer	2
Designer	2 10 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 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 Managed Altair Units License Via Altair One

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 Altair Units, it is required to configure the following properties in the <u>Panopticon.properties</u> file located in the `AppData` folder or `C:\vizserverdata`:

Property	Licensing
Attribute	`license.hwu.hosted`
Description	Boolean stating if you wish to use Managed or Local Altair Units licensing. Set to **true** if you wish to use managed licensing.
Default Value	**false**
Property	Licensing
Attribute	`license.hwu.hosted.authorization.username`
Description	Username to the Altair One account.
Default Value	
Property	Licensing
Attribute	`license.hwu.hosted.authorization.password`
Description	Password to the Altair One account.
Default Value	
Property	Licensing
Attribute	`license.hwu.hosted.authorization.token`
Description	An authorization token generated through the Altair One admin portal. Used to authorize a machine to the managed Altair Units system.
Default Value	

NOTE	• To use the managed Altair Units licensing, set the following properties:
	
	`license.hwu.hosted=true`
	`license.mode=HWU`
	`authentication.required=true`
	
	• Add the Panopticon application to your Altair One account.
	
	• When using Managed Altair Units via Altair One, you do not need to configure the `license.hwu.uri` attribute. This is because, the URL or host where the Managed Altair Units servers are running is already set internally in the Altair License JDK and any value entered by the product user will be ignored.

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

Option 1

Use an authorization code generated through Altair One:

1. Log on to Altair One (https://admin.altairone.com) then navigate to **User Profile** and select **Authorized Machines.**
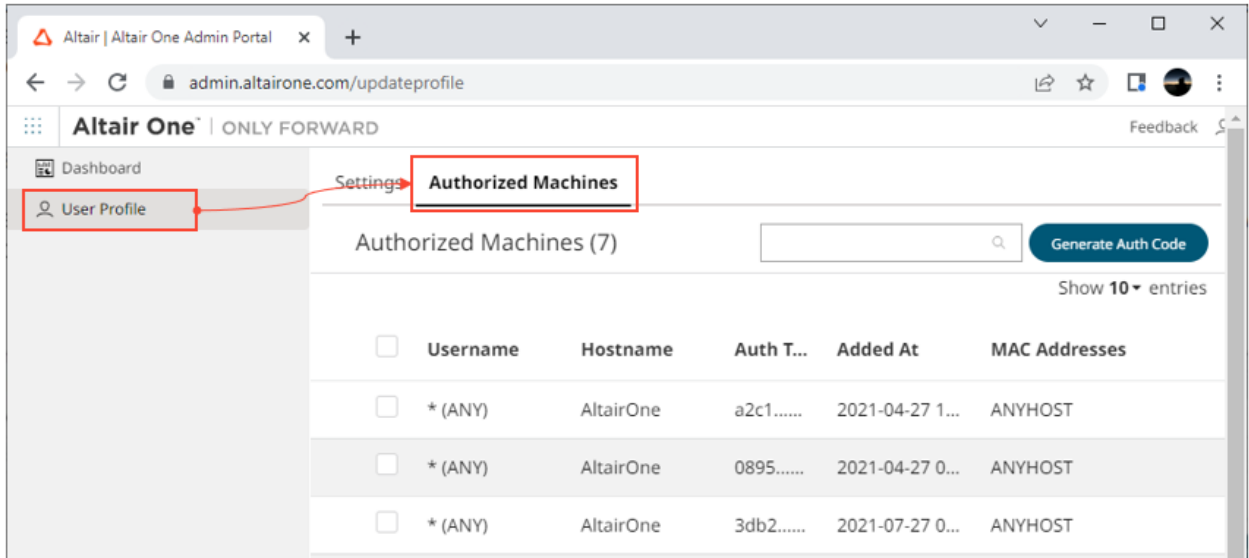
2. Click **Generate Auth Code** .

   The page displays the auth code and a timer indicating the code's expiration.
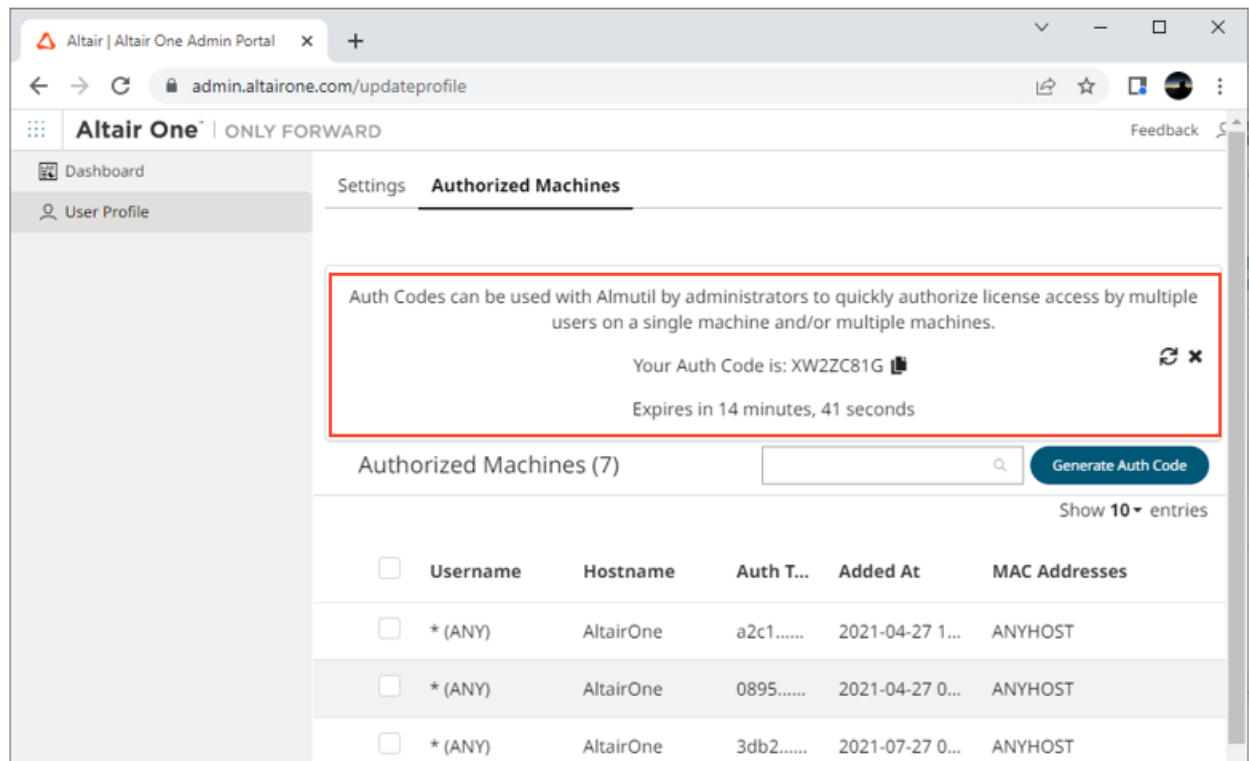
   Click **Copy to Clipboard** 📋 to copy the generated auth code.

3. Paste the generated code into the `license.hwu.hosted.authorization.token` property in the `Panopticon.properties` file.

4. Start the server.

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

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 managed Altair Units system.
	• In Option 1, the generated auth code is only valid for 15 minutes and you should restart your server within that timeframe to properly get access to your licenses.
	• In Option 2, Altair One credentials are only required at first restart of the server to generate the auth token and should be removed from the `Panopticon.properties` file going forward to avoid exposing these credentials.
	• A working Internet connection is required to use Altair Units licensing.
	• If your company uses proxy, you might need to add exception in your proxy to allow access to the Managed Altair Licenses. Please refer to this link for more information: https://community.altair.com/kb_view.do?sys_kb_id=bb9bf3fc97205590e3b0361e6253af03 Or see Managed Altair Units License SSL Error section below.
	• 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.

Managing File Handles

In some cases, the Tomcat process that runs the Panopticon server may run out of file descriptors, which are handles used by the operating system to access a file (file handles). Panopticon data connections and license units checkouts count towards the total number of file handles in use.

When the maximum limit of file handles (open files) is reached, you cannot open any additional files, or make any additional TCP connections, or check out additional Altair license units.

On a Linux system, maximum file handles limit (open files) and other limits such as memory usage are called Resource Limits or **rlimits**. The `rlimit` values can be viewed and changed using the `ulimit` command and various arguments to that command. Please refer to Linux documentation for details on using the `ulimit` command, for example https://linuxhint.com/linux_ulimit_command/.

The system will have an overall limit (global system maximum) as set in the kernel variable file `/proc/sys/fs/file-max`. In addition, there are `rlimits` set for each process, or for each user running a process. The `rlimits` for any given process will have a soft limit and a hard limit. The soft limit is what will be in effect, and the hard limit says what the maximum is if you want to raise the soft limit.

The reason for setting `rlimits` specific for a process that are lower than the overall system `rlimits`, is that you want to economize system resources between several processes to ensure that no single process consumes all

available system resources, thereby causing problems for other processes. Likewise, a process specific `rlimit` set lower than the system maximum is also useful for protecting the system, avoiding system total resource depletion.

Identifying the Problem

When a user encounters the maximum limit of file handles in Panopticon, data connection or logging into Panopticon could fail. To confirm that Panopticon has hit the file handle limit, you should look for this message in the Panopticon log:

(Err: 60) Out of file descriptors suitable for socket operation

In conjunction with the error message above, you may also see the following messages in the Panopticon log:

com.panopticon.server.core.exception.HyperWorksUnitsLicenseException: Unable to check out a license java.lang.Throwable: Altair License Manager: License error

Other things that could indicate that Panopticon has reached the maximum limit for file handles, but can also have other reasons, are:

❑ The browser console may show error messages like:

Unable to check out a license

❑ The dashboard may show the following message when you click the **Edit** button:

Unable to edit – HWU license error

❑ The dashboard may show the following when a Viewer wants to open a workbook:

AW Snap ! Unable to checkout license; Go to workbook overview

Enabling Event Poll to Avoid False Warnings

Altair Units License checkouts count towards the total number of open files. The Altair Units License SDK by default uses the **select(2)** interface in Linux to monitor the number of available file handles or sockets. The highest number that select(2) can return is **1024**. This means that as soon as 1024 or more file handles are open, the license server process will not see any available file descriptors.

However, the Altair Units License SDK can also use the **epoll(7)** interface in Linux, which is not suffering from the limitation of select(2). To make the ALSDK use epoll(7) instead of select(2), you must set an environment variable as follows:

LMX_USE_EPOLL=1

Still, epoll(7) is nonetheless bound by the `rlimit` settings, which means that in addition to setting the above environment variable, you may also need to raise the open files `rlimit` for the Tomcat process, by using the `ulimit` command.

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 set up 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
   ```

<table>
<tr><td>**NOTE**</td><td>In the `httpd.conf` file:</td></tr>
<tr><td></td><td>

- Ensure that the line `Include conf/extra/httpd-vhosts.conf` is uncommented.

- Replace Listen 80 with Listen 10088.
</td></tr>
</table>

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 is 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`

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

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 set up 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 store.
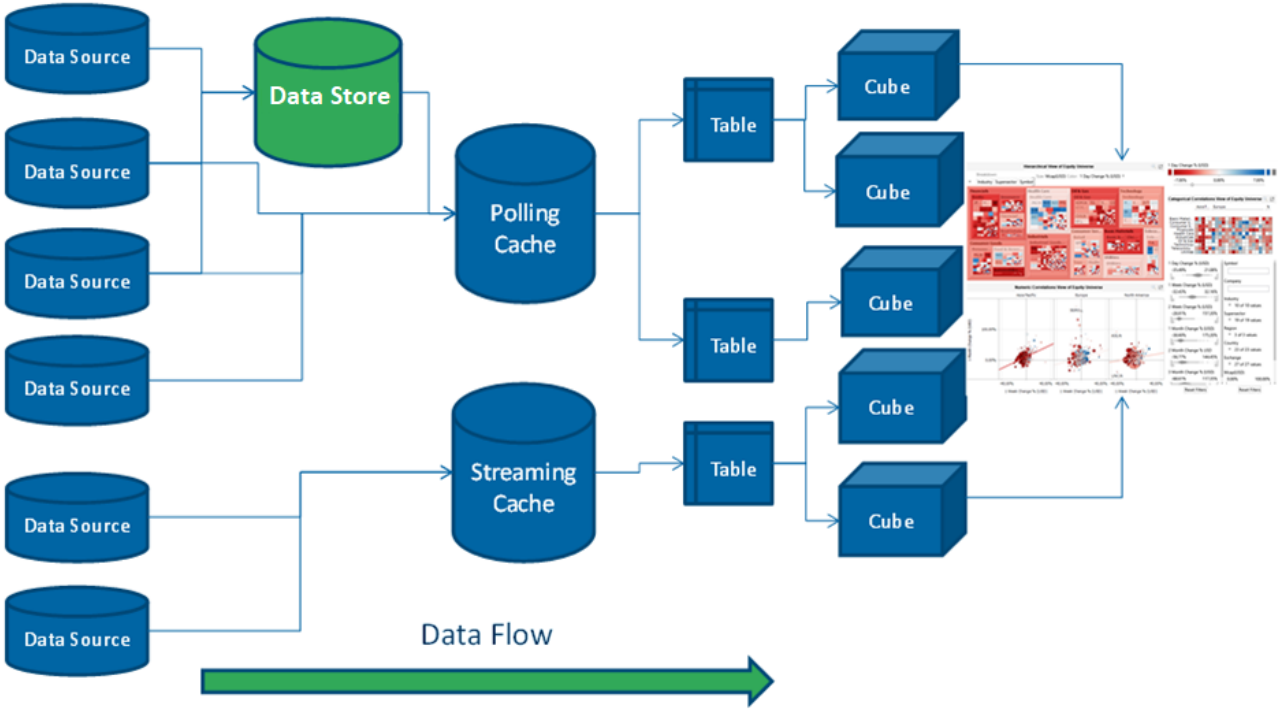
Consequently:

❑ 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)

- Import data into the Data Store

Usage is typically **Hybrid**. Based on the characteristics of the underlying data, you choose whether to import to Data Store, 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 it 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 multiple 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 by a few seconds. The leader is also special if there are conflicting changes, for example, if two users edit and save the same dashboard. In this case, the leader's changes always win.

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 no 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 changes, they adapt.

To call the Kubernetes API, the server needs to know the address of the API Server and 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 are being used as a cluster with a load balancer in front, they should be configured to have synchronized tokens. Even if the load balancer uses sticky sessions as it should, a token is typically valid for a longer time than a session 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, the 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&replace
ExistingWorkbooks=true

Example:

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

❑ or perform the following steps

1. Set the `documentation.enabled` property in <u>`Panopticon.properties`</u> 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]/swagger-ui.html`

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

   The Panopticon REST API page displays.

4. Select **legacy** in the *Select a Definition* drop-down list.

5. Expand the *Server* section and find the **Export Legacy Files** service then run it.

   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 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**.	**BASIC**
authentication.domain	The default domain information for user authentication.	

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. A blank value for `access.default.roles` is equivalent to ANONYMOUS. A blank value or the value ANONYMOUS will NOT block users from authenticating. **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.	

Role	Altair Unit Draw
Viewer	2
Designer	2 10 when designing a workbook
Administrator	2

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

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

Configuring Which Users are Allowed to Log On to Panopticon

When `access.default.roles` is set to blank (nothing), it is equivalent to setting it to the role **ANONYMOUS**. This means that, authenticated users will get the role **ANONYMOUS** when they don't have any of the roles that give them any of the following capabilities such as **VIEWER**, **DESIGNER**, or **ADMINISTRATOR** in Panopticon. The role **ANONYMOUS** in practice gives a user **VIEWER** capabilities in any folders where "Everyone" is allowed to read.

This means that setting `access.default.roles` to blank is NOT a valid way of preventing users from successfully authenticating and getting access to Panopticon. It is only related to default authorization of users, while it does not affect authentication.

In an organization where only, selected users should have access to Panopticon, as opposed to letting any existing user have access, one of the following approaches should be taken:

a. The authentication approach (preferred)

Configure the authentication layer integration to specify which group memberships are required to be allowed to authenticate, so that only users to whom you intend to give access to Panopticon are allowed to authenticate. For example, when using LDAP, add a userPattern specification of an OU (Organizational Unit) that is allowed to authenticate in the Realm configuration.

b. The content access control approach (use only of option **a** is unavailable)

Change the general content access control on Panopticon by changing the settings on its root folder, so that the role names associated with **VIEWER** and **DESIGNER** capabilities are listed as Allowed to Read and remove "Everyone" from the *Allowed* section. (**NOTE:** Users with roles that are associated with ADMINISTRATOR capability will always have access to all folders.) The disadvantage of using this approach is, any existing user will still be able to successfully authenticate and view any content where "Everyone" is allowed to read.

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** or **SAML**. The token is used to identify the user and represent the user's ongoing session. This is done to prevent user credentials being sent between the user and server more than necessary.

The token is returned from Panopticon 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 requirements. 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 will expire and invalidate the user session if the property is set to **false**.	**true**
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	**false**

	receives a secure cookie (HttpOnly cookie), you will not be able to transmit it unless the connection is secure.	
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 be added in the `server.xml` file in the Apache Tomcat `conf` folder:

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

NOTE	The Tomcat User Database Realm is used as the default. No configurations are

> required in the `server.xml` file to be able to use the Tomcat Database Realm.

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

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**

Encrypting Passwords in tomcat-users.xml

Tomcat supports encrypted user credentials via the Digested Passwords feature:

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

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

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

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

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

4. Edit the `UserDatabaseRealm` element into the following:

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

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

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

    Linux example:

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

    Windows example:

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

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

    Linux example:

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

    Windows example:

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

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

    **asd123:74807befd6bdc1c937dc931a3dfadf015da1df1b99b74cd8d91210788e0141a5$1$f21cb2dd667209d6
    39f6be48cf83826a657730032bdacb04465262d221bfc509**

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

7. Start Tomcat.

LDAP

Panopticon 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 these 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 further information,
> please refer to Apache Tomcat 9 documentation on
> https://tomcat.apache.org/tomcat-9.0-doc/realm-howto.html.

Using LDAPS

To use TLS/SSL encrypted communication between Panopticon Real Time and the LDAP directory, we need to make
the following changes in configuration:

❑　Change the protocol in the LDAP URL to **ldaps**

❑　Change the port in the LDAP URL to an SSL enabled port, typically **636**

❑　If the LDAP directory is configured with a self-signed certificate; the certificate needs to be imported into a
TrustStore.

See an example of a Tomcat Realm using LDAPS below:

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
    connectionURL="ldaps://ldap-server:636"
    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"
/>
```

There are two options for trusting a self-signed LDAP certificate:

❑　Import the certificate into the JVM TrustStore

The JVM TrustStore is located at `JAVA_HOME/lib/security/cacerts`. Use the command below to add
a new trusted certificate to the TrustStore:

```
keytool -import -alias ldap -keystore cacerts -trustcacerts -file ldap.crt
-noprompt -storepass changeit
```

❑　Create an application TrustStore

1.　Import the chain certificate:

```
keytool -import -alias root -keystore ldaptruststore.jks -trustcacerts
-file ca.pem -noprompt -storepass changeit
```

2.　Import the LDAP server certificate:

```
keytool -import -alias ldap -keystore ldaptruststore.jks -file ldap.crt
-noprompt -storepass changeit
```

3.　Check the TrustStore.

```
keytool -list -keystore ldaptruststore.jks -storepass changeit
```

4. Configure Tomcat to use the new TrustStore by editing `setenv.bat/.sh`:

- setenv.bat (Windows)

```
set JAVA_OPTS=%JAVA_OPTS% "-Djavax.net.ssl.trustStore=[path to
ldaptruststore.jks]" "-Djavax.net.ssl.trustStorePassword=changeit"
```

- setenv.sh (Linux)

```
export JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStore=[path to
ldaptruststore.jks]-Djavax.net.ssl.trustStorePassword=changeit"
```

With either approach, Tomcat need to be restarted for the changes to have effect.

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"
    adCompat="true"
    connectionURL="ldap://ad.test.com:3268"
    alternateURL="ldap://ad.test.com:389"
    authentication="simple"
    referrals="follow"
    connectionName=admin@test.com
    connectionPassword="admin"
    userBase="cn=Users,dc=test,dc=com"
    userSearch="(sAMAccountName={0})"
    userSubtree="true"
    roleBase="cn=Users,dc=test,dc=com"
    roleName="cn"
    roleSearch="(member={0})"
    roleSubtree="true"
    roleNested="true"

/>
```

<table>
<tr>
<td>**NOTE**</td>
<td>

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

- For the `userSearch` attribute you can use either `"(sAMAccountName={0})"` or `"(UserPrincipalName={0})"`.

  - `sAMAccountName` supports clients and servers pre-Windows 2000 and expects the input format `DomainName\userName`.

  - `UserPrincipalName` is a modern, internet-style user name and expects input on the format `userName@DomainName.com`.

Example:

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

/>
```

</td>
</tr>
</table>

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.

<table>
<tr>
<td>**USEFUL TIP**</td>
<td>

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

</td>
</tr>
</table>

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 <u>Panopticon.properties</u>.

OAUTH 2.0

This section discusses how to configure Panopticon Real Time to use OAuth 2.0 for authorization. Upon a logon request, Panopticon Real Time will redirect the user to the Login page provided by 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.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/panopt
icon/server/rest/auth/login
authentication.oauth2.login.redirect.url=
authentication.oauth2.login.response.type=code
authentication.oauth2.login.scope=
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/t
okeninfo
authentication.oauth2.login.callback.url=http://localhost:8080/panoptico
n/server/rest/auth/login
authentication.oauth2.login.redirect.url=
authentication.oauth2.login.response.type=code
authentication.oauth2.login.scope=uid isMemberOf
authentication.oauth2.login.url=http://localhost:9080/openam/oauth2/auth
orize
authentication.oauth2.logout.redirect.url=http://localhost:8080/panoptic
on
authentication.oauth2.logout.url=http://localhost:9080/openam/oauth2/tok
en/revoke
authentication.oauth2.token.method=QUERY
authentication.oauth2.token.url=http://localhost:9080/openam/oauth2/acce
ss_token
authentication.type=OAUTH2
```

OAuth 2.0 Integration with Azure Active Directory

Below is an example of configuration values in `Panopticon.properties` when doing SSO integration of Panopticon with Azure Active Directory. The example values are explained in this table:

Value	Explanation
https://myhost:8443/panopticon	The URL where Panopticon is accessed.
dfe52dce-6a65-4ebf-85c0-155c45782ad6	Panopticon client id from Azure app.
ptB8Q~lKsqdSOLEKDFKc0eUgsDazpE5aEgvFUebG3	Panopticon client secret from Azure app.
2bae5b57-0eb8-48fb-ba47-990259sk45kj	Azure tenant id.
https%3A%2F%2Fmyhost%2Flogout	URL-encoded https://myhost/logout.

`Panopticon.properties` example configuration:

```
authentication.oauth2.client.id=dfe52dce-6a65-4ebf-85c0-155c45782ad6

authentication.oauth2.client.secret=ptB8Q~lKsqdSOLEKDFKc0eUgsDazpE5aEgvFUebG3

authentication.oauth2.identity.attribute.roles.pattern=

authentication.oauth2.identity.attribute.roles=groups

authentication.oauth2.identity.attribute.username=email

authentication.oauth2.identity.method=HEADER

authentication.oauth2.identity.url=https\://graph.microsoft.com//oidc/userinf
o

authentication.oauth2.login.callback.url=https\://myhost\:8443/panopticon/ser
ver/rest/auth/login

authentication.oauth2.login.redirect.url=https\://myhost\:8443/panopticon

authentication.oauth2.login.response.type=code

authentication.oauth2.login.scope=openid profile

authentication.oauth2.login.url=https\://login.microsoftonline.com/2bae5b57-
0eb8-48fb-ba47-990259sk45kj/oauth2/v2.0/authorize

authentication.oauth2.logout.redirect.url=

authentication.oauth2.logout.url=https://login.microsoftonline.com/2bae5b57-
0eb8-48fb-ba47-
990259sk45kj/oauth2/v2.0/logout?post_logout_redirect_uri=https%3A%2F%2Fmyhost
%2Flogout

authentication.oauth2.token.method=

authentication.oauth2.token.url=https\://login.microsoftonline.com/2bae5b57-
0eb8-48fb-ba47-990259sk45kj/oauth2/v2.0/token

authentication.required=true
```

NOTE	• The value of the `authentication.oauth2.logout.url` property (i.e., `https://login.microsoftonline.com/2bae5b57-0eb8-48fb-ba47-990259sk45kj/oauth2/v2.0/logout?post_logout_redirect_uri=https%3A%2F%2Fmyhost%2Flogout`) must also be registered as the Logout URL value in the Azure app configuration.
	• The parameter **post_logout_redirect_uri** is set to a URL different from the URL of the Panopticon server, this is because, if the user is redirected back to the Panopticon server at logout, there will be an immediate re-login of the user in the Panopticon application due to the SSO integration with Azure AD.

- For the `authentication.oauth2.identity.attribute.roles` property, Microsoft recommends that user roles are handled by creating app roles that map to AD user and/or groups, instead of using AD groups directly. The maximum number of groups that the AD will return is **50**.

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 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.

   ```
   package com.panopticon.server.filter;

   import org.apache.catalina.realm.GenericPrincipal;
   import jakarta.servlet.http.HttpServletRequestWrapper;
   import jakarta.servlet.http.HttpServletRequest;

   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.

```
package com.panopticon.server.filter;

import org.apache.catalina.realm.GenericPrincipal;
import jakarta.servlet.*;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.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 doFilter(
            final ServletRequest servletRequest,
            final ServletResponse servletResponse,
            FilterChain filterChain)
            throws IOException, ServletException {

        if (!(servletRequest instanceof HttpServletRequest) ||
                !(servletResponse instanceof HttpServletResponse)) {
            return;
        }

        HttpServletRequest request = (HttpServletRequest) servletRequest;
        HttpServletResponse response = (HttpServletResponse)
servletResponse;

        final String username = "username";
```

```
        final List<String> groups = Arrays.asList("Group1", "Group2");
        final GenericPrincipal principal =
                new GenericPrincipal(username, groups);

        filterChain.doFilter(
                new FilterRequestWrapper(request, principal),
                response);
    }
}
```

3. When these classes have been created, you can compile them and package them in a jar file.

4. Copy the jar file to the `WEB-INF/lib` folder in the `panopticon` war file (or the extracted folder).

5. Enable the filter by adding the following code to the `web.xml` file in `panopticon WEB-INF` folder:

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

HEADER

It is possible to use a web-facing Panopticon 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 about 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] TOMCAT SECURITY CONFIGURATION

Any Apache Tomcat server used in production must be properly configured to meet the security requirements of your organization. This section describes security configuration options that should be covered before any security scan or penetration test is performed.

The first step in vulnerability mitigation is to always install the latest release version of Apache Tomcat 9.

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 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) must 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 throw an exception on initialize.

CONFIGURING HTTP HEADER SECURITY FILTER

HTTP Header Security Filter settings are part of the Apache Tomcat configuration and by default, the HTTP Header Security Filter is disabled.

In some cases, it is necessary to prevent the possibility of Panopticon being loaded inside an iframe of a web page that is running on a different server. If a web application can be loaded in any iframe, it can become a target of a clickjacking attack. A web application that allows loading into an iframe is said to give a frameable response. The HTTP Header Security Filter can also be used for enabling HTTP Strict Transport Security and Cross-Site Scripting protection.

To enable a HTTP Header Security Filter in Apache Tomcat, select one of the following:

❑   To create a filter that applies to all web applications running on the same Tomcat server, configure the filter in the file `$CATALINA_BASE/conf/web.xml`.

❑   To create a filter that applies to Panopticon only, configure the filter in the file `$CATALINA_BASE/webapps/panopticon/WEB-INF/web.xml`.

Apache Tomcat 10 documentation is available on `https://tomcat.apache.org/tomcat-10.0-doc/config/filter.html#HTTP_Header_Security_Filter`.

The filter class name is `org.apache.catalina.filters.HttpHeaderSecurityFilter` and in the default `$CATALINA_BASE/conf/web.xml`, you will find a commented-out filter of this kind by searching for that filter class name. It has a **filter-name** which is `httpHeaderSecurity`.

In addition to the filter configuration, mapping must also be enabled for the filter. The default `$CATALINA_BASE/conf/web.xml` contains a commented-out filter mapping for the **filter-name** `httpHeaderSecurity`.

<table>
<tr>
<td>NOTE</td>
<td>

- The filter and the filter mapping must match by the **filter-name**.

- Filter mapping comes *after* the filter definition.

- By simply de-commenting the filter named `httpHeaderSecurity` and the filter mapping for the same **filter-name**, and then restarting Tomcat, you will enable a HTTP Header Security Filter where:
    - The parameter `hstsEnabled` is applied with a default value **true.**
    - The parameter `hstsMaxAgeSeconds` is applied with a default value **0**.
    - The parameter `hstsIncludeSubDomains` is applied with a default value **false**.
    - The parameter `hstsPreload` is applied with a default value **false**.
    - The parameter `antiClickJackingEnabled` is applied with a default value **true**.
    - The parameter `antiClickJackingOption` is applied with a default value **DENY**.
    - The parameter `blockContentTypeSniffingEnabled` is applied with a default value **true**.
    - The parameter `xssProtectionEnabled` is applied with a default value **true**.
    - The filter-mapping applies to all URLs on the server.

</td>
</tr>
</table>

DISABLING SPECIFIC HTTP METHODS

For security reasons, you may want to disable or restrict one or several HTTP request methods on your Apache Tomcat server. A method which is often restricted is the `HTTP OPTIONS` request method. This is because a `HTTP OPTIONS` request can sometimes expose internal server configuration details and reveal vulnerabilities.

The `HTTP OPTIONS` method of course has a legitimate purpose, besides any abuse for malicious purposes. Browsers send an `HTTP OPTIONS` request to find out the supported HTTP methods and other options supported by the server before sending the actual request.

To restrict HTTP methods, such as **OPTIONS**, add a `<security-constraints>` element inside `<tomcat>/conf/web.xml`. Below is an example where the methods **OPTIONS** and **DELETE** are disabled. The tag `<auth-constraint/>` in the example means that no role can access the specified methods and the methods are completely disallowed.

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>restricted methods</web-resource-name>
    <url-pattern>/*</url-pattern>
    <http-method>OPTIONS</http-method>
    <http-method>DELETE</http-method>
  </web-resource-collection>
  <auth-constraint/>
</security-constraint>
```

PREVENTING CACHEABLE HTTP RESPONSE

Browsers may store a local cached copy of content received from web servers. Cached content could be retrieved by other users of the same computer later, and this could be a problem if the content contains sensitive information. A web server – or a specific web application on a server - can be configured to give directives to browsers to not store local cache copies of response content.

In Tomcat, you can control client-side caching by creating a filter of the type **ExpiresFilter**, which is a Java Servlet API port of Apache Module **mod_expires**. This filter controls the setting of the following properties in server responses:

❑ Expires HTTP header

❑ Cache-Control: max-age HTTP header

The expiration date can be set relative to either the time the source file was last modified, or to the time of the client access. The Cache-Control header turns on client-side caching and sets the max-age of a resource before it is expired. The Expires header is used to specify a specific point in time the resource is no longer valid. In practice, when both the Expires header and the Cache-Control: max-age header are set, the max-age will take precedence.

An **ExpiresFilter** in Tomcat is created in the configuration file `web.xml` to do the following:

❑ Create a filter that applies to all web applications running on the same Tomcat server, you configure the filter in the file `$CATALINA_BASE/conf/web.xml`.

❑ Create a filter that applies to Panopticon only, you configure the filter in the file `$CATALINA_BASE/webapps/panopticon/WEB-INF/web.xml`.

The Apache Tomcat 9 documentation on https://tomcat.apache.org/tomcat-9.0-doc/config/filter.html#Expires_Filter has some examples of how such a filter could be constructed. In addition to the filter definition, there must also be a filter-mapping. The Tomcat documentation has examples of this as well.

In addition to the Apache Tomcat documentation examples, here is another one that shows how a filter can be constructed, which makes any content not specified explicitly to expire immediately:

```
<filter>
  <filter-name>ExpiresFilter</filter-name>
  <filter-class>org.apache.catalina.filters.ExpiresFilter</filter-class>
  <init-param>
    <!-- specific content type expiry rules go here -->
  </init-param>
  <!-- Let everything else expire immediately -->
  <init-param>
    <param-name>ExpiresDefault</param-name>
    <param-value>access plus 0 seconds</param-value>
  </init-param>
</filter>


<filter-mapping>
  <filter-name>ExpiresFilter</filter-name>
  <url-pattern>/*</url-pattern>
  <dispatcher>REQUEST</dispatcher>
</filter-mapping>
```

NOTE	The filter-mapping comes after the filter definition.

Additional Cache-Control Directives

There are additional instructions about caching that can be given from the server to the client, through directives of the Cache-Control header. Here are some examples:

❑ **Cache-Control: public** means resources can be cached by any intermediate proxies along the way between server and end-client.

❑ **Cache-Control: private** means resources can only be cached by the end-client.

❑ **Cache-Control: no-cache** means that the resource may indeed be cached, but it is an instruction to the client that it must revalidate with the server every time before using a cached version of the resource.

❑ **Cache-Control: no-store** means that the client is now allowed to cache any resource. The resource must be requested, and a full response downloaded, from the server each time. This is a directive commonly used with sensitive data.

Legacy HTTP Header Pragma

Pragma is the HTTP/1.0 implementation and cache-control is the HTTP/1.1 implementation (since 1999) of the same concept. They both are meant to prevent the client from caching the response. Older clients may not support HTTP/1.1 which is why that header is still in use. Pragma is a legacy of HTTP/1.0 and hasn't been needed since Internet Explorer 5, or Netscape 4.7.

Creating a Custom filter for Cache-control with Tomcat

To use Cache-Control directives in the Cache-Control header with Tomcat, you must write a custom filter. Below is an example of such a filter. It also sets the Expires header to a time in the past, assuring that any cached content is immediately expired. It also sets the legacy HTTP 1.0 Pragma header.

```
import javax.servlet.*;
import javax.servlet.http.HttpServletResponse;

public class CacheControlFilter implements Filter {

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

    @Override
    public void destroy() {
    }

    public void doFilter(ServletRequest request, ServletResponse response,
                         FilterChain chain) throws java.io.IOException,
ServletException {
        HttpServletResponse resp = (HttpServletResponse) response;
        resp.setHeader("Expires", "Tue, 03 Jul 2001 06:00:00 GMT");
        resp.setDateHeader("Last-Modified", new java.util.Date().getTime());
        resp.setHeader("Cache-Control", "no-store, no-cache, must-revalidate,
max-age=0, post-check=0, pre-check=0");
        resp.setHeader("Pragma", "no-cache");

        chain.doFilter(request, response);
    }
}
```

Copy the filter code to a file named CacheControlFilter.java, compile and package using the commands below:

```
javac -cp /tomcat/lib/servlet-api.jar CacheControlFilter.java
jar cf CacheControlFilter.jar CacheControlFilter.class
```

Place the .jar file in `$CATALINA_BASE/lib/`

The filter is enabled by adding the below in <tomcat>/conf/web.xml:

```
<filter>
    <filter-name>SetCacheControl</filter-name>
    <filter-class>CacheControlFilter</filter-class>
</filter>

<filter-mapping>
    <filter-name>SetCacheControl</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

[5] ADDITIONAL OR OPTIONAL STEPS

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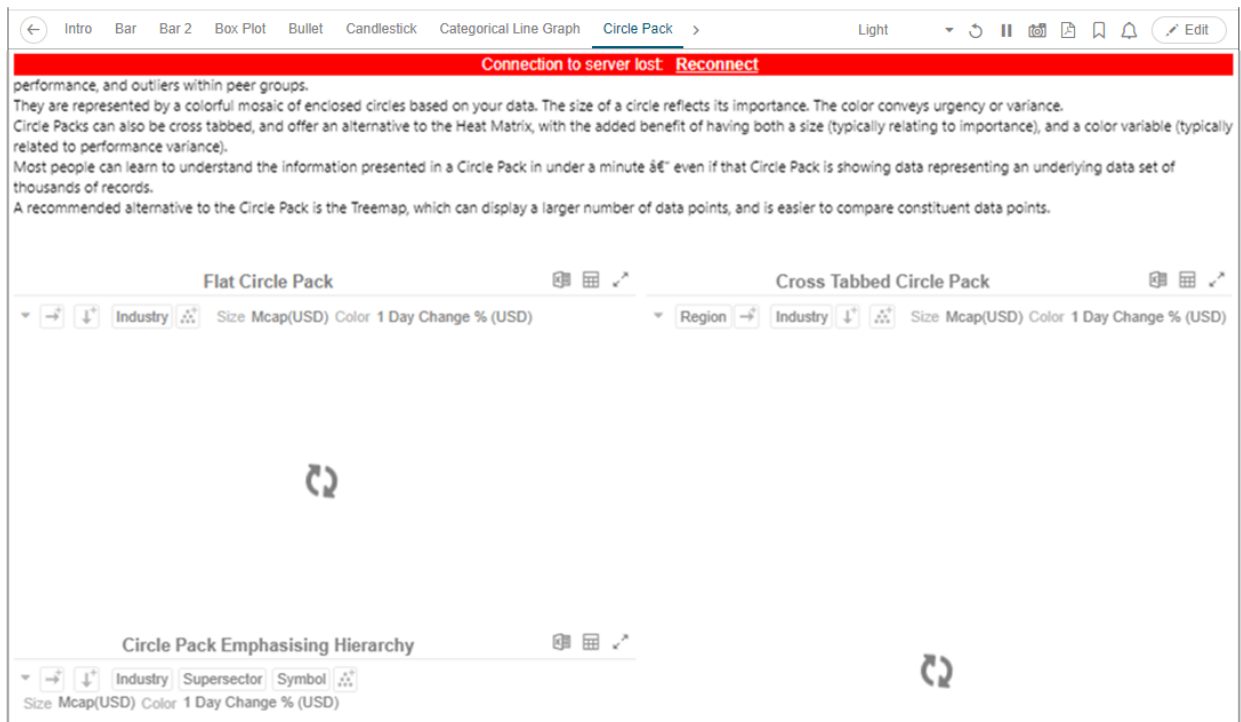
The `AppData` folder of Panopticon Real Time has a subfolder named `JavaScriptConfiguration` which contains the file named `workbook.json`.

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 is displayed below. **Connection to server lost: attempting to reconnect…**
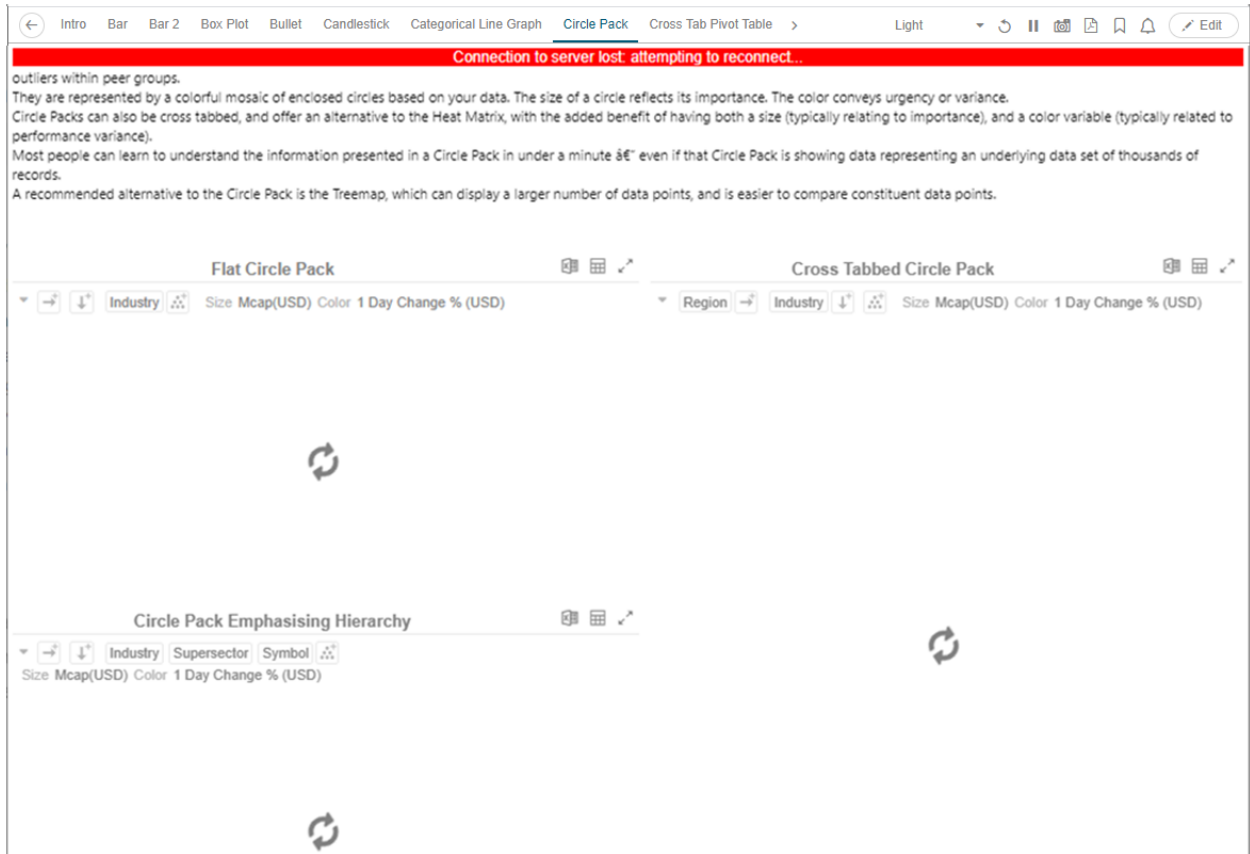
AUTOMATIC LOGOUT OF USERS ON TIMEOUT TO SAVE UNUSED LICENSES

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. **NOTE:** When using a Gmail account, you must use an app password to authenticate with the Gmail SMTP server. See Sign in with app passwords for more information.
`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`.

[6] ADVANCED SERVER DEPLOYMENTS

USAGE IN SSL ENABLED ENVIRONMENTS

Enabling SSL for Panopticon Real Time

The steps shown in this guide use the keytool and OpenSSL commands for managing keystores and certificates. The keytool command is part of the Java distribution and can be found in `JAVA_HOME\bin`. Make sure you have the `JAVA_HOME\bin` folder in your `PATH` environment variable 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

❑   Details on the OpenSSL command can be found here:

> https://www.openssl.org/docs/manmaster/man1/openssl.html

Follow the steps below to configure SSL for Panopticon Real Time using JKS/PKCS12 formats. For using the PEM formats directly inside Connector, please refer to the Apache Tomcat SSL document link at the end of this section.

Steps:

1. Change directory to the `CATALINA_HOME\conf` folder, which is where we want to generate the Tomcat keystore.

    Do one from the following options:

    Option 1. If you don't have a certificate and private key file

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 this command.

    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
    ```

    Option 2. If you already have a certificate and private key

NOTE	You may need to have Administrator rights to run this command.

```
openssl pkcs12 -export -name cast -in [certname].cer -inkey [certname].key
-out keystore.jks
```

If the above OpenSSL command gives an "Unable to load certificate" error, the certificate may be in binary format. Then use the below two commands, to generate the JKS file.

```
openssl x509 -inform DER -in [certname].cer -out [certname].crt
```

```
openssl pkcs12 -export -name cast -in [certname].crt -inkey [certname].key
-out keystore.jks
```

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

> **NOTE**   The `keystorepass` should be the password you used while creating the keystore in Step 1.

```
<!-- 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"/>
```

3. Disable HTTP ports.

   a. If you don't want to allow plain HTTP protocol, 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" />
-->
```

   b. If you want to redirect all HTTP traffic to HTTPS.

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

Then add a security constraint entry in `$CATALINA_BASE/conf/web.xml`.

```
<security-constraint>
      <web-resource-collection>
            <web-resource-name>All</web-resource-name>
            <url-pattern>/*</url-pattern>
            </web-resource-collection>
            <user-data-constraint>
                  <transport-guarantee>CONFIDENTIAL</transport-guarantee>
            </user-data-constraint>
</security-constraint>
```

4. 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 [certname].cer
```

3. Import the certificate into the trustStore:

```
keytool -import -trustcacerts -alias myalias -file [certname].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'";
```

[7] AUTHORIZATION

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

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

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

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

Then in the `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 in 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 role can create folders on the *Workbooks* and *Data Library* pages.

NOTE	Users that log on with a Designer role will have their personal folder created and displayed on the *Workbooks*, *Data Library*, *Webhooks*, and *Themes* pages (e.g., ~designer).
	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** or **Data Library** tab, right-click on the topmost folder, and select **New Folder**.
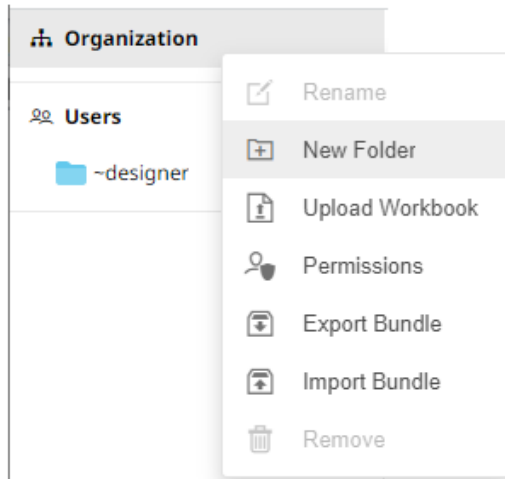
| NOTE | Only Administrators are allowed to change the permissions on the root folder. |

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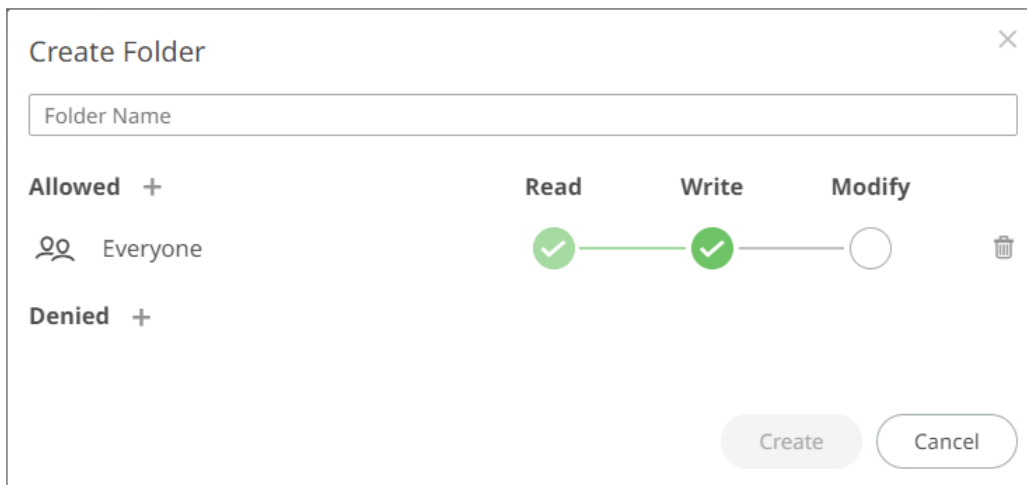
NOTE	• **Everyone** is available in the *Allowed* section by default.
	• Removing the **Everyone** group will mean that the folder and its subfolders will not be available for public access.
	• The default group permissions on the root folder are **WRITE + READ**.
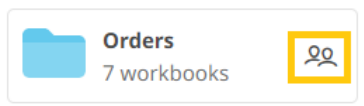
2. Enter the *Folder Name*.

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.
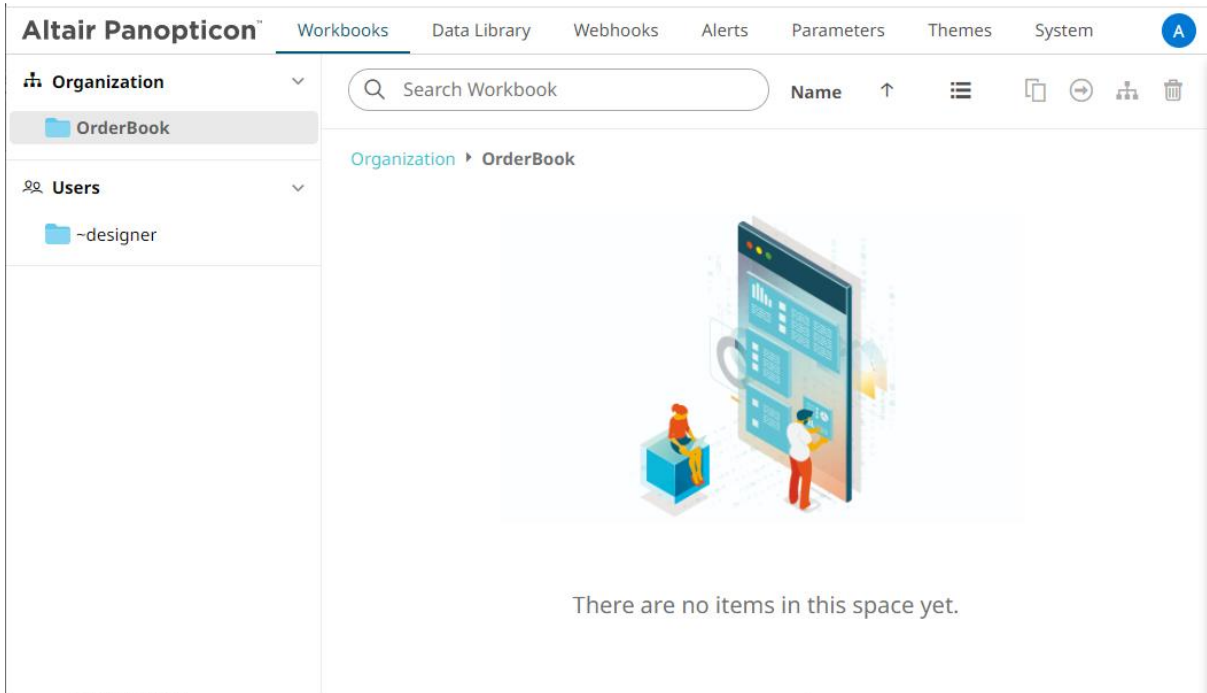
NOTE	Folders that have permission settings different from the default will display with the following icon:
	Default permission settings mean the workspace owner is the only allowed or denied user or group.

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** in the context menu.
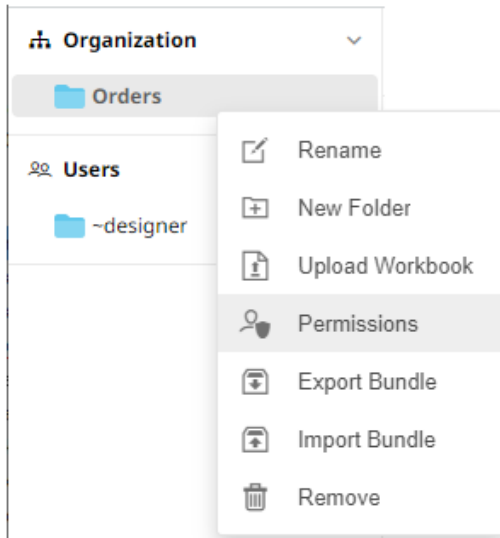
The *Permissions* dialog displays.
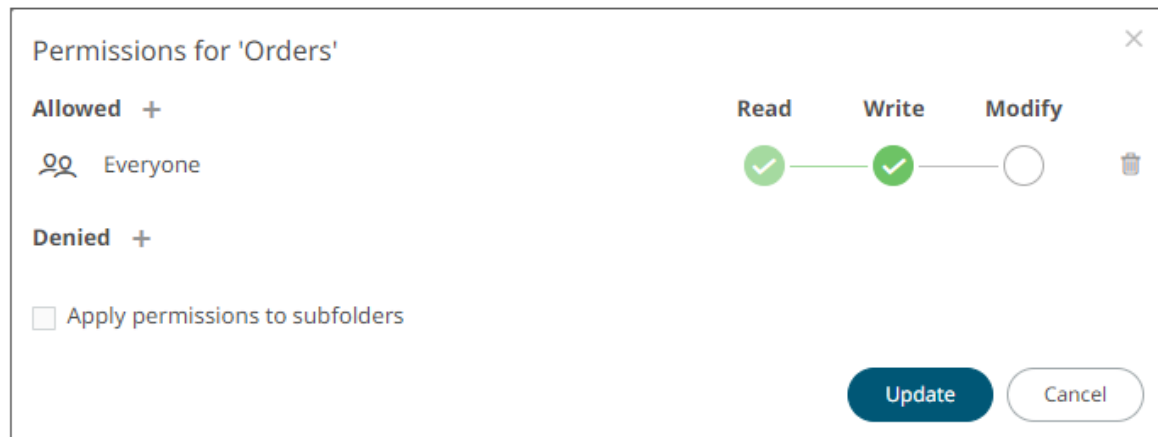
2. Under the *Allowed* section, click the **Add** ➕ icon.

    A new *User/Group Allowed* section is displayed.
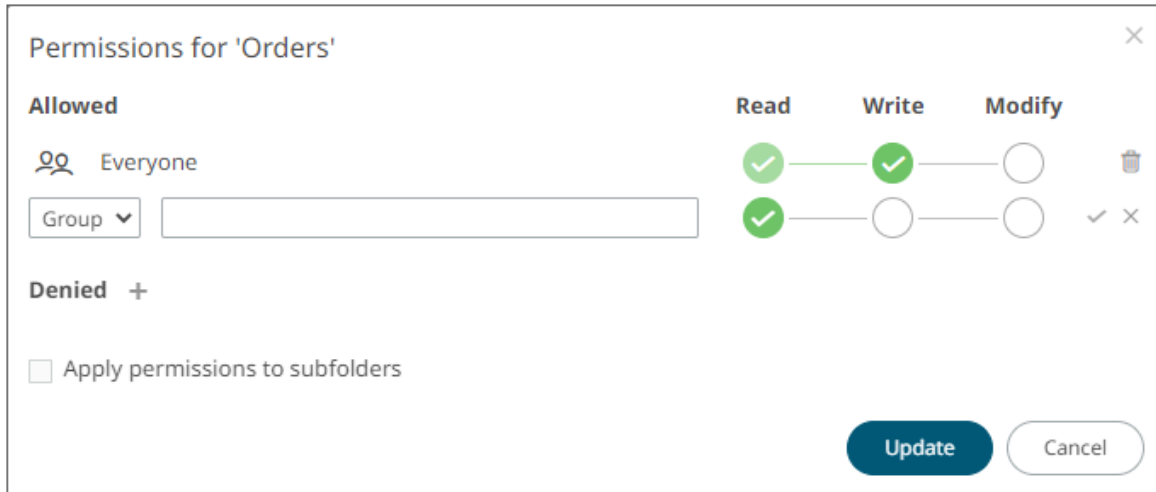
3. Select **User** or **Group** to be given permission in the drop-down list.

4. Enter the user or group *Name*.

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

   - READ

     Permission to read the folder.

   - READ + WRITE

     Permission to write to the folder and read.

   - MODIFY + WRITE + READ

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

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

7. You can either:

- Select the **Apply Permissions to Subfolders** checkbox
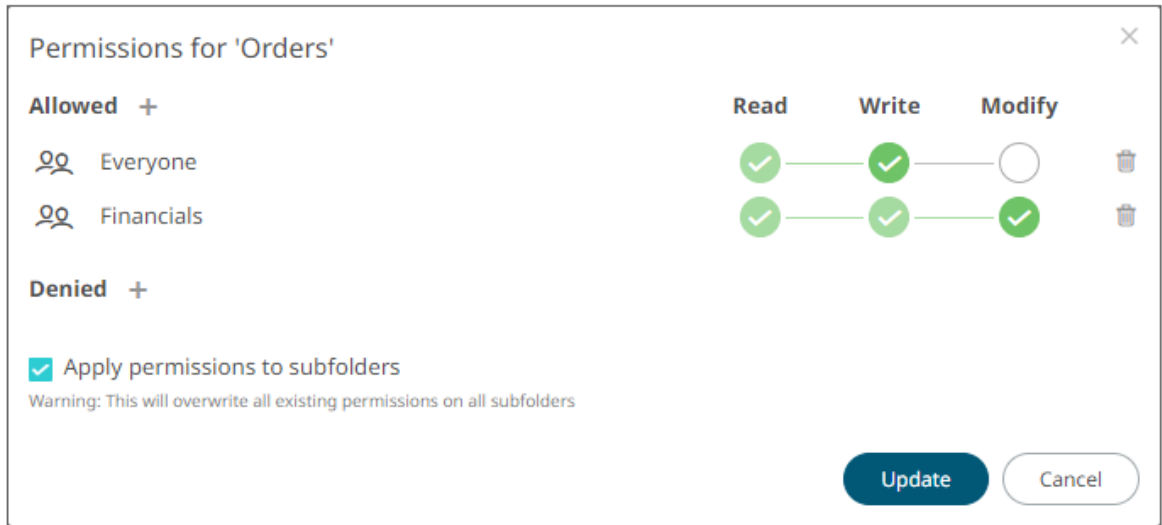
Permissions for 'Orders'   ×

Allowed +   Read   Write   Modify

   👥 Everyone   ✓ —— ✓ —— ◯   🗑

   👥 Financials   ✓ —— ✓ —— ✓   🗑

Denied +

☑ Apply permissions to subfolders
Warning: This will overwrite all existing permissions on all subfolders

                                           Update   Cancel

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** option:
	- Is only enabled when there is an <u>existing subfolder.</u>
	- 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.

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

Adding Groups and Users with Denied Access

Steps:

1. Right-click on a folder and select **Permissions** on the context menu.
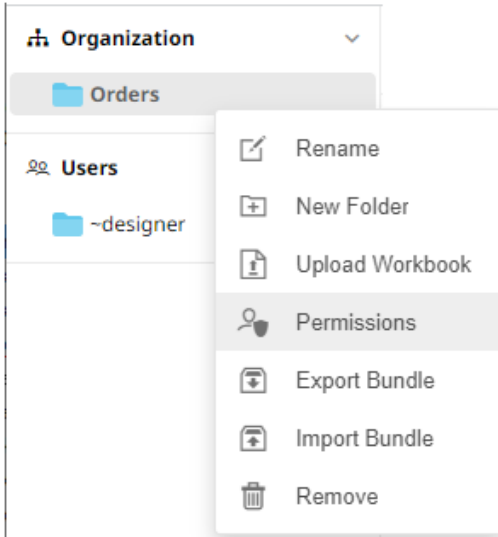
The *Permissions* dialog displays.
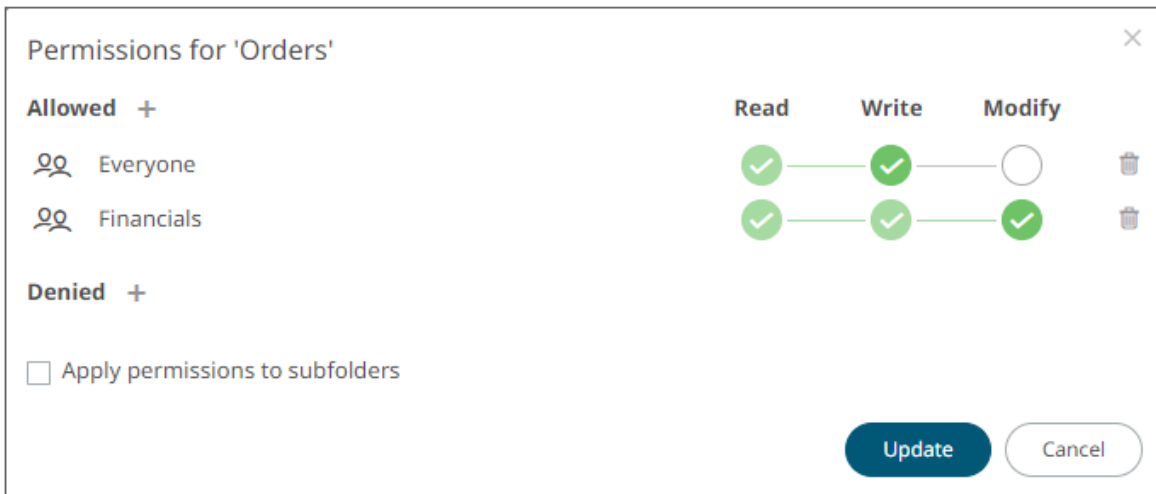
2. Under the *Denied* section, click the **Add** ✛ icon.

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 **Update** to save the changes.

Creating Subfolders

Steps:

1. To create subfolders, you can either click a folder:

   • On the expanded *Folder* hierarchy list
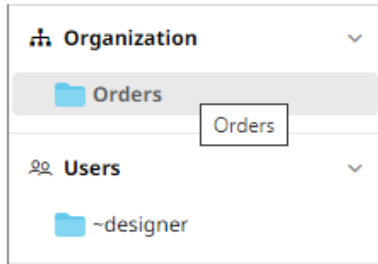
   • On the Root workbooks/folders list
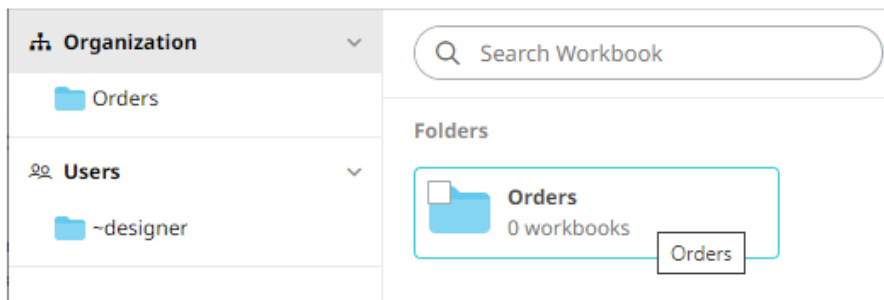
   The *Folders* page is displayed.

2. Right-click on the folder and select **New Folder**.
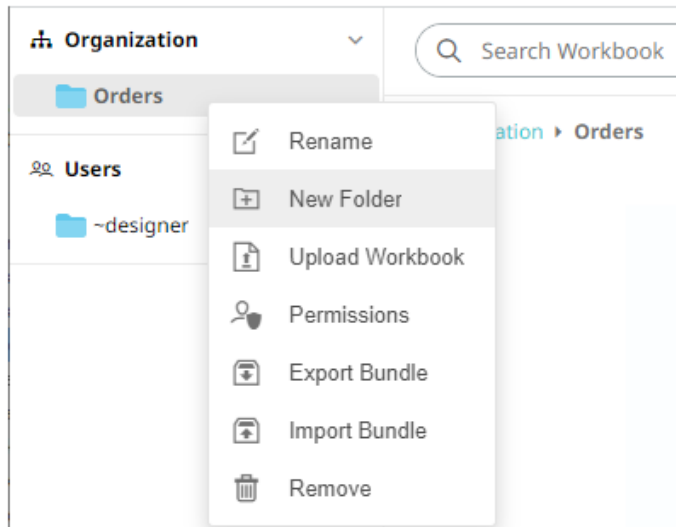
   Refer to Creating Folders for the steps in creating the subfolders. Also, Adding Groups and Users with Allowed Authorization and Adding Groups and Users with Denied Access for more information on adding Users and Groups with allowed or denied authorization.

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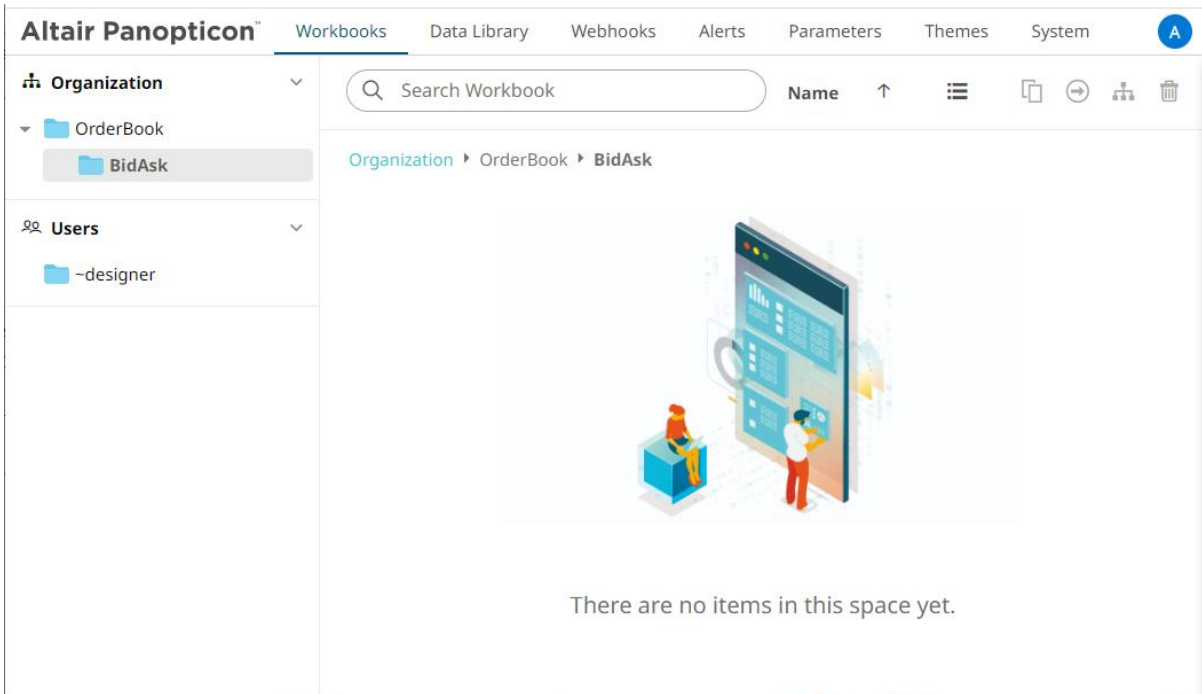
3. You can also opt to delete a subfolder by right-clicking on the folder and selecting **Remove** on the context menu as long as it does not contain published workbooks.

A confirmation message displays.

Click **Yes** .
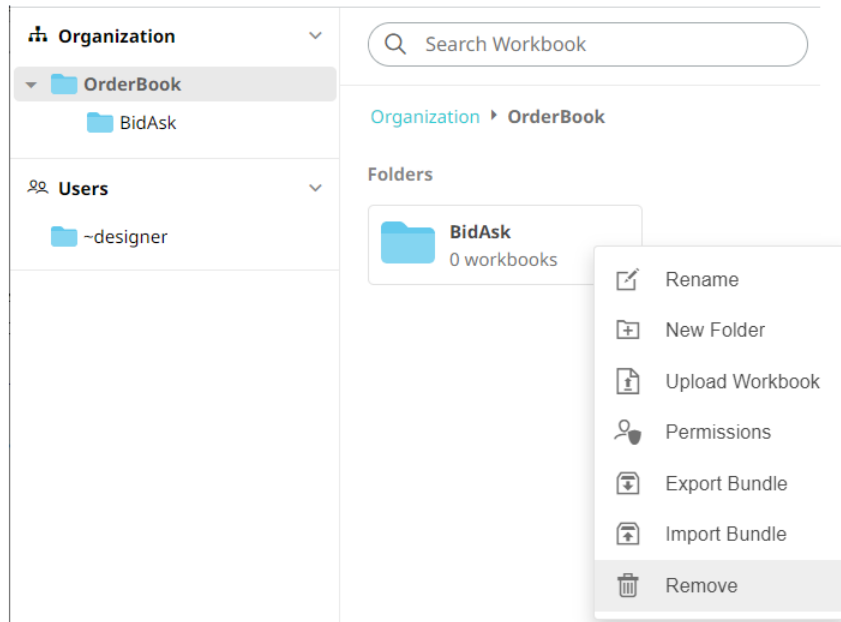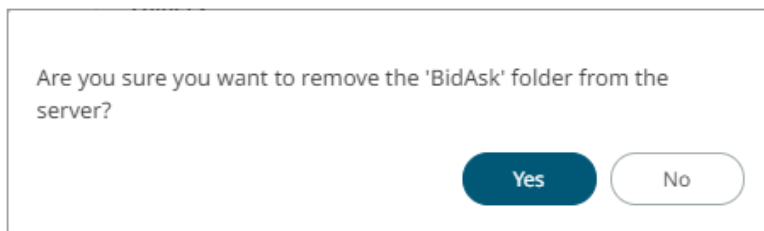
Updating Folder or Subfolder Properties

Steps:

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

2. Right-click on the folder or subfolder and select **Permissions**.
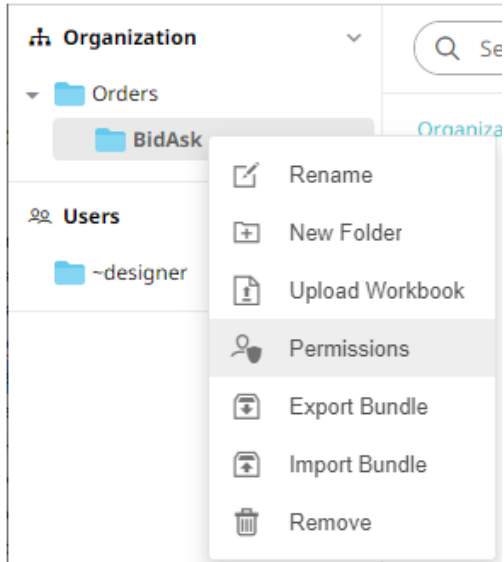
The corresponding *Permissions* dialog displays.
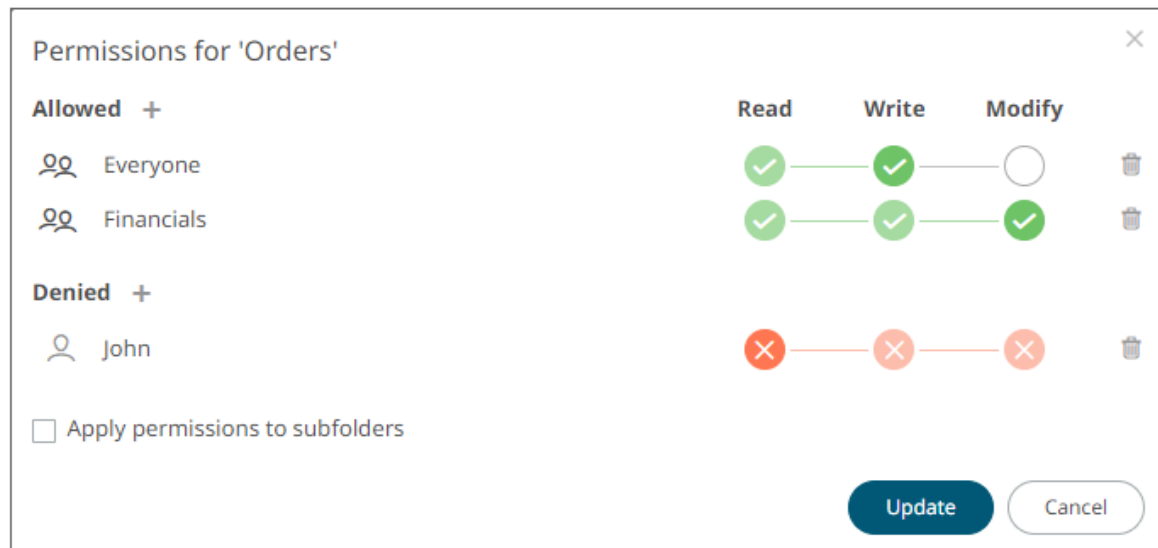
3. Make the necessary changes such as new folder name, add or delete users and groups.

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 **Update** 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.
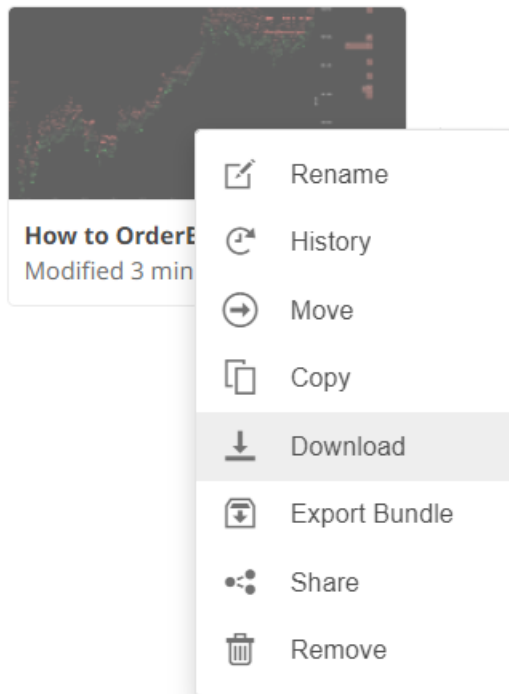
A copy of the workbook is downloaded.

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
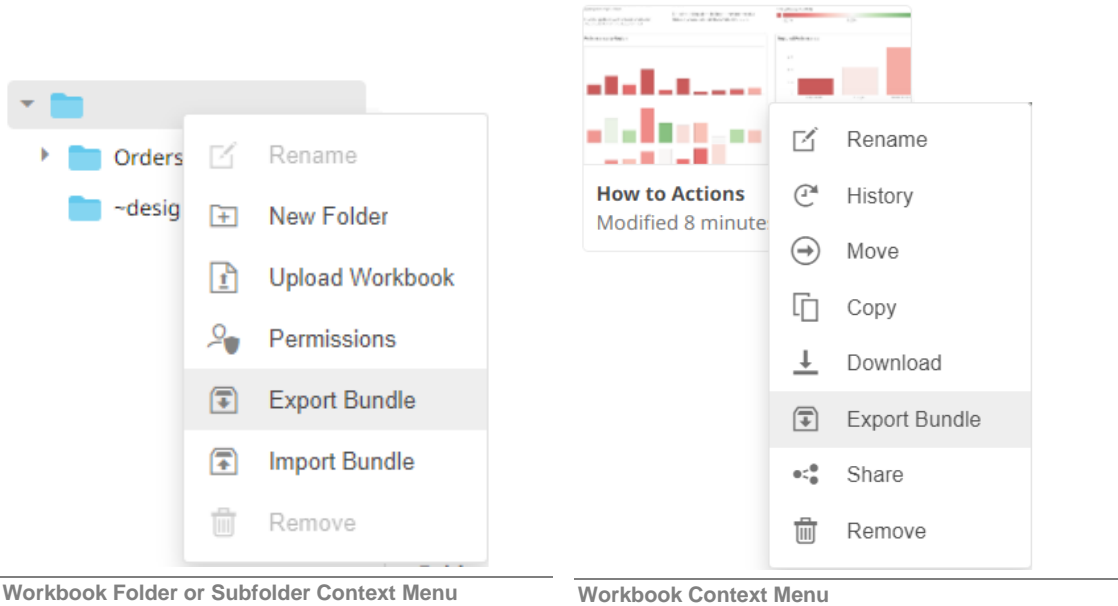
A notification message displays.
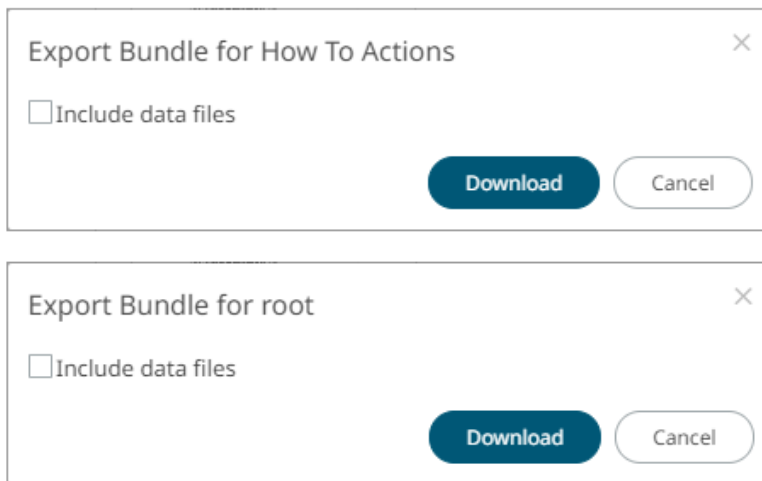
2. Check the **Include Data Files** box to include the associated workbook data files in the download.

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.
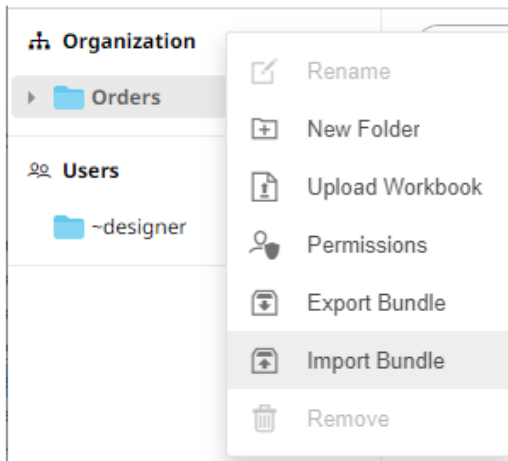
The *Import Bundle* dialog displays.
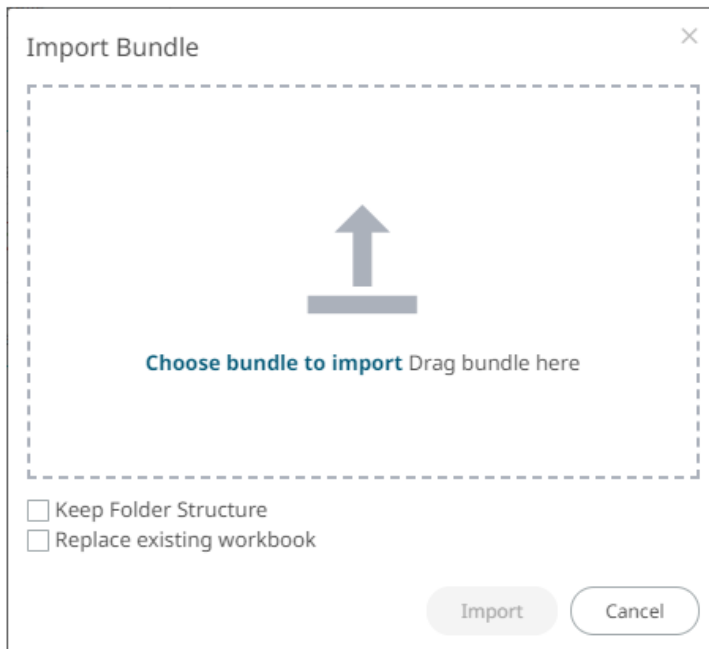
2. To import a bundle, you can either:

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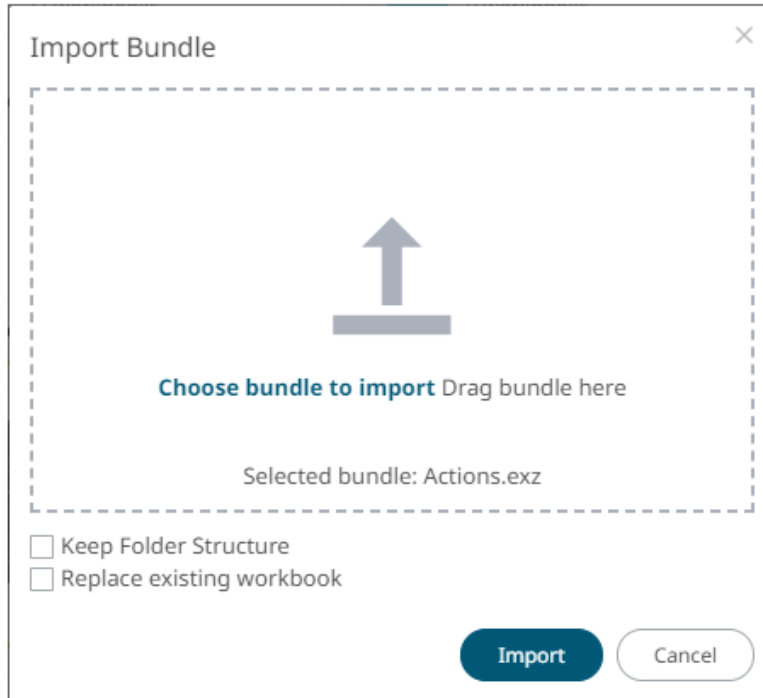
3. Check the **Keep Folder Structure** box.

   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 .

Data Level Secure Access

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

Data is filtered using the special parameter **_user_id.**

This **_user_id** parameter is replaced at run time by the authenticated user id in lower case.
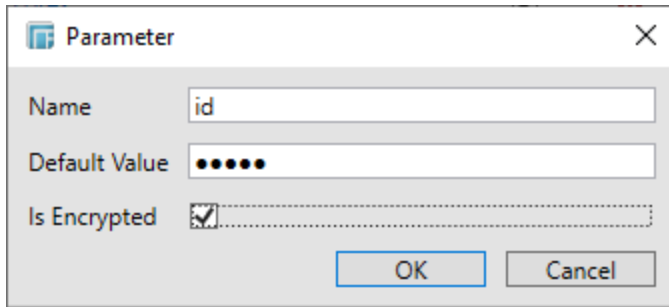
This parameter can then be used to restrict the data being retrieved, though use in either:

❑   Connection Details to Data Sources

❑   Filter constraints on data queries (e.g., SQL WHERE Clauses)

[8] SYSTEM ADMINISTRATION

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

Page	Description
System Settings	Allows to 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.
Scheduler	Allows scheduling of email send outs and extracting of data.
Logged In User	Allows to view and manage logged in users.
API Tokens	Allows 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

❑ 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.

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 and columns

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.
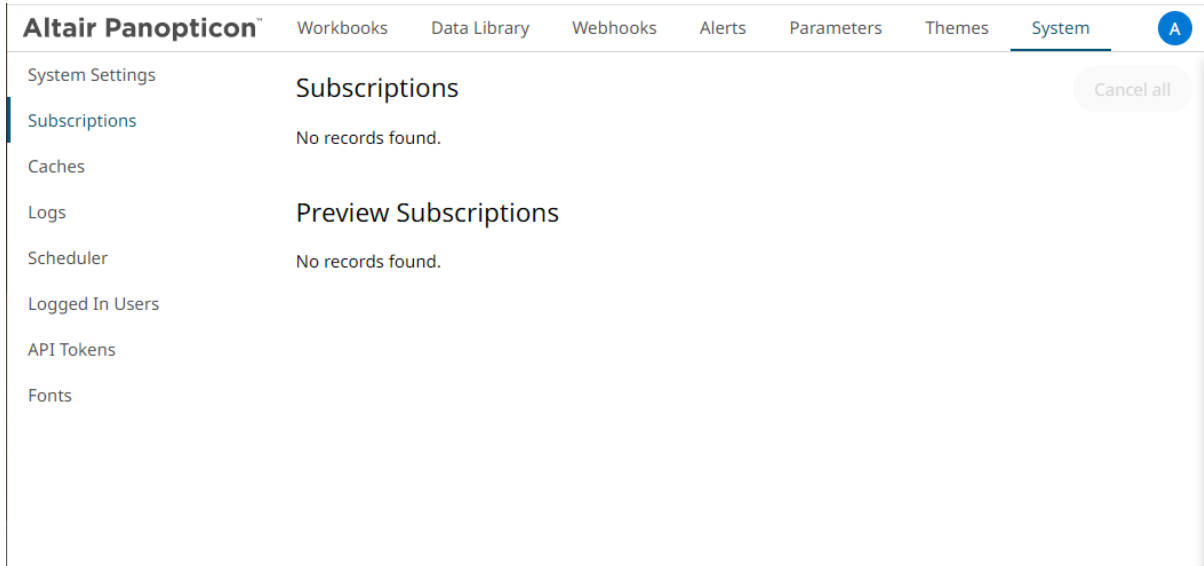
❑ Cancel a plugin subscription by clicking
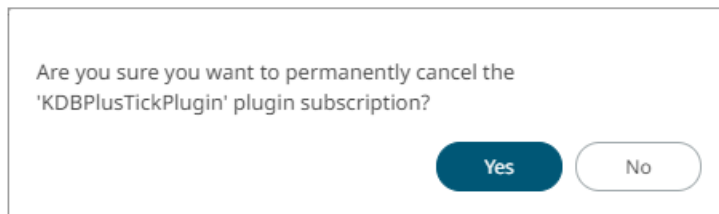
A notification message displays.

Click to cancel.

❑ Move to other pages

WORKING WITH CACHES

On the *Caches* page, you can view, refresh, or delete caches that are currently running on the server.

The *Caches* list includes the following information:

- ❏ Cache rendering type
- ❏ Workbook Name
- ❏ Data Table Title
- ❏ Data Source Name
- ❏ Read Count
- ❏ Cell Count
- ❏ Cache Age
- ❏ Time to Live (Expires In)
- ❏ Response Length

You can also do the following:

- ❏ Display Data Table Cache
- ❏ Display Data Source Cache
- ❏ Display Query Cache

Select a *Cache Type* to display in the list.

- Data Table Cache Type

	Read Count	Cell Count	Cache age	Expires in	Response Length
Axis Graphs ‣ AIDA Disciplines					
🔳 Data Table	1	2	8 minutes	7 minutes	-
Axis Graphs ‣ AIDA world records 1993-2021					
🔳 Data Table	1	2112	8 minutes	7 minutes	-
Axis Graphs ‣ Example Data					
🔳 Data Table	3	630	8 minutes	6 minutes	-

10 20 50 100

- Data Source Cache Type

	Read Count	Cell Count	Cache age	Expires in	Response Length
Axis Graphs ‣ AIDA Disciplines					
Datasource	1	2	8 minutes	6 minutes	-
Axis Graphs ‣ AIDA world records 1993-2021					
Datasource	1	498	8 minutes	6 minutes	-
Axis Graphs ‣ Example Data					
Datasource	1	140	8 minutes	6 minutes	-
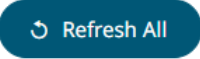
10 20 50 100

- Query Cache Type

		Read Count	Cell Count	Cache age	Expires in	Response Length
☐	Axis Graphs ‣ AIDA Disciplines					
	🗨 Query	2	-	9 minutes	5 minutes	500
☐	Axis Graphs ‣ AIDA world records 1993-2021					
	🗨 Query	2	-	9 minutes	5 minutes	128
☐	Axis Graphs ‣ Example Data					
	🗨 Query	5	-	9 minutes	5 minutes	2060

❑   Refresh page

Click the **Refresh** [Refresh All] button to refresh the list.

❑   Delete individual caches

Select the checkbox of a cache or several caches, then click [Delete] . A notification displays.

Are you sure you want to remove selected caches?

[Yes]   [No]

Click [Yes] to remove the cache or caches in the list.

❑   Move to other pages

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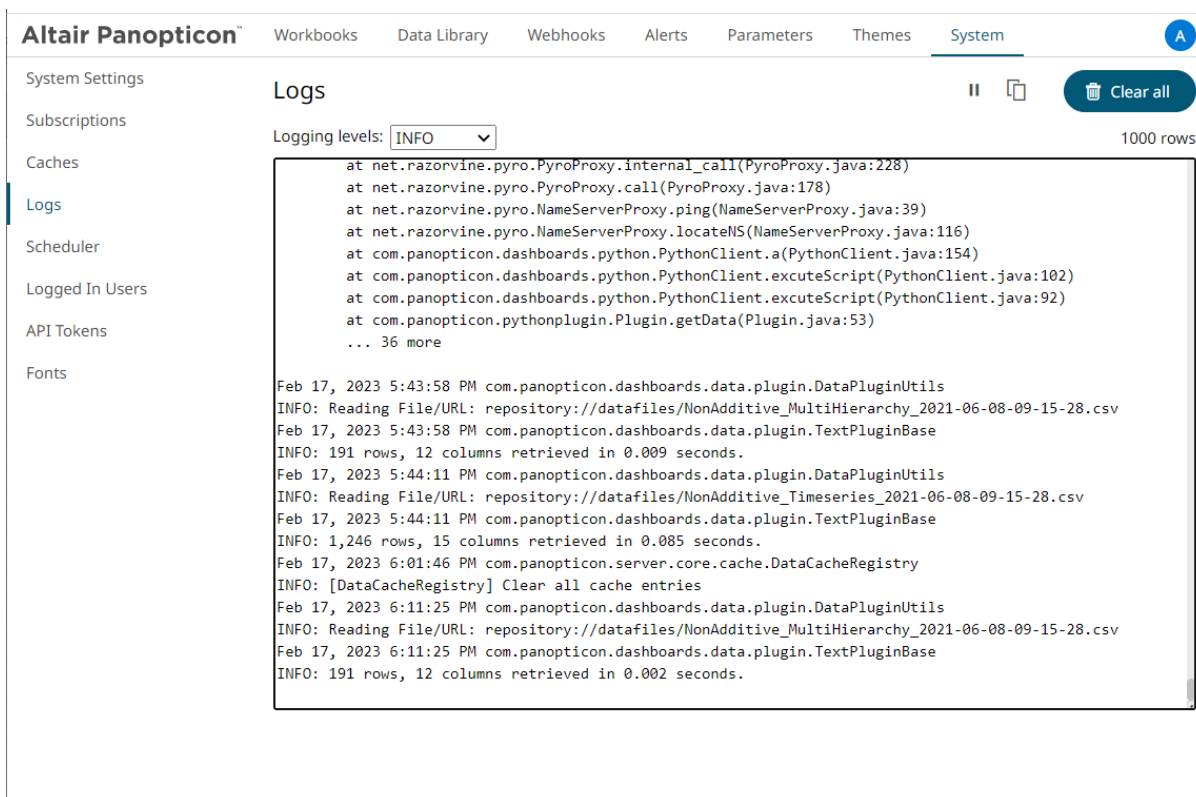
2. Select another *Logging Level* in the drop-down.

   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 to the list with the properties.

Scheduler Property	Description
1	**Run Task** Manually run scheduler task.
2	**Upload Task** Upload scheduler task.
3	**New Task** Create a new scheduler task.
4	**Delete Task** Delete a scheduler task.
5	**Download Task** Download a scheduler task.

6	**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:

- ❑ [Clearing and Importing Data Table to Data Store](#)

- ❑ [Importing Data Table to Data Store](#)

- ❑ [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.

Creating Task to Clear and Import Data Table to Data Store

Allows you to clear the earlier imported data and import again to the data store.

Steps:

1. On the *New Task pane*, enter the *Name* of the task. 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

Trigger	Period	CRON
Interval (sec)	3600	

   • **CRON** then enter a *CRON Expression* on the format:

   sec mins hours dayofmonth month dayofweek (e.g., 09 02 18 ? * MON-FRI)

Trigger	Period	CRON
CRON Expression ⑦		

4. Select the task *Type*: **Data Store – Clear and Import.**

5. Enter the *Description* of the task.

6. Select the *Data Table Name* that will be cleared in the data store and imported again.

7. Click 🖫 Save .

   • Once saved, you can opt to click ⊙ Run to manually run the task.

   • Click ← to go back to the *Tasks* pane. A new task is added to the list.

Creating Task to Import Data Table to Data Store

Allows you to store data closer to Panopticon server in an embedded database.

Steps:

1. On the *New Task pane*, enter the *Name* of the task. 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

Trigger	Period	CRON
Interval (sec)	3600	

   • **CRON** then enter a *CRON Expression* on the format:

   sec mins hours dayofmonth month dayofweek (e.g., 09 02 18 ? * MON-FRI)

Trigger	Period	CRON
CRON Expression ⑦		

4. Select the task *Type*: **Data Store – Import Data.**
5. Enter the *Description* of the task.
6. Select the *Data Table Name* that will be imported in the data store.
7. Click 🖫 Save .

   • Once saved, you can opt to click ▶ Run to manually run the task.

   • Click ← to go back to the *Tasks* pane. A new task is added to the list.

Creating 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. 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 a *CRON Expression* on the format:

   sec mins hours dayofmonth month dayofweek (e.g., 09 02 18 ? * MON-FRI)

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.

    - 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	• See [Setting Parameter Values in Scheduler Tasks](#) for more information on the different syntax you can use to enter parameters with multiple values, as well as parameter values that contain comma.
	• 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.

- 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**.**

7. You can opt to tap the **Zip CSV Data** slider to attach a zipped copy of the CSV data in the email.

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 .

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

- Click ← to go back to the *Tasks* pane. A new task is added to the list.

Creating 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. 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

Trigger	Period	CRON
Interval (sec)	3600	

   * **CRON** then enter a *CRON Expression* on the format:

     sec mins hours dayofmonth month dayofweek (e.g., 09 02 18 ? * MON-FRI)

Trigger	Period	CRON
CRON Expression ⑦		

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`

- 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 .

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

   - Click ← to go back to the *Tasks* pane. A new task is added to the list.

Creating 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. 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

Trigger	Period	CRON
Interval (sec)	3600	

   * **CRON** then enter a *CRON Expression* on the format:

   sec mins hours dayofmonth month dayofweek (e.g., 09 02 18 ? * MON-FRI)

Trigger	Period	CRON
CRON Expression ⓘ		

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	• See <u>Setting Parameter Values in Scheduler Tasks</u> for more information on the different syntax you can use to enter parameters with multiple values, as well as parameter values that contain comma.
	
	• 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 .

   • Once saved, you can opt to click ⏵ Run to manually run the task.

   • Click ← to go back to the *Tasks* pane. A new task is added to the list.

Creating 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. 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

Trigger	Period	CRON
Interval (sec)	3600	

   - **CRON** then enter a *CRON Expression* on the format:

   sec mins hours dayofmonth month dayofweek (e.g., 09 02 18 ? * MON-FRI)

Trigger	Period	CRON
CRON Expression ⓘ		

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	• See [Setting Parameter Values in Scheduler Tasks](#) for more information on the different syntax you can use to enter parameters with multiple values, as well as parameter values that contain comma.
	• 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.

- enter the *Width* and *Height* of the Image file. Default values are **1024** and **768,** respectively.
- check the **Hide Scrollbars** box.
- check the **Hyperlinks** box. This makes the Image file in the email will be clickable.

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

7. 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.

8. Click 💾 **Save** .

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

- Click ← to go back to the *Tasks* pane. A new task is added to the list.

Creating Task to Send PDF File via Email

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

Steps:

2. On the *New Task pane*, enter the *Name* of the task. 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

Trigger	Period	CRON
Interval (sec)	3600	

- **CRON** then enter a *CRON Expression* on the format:

   sec mins hours dayofmonth month dayofweek (e.g., 09 02 18 ? * MON-FRI)

Trigger	Period	CRON
CRON Expression ⑦		

5. Select the task *Type*: **Email PDF**.

6. 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.

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

8. 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	• See [Setting Parameter Values in Scheduler Tasks](#) for more information on the different syntax you can use to enter parameters with multiple values, as well as parameter values that contain comma.
	• 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.

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

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

10. 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.

11. Click 🖫 Save .
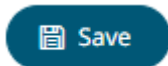
- Once saved, you can opt to click ⊙ Run to manually run the task.

- Click ← to go back to the Tasks pane. A new task is added to the list.

Creating Task to Extract Data

Tasks can be created to reload workbooks or global extracts.

Steps:

1. On the *New Task pane*, enter the *Name* of the task. 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

Trigger	Period	CRON
Interval (sec)	3600	

   - **CRON** then enter a *CRON Expression* on the format:

     sec mins hours dayofmonth month dayofweek (e.g., 09 02 18 ? * MON-FRI).

Trigger	Period	CRON
CRON Expression ⓘ		

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.

   Extract name ⓘ

   BitcoinOrders
   Orders
   Orders\Order
   Orders\StocksStaticExtract
   Orders\WebDataExtract
   StocksStatic

7. Click 🖫 **Save** .

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

   - Click ← to go back to the *Tasks* pane. A new task is added to the list.

Setting Parameter Values in Scheduler Tasks

Use any of the following syntax to define parameter values in scheduler tasks:

❑ For multiple values (array parameter), use bracket syntax

Example: `parameter1=[Value1, Value2, Value3]`

❑   For a parameter with a value containing comma, quote the value in double quotes

Example: `parameter1="Parameter value, containing comma"`

❑   The double quoting can also be used inside arrays

Example: `parameter1=[Value1, "Value2, containing comma"]`

❑   Normal parameters, quoted parameters, and array parameters can be mixed

Example: `parameter1=Normal,parameter2=[Val1,Val2],parameter3="Quoted Value"`
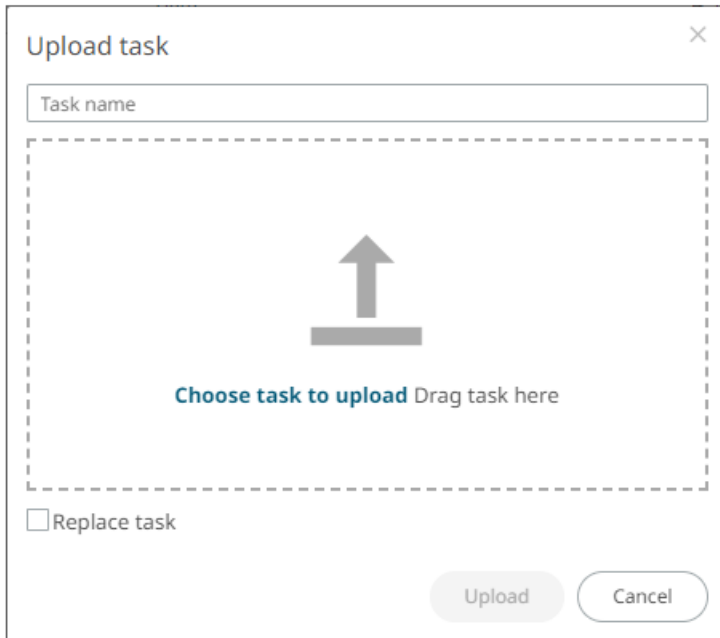
Uploading a Scheduler Task

Users with an Administrator role can upload scheduler task definitions.

Steps:

1.   On the **Scheduler** tab, click **Upload**   .

The *Upload Task* dialog displays.

2.   To upload a task, you can either:

• drag it from your desktop and drop on the dialog, or

• click **Choose task to upload** and select one on the *Open* dialog that displays.

The name of the task is displayed on the uploaded task area and in the *Name* box.
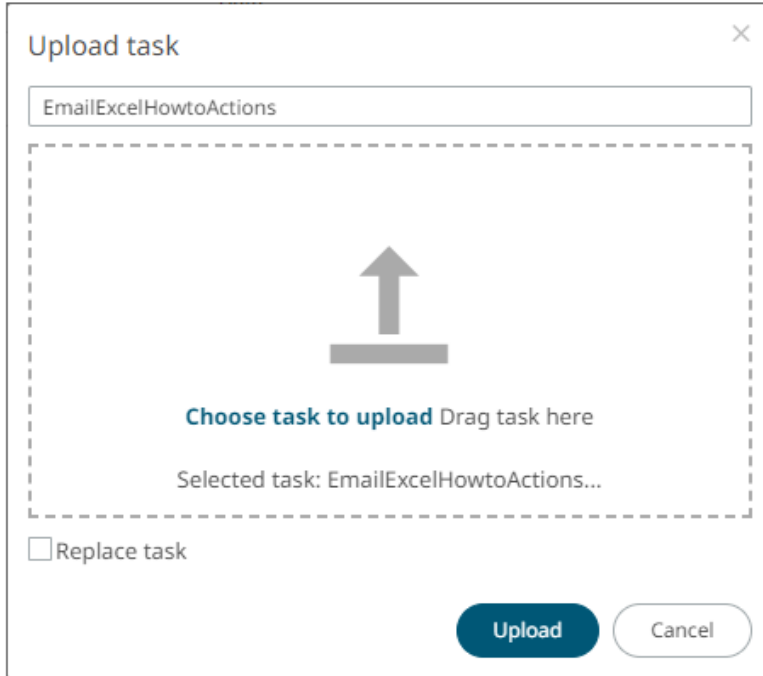
3. You can opt to rename the task.

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

5. Click .

   You will be notified once the task is uploaded.
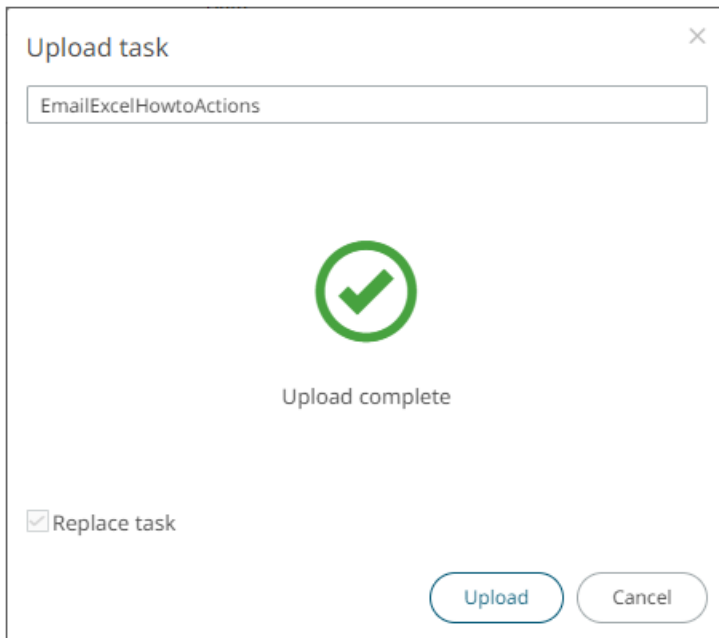
The task is added and displayed in the *Scheduler* list.

Downloading a Scheduler Task

Users with an Administrator role can download scheduler task definitions.

Click the **Download** ⤓ icon of a task.

Other Scheduler Tasks Operations

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

❑ Sort tasks

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 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 ▶.

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

❑ [Modify](#) tasks

❑ Delete tasks

Click 🗑 of a task. A confirmation message displays.

Are you sure you want to delete the task?

[Yes] [No]

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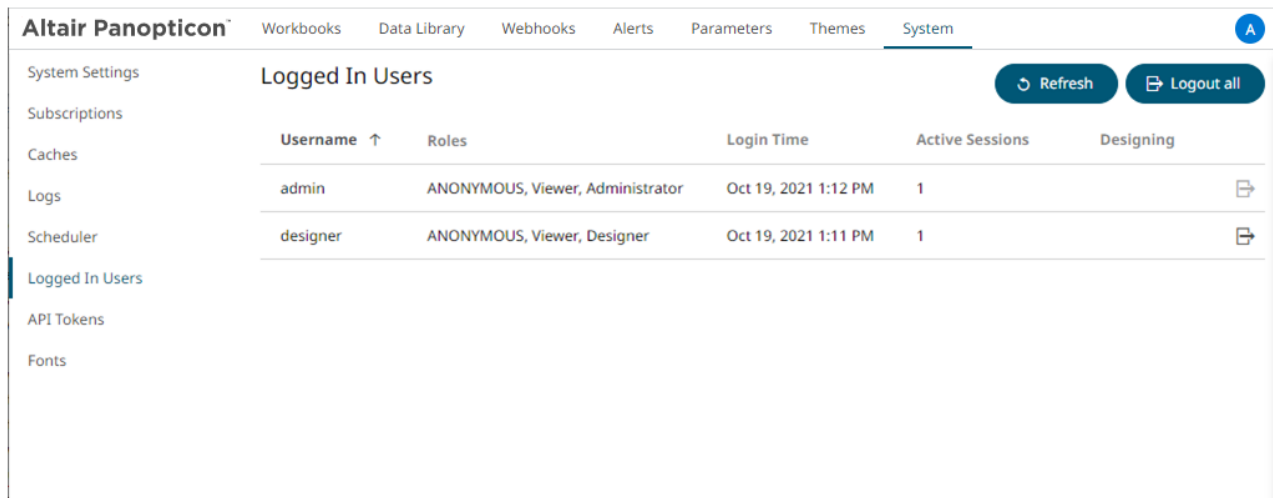
In the list, the following properties are displayed for each user:

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
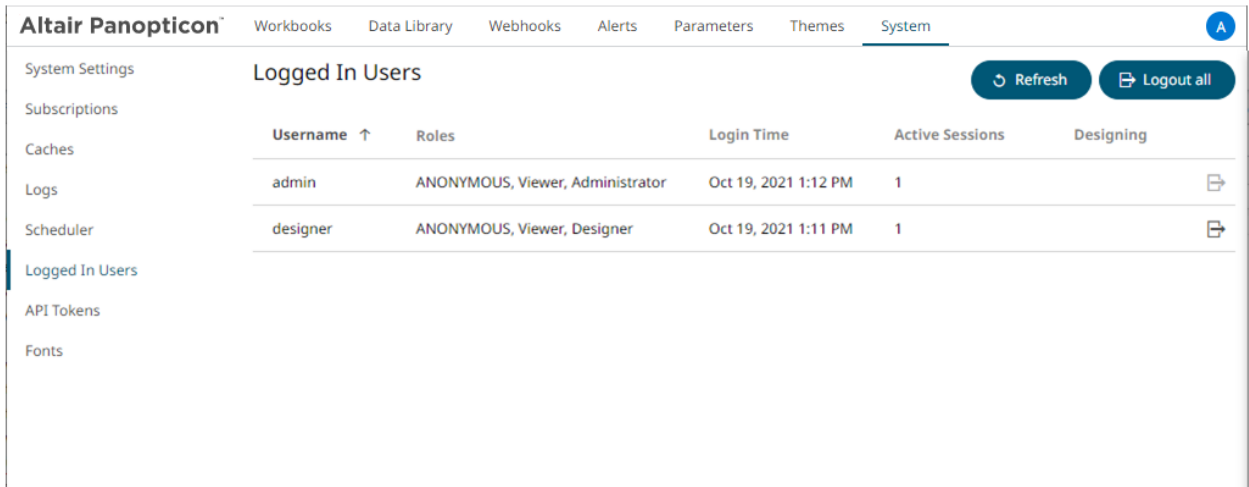
Logging Out Users

Logging out users on the server consequently deletes their tokens.

Logging Out All Users

Steps:

1. Click ⏻ Logout all

A notification message displays.

2. Click Yes .

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.
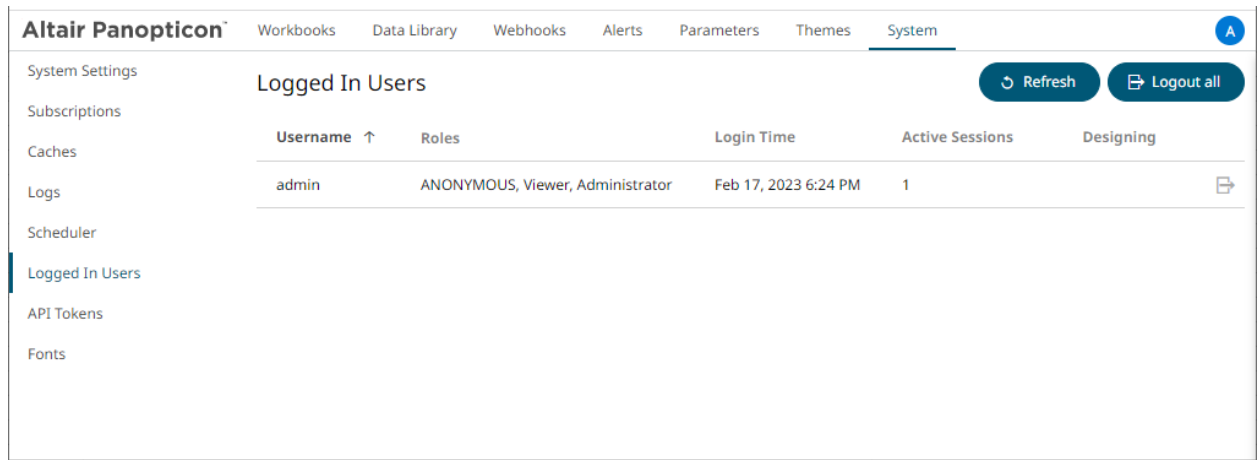
Logging Out Individual Users

Steps:
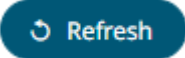
1. Click the ⯮ button of a user in the list.

   A notification message displays.

2. Click **Yes** .
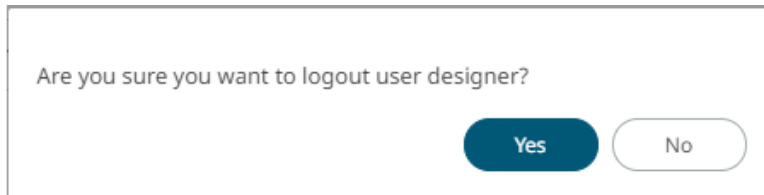
   The user is logged out and their token is deleted.

Refreshing the Logged In Users List
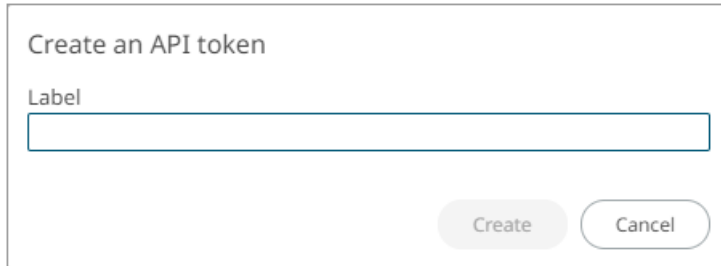
Click **↻ Refresh** to refresh the list of logged in users.

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:
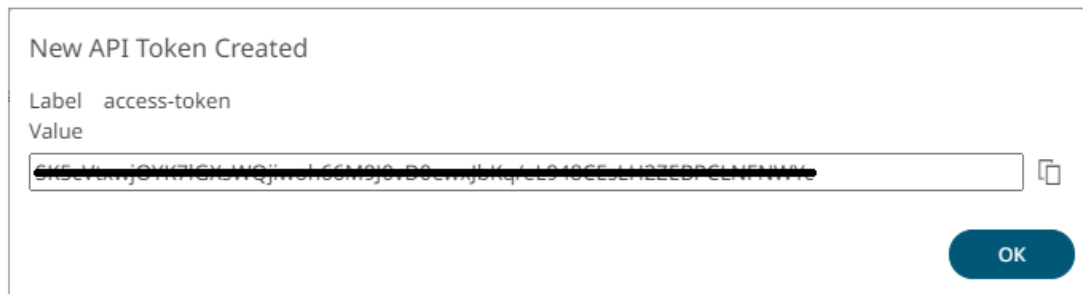
1. Click ![+ Create]. The *Create an API Token* dialog displays.

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 ![copy icon] to ensure you have a copy of the key and paste in a secure location.

5. Click ![OK]. The new API Token is displayed on the list.
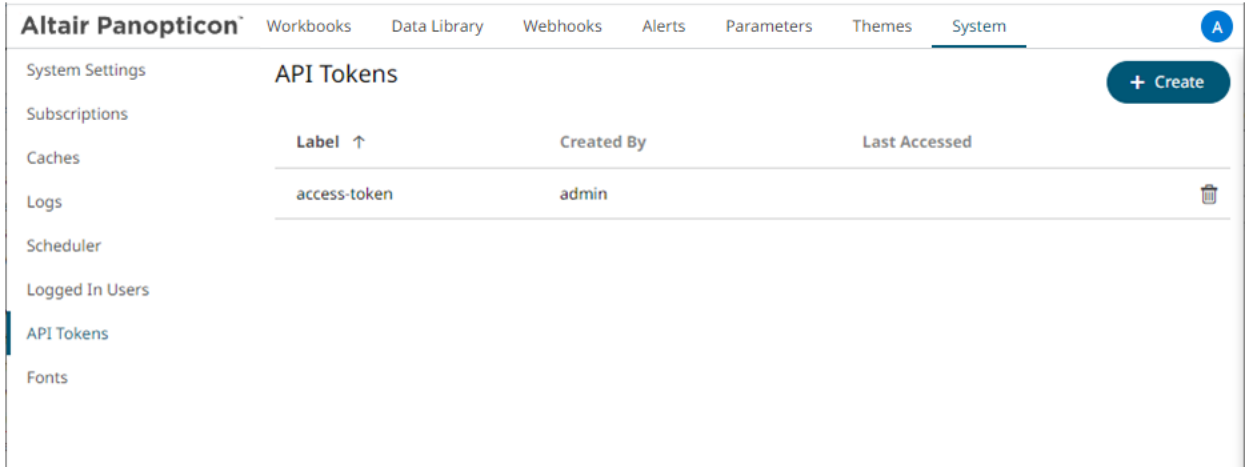
In the list, the following properties are displayed for each API Token:

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.
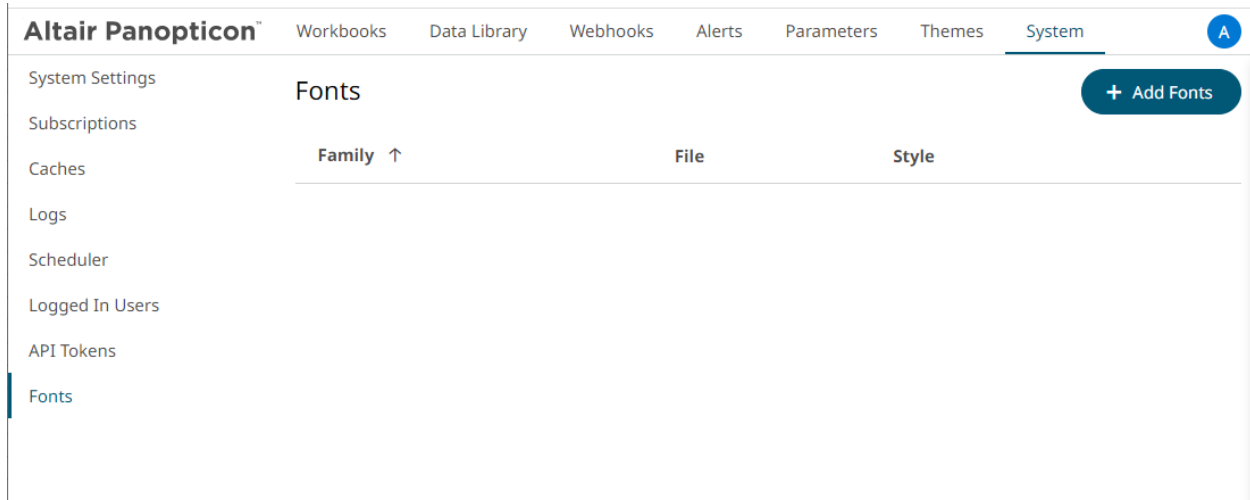
The supported custom font files include the following:

❏ 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 ![Add Fonts]. The *Add Fonts* dialog displays.
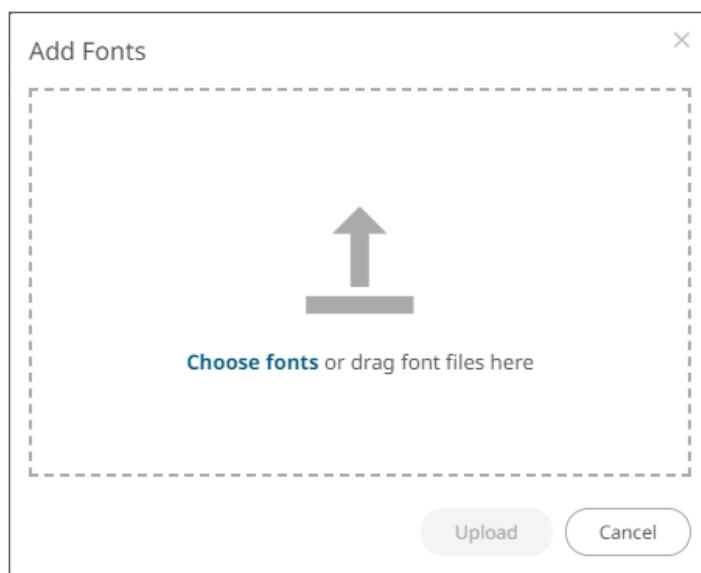
2. To add fonts, you can either:

- 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 is displayed on the uploaded font area.
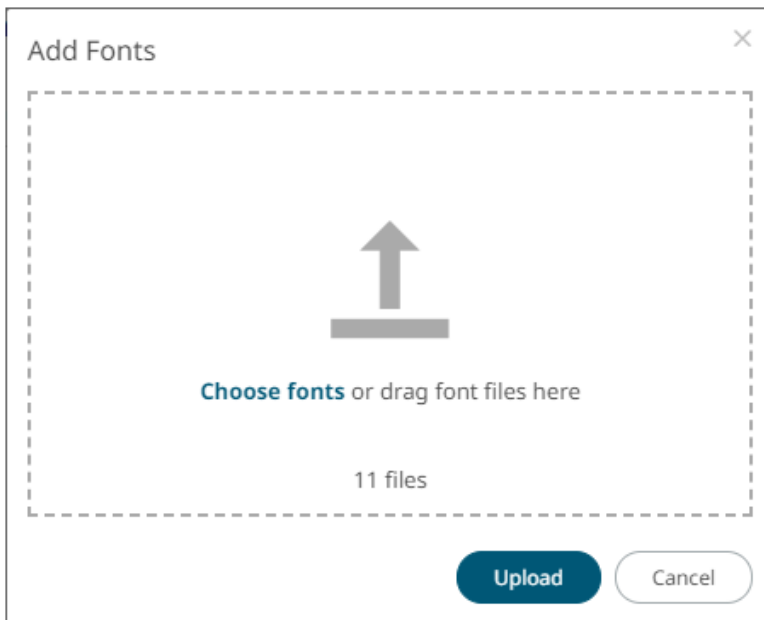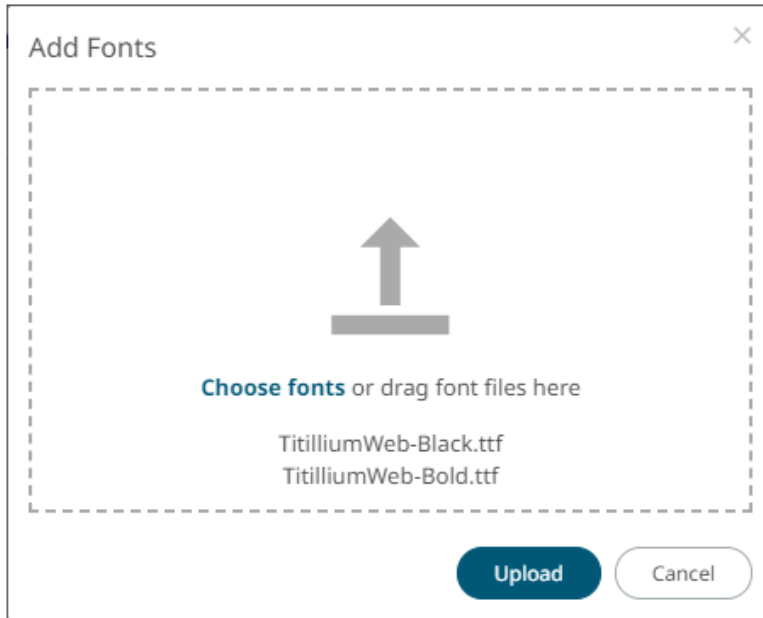
3. Click **Upload** .

A notification prompt will be displayed once the fonts are uploaded.
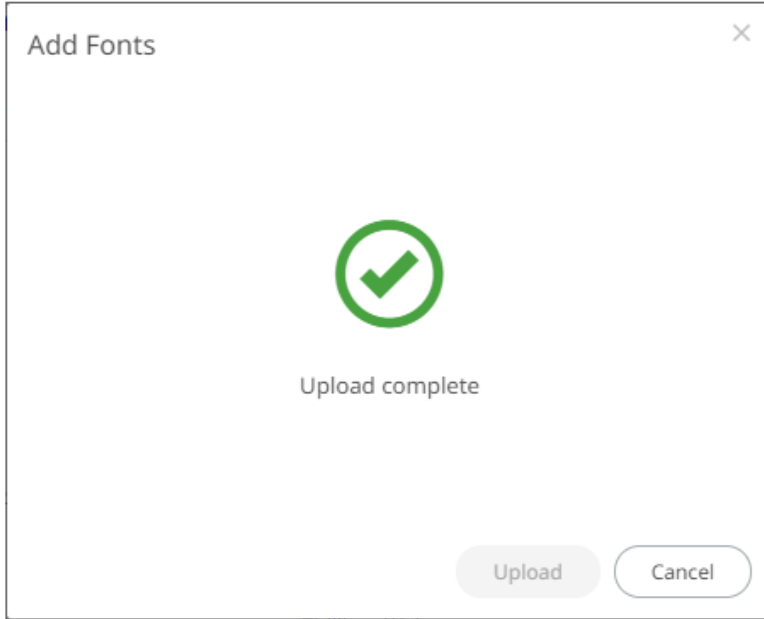
The added custom fonts are displayed on the *Fonts* page.

Family ↑	File	Style	
	Titillium Semibold	Bold	🗑
	Titillium Semibold Italic	Bold Italic	🗑
Titillium	Titillium	Regular	🗑
	Titillium Regular Italic	Italic	🗑
	Titillium Black	Regular	🗑
	Titillium Bold	Bold	🗑
Titillium Bd	Titillium Bold Upright	Italic	🗑
	Titillium Bold Italic	Bold Italic	🗑
	Titillium Thin	Regular	🗑
	Titillium Light Italic	Bold Italic	🗑
Titillium Lt	Titillium Light	Bold	🗑
	Titillium Thin Italic	Italic	🗑
	Titillium Regular Upright	Bold	🗑
	Titillium Light Upright	Italic	🗑
Titillium Up	Titillium Thin Upright	Regular	🗑
	Titillium Semibold Upright	Bold Italic	🗑

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

Click .

[9] 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 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 connector

Dependencies for Elasticsearch are included in Panopticon Real Time installation.

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`)

❑   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 Panopticon Real Time.

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

Includes: Cassandra, Elasticsearch 7.x, InfluxDB 1.x, 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.

| JNDI Name | ▼ | jdbc/Postresql | (JNDI resource name as defined inside Context eg. jdbc/MyDB) |

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*.

URL	▼	jdbc:postgresql://localhost:5432/stc
Driver Class Name		org.postgresql.Driver
User Id		buyer
Password		······	☐ Show characters

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, FastAPI is the recommended integration method for Linux systems, while Pyro4 (Python Remote Data Objects) can be used on either Windows or Linux.

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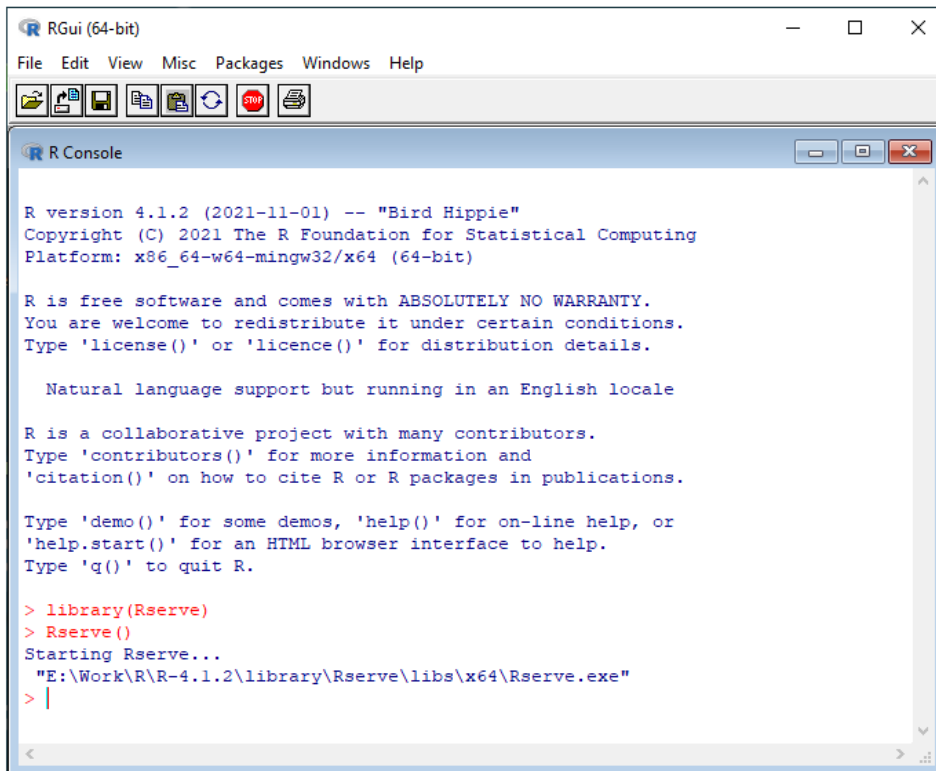
```
install.packages("Rserve")
```

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 with Pyro4

Panopticon can use Python for both data transforms and as a primary data source. The server part of Panopticon will send requests to Python, with data and/or Python code, via Pyro4 - Python Remote Objects. Pyro4 installs as a Python package in your Python environment, and connectivity is enabled by starting a Pyro process with a shell script file (.BAT script file) which is included in the Panopticon distribution zip-archive.

To set up a Python environment that can be used from Panopticon, follow these steps:

1. Download and install Python.
2. Install Pyro4.
3. Install pandas.
4. Install additional packages.
5. Set the Pyro HMAC key.
6. Start Pyro4.

Downloading and Installing Python

Download Python from https://www.python.org/downloads/ and select the release version you require as well as the right version for the operating system of your server. Note that Linux systems often have Python included out of the box. Install Python as described by documentation from Python.org. You can install Python on the same host that runs the Panopticon server, or a different host, if firewall settings and port mapping allow communication between the two hosts. If you are running Panopticon for development, testing or personal use on your workstation, install Python on your workstation as well.

Installing Pyro4

When Python is installed, add Pyro4 by installing it like a Python package. The Pyro4 version must be 4.71 or higher. On the command prompt, type **python** and press **Enter** to start a Python prompt. Then run this command:

```
pip install Pyro4>=4.71
```

Installing Pandas

When working with a data table in Python, the pandas package and the pandas DataFrame object provides many useful advantages and is highly recommended. Panopticon's integration with Pyro4 will check if the object returned from Python is a pandas DataFrame and therefore the pandas package is required. On the Python prompt, run this command:

```
pip install pandas>=1.1.5
```

This will also automatically give you the NumPy package.

Installing Additional Packages

To be able to view and use the examples in the Panopticon example workbook "How to Python", you are also required to install a few additional packages using these commands:

```
pip install scikit-learn>=1.1.3
```

```
pip install pyarrow>=10.0.0
```

Setting the Pyro HMAC Key

When you send a request to Pyro4, you are required to supply the correct password which is called the Pyro HMAC Key. This password protects the Python environment from unauthorized remote calls via Pyro4. You should create an environment variable named **PYRO_HMAC_KEY** on the host where Python and Pyro4 are installed. However, if **PYRO_HMAC_KEY** is not found or created, the environment variable will be created by the script used for starting Pyro4. The default value is **password**. You have the option of either:

❑ creating the **PYRO_HMAC_KEY** and setting a password value of your choice

❑ editing the start script and entering your password value instead of the default value **password**

When using Python from Panopticon, either as a transform or as a primary data source, you will supply the password as part of the connection settings in Panopticon. The password can also be saved in the Panopticon.properties file, by an Administrator, which will let Designer users create Python connections without knowledge of the password.

For details about the configuration of Python connectivity properties, see Properties: Panopticon.

Starting Pyro4

Before you can use Python from Panopticon, you must start the Pyro4 process that will receive requests from Panopticon and pass them on to Python. This is done by running a script included with Panopticon on the host where Python and Pyro4 are installed.

On Linux, you run the file **start_Python_connectivity.sh** which in turn runs the file **pyro.py.**

On Windows, you run the file **start_Python_connectivity.bat** which in turn runs the file **pyro.py**.

Multiple Python Environments on Windows

On Windows, you can install multiple Python versions in parallel, resulting in installation folders like the following examples:

❏ C:\Program Files\Python38

❏ C:\Program Files\Python39

❏ C:\Program Files\Python310

Each of these versions has its own package installations. For example, you can have one version of a package installed for Python 3.9 and another package version for Python 3.10.

> **NOTE** When installing packages for different versions of Python on a host which will serve multiple users, make sure you install from a command prompt with elevated privileges (run As Administrator), otherwise, packages will be installed under your own Windows user profile folder.

With Python for Windows downloaded from Python.org, you also get `py.exe` which is a Python launcher. When installing Python for all users, it is placed in `C:\Windows\py.exe` or `C:\Users\<username>\AppData\Local\Programs\Python`.

With the Python launcher `py.exe`, you can start a specific Python version as follows:

```
py -3.9
```

To make a package installation for a specific Python version, open a command prompt as Administrator and run:

```
py -3.9 -m pip install <packagename>
```

To start a Pyro4 process with a specific Python version (in this example, 3.9) you can launch Pyro4 as follows:

```
start_Python_connectivity.bat -3.9
```

Python Integration with FastAPI

On Linux OS, Panopticon can be integrated with Python using FastAPI instead of Pyro4.

Python integration with FastAPI has the advantage of supporting parallel processing.

Requirements:

❏ Python 3.8 or higher

❏ Python packages:

- uvicorn >= 0.20.0

- gunicorn >= 20.1.0

- fastapi >= 0.88.0

- pandas >= 1.5.1

- serpent >= 1.41

Steps:

1. Copy the **panopticon_fastapi_py** folder to the host where you have your Python environment.

   This can be the same host as where Panopticon is running, i.e., localhost from the point of view of Panopticon. If you are using different hosts, ensure that the firewall portmapping settings allow traffic on port **9090**.

   You can place the folder to your preferred location.

2. Set the following values in the `Panoption.properties` file:

   ```
   connector.python.host=<your_python_host>
   connector.python.mode=fast_api
   connector.python.port=9090
   ```

3. Restart Tomcat.

NOTE	• To start the FastAPI service, run the `start.sh` file with elevated privileges:
	``` sudo ./start.sh ```
	• To get back to the Linux prompt, hit **Ctrl+Z (^Z)** to pause the process and immediately run the command `bg` to send the process to the background.
	• To shut down the FastAPI service, run the `stop.sh` file with elevated privileges:
	``` sudo ./stop.sh ```

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.

[10] VIEWING AND MANAGING WORKBOOKS

ACCESSING WORKBOOKS

The **Workbooks** tab 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.
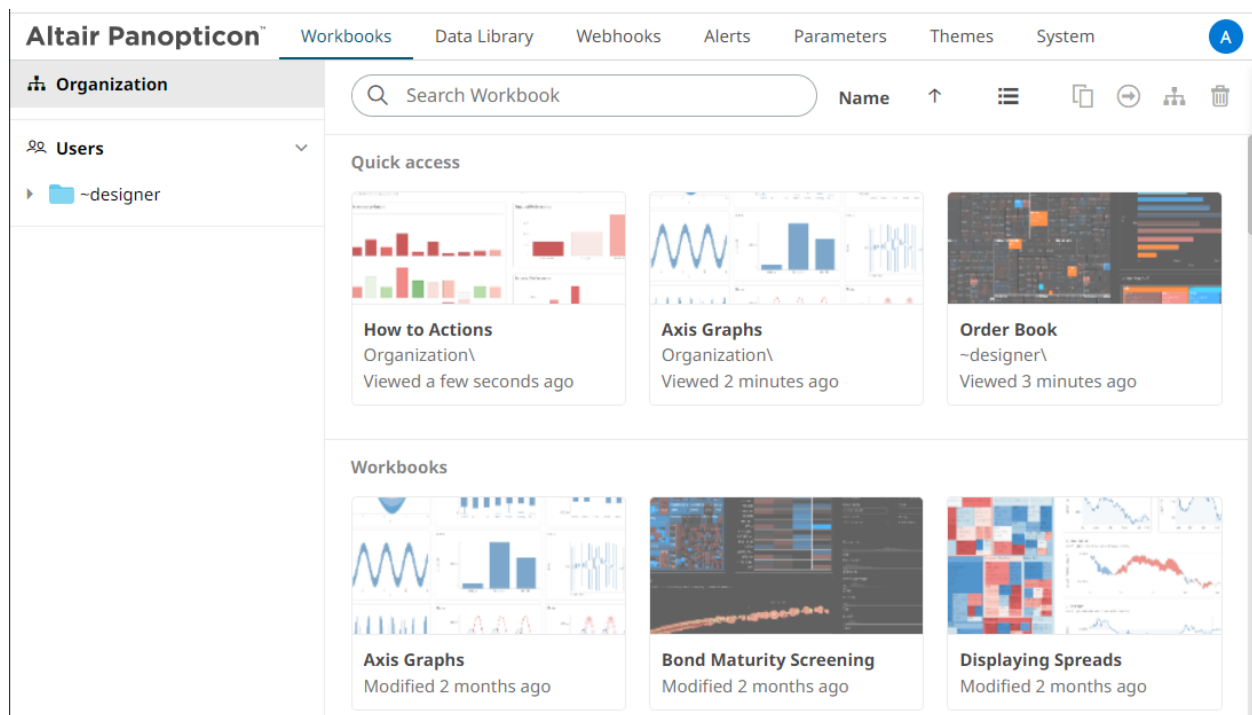
Clicking on the workbook thumbnail opens it on the web browser.

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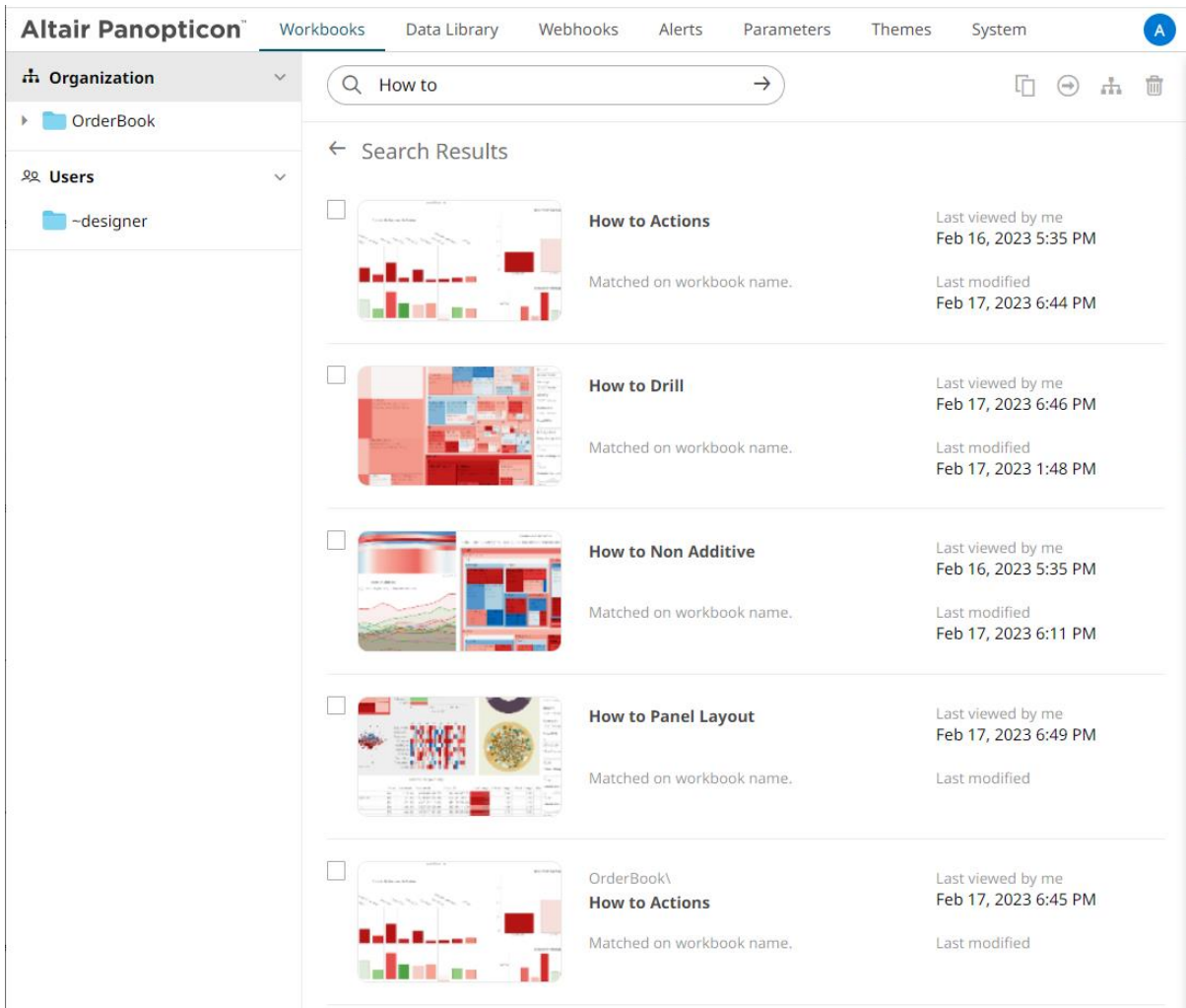
Click on a workbook thumbnail to open and display it on the web browser.

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.
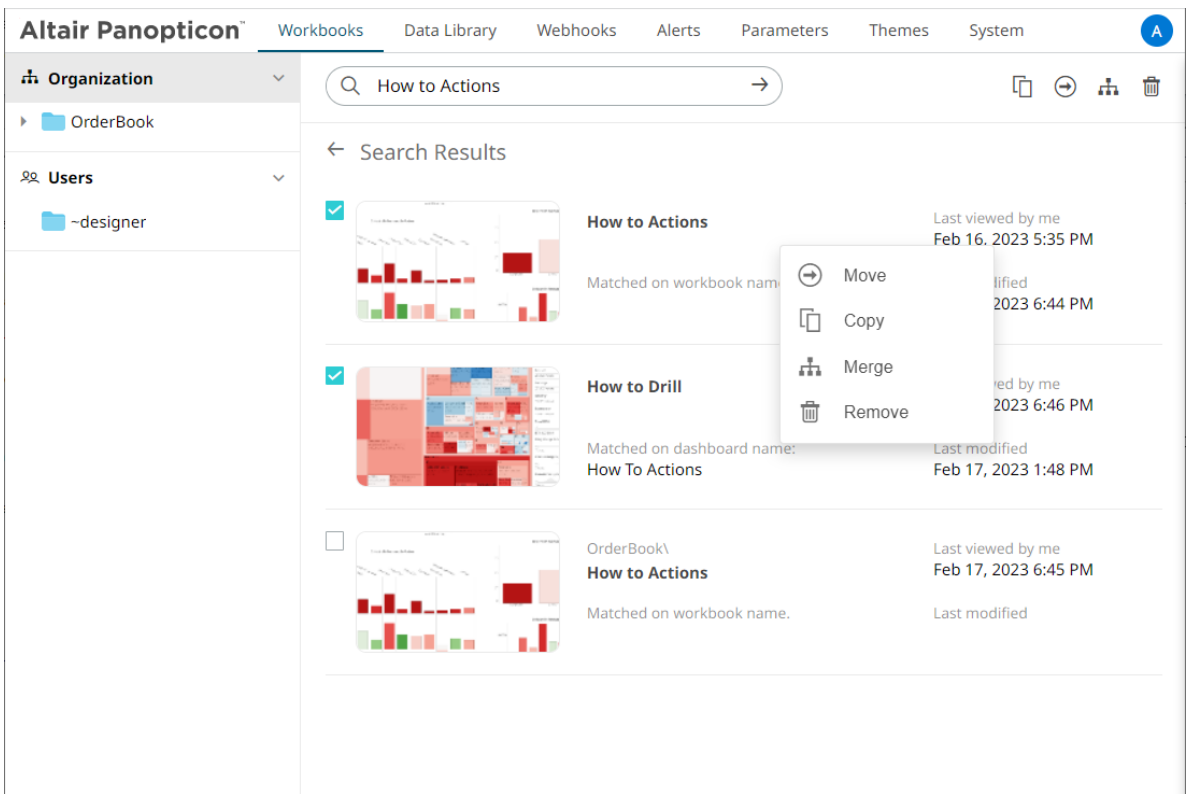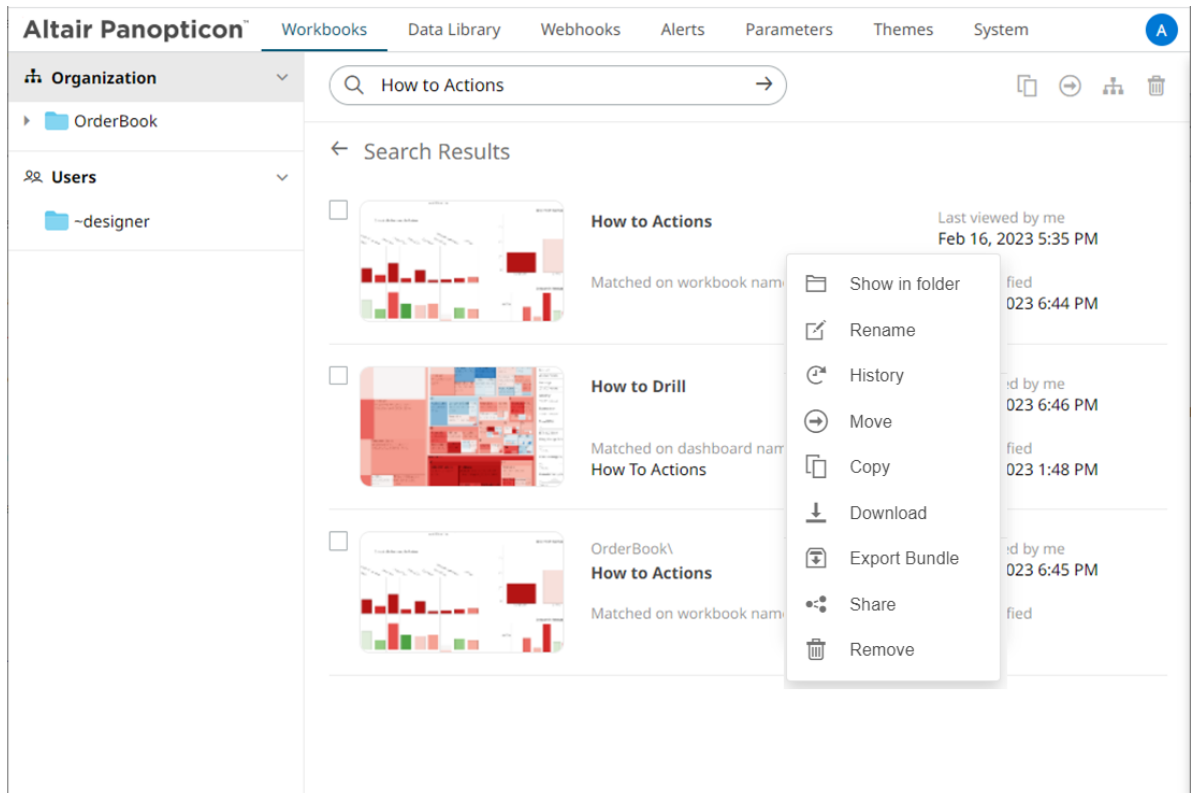
To display the workbook in its location, click **Show in Folder** on the context menu.
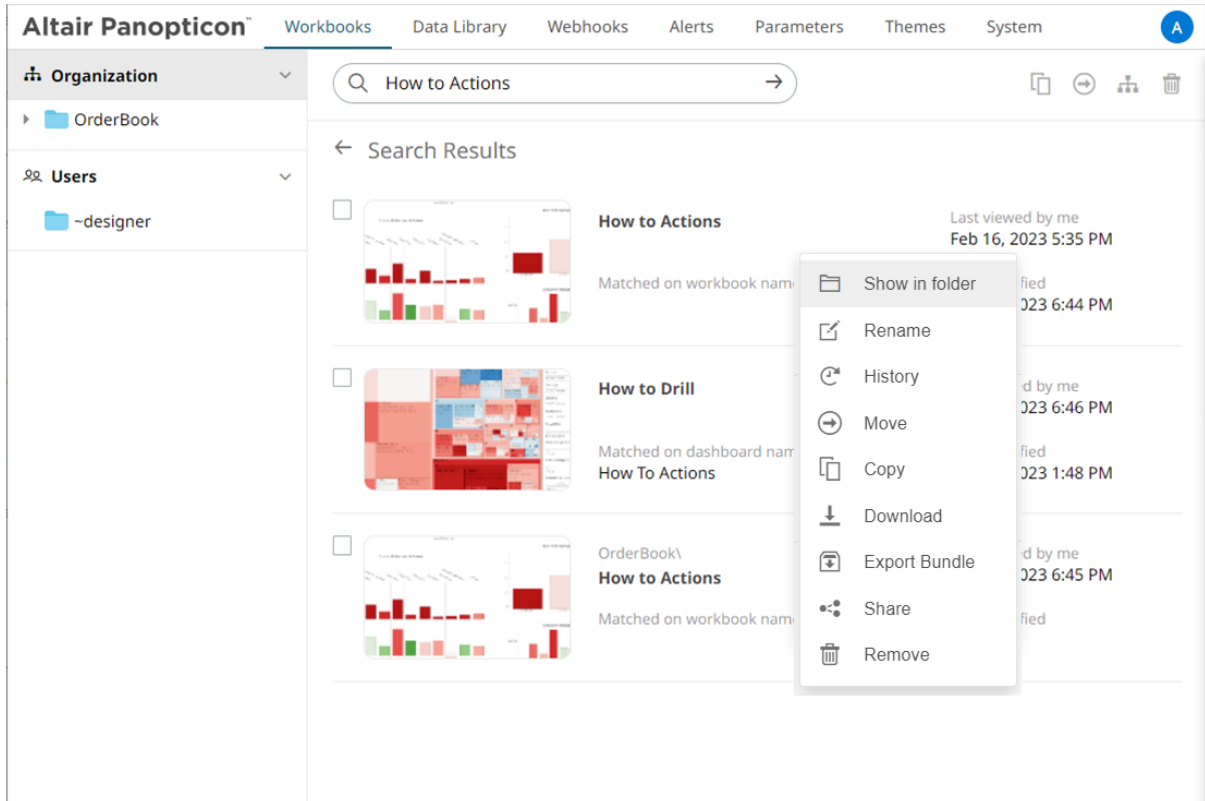
The other context menu options are discussed in the sections below.

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.
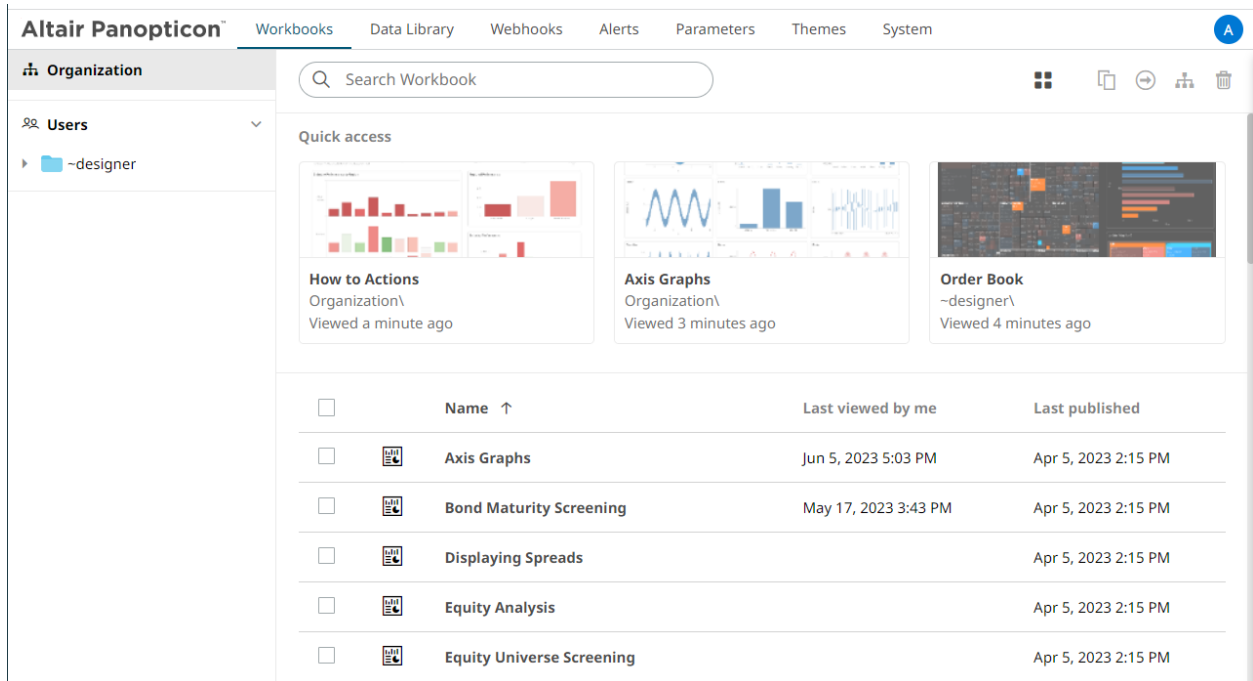
Or click **Grid View** . The folders and workbooks are displayed as thumbnails.
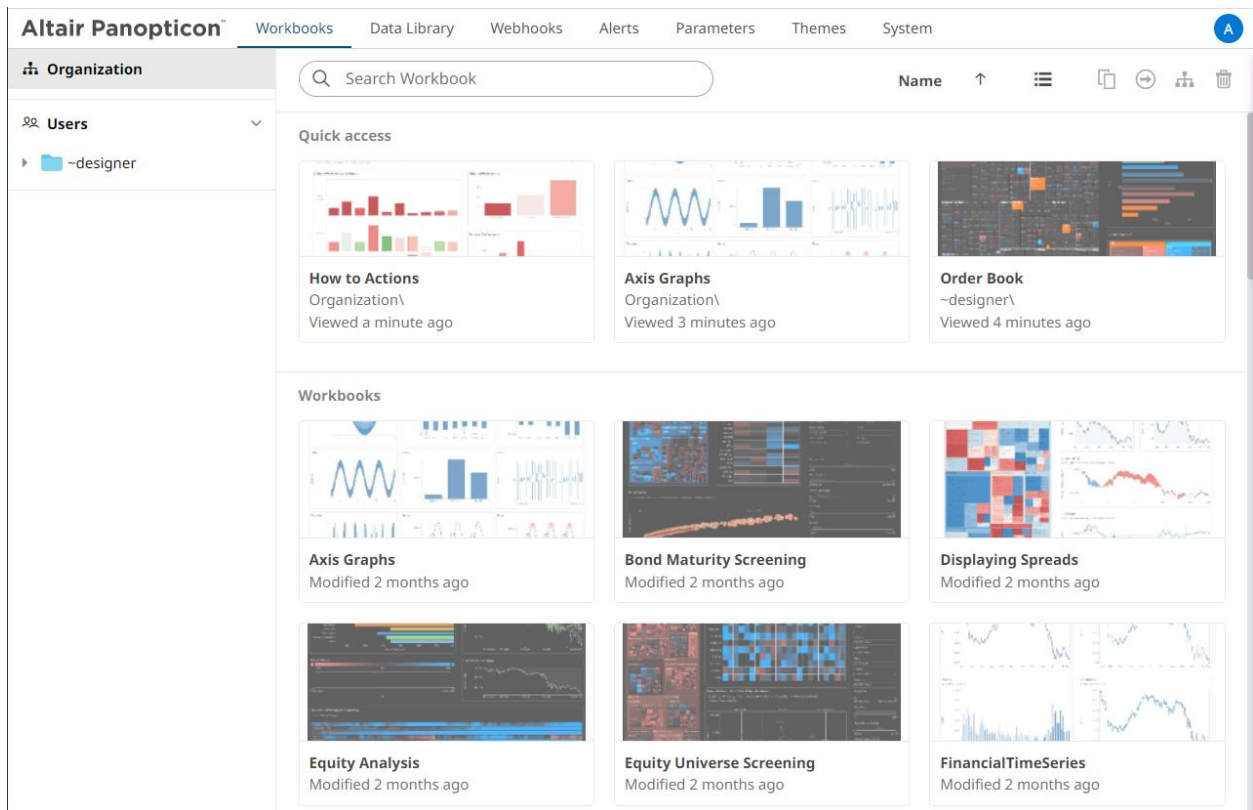
On either display view style, clicking on a workbook title or thumbnail displays the workbook on the *Open Workbook in View Mode.* For more information on how to analyze interactive dashboards, refer to the Client User Guide.

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**.
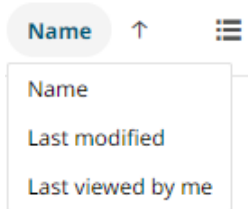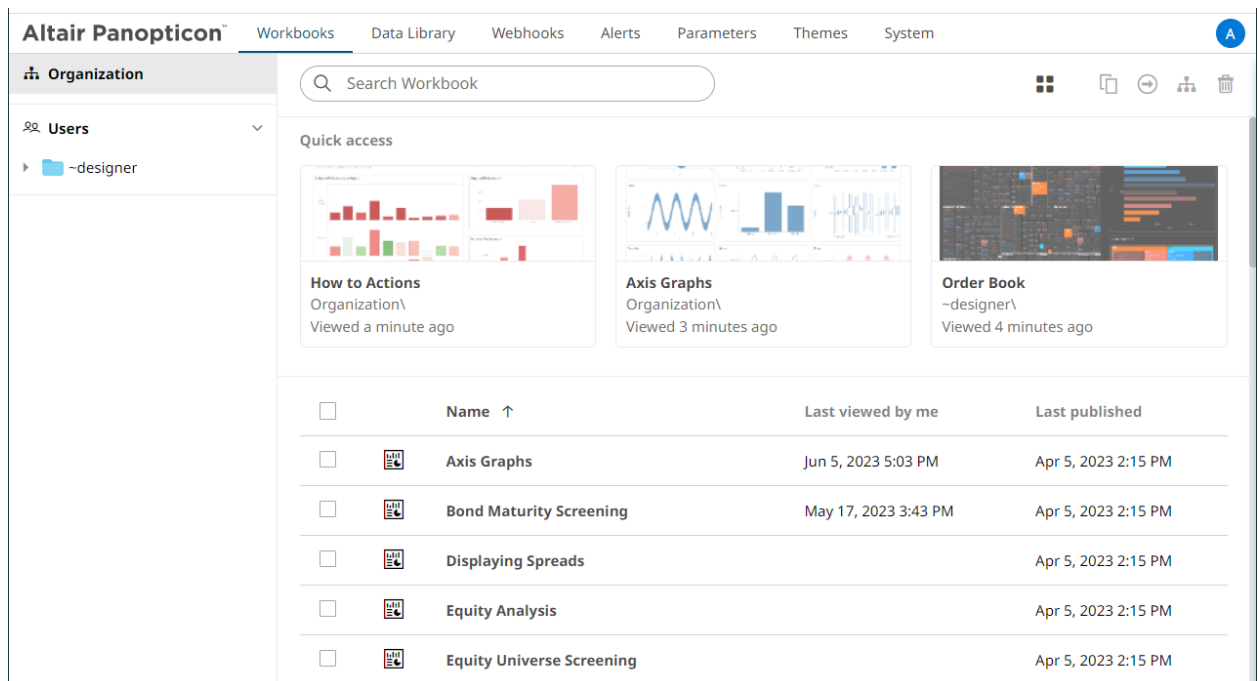
- Name
- Last Modified
- Last Viewed By Me

  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.
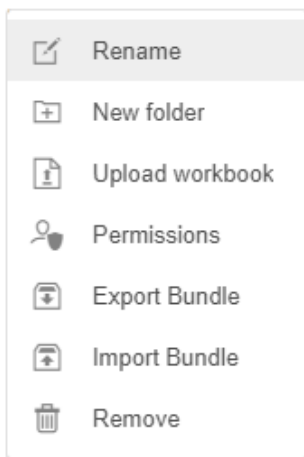
Workbook Folder or Subfolder Context Menu
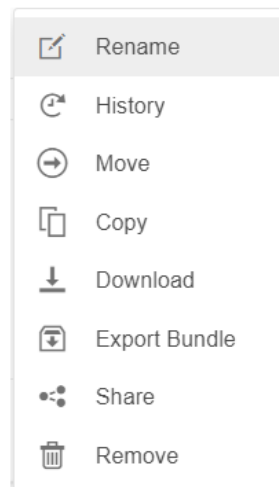
Workbook Context Menu

The *Rename Workbook* or *Rename Folder* dialog displays.
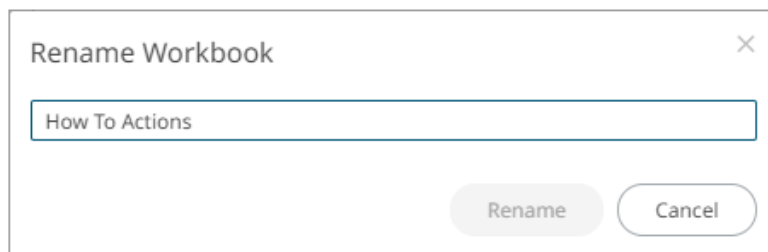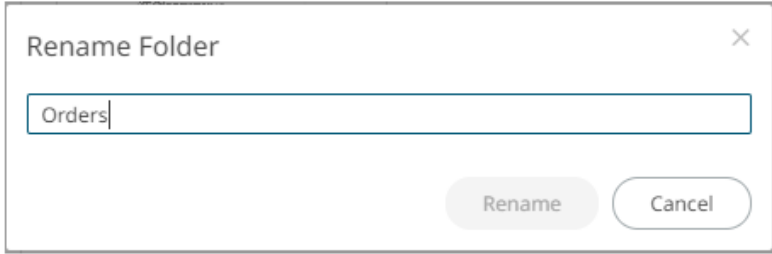
2. Enter a new name then click .

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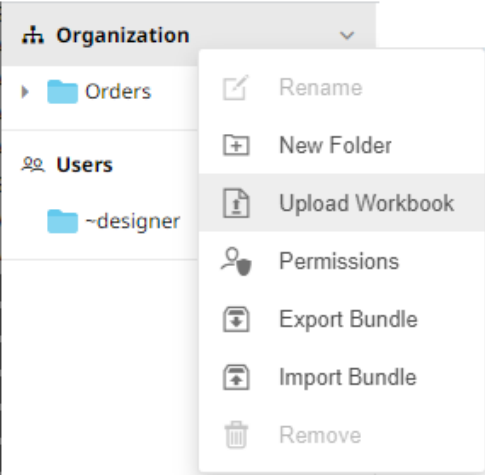
The *Upload Workbook* dialog displays.
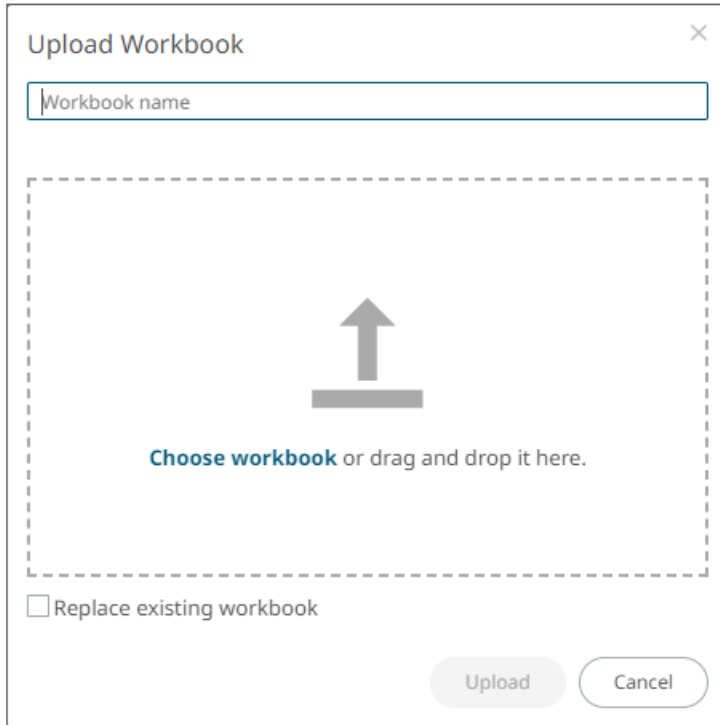
2. To upload a workbook, you can either:
   - drag it from your desktop and drop on the dialog, or
   - click **Choose Workbook** and select one on the *Open* dialog that displays.

   The name of the workbook is displayed on the uploaded workbook area and in the *Name* box.
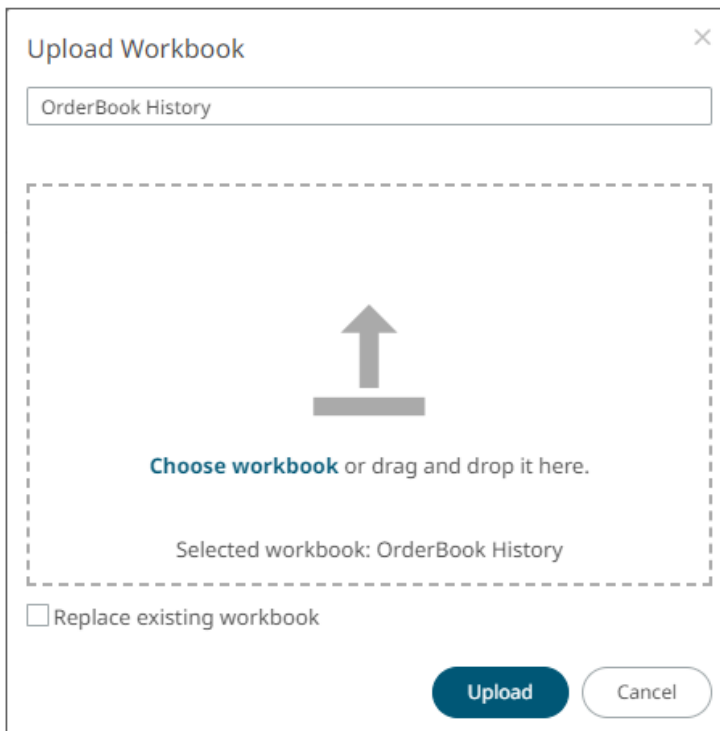
3. You can opt to rename the workbook.

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

5. Click [Upload] .

    You will be notified once the workbook is uploaded.

    Upload Workbook   ✕

    OrderBook History

                              ✓

                        Upload complete

    ☑ Replace existing workbook

                                    Upload   Cancel
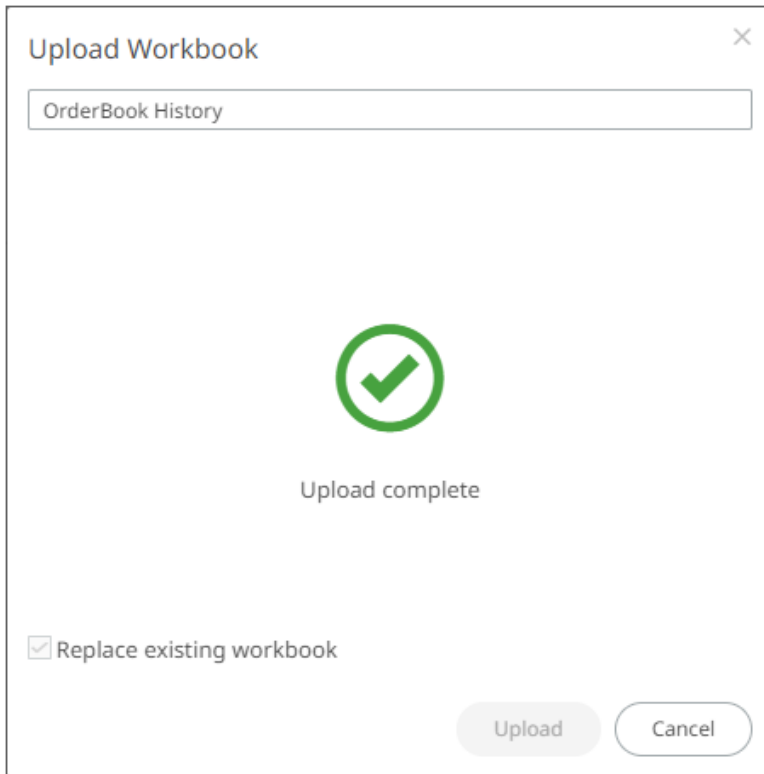
    The workbook is added and displayed.

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.
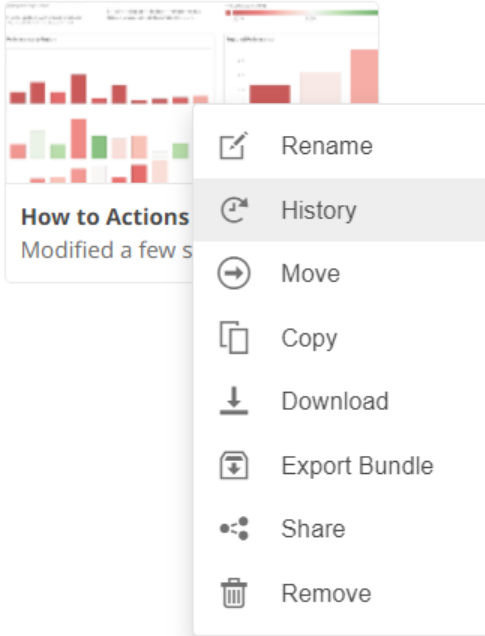
The *History of Workbook <Name>* dialog is displayed with the current version of the workbook indicated.
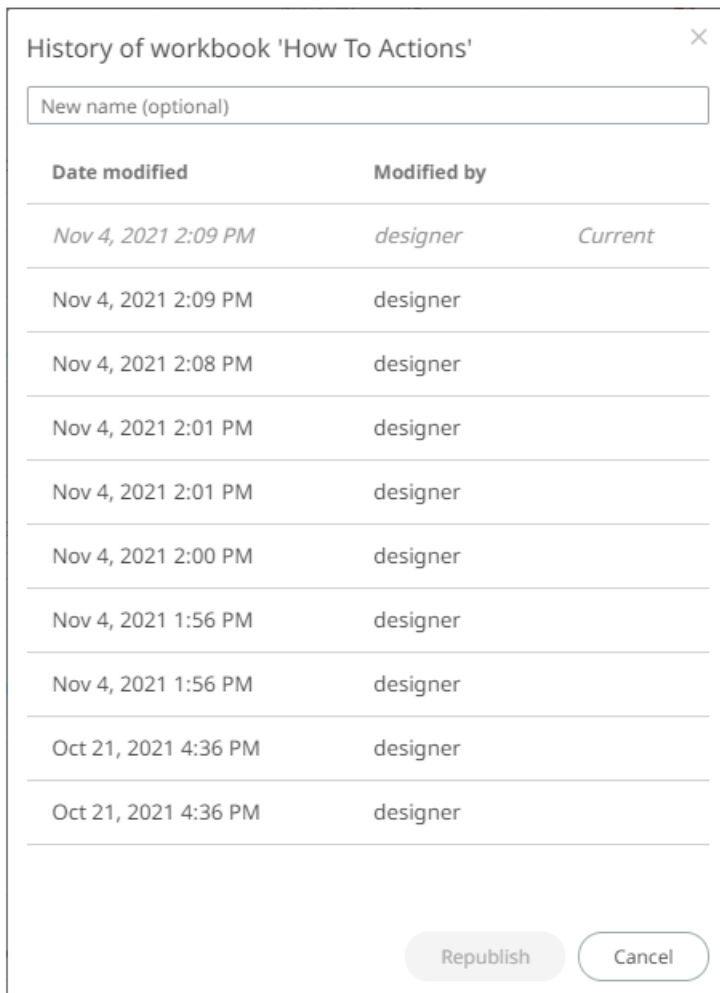
Sort the archival list either through the *Date Modified* or *Modified By* by clicking on the ↓ or ↑ button.

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 **Republish** . A notification message displays.

4. Click **Yes** .

A confirmation message displays.

The earlier version of the workbook was successfully republished.

OK

5. Click **OK** .

The republished workbook version is added in the history list.

History of workbook 'How To Actions'   ×

New name (optional)

Date modified	Modified by	
Nov 4, 2021 2:22 PM	_designer_	_Current_
Nov 4, 2021 2:09 PM	designer	
Nov 4, 2021 2:09 PM	designer	
Nov 4, 2021 2:08 PM	designer	
Nov 4, 2021 2:01 PM	designer	
Nov 4, 2021 2:01 PM	designer	
Nov 4, 2021 2:00 PM	designer	
Nov 4, 2021 1:56 PM	designer	
Nov 4, 2021 1:56 PM	designer	
Oct 21, 2021 4:36 PM	designer	
Oct 21, 2021 4:36 PM	designer	

Republish   Cancel

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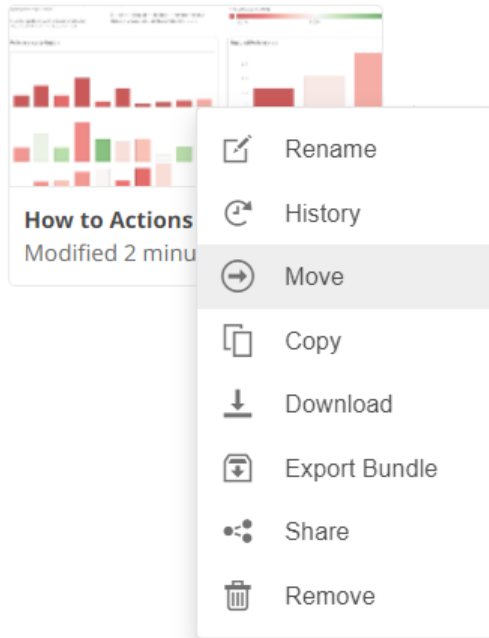
The *Move Workbook* dialog displays with the folder or subfolders the user is allowed to move the workbook.

2. Select the folder or subfolder.

3. Click ![Move] .
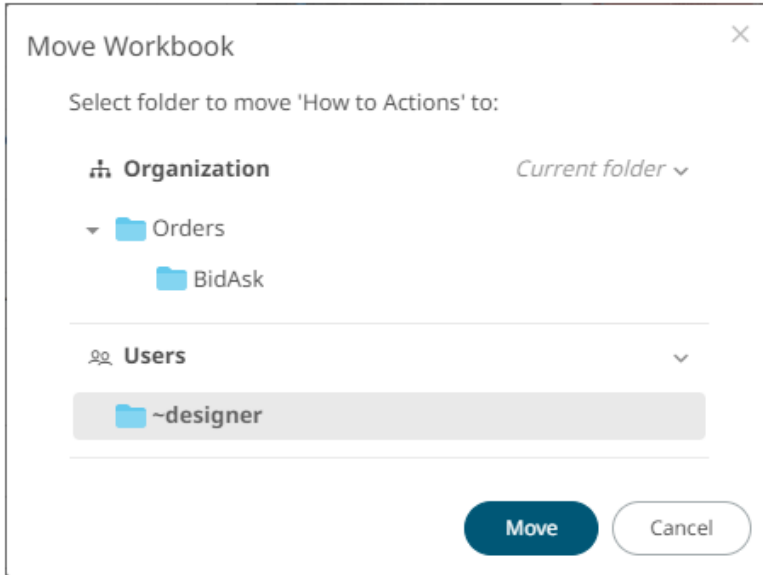
The workbook is moved and displayed on the selected folder.

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.
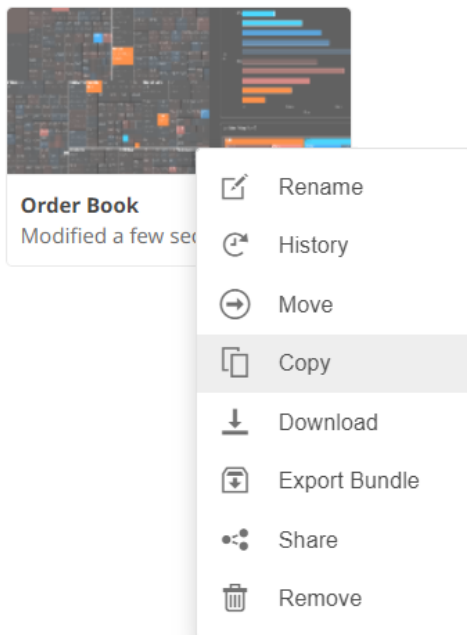
The *Copy Workbook* dialog displays with the folder or subfolders the user is allowed to copy the workbook to.
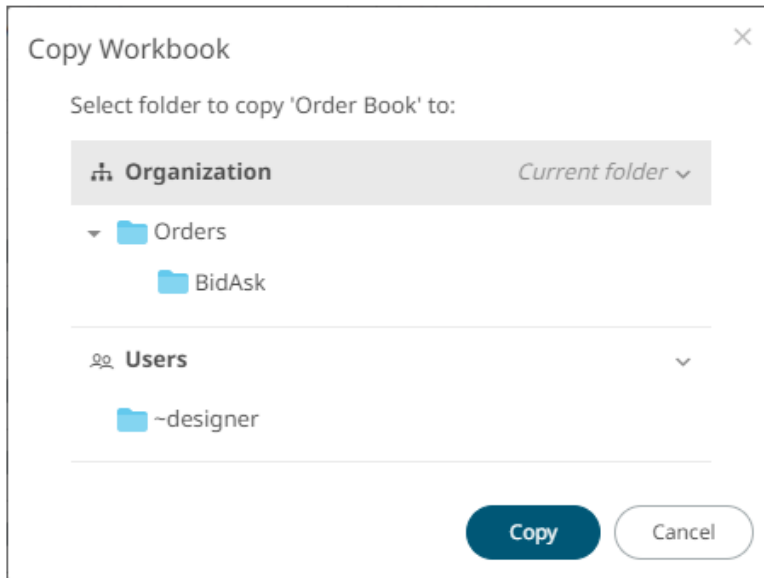
2. Select the folder or subfolder.
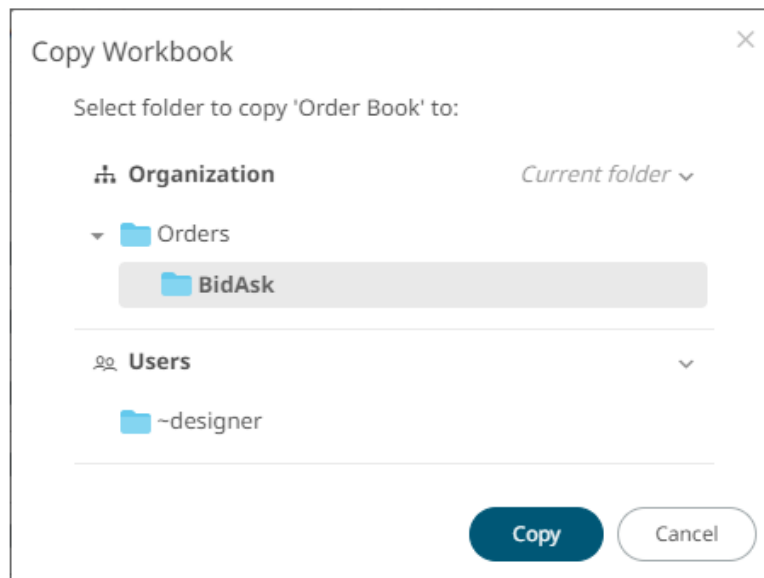
3. Click .

The workbook is copied and displayed on the selected folder.

Sharing Workbooks

You can generate a shareable link of a workbook.

Steps:

1. Right-click on a workbook and select **Share** in the context menu.
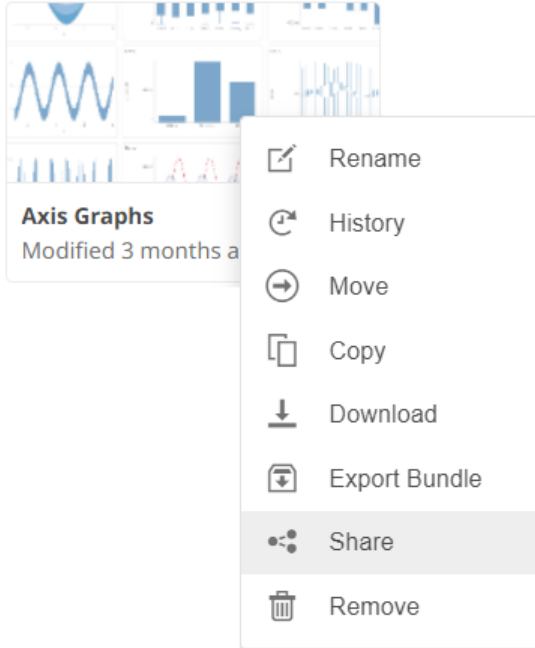
The *Share Workbook* dialog displays.
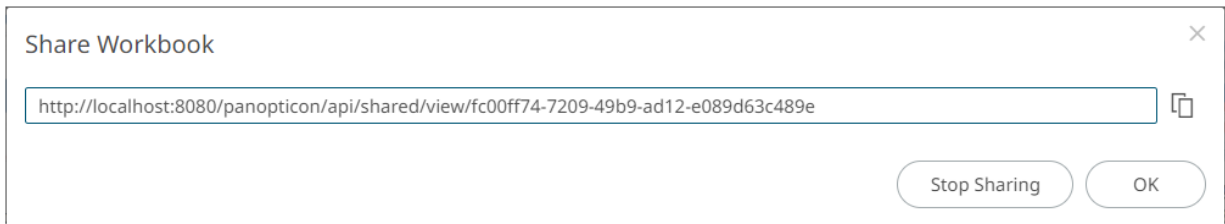
2. Click **Copy Shared Link** to copy the generated workbook link that can be shared with other users.

3. Click OK .

    The workbook displays a **Shared Link** icon.
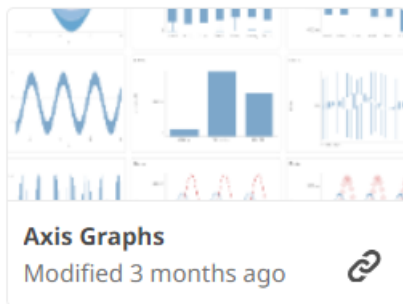
You can click the **Shared Link** icon to copy the link.

4. To stop sharing the workbook, right-click on the shared workbook and select **Share** in the context menu.

    The *Share Workbook* displays.

5. Click [Stop Sharing] .

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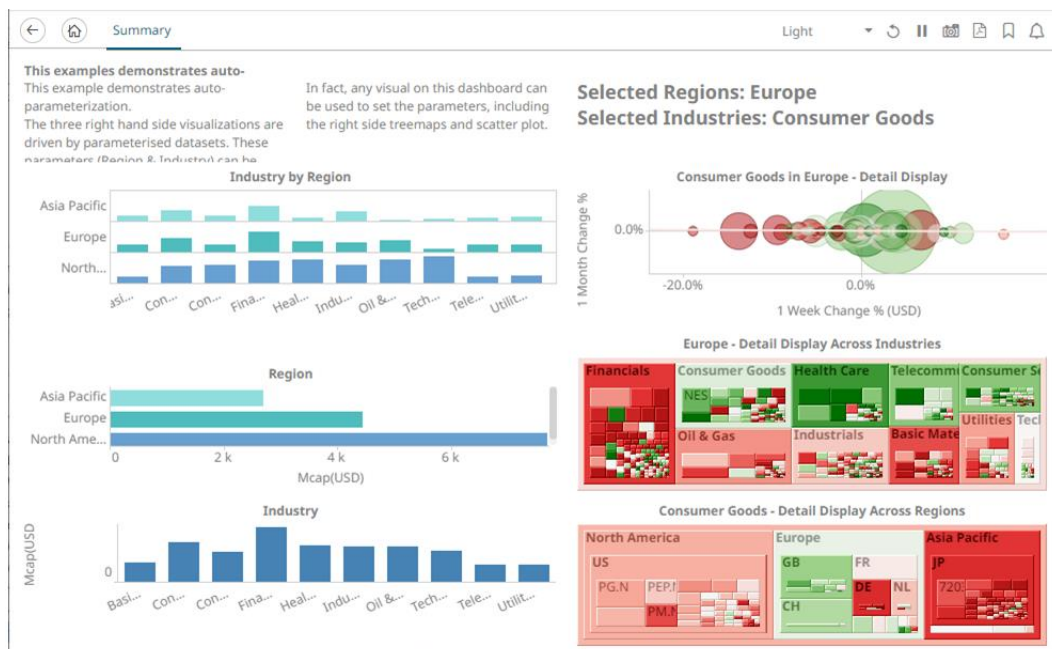
This workbook can also be displayed on the web browser using this URL:

```
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"

[11] DATA LIBRARY

The *Data Library* page allows creation and management of reusable data tables outside workbooks. Data tables from the data library can be used by multiple workbooks server wide.

Some important concepts to remember:

❑ Bringing data inside workbooks is done through data tables.

❑ A data table contains metadata for data source connection settings, data query, schema definition, calculated columns, transforms, etc.

❑ One data table can use only one data connector to connect to a data source.

❑ Two or more data tables can be joined to create a new joined data table.

❑ Data table permissioning and sharing between users or groups is done similarly like workbooks (i.e., using folder tree).

❑ There are four types of data tables. The first three reside at the Data Library.

Data Table Type	Description
Data Store	Users can opt to store data closer to Panopticon server in an embedded database.
Live	Direct connection to source data.
Joined	Two or more different types of data tables joined together creating a new data table.
Uploaded	Uploaded files through any of the file connectors, that consequently becomes a Data Store, after importing to data store.

The *Data Library* page is composed of the following sections.
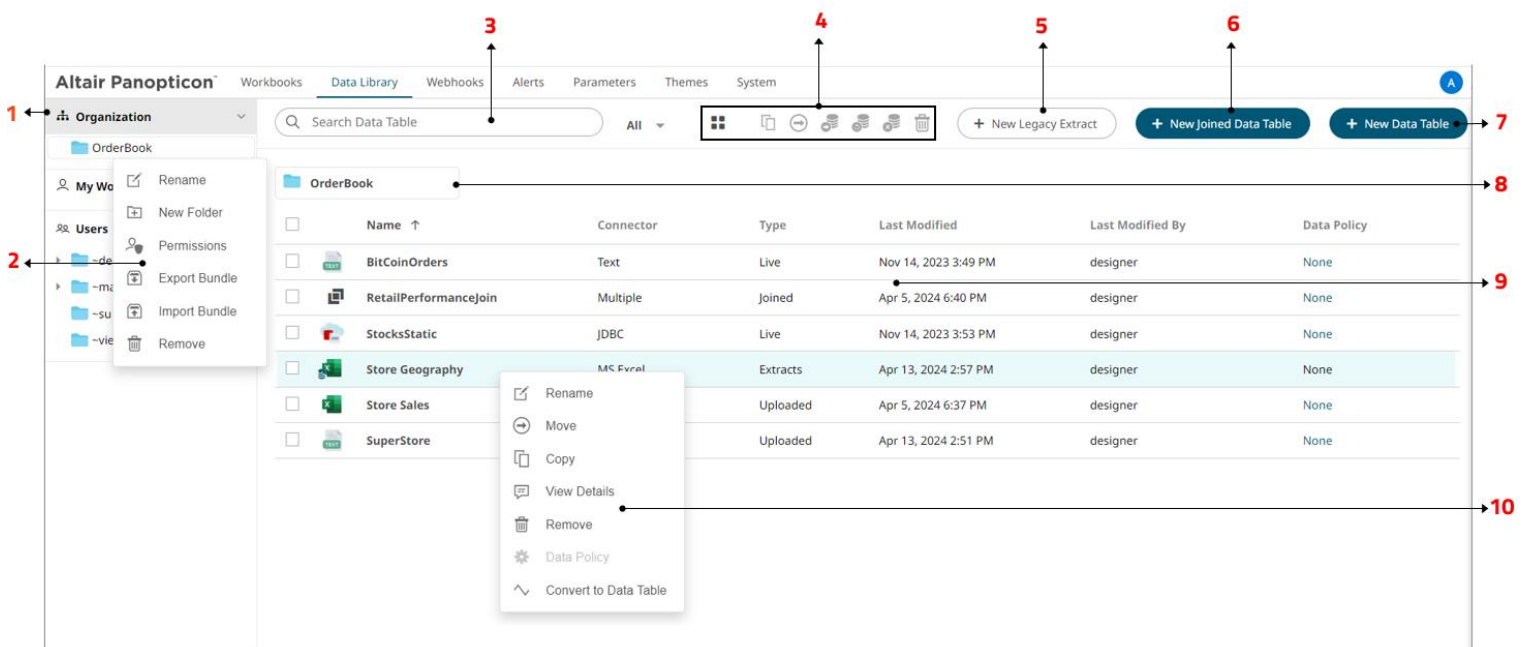
Data Library Page Sections and Descriptions

Section	Description
1	**Folders** List of folders where data tables can be saved, exported, or imported.
2	**Folder Context Menu** Allows you to: • Create a data table and joined data table • Assign folder permissions on your workspace • Import or export data table bundles • Create, rename, or remove folders
3	**Search Data Table** Entering text will filter data tables which can include: • Those that are available in data store • Live data tables • Joined data tables • Extracts
4	**Toolbar** Allows you to: • Display the data tables list either on List View or Grid View • Copy or move data tables to other folders • Import data table to data store • Clear and import data table to data store • Delete data tables
5	**New Legacy Extract** Allows accessing data 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.
6	**New Joined Data Table** Allows you to join data tables created in the data library.
7	**New Data Table** Allows you to create a data table.
8	**Folders List** Available folders.
9	**List of Data Tables and Data Extracts** Data tables and data extracts created in the data library.
10	**Data Table Context Menu** Allows you to: • Export data table bundles • Copy or move data tables to other folders • Rename or remove data tables

Section	Description
	• View details of the data table • Set the data policy for data tables in the Data Library • Convert data extract to data table and import into data store

For more information on using this page, see *[4] The Data Library Page* section in the Web Authoring Guide.

SETTING UP DATA STORE

To be able to use data store, you would need to set the following properties. By default, Panopticon supports MonetDB, so default values correspond to it.

Also. MonetDB JDBC driver is packaged with Panopticon server. For other data store types, refer to *JDBC Driver Installation* section.

Property	Data Store
Attribute	`datastore.connection.schema`
Description	Name of the database schema to be used for creating or managing objects inside database.
Default Value	**dbo**
Property	Data Store
Attribute	`datastore.type`
Description	Controls which data store connector should be used. Valid values are **MonetDB**", **MSSQLServer** and **PostgreSQL.**
Default Value	**MonetDB**
Property	Data Store
Attribute	`datastore.connection.jndi`
Description	JNDI resource name for the connection e.g., **jdbc/MyDB**. More details on how to configure JNDI is at *JNDI Connection Details* section.
Default Value	
Property	Data Store
Attribute	`datastore.connection.url`
Description	JDBC connection URL for the database e.g., **jdbc:monetdb://localhost:49153/PanopticonDataStore** This property value is discarded If `datastore.connection.jndiproperty` is set.
Default Value	
Property	Data Store

Attribute	`datastore.connection.driverclassname`
Description	Fully qualified Java class name of the JDBC driver used for the connection.
Default Value	**`org.monetdb.jdbc.MonetDriver`**
Property	Data Store
Attribute	`datastore.connection.username`
Description	Username for the connection. Only required when using connection URL.
Default Value	
Property	Data Store
Attribute	`datastore.connection.password`
Description	Password for the connection. Only required when using connection URL.
Default Value	

CACHING

Panopticon Real Time supports five levels of caching:

❑ Data Store

❑ 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 unnecessary 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.

[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 publicly, 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 webhooks. 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, like 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.
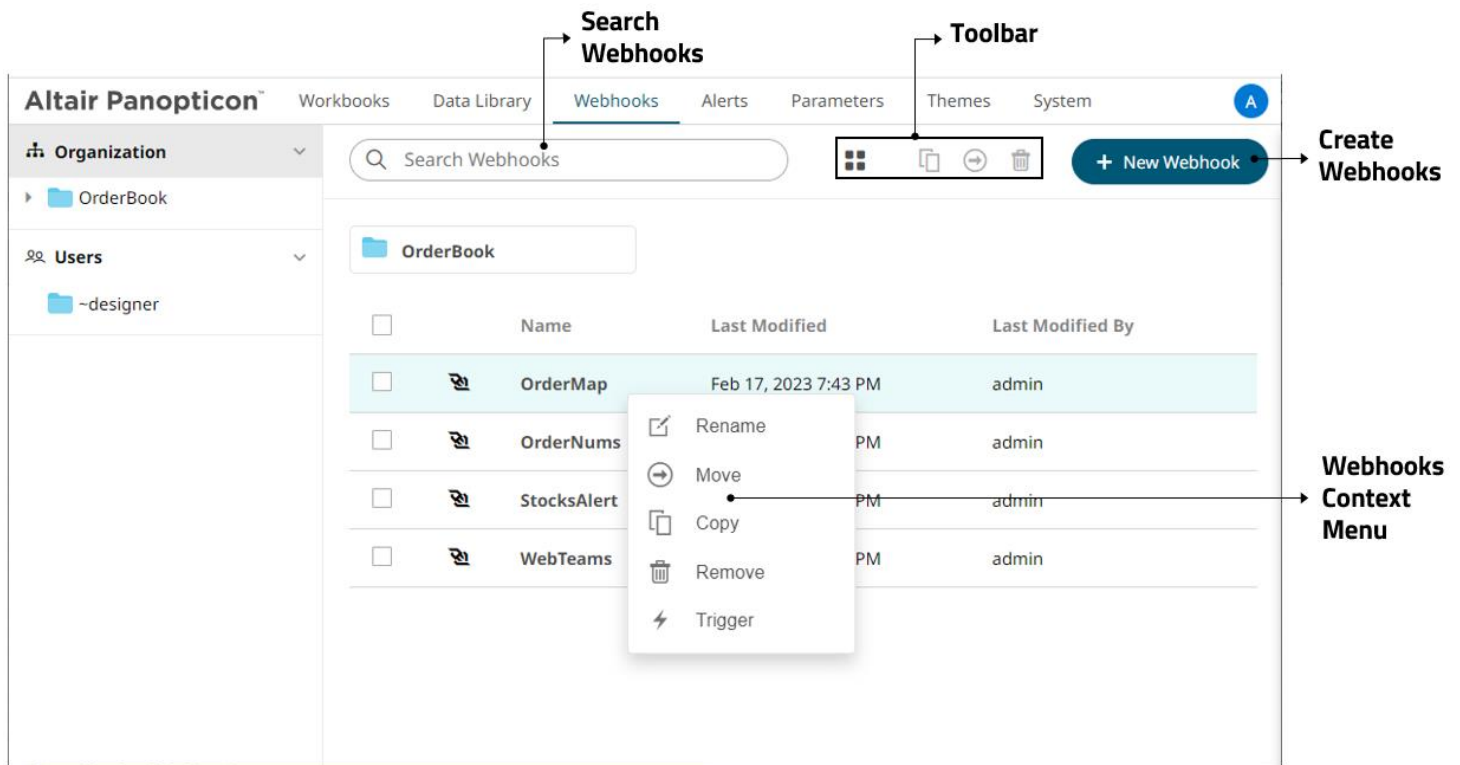
Property	Description
Search Webhooks	Entering text will filter the webhooks.
Toolbar	Allows copying, moving, and removing of webhooks. Also, to display the webhooks list either on *List View* or *Grid View*.

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.
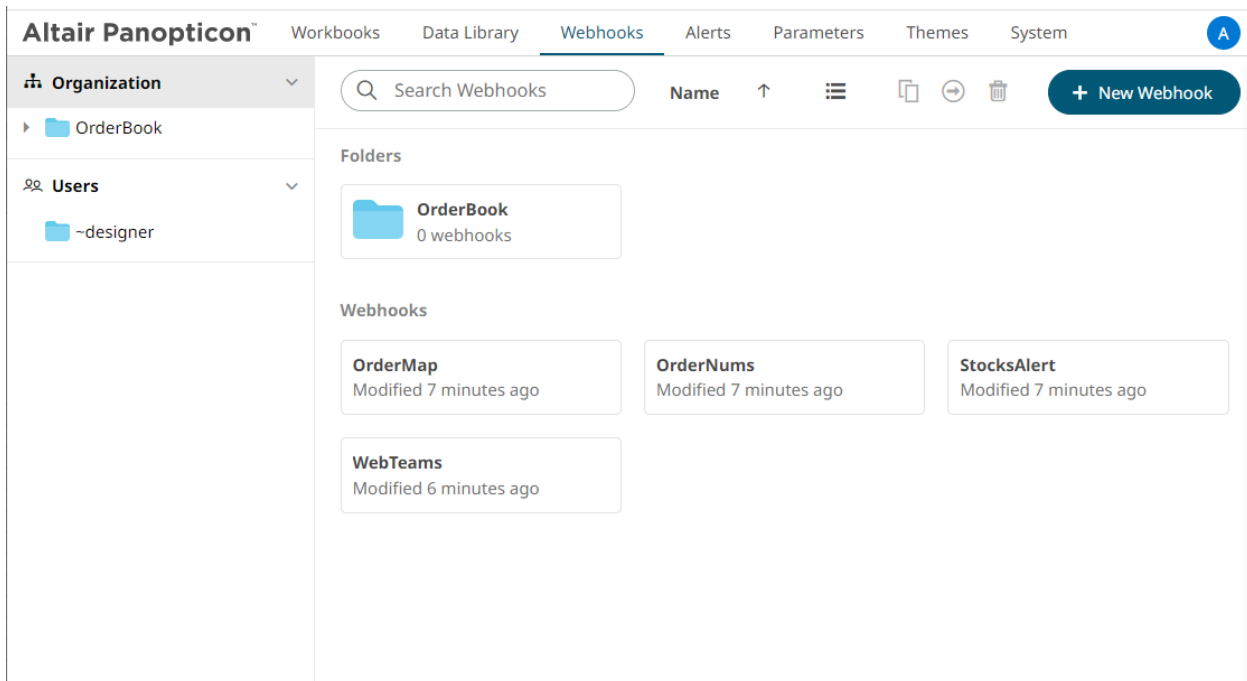
Or click **List View** , the folders and webhooks are displayed in a standard listing.
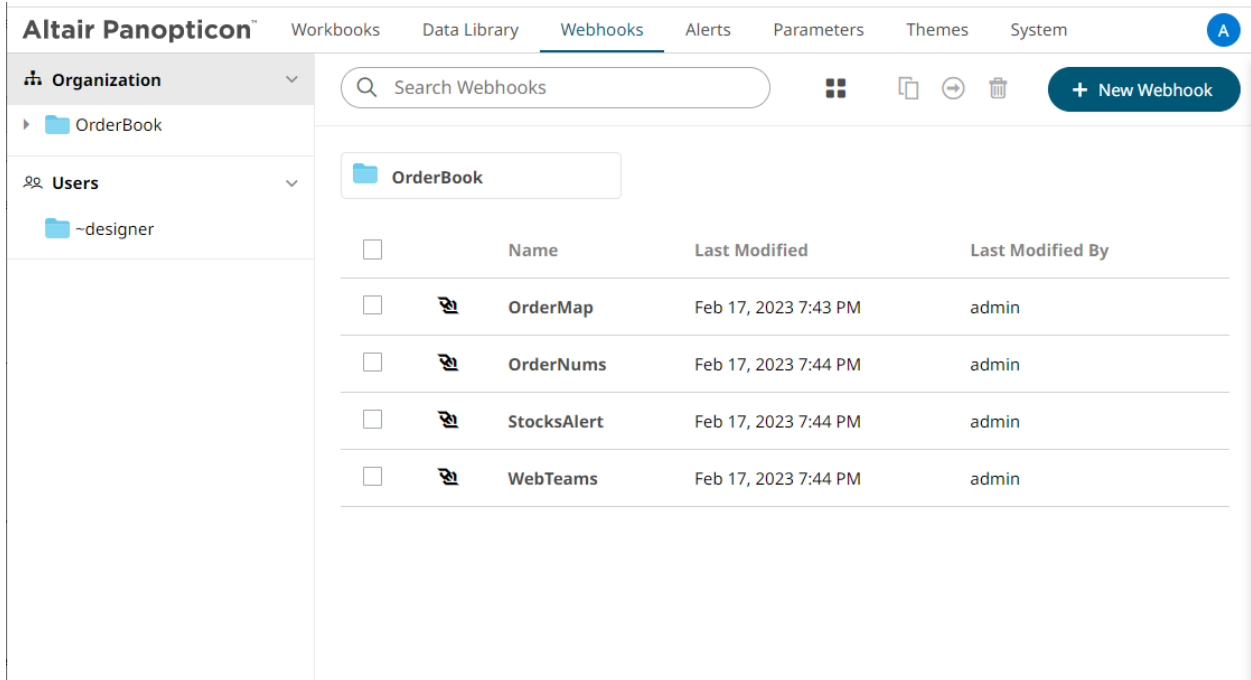
On either display view style, clicking on a webhooks title or thumbnail displays the *Webhooks* page.

Searching for Webhooks

On the *Webhooks* tab, to search for a particular webhook, enter it in the *Search Webhooks* box.
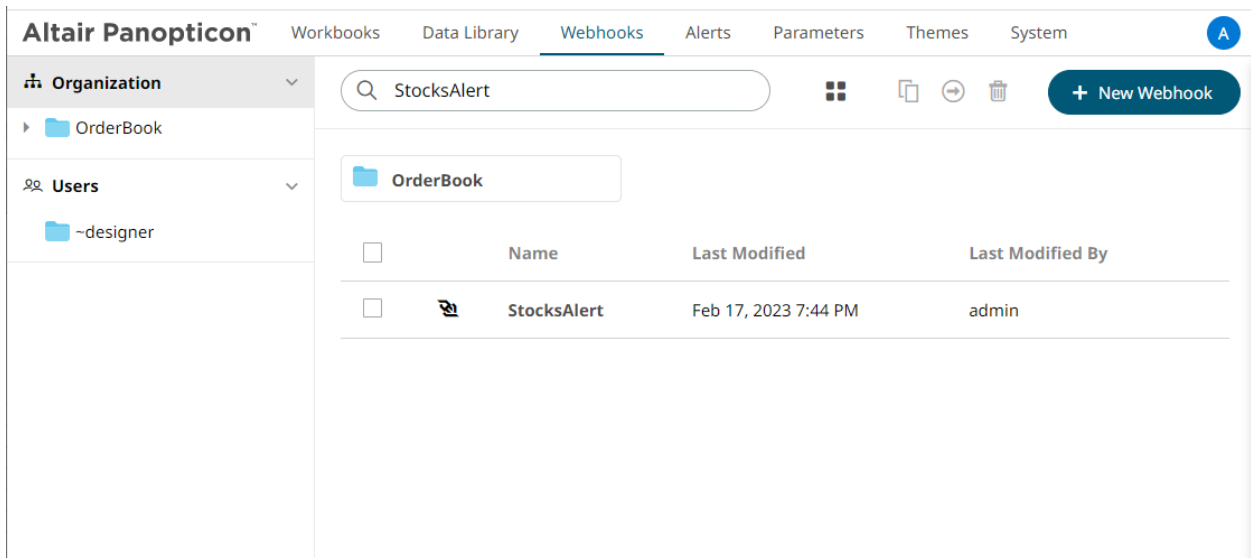
You can also enter one of more characters into the *Search Webhooks* box then click **Enter**. The suggested list of webhooks that matched the entries will be displayed.
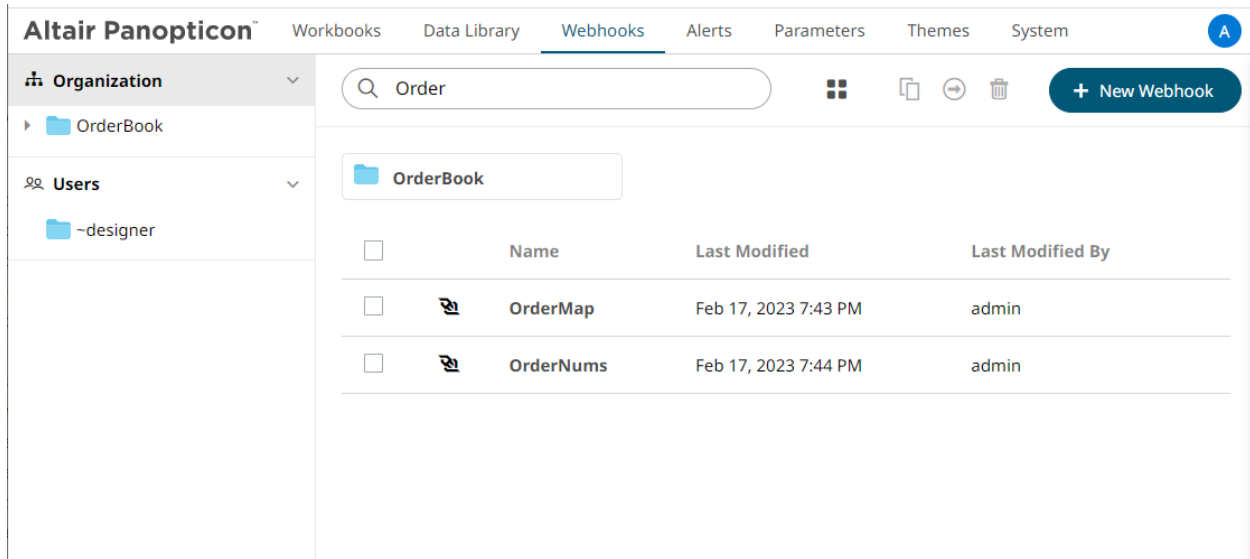
Click on a webhooks to open and display.

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 .

   The new webhook is displayed on the *Webhook* page.
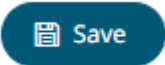
If you want to change the name of the webhook, just enter a new one then click ✓ .

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 **📄 Save** to save the new webhook.

5. You may opt to click **⚡ Trigger** 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:
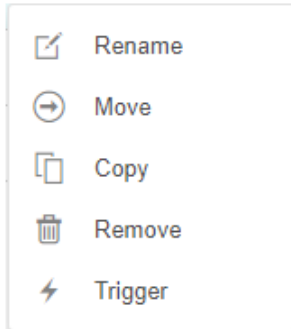
❏ Context menu

Webhook Context Menu
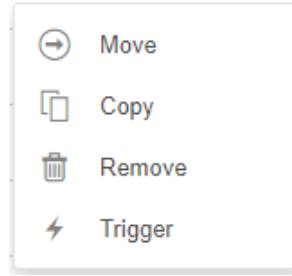
Webhooks Folder Context Menu

❏ 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**.
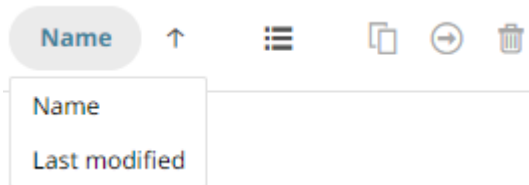
- Name
- Last Modified

Then click the *Sort Order*:

- Ascending

- Descending

❑ click on the **Name**, **Last Modified**, or **Last Modified By** column header of the *List View.*
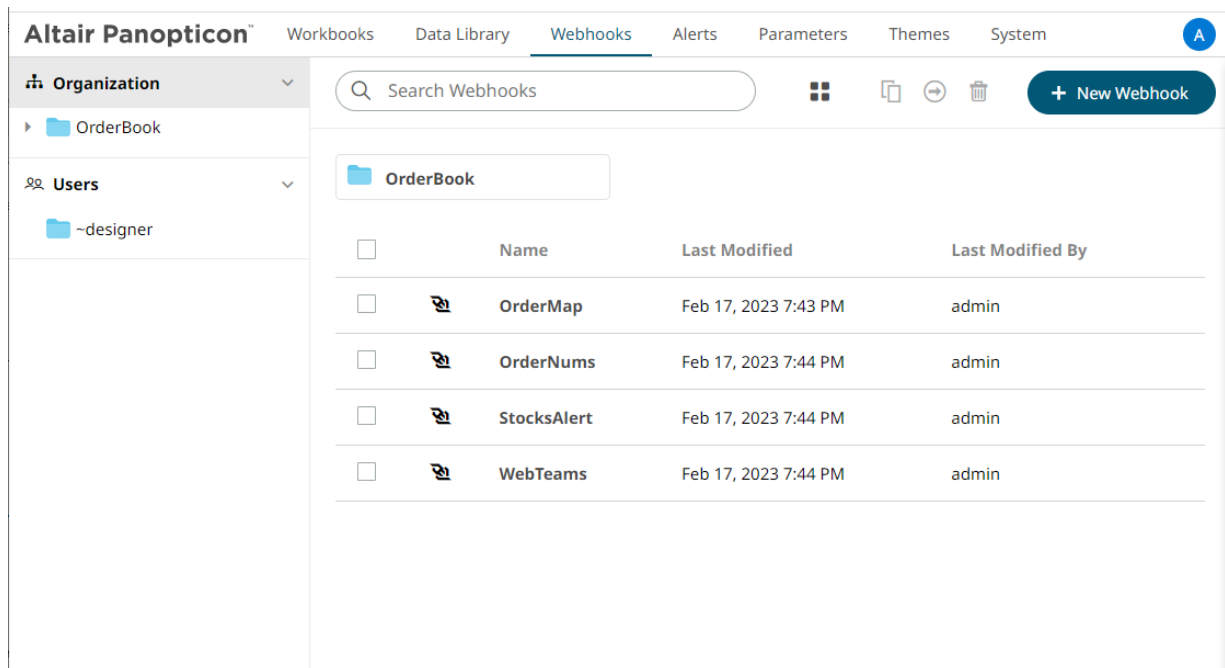
Then click the *Sort Order*:

- ↑ Ascending
- ↓ Descending

Renaming a Webhook
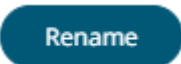
Steps:

1. Right-click on a webhook then select **Rename** on the context menu.

☑ Rename
⊕ Move
⧉ Copy
🗑 Remove
⚡ Trigger
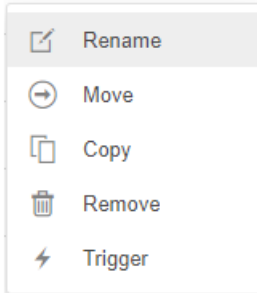
The *Rename Webhook* dialog displays.

Rename Webhook   ×

StocksAlert

Rename   Cancel
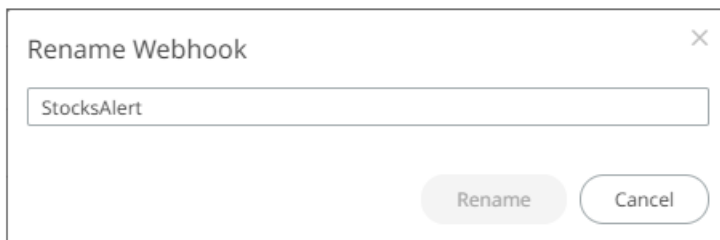
2. Enter a new name then click **Rename** .

Moving Webhooks

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

Steps:

1. Select the checkbox of one or several webhooks either on the *Grid View* or *List View*.
2. Then select either:

- **Move** ⊕ icon on the toolbar, or
- **Move** on the content menu.
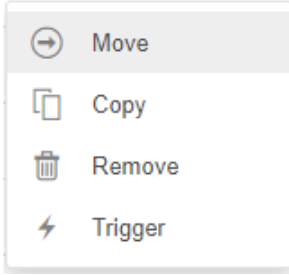
The *Move Webhook* dialog displays with the folder or subfolders that the user is allowed to move the webhooks. Select the folder or subfolder.

3. Click [Move] .
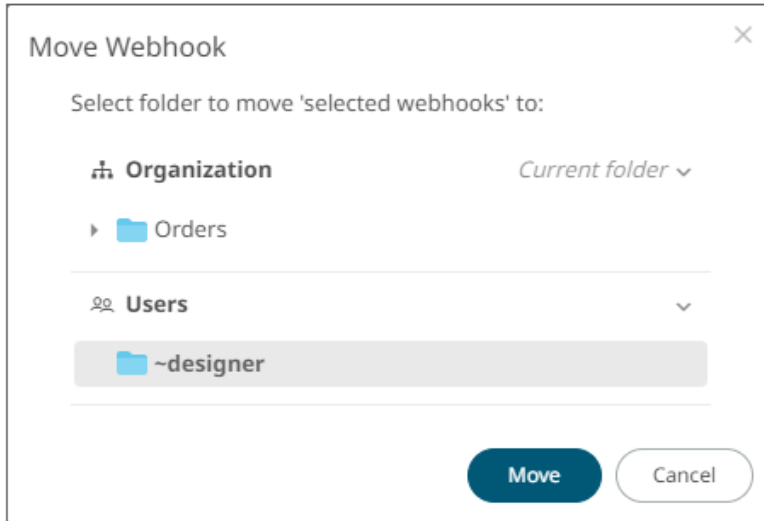
The webhooks are moved and displayed on the selected folder.

Copying Webhooks

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

Steps:

1. Select the checkbox of one or several webhooks either on the *Grid View* or *List View*.
2. Then select either:

    • C**opy** [icon] icon on the toolbar, or
    • **Copy** on the content menu.
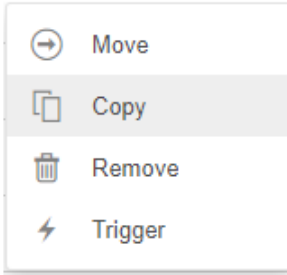
The *Copy Webhook* dialog displays with the folder or subfolders the user is allowed to copy the webhooks to. Select the folder or subfolder.

3. Click **Copy** .
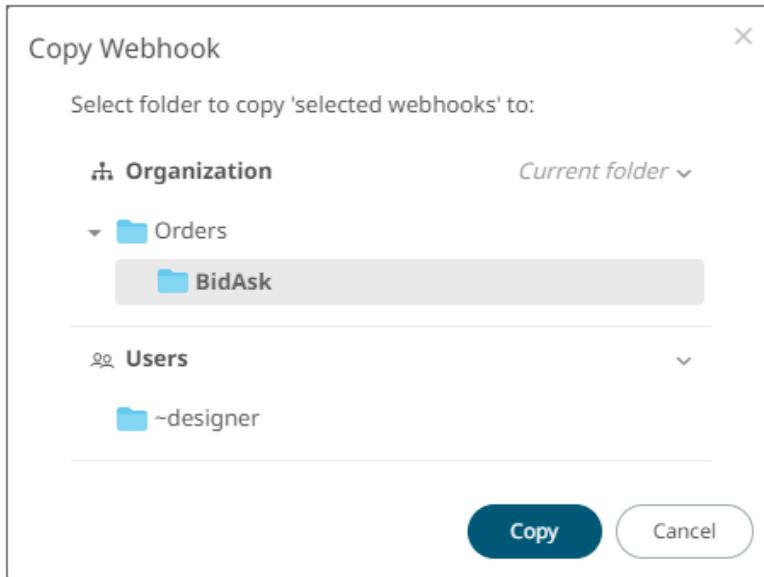
The webhooks are copied and displayed on the selected folder.

Deleting Webhooks

Users with an Administrator role can remove webhooks.

Steps:

1. Select the checkbox of one or several webhooks either on the *Grid View* or *List View*.

2. Then select either:

    • **Remove** 🗑 icon on the toolbar, or
    • **Remove** on the content menu.
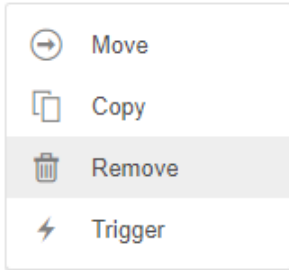
A notification message displays.
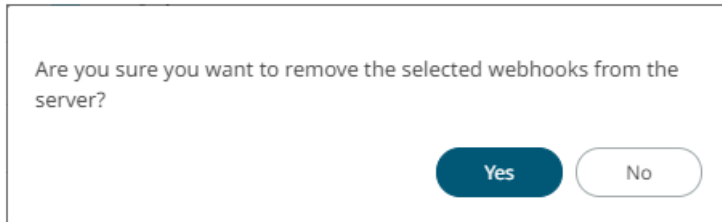
3. Click to remove.

Triggering Webhooks

To trigger a webhook, right-click on it and select **Trigger** on the context menu.
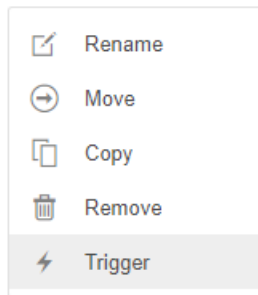
Any parameter in the request body will be replaced by its value when triggering the webhook request.

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:
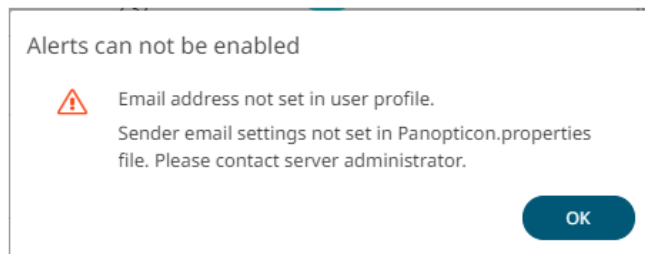
In addition, you can also set the `alert.creation.only.by.administrators property` to **true** for only the Administrators to create alerts.

SETTING UP ALERTS

Alerts can be defined against:

❑ Streaming data sources (including CEP Engines and message queues)

❑ Periodically refreshed data sources (like REST services, files on HTTP URLs, databases, Python or R)

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	The alert criteria will be evaluated across the full series span along the axis, not on the latest value alone, when using an X-Axis or Y-Axis visualization and the domain axis is using any of the following: • Text column and is set to Axis Data: Series • Numeric column • Time column while the data table does not have a Timeseries Transform applied to it In contrast, the alert criteria will be evaluated data point by data point as new data is received when using any of the following: • None-series graphs (such as the Treemap and the Scatterplot) • Axis graphs that use a nominal text axis or a Timeseries Transformed data table

IMPORTANT	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 user icon.
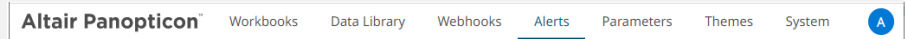
The *Profile* pane displays the name of the user and the role.
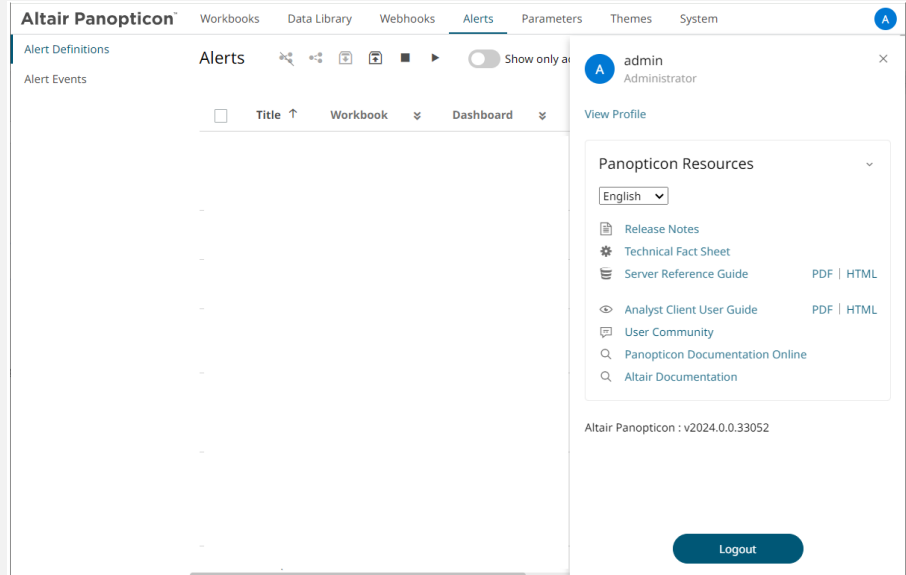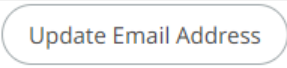
2. Click *View Profile*.

The *User Profile* page displays.
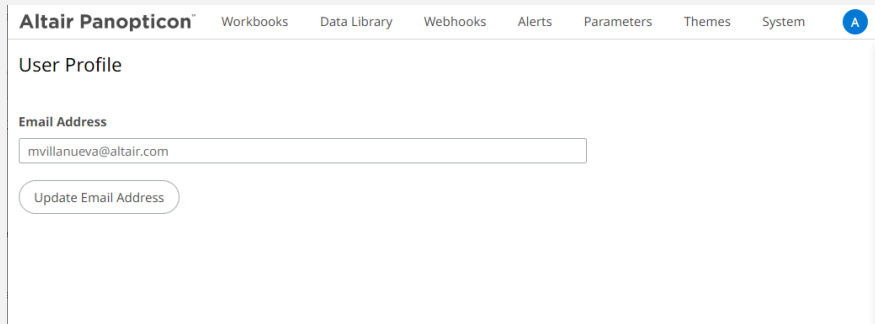
3. Enter the *Email Address*.

4. Click Update Email Address.

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.
Search	Search for columns.
Criteria	Criteria set of the alert. Can be multiple, in which case all criteria sets are evaluated in parallel, each triggering respective alerts. Additional criteria sets can be added by clicking . You can also right-click a **Criteria** tab and select **Rename** to rename the criteria or select **Remove** to delete. The **Remove** option is disabled when only one criteria set is available.
Variable	Available variable columns in the visualization where the alert is set.

Condition	Allows setting the following *Limit* of all the available numeric variables in the visualization:
	• Upper or Equal To (<=)
	• Lower or Equal To (>=)
	• Upper values (<)
	• Lower values (>)
	• Between – values between the *Lower* and *Upper* values
	For text variables, there are four types of conditions:
	• Equals - The string is equal to another string, e.g., Country=Sweden
	• Not Equals – The string is not equal to another string
	• 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.
CC	CC mailing groups that will receive the alert, separated by a comma.
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).

	☑ Sound ▼ ☐ Webhook alarm_clock beep_short ☐ Active Hours bell_ping_1s Default is **None**.
[Webhook](#)	Webhooks that will be 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 **OK** . The new alert is added on the *Alerts Definition* page.

NOTE	When creating alerts for grand total, ensure that no breakdown is set.

WORKING WITH ALERTS

Working with alerts can be done either on the *Alerts* page or on the *Alerts* pane of a workbook.

Alerts Page

This page is accessible through the **Alerts** tab with two sections: **Alert Definitions** and **Alert Events**.
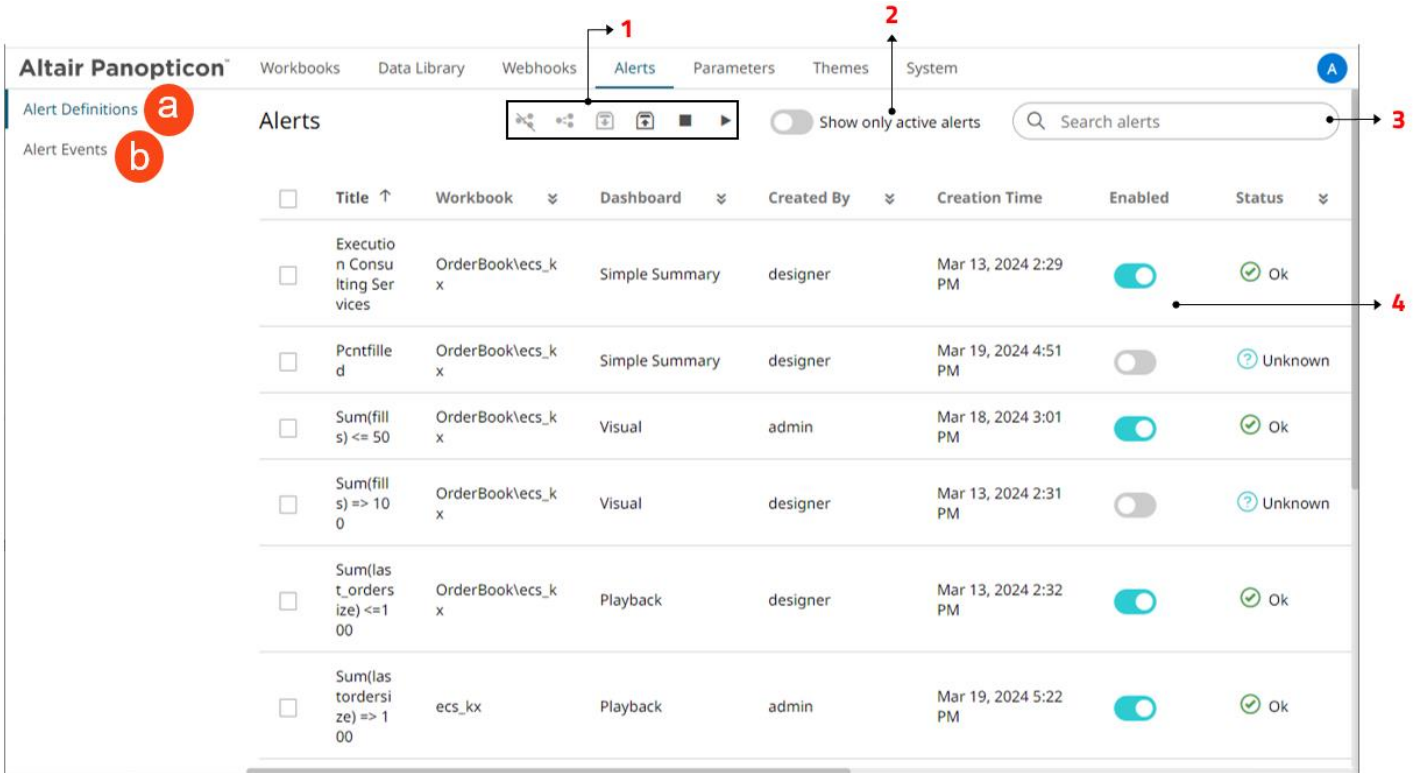
Clicking **Alert Definitions** (a) displays the list of alerts.
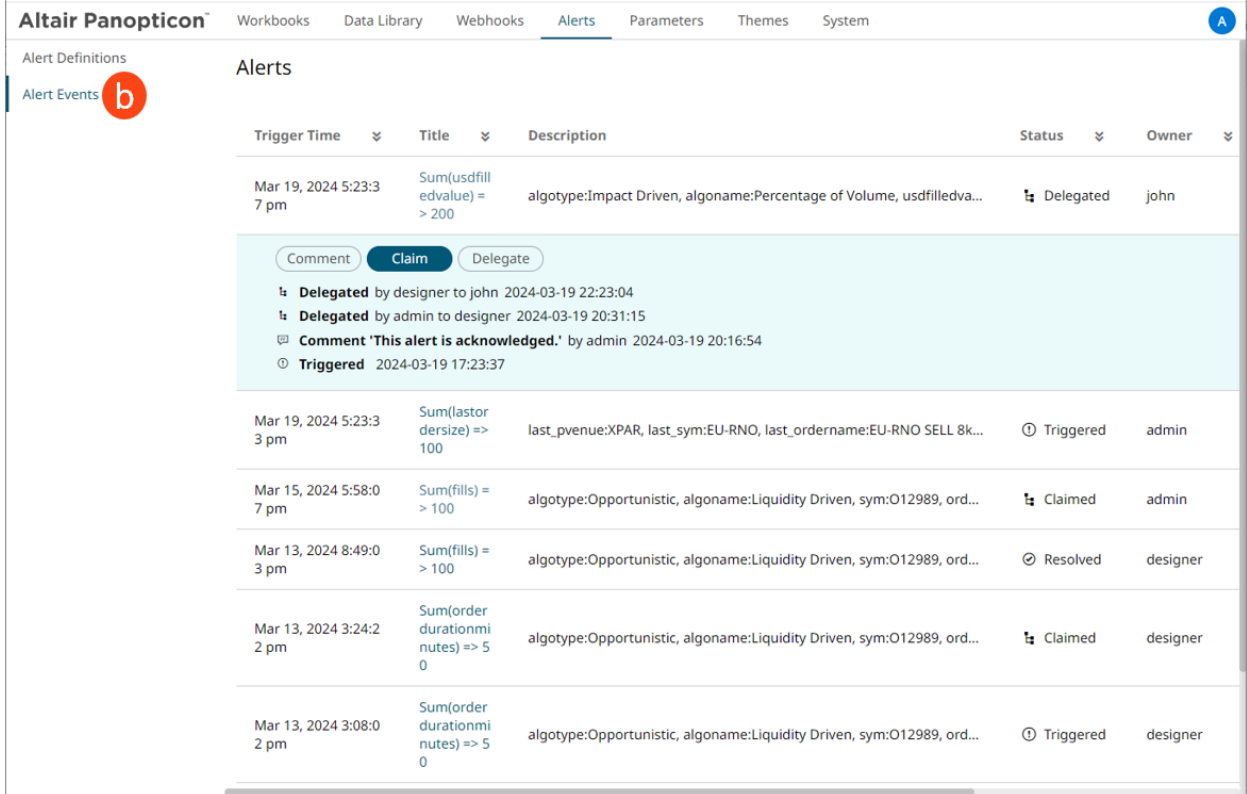
Option	Description
1	**Alert Definitions Toolbar** Allows you to <u>import</u>, <u>export</u>, <u>deactivate all</u>, or <u>activate all</u> alerts. As an administrator, you are also allowed to <u>share</u> or <u>stop sharing</u> alerts.
2	**Show Only Active Alerts** Tap the **Show only active alerts** slider to turn it on. Only the active or enabled alerts are displayed on the *Alert Definitions* list.
3	**Search Alerts** Entering text will filter the alerts. You can also enter one or more characters into the *Search Alerts* box and the suggested list of alerts that matched the entries will be displayed. You can also click ⌄ of a column to display a multi-select checkbox list. By default, all checkboxes are selected (**Select All**). Selecting or deselecting items in the list controls the filter.
4	**List of Alerts** Displays the *Title*, *Workbook*, *Dashboard*, *Created By*, *Creation Time*, *Enabled*, *Status*, *Shared*, *Times Triggered*, *Sent Emails*, *Notifications*, and *Triggered Webhooks* settings of the alerts. By default, the list of alerts is sorted by *Title* in 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, Shared, Times Triggered, Sent Emails, Notifications, or Triggered Webhooks* columns. The icon beside the

	column that was used for the sorting will indicate if it was in ascending or descending order.
	You can also:
	• Click 🗑 to delete an alert.
	• Tap the **Enabled** slider of an alert to turn it on.

Clicking **Alert Events** 🅑 displays the details of all triggered events of activated alerts.

Property	Description
Trigger Time	The Date/Time when the alert was triggered.
Title	Title of the alert. Click a **Title** link to go to the workbook where the alert was triggered.
Description	Description of the alert.
Status	Status of the alert. Can be any of the following: • Triggered • Resolved • Claimed • Delegated
Owner	Owner of the alert.
Workbook Name	The workbook name where the alert was set.

Dashboard	The dashboard name where the alert was set.
Alert Events Properties	Displays the workflow changes. Also allows you to: • Add a comment • Resolve an alert • Delegate an alert • Claim an alert

You can also do any of the following options:

❑ Click ↓ or ↑ of a column title to sort the list.

❑ Click ⩢ of a column to display a multi-select checkbox list. By default, all checkboxes are selected (**Select All**). Selecting or deselecting items in the list controls the filter.

Alerts Pane

This pane is accessed by clicking the **Alerts** 🔔 icon in the workbook or selecting an alert in the pop-up notification. Like the *Alerts* page, there are two screens: **Alert Events** and **Alert Definitions**.

Clicking **Events** displays the list of alert events.

Option	Description
1	**Search Alerts** Entering text will filter the alert events.
2	**Alert Event State** Allows you to view any of the following alert event states: • All Open • My Open • Shared Open • All Resolved • Assigned to me
3	**Alert Event Properties** Displays the following information: • Title • Breakdown details • Dashboard where the alert is defined • Date/Time of alert creation • Workflow changes Also allows you to: • Add a comment • Resolve an alert • Delegate an alert • Claim an alert

Clicking **Definitions** displays the list of alerts (**Shared Alerts** or **My Alerts**).

Option	Description
1	**Stop Sharing**
	Allows you to stop sharing the alert.
2	**Shared Alerts**
	Alerts that are shared by administrators.
	Allows you to:
	• Open the alert where you can opt to edit, activate/deactivate, or delete it
	• Tap the **Enable** slider to activate the alert
	NOTE: Non-administrators are not allowed to deactivate a running shared alert.
3	**My Alerts Toolbar**
	Allows you to share, import, export, or delete alerts.
4	**My Alerts**
	Allows you to:
	• Open the alert where you can opt to edit, activate/deactivate, or delete it.
	• Activate/deactivate the alert.

Activating or Deactivating All Alerts

Alerts can be activated or deactivated in one click.

To activate all deactivated alerts, click **Activate All** ▶ . All of the deactivated alerts are activated.

Activating Alerts

Number of updated alerts: 3

OK
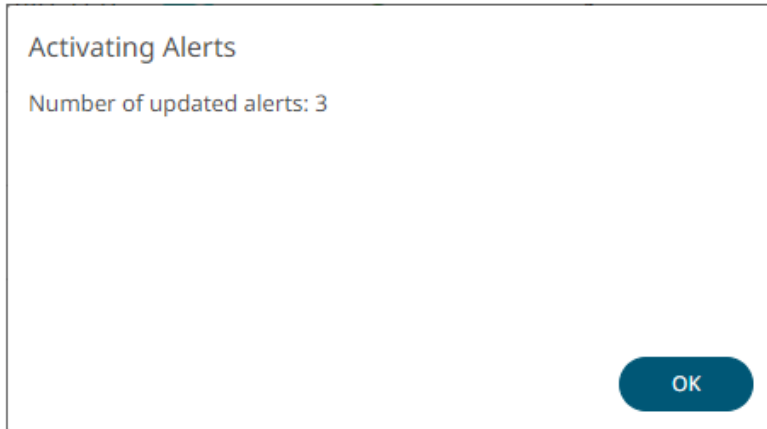
To deactivate all activated alerts, click **Deactivate All** ■ . All of the activated alerts are deactivated.

Deactivating Alerts

Number of updated alerts: 8

OK
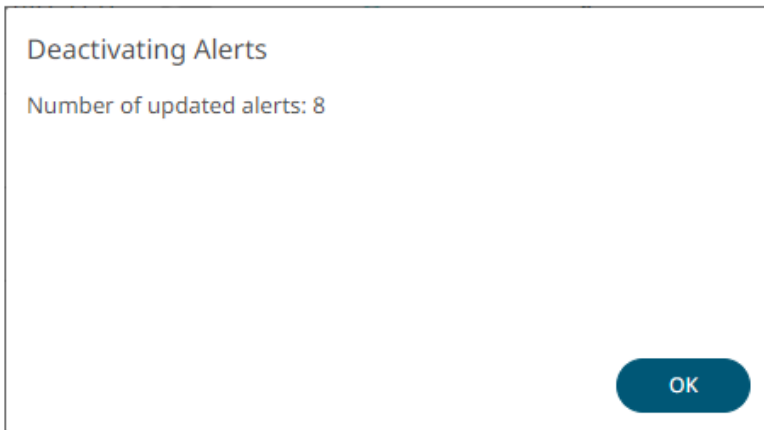
Click OK to close the dialog.

Importing Alerts

Allows you to import alerts shared by other others.

Steps:

1. On the *Alert Definitions* section of the *Alerts* page or *Alerts* pane, click the **Import Alerts** icon.
   The *Upload Alert* dialog displays.
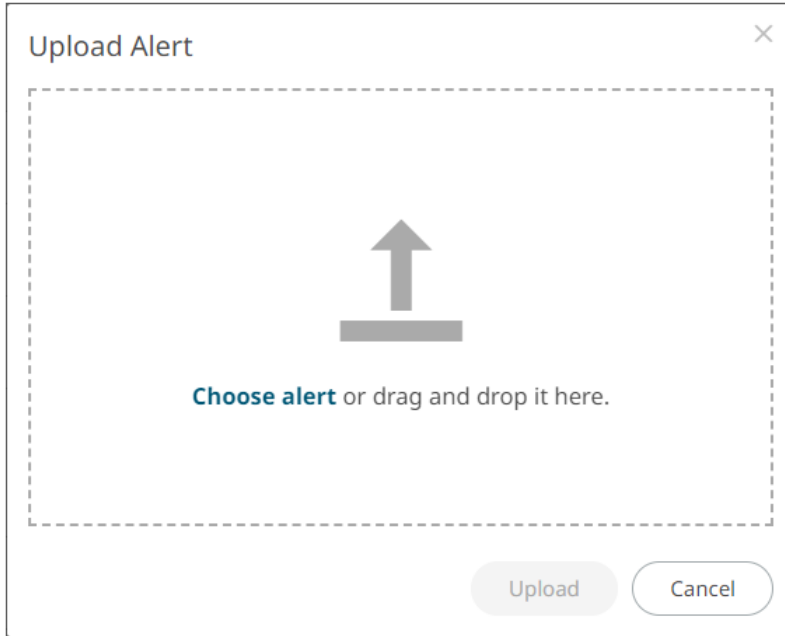
2. To upload an alert, do one of the following:

   - Drag the file from your desktop and drop on the dialog, or

   - Click **Choose alert** and then browse and select one on the *Open* dialog that displays.

3. Click .

   A notification displays once the alert is uploaded.

   Click to close the dialog. The uploaded alert is added in the list.

Exporting Alerts

You can download a copy of any of the alerts.

Steps:

1. On the *Alert Definitions* section of the *Alerts* page or *Alerts* pane, select the checkbox of the alerts you want to export.

2. Click **Export Alerts** icon.

   The selected alerts are downloaded.

Viewing and Managing Alerts for Non-Administrator Users

There are three tabs that are available for non-Administrator users:
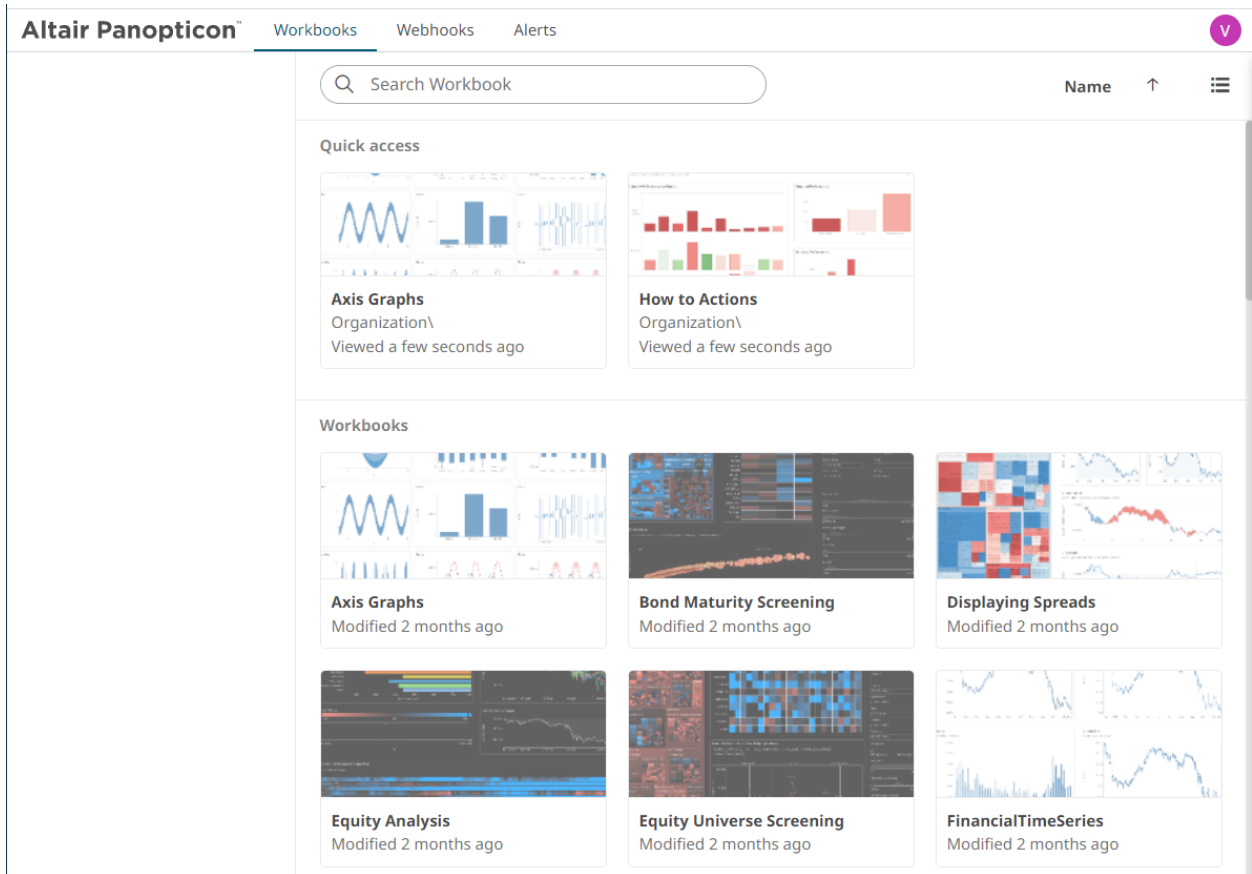
Click on the **Alerts** tab to view and manage the available alerts.
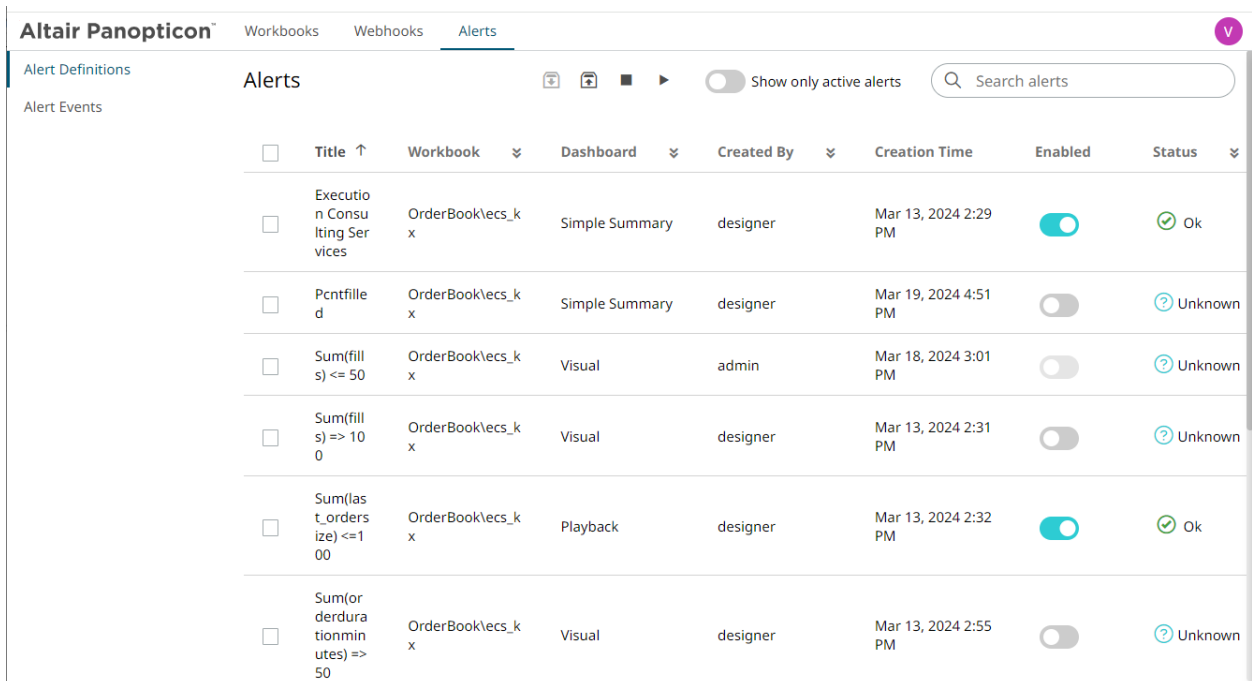
See Working with Alerts for more information.

Click the user icon and click **View Profile**. Then enter the email of the user or group who will receive the alert.

User Profile

Email Address

mvillanueva@altair.com

(Update Email Address)

Click **Update Email Address**.

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

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
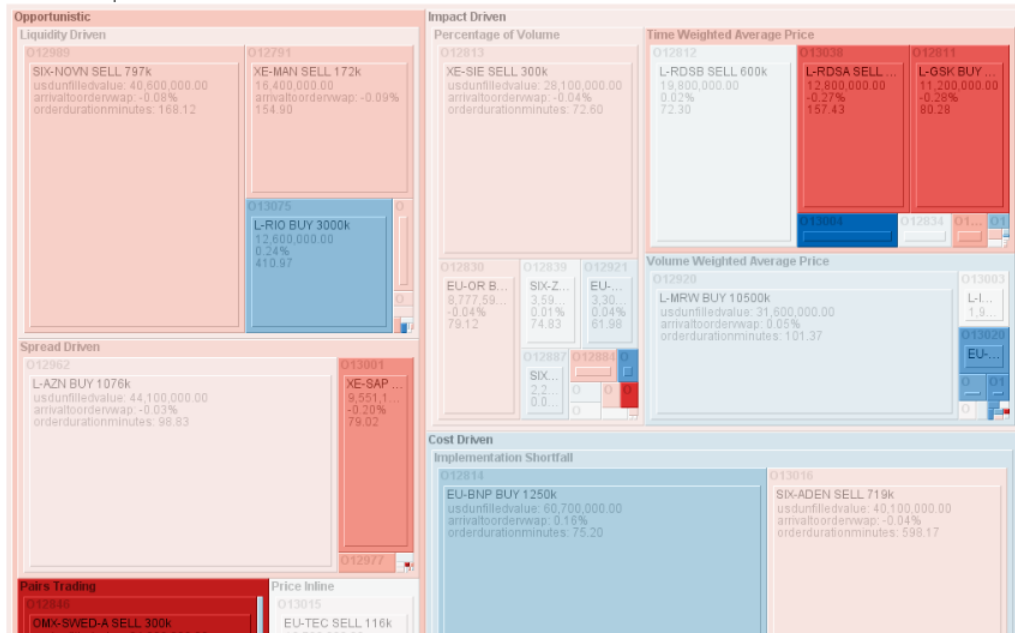
SAMPLE WEB CLIENT ALERTS

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, alerts initially display highlighted in blue:

The blue highlight eventually fades away.

The pop-up stays on screen until it is closed by clicking the ✕ button.

You can open an alert either by:

- ❑ Clicking the **Alert** 🔔 icon
- ❑ Clicking an alert on the pop-up

The item in the workbook that triggered the alert is highlighted.

You can now add comments, resolve, delegate, or claim the alert.

Adding Comments to an Alert

You can add comments to **Open** alerts that you own or are assigned to you.

Steps:

1. You can either:

   * Open a workbook and click the **Alerts** 🔔 icon to display the *Alerts* pane

- Click an alert on the *Alerts Events* section of the **Alerts** tab

2. Click of an alert.

    The *Comment* dialog displays.

3. Enter the comment then click .

    The comment is added to the alert.
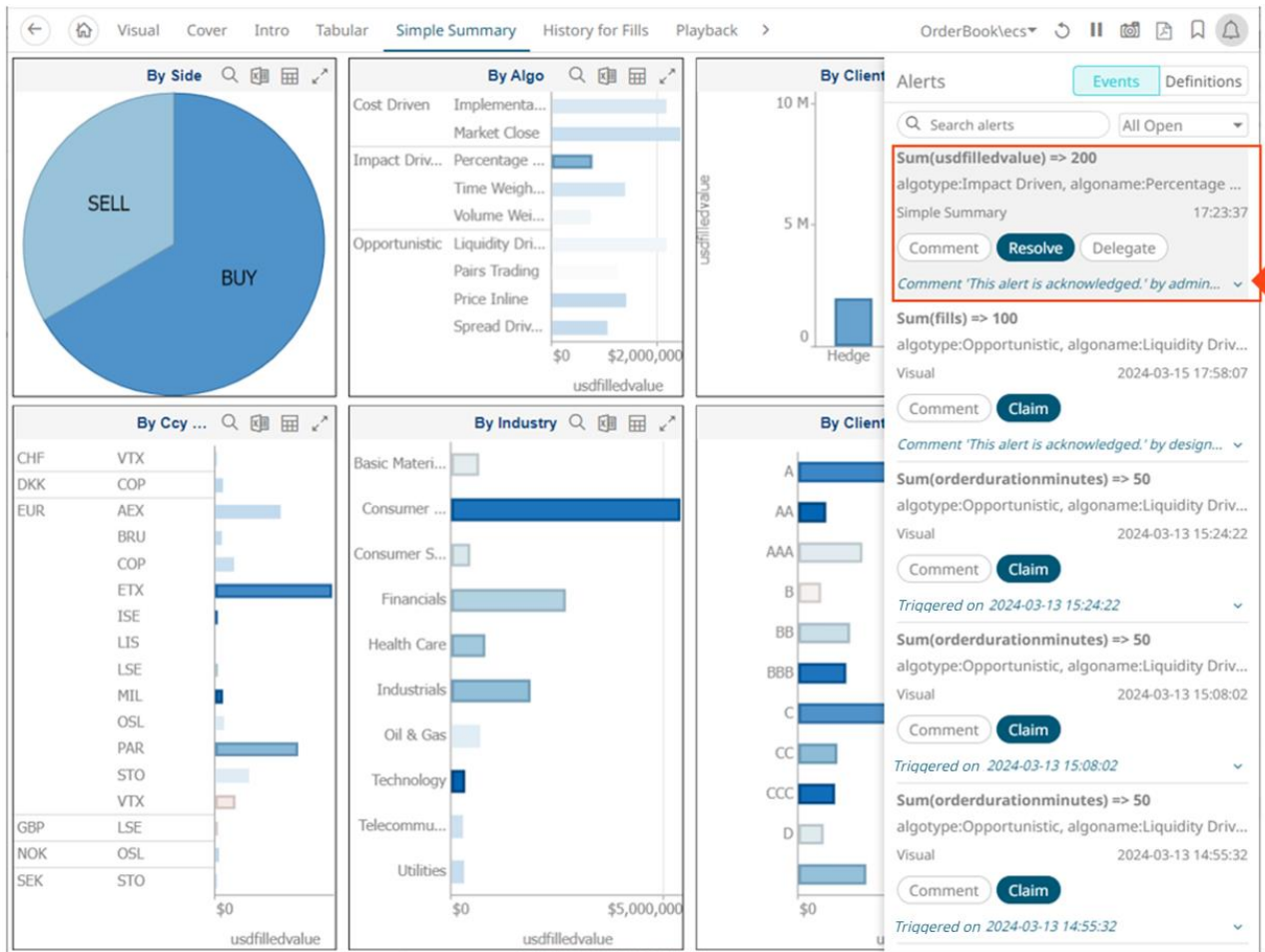
4. Click ⌄ to expand the alert and view all the workflow changes.

5. Add more comments if required.

Resolving an Alert

You can resolve **Open** alerts that you own or are assigned to you.
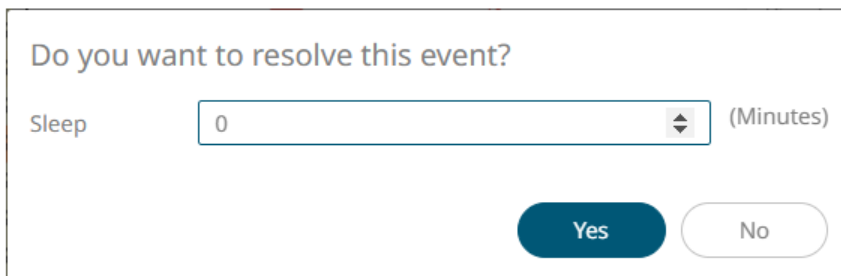
Steps:

1. You can either:

   - Open a workbook and click the **Alerts** 🔔 icon to display the *Alerts* pane. Then select an alert.
   - Click an alert on the *Alerts Events* section of the **Alerts** tab.

2. Click **Resolve** .

   A notification dialog displays.

   > **Do you want to resolve this event?**
   >
   > Sleep   0   ⬍   (Minutes)
   >
   > **Yes**   No

3. Enter the *Sleep* time to snooze the alert notification.

4. Click **Yes** .

Delegating an Alert

Administrators are allowed to delegate alerts to other users.
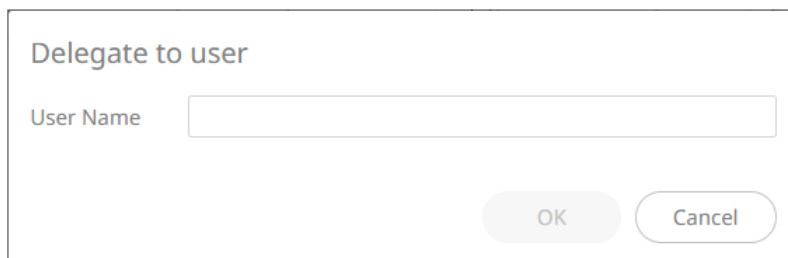
Steps:

1. You can either:

   - Open a workbook and click the **Alerts** 🔔 icon to display the *Alerts* pane. Then select an alert.
   - Click an alert on the *Alerts Events* section of the **Alerts** tab.
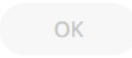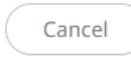
2. Click **Delegate** .

   The *Delegate to User* dialog displays.

   > **Delegate to user**
   >
   > User Name   [   ]
   >
   > OK   Cancel

3. Enter the *User Name* then click **OK** .

The workflow changes are updated.

The assigned user (e.g., **designer**) will get a notification and will be able to view the alert.

For example:
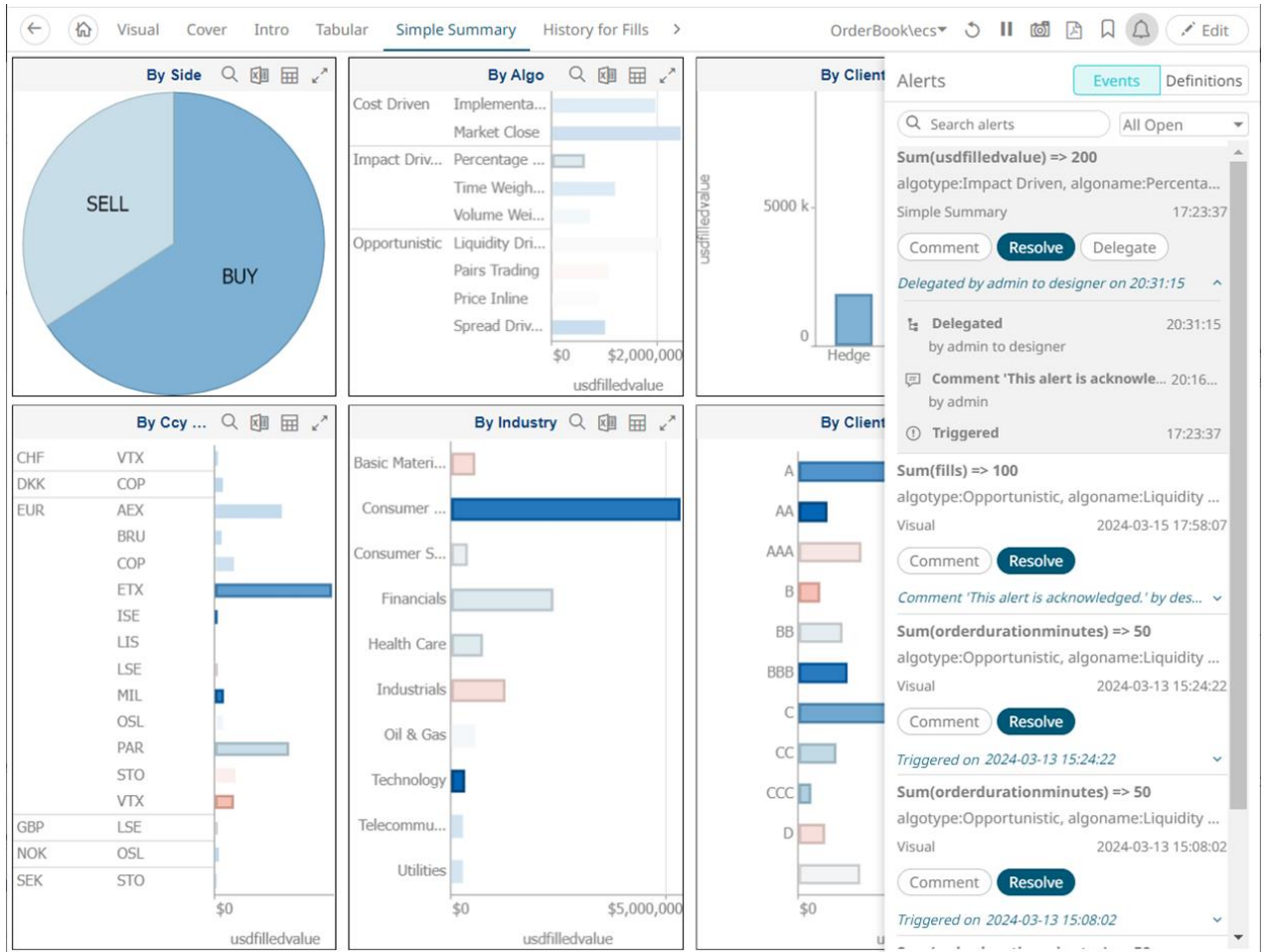
Claiming an Alert

Administrators can claim alerts that are not assigned to them.

Steps:

1. You can either:

   - Open a workbook and click the **Alerts** 🔔 icon to display the *Alerts* pane. Then select an alert.
   - Click an alert on the *Alerts Events* section of the **Alerts** tab.

2. Click **Claim** .

   A notification displays.

3. Click **Yes** .

The workflow changes are updated.
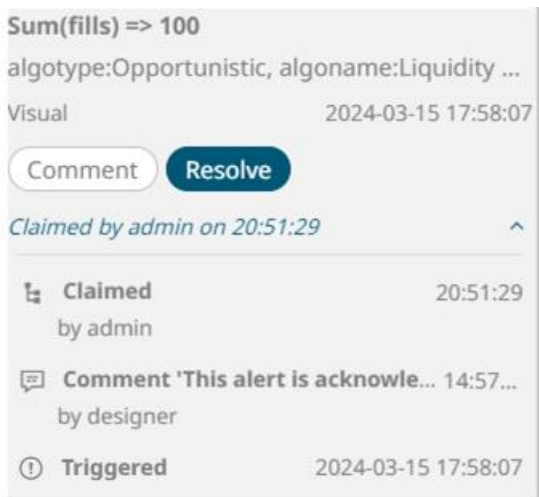
You can resolve or leave a comment on the alert.

Sharing Alerts

Administrators can share alerts.

Steps:

1. On the *Alert Definitions* section of the *Alerts* page or *Alerts* pane, select one or more alerts then click **Share Alerts** .

A notification displays.

Do you want to share the selected alert(s)?

[Yes] [No]

2. Click **Yes** .
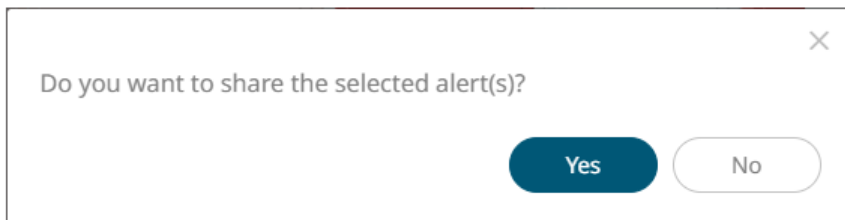
The selected personal alerts are now moved to the *Shared* section.

Stopping the Sharing of Alerts

Administrators can also stop the sharing of alerts.

Steps:

1. On the *Alert Definitions* section of the *Alerts* page or *Alerts* pane, select one or more alerts under Shared Alerts

   then click **Stop Sharing**   .

   A notification displays.

   Do you want to stop sharing the selected alert(s)?

   [Yes]   [No]

2. Click [Yes].
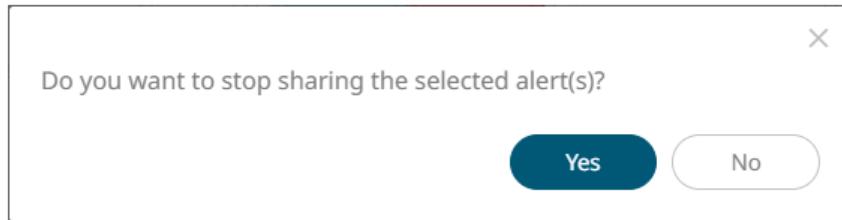
   The selected shared alerts are moved to the *My Alerts* section.

[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.

For example:

Parameters Set In	Description
Organization's root folder (i.e., **Global**)	Inherited by all of the available folders and applied to all workbooks
Public root folder (e.g., **Global > Orders**)	Inherited by the public root folder's subfolders and applied to all public workbooks.
User's root folder (i.e., **Global > ~**)	Inherited by the user root folder's subfolders and applied to all private workbooks.

ADDING PARAMETERS

Follow the steps below to add global parameters with an Administrator role.

Steps:

1. On the **Parameters** tab, click the Add ➕ icon of a global folder (root or subfolder).

   A new parameter entry displays.
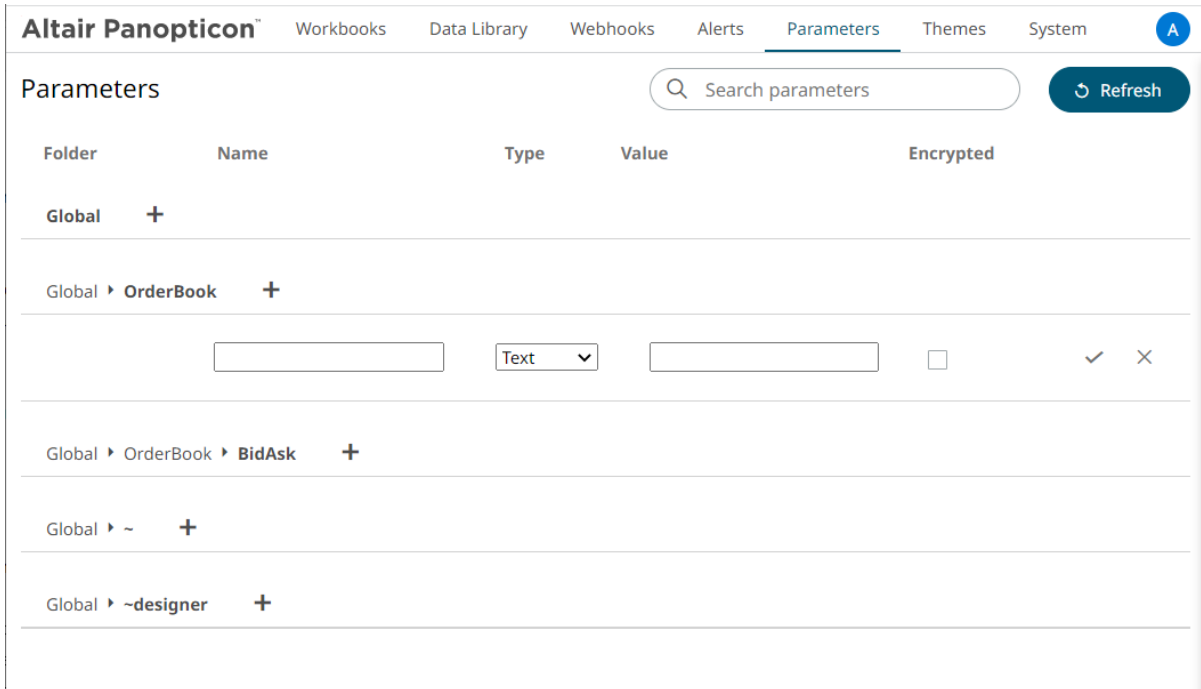
2. Enter a *Name* for the new parameter.

3. Select the *Type*: **Text**, **Numeric**, or **Time**.

4. Enter the *Default Value*.

<table>
<tr><td>**NOTE**</td><td>

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"

</td></tr>
</table>

5. Check the *Encrypted* box to encrypt the value, if required.

<table>
<tr><td>**NOTE**</td><td>Encryption is only supported for text parameters.</td></tr>
</table>

6. Click ✓ . The new parameter is added in the list.

Global parameters added in the parent/root folder are inherited by the corresponding subfolders.

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 to delete.

Refresh Parameters

Click 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.

Folder	Name	Type	Value	Encrypted		
Global ✚						
	Industry	Text	Telecommunications	☐	✎	🗑
Global ▸ **OrderBook** ✚						
	Industry	Text	*****	☑	✎	🗑
Global ▸ OrderBook ▸ **BidAsk** ✚						
	Industry	Text	Industrials	☐	✎	🗑
Global ▸ ~ ✚						
	Industry	Text	Telecommunications	☐	✎	🗑
	Interest	Text	Charged	☐	✎	🗑
Global ▸ **~designer** ✚						
	Industry	Text	Telecommunications	☐	✎	🗑
	Interest	Text	Charged	☐	✎	🗑

[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 to one of the provided default workbook themes: **Light, Light2023, Dark,** or **Dark2023**.
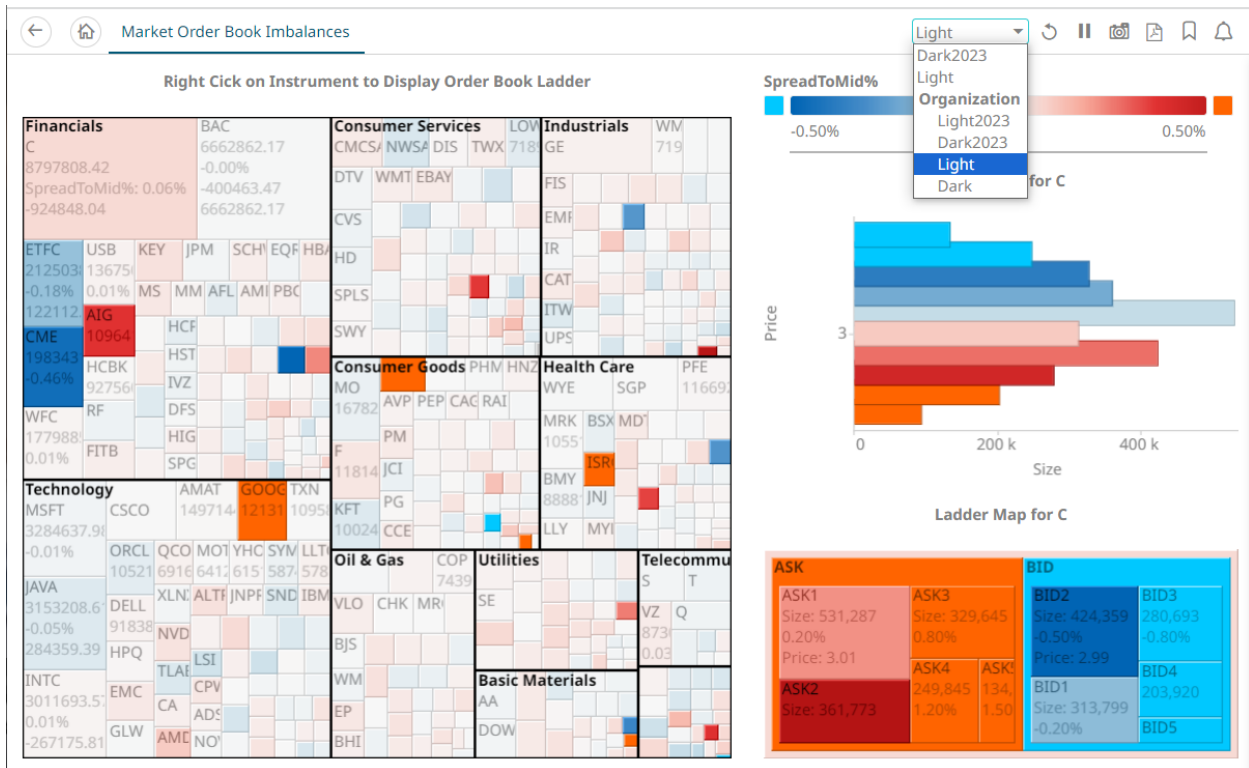
The **Themes** tab allows management of these workbook themes which are stored in the repository on the server.
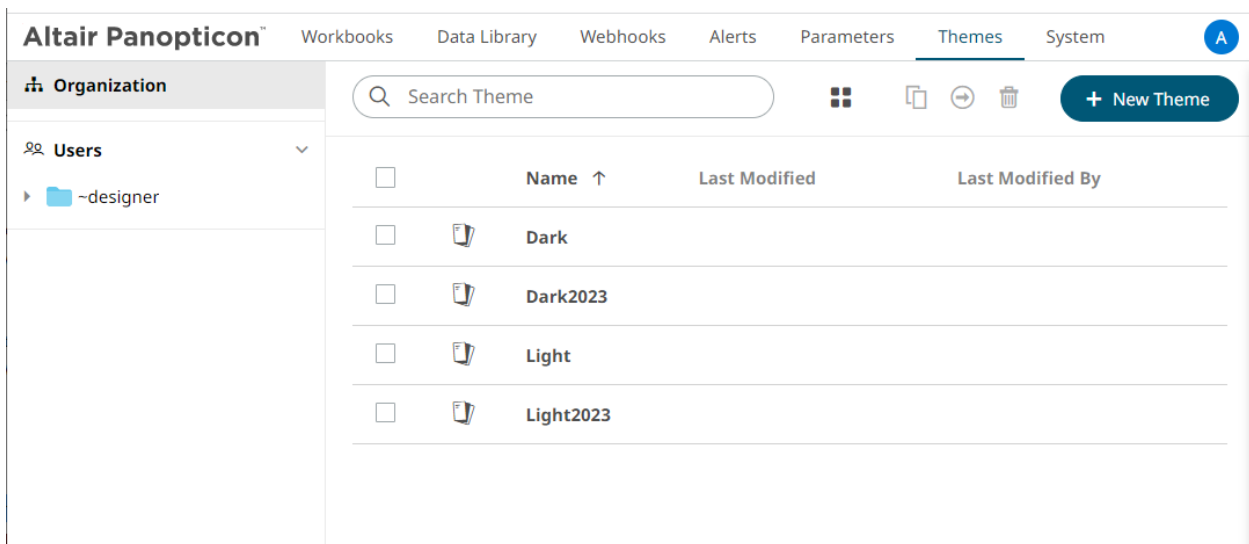
	NOTE	In the previous versions of Panopticon, all of theme-related settings are part of the workbook style, making it difficult to dynamically switch styles (e.g., colors, fonts etc.)
		When a workbook (created using versions before 17.5) is opened, all of the existing styles are extracted then saved 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.
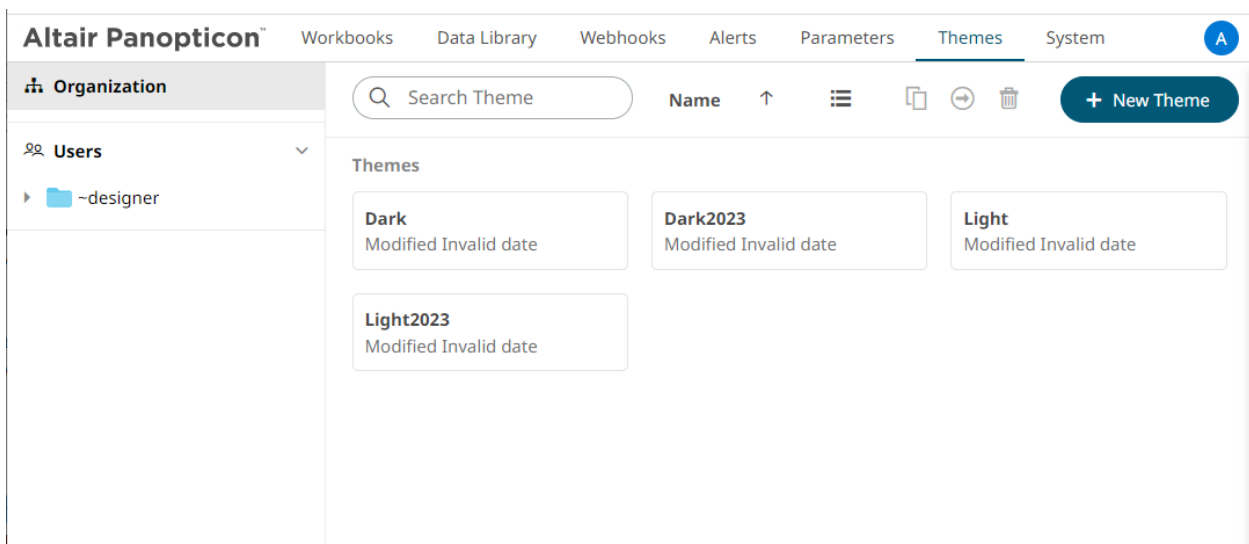
Or click **List View** , the themes are displayed in a standard listing.
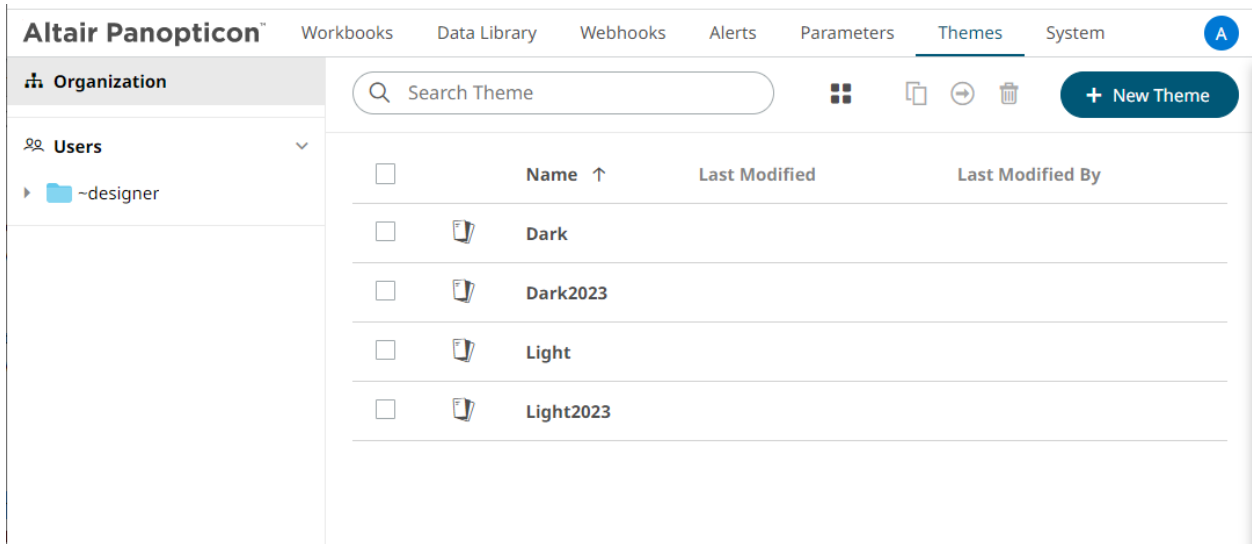
On either display view style, clicking on a themes title or thumbnail displays the *Theme* page.

SEARCHING FOR THEMES

On the *Themes* tab, to search for a particular theme, enter it in the *Search Theme* box.
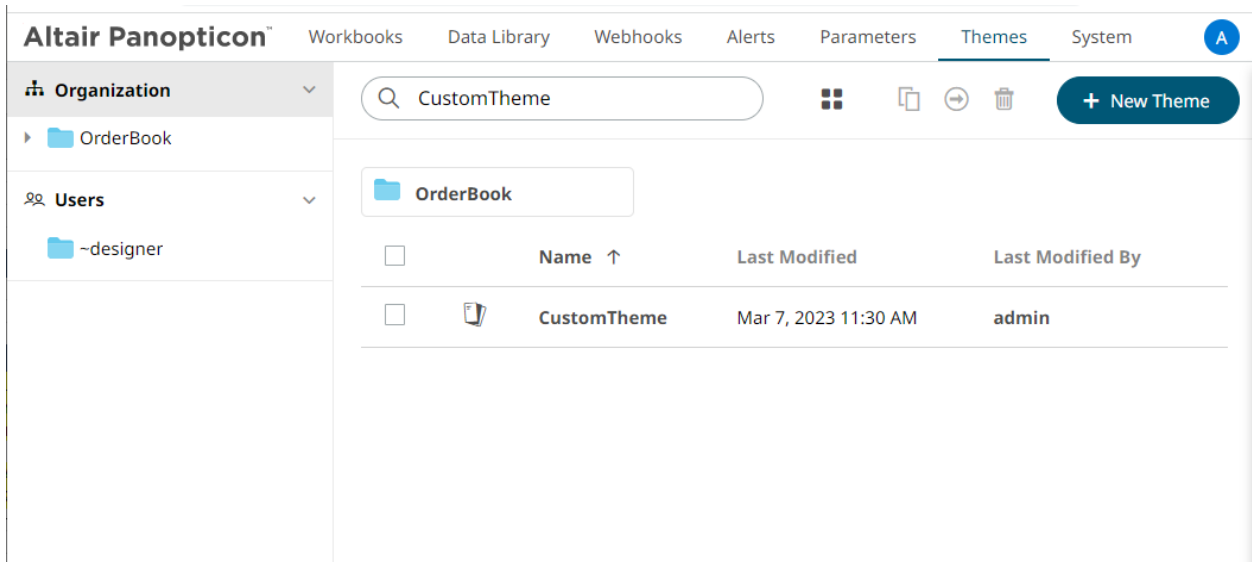
You can also enter one of more characters into the *Search Theme* box then click **Enter**. The suggested list of themes that matched the entries will be displayed.
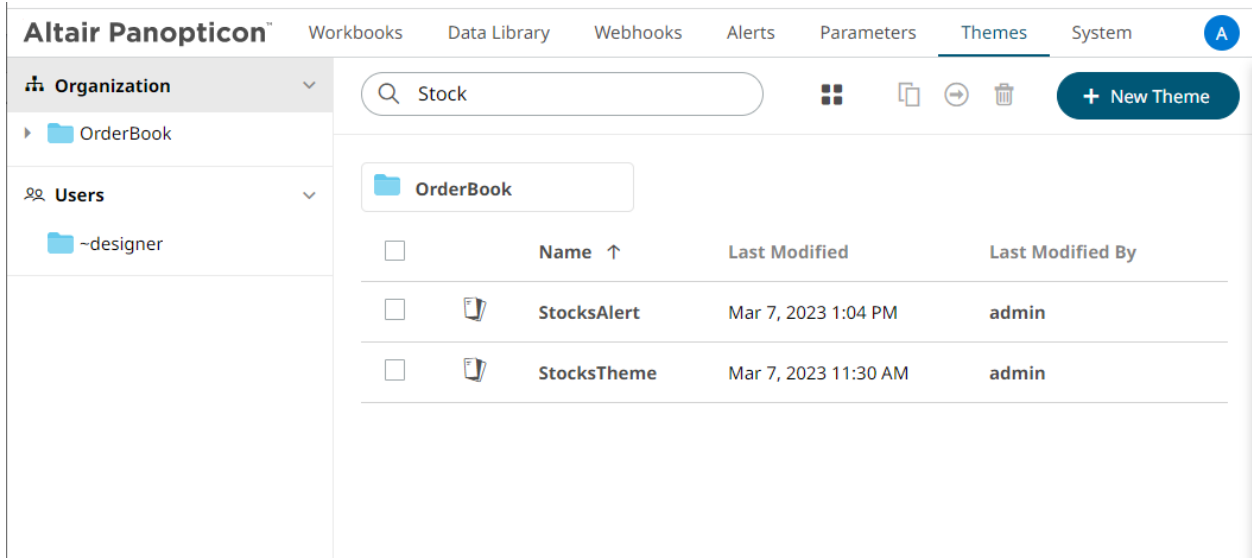
Click on a theme to open the settings page.

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

CREATING A NEW THEME

Creating a new theme allows setting the default or custom styles, color palettes, general colors, editor, and shape palettes to be used in workbooks and parts.

Steps:

1. On the *Themes* page, click .

   The *New Theme* dialog displays.
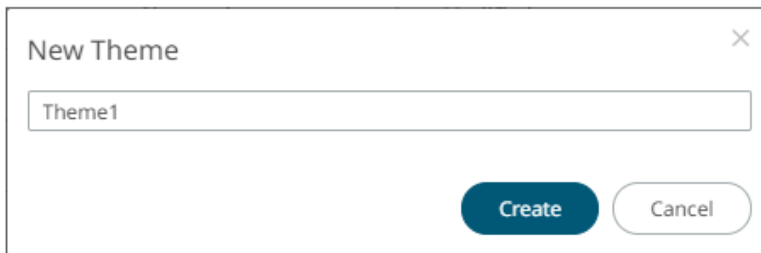
2. Enter the name of the theme then click .

   The new theme is displayed on the *Theme* page.
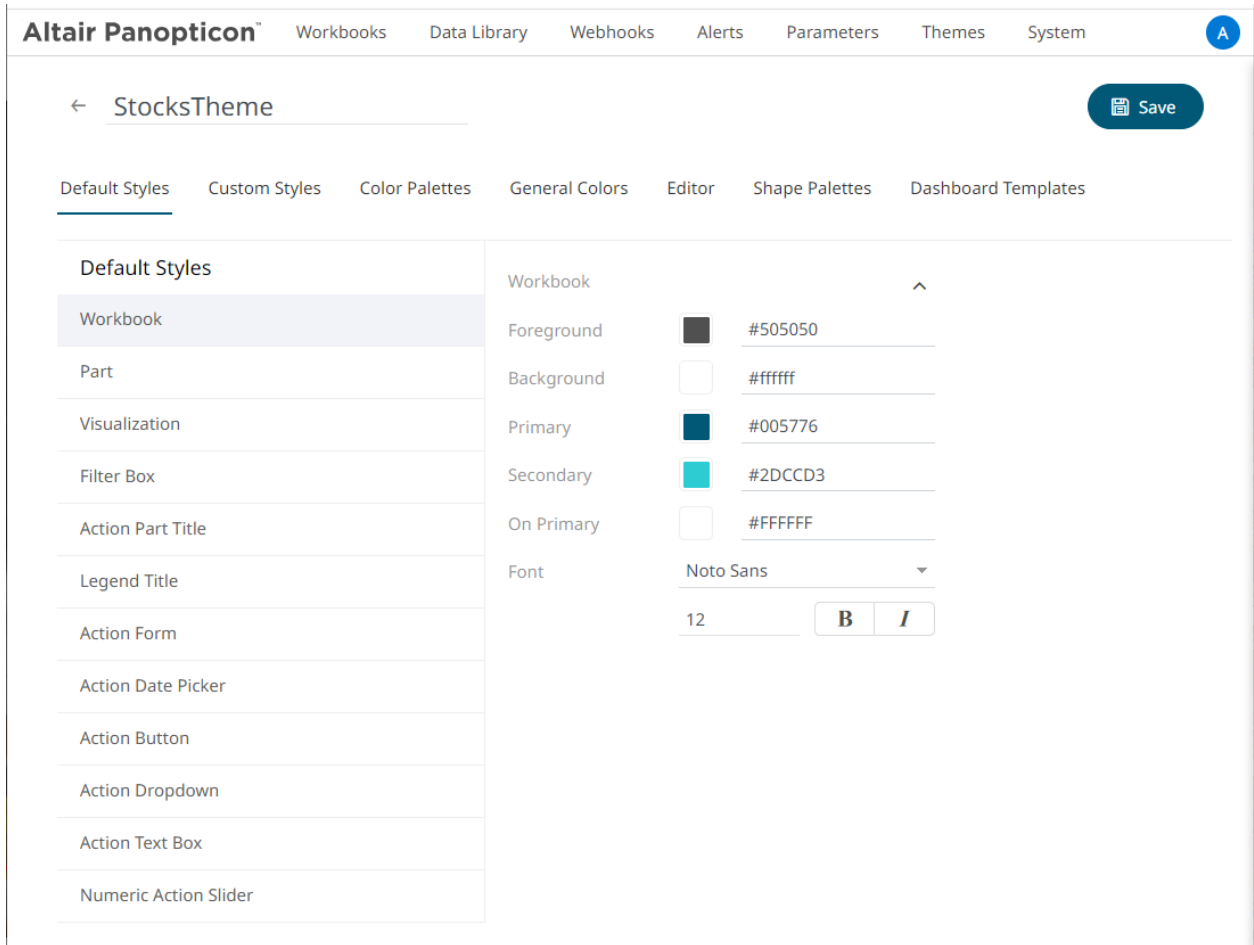
If you want to change the name of the theme, just enter a new one then click ✓ .

3. When creating a new theme, you may specify the following properties:

- Default Styles – Define the default style settings of the workbook, parts, visualizations, filter box, action part title, legend title, and actions.

- Custom Styles – Define the settings of the custom styles.

- Color Palettes – Manage, import, or export Text, Sequential, and Diverging color palettes.

- General Colors – Define or create duplicate general color.

- Editor – Define the editor style settings.

- Shape Palettes – Define the settings of shape palettes and add, upload, download, duplicate, or remove them.

- Dashboard Templates – Update or delete default and new dashboard templates.

Defining the Default Style Settings of a Theme

When you define the default settings of a theme, you specify the colors and fonts of the workbook, visualizations, filter box, action part title, legend title, and action form.

Steps:

1. To define the default styles of the workbook, click **Workbook o**n the **Default Styles** tab. The *Workbook Settings* are displayed.

You may opt 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.
Secondary	Secondary color of the workbook.
On Primary	Foreground color within the primary color.

1.1. You can either:

♦ click the corresponding *Color* box to display the *Color* dialog to:
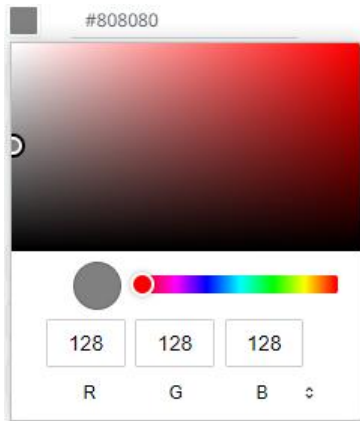
- select the color, or

- click ⚒ to enter the values

  for RGB
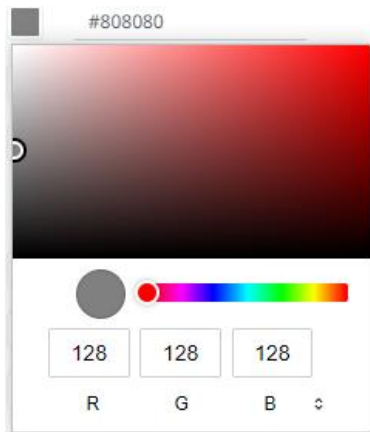
  for HSL
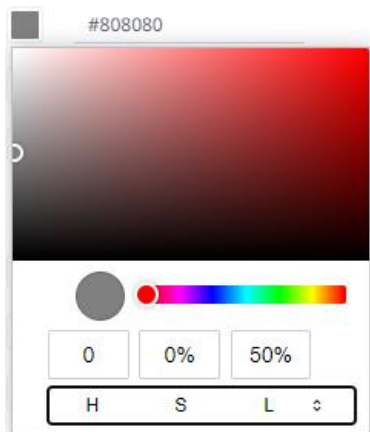
  for the Hex color code
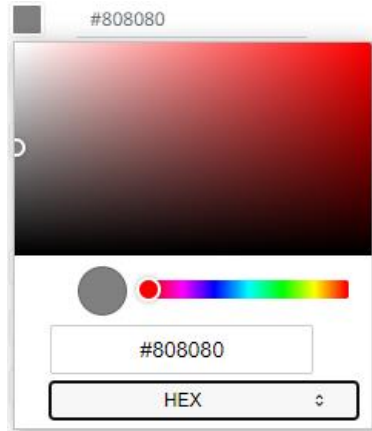
- ♦ or enter the Hex color code

1.2. Select the *Font.*

NOTE	The available custom fonts in Panopticon Real Time can be selected in the *Font* drop-down list.

1.3. Specify the *Font Size.*

1.4. Specify whether **Bold** and **Italic**.

2. To define the default styles of the parts, click **Part** on the *Default Styles* pane. The *Part Settings* are displayed.

You may opt to modify the colors of the following properties:

Property	Description
Foreground	Foreground color of the parts.
Background	Background color of the parts.
Border	Border color of the parts.

2.1. Follow step 1.1 to define the visualization, title, and border colors.

2.2. Select the part's *Font.*

2.3. Specify the part's *Font Size.*

2.4. Specify whether **Bold** and **Italic**.

2.5. Specify the *Padding* of the parts.

2.5.1. To modify the *Top*, *Right*, *Left*, and *Bottom* padding values, click 　.

The page updates to display the following fields:

Padding	8	
Top	Right	
8	8	
Left	Bottom	
8	8	

2.5.2. Set the desired padding values. If the values are not the same, **Mixed** is displayed in the *Padding* field.

2.6. Specify the *Border Radius*. When set to **0px**, the border is displayed as a sharp corner. Setting to higher values makes the border more rounded.

2.7. Specify the *Margin* of the parts.

2.7.1. To define the *Top*, *Right*, *Left*, and *Bottom* margin values, click 　.

Margin	8	
Top	Right	
8	8	
Left	Bottom	
8	8	

2.7.2. Set the desired margin values. If the values are not the same, **Mixed** is displayed in the *Margin* field.

3. To define the default styles of the visualizations, click **Visualization** on the *Default Styles* pane. The *Visualizations Settings* are displayed.

Title ^

Foreground

Background

Font ▾ ↺

B *I*

Alignment ☰ ☰ ☰

Part ^

Foreground

Background

Font ▾ ↺

B *I*

Border #dddddd

2

Padding 8 ⌞⌝

Border Radius 8

Margin 8 ⌞⌝

Title Row ^

Foreground

Font ▾ ↺

14 **B** *I*

🗑 Remove Style

You may opt to modify the colors of the following properties:

Property	Description
Foreground	Foreground color of the visualizations and title.
Background	Background color of the visualizations and title.
Border	Border color of the visualizations.

3.1. Follow step 1.1 to define the visualization, title, and border colors.

3.2. Select the visualization and title's *Font.*

3.3. Specify the visualization and title's *Font Size.*

3.4. Specify whether **Bold** and **Italic**.

> **NOTE**   For the part title, **Bold** is selected by default.

3.5. Specify the *Border Size* of the visualizations.

3.6. Select the visualization title *Alignment,* **Left, Center,** or **Right**.

3.7. Specify the *Padding* of the visualizations.

    3.7.1. To modify the *Top*, *Right*, *Left*, and *Bottom* padding values, click ⬚ .

        The page updates to display the following fields:

Padding	8	⬚
Top	Right	
8	8	
Left	Bottom	
8	8	

    3.7.2. Set the desired padding values. If the values are not the same, **Mixed** is displayed in the *Padding* field.

3.8. Specify the *Border Radius*. When set to **0px**, the border is displayed as a sharp corner. Setting to higher values makes the border more rounded.

3.9. Specify the *Margin* of the visualizations.

    3.9.1. To define the Top, Right, Left, and Bottom margin values, click ⬚ .

Margin	8	⬚
Top	Right	
8	8	
Left	Bottom	
8	8	

    3.9.2. Set the desired margin values. If the values are not the same, **Mixed** is displayed in the *Margin* field.

3.10. You can opt to define the settings of the *Title Rows*.

Remove Style

Title Row   ∧

Foreground   ☐   _____

Font   ▾ ↺

                    12   **B** *I*

Remove Style

Title Row   ∧

Foreground   ☐   _____

Font   ▾ ↺

                    16   **B** *I*

Remove Style

Title Row   ∧

Foreground   ☐   _____

Font   ▾ ↺

                    12   **B** *I*

Remove Style
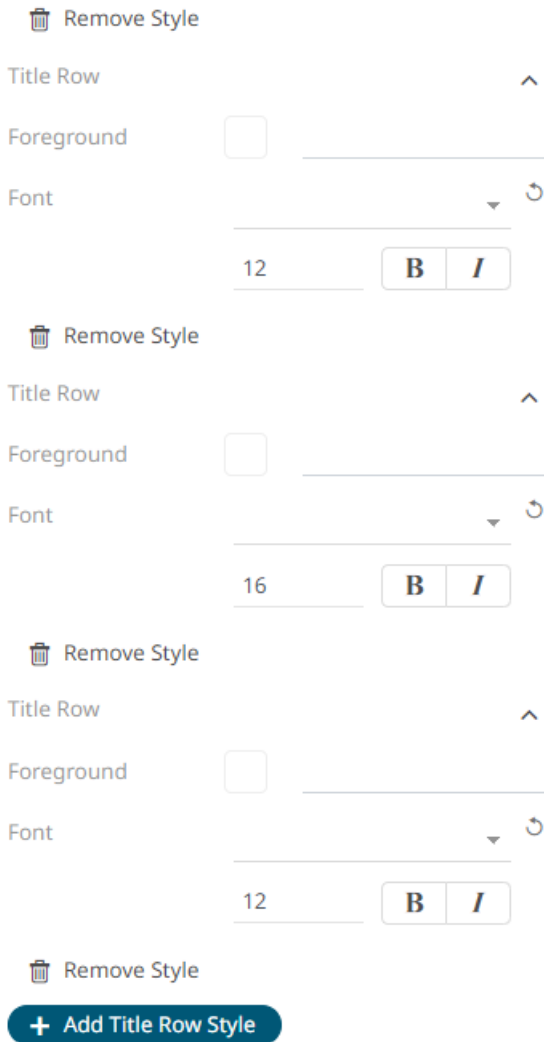
+ Add Title Row Style

By default, there are three title rows. You can do one of the following:

♦ Click 🗑 Remove Style to delete, or

♦ Click **+ Add Title Row Style** to add more title rows and define their settings.

4. To define the default styles of the filter box, click **Filter Box** on the *Default Styles* pane. The *Filter Box Settings* are displayed.

Title ^

Foreground ☐ _____

Background ☐ _____

Font ▼ ↺

_____ **B** *I*

Alignment ☰ ☰ ☰

You may opt to modify the colors of the following properties:

Property	Description
Foreground	Foreground color of the filter box.
Background	Background color of the filter box.

4.1. Follow step 1.1 to define the colors of the filter box.

4.2. Select the filter box title's *Font.*

4.3. Specify the filter box title's *Font Size.*

4.4. Specify whether **Bold** and **Italic**.

> **NOTE**   For the filter box title, **Bold** is selected by default.

4.5.   Select the filter box title *Alignment*: **Left, Center,** or **Right**.

5.   To define the default styles of the action part title, click **Action Part Title** on the *Default Styles* pane. The *Action Part Title Settings* are displayed.

Title ^

Font ▼ ↺

_____ **B** *I*

5.1. Select the action part title's *Font.*

5.2. Specify the action part title's *Font Size.*

5.3. Specify whether **Bold** and **Italic**.

> **NOTE**   For the action part title, **Bold** is selected by default.

6. To define the default styles of the legend title, click **Legend Title** on the *Default Styles* pane. The *Legend Title Settings* are displayed.

Title   ^

Font   ▾   ↺

                            B | I

6.1. Select the legend title's *Font.*

6.2. Specify the legend title's *Font Size.*

6.3. Specify whether **Bold** and **Italic**.

> **NOTE**   For the legend title, **Bold** is selected by default.

7. To define the default styles of the different actions (i.e., Action Form, Action Date Picker, Action Button, Action Dropdown, Action Text Box, Numeric Action Slider), click one and on the *Default Styles* pane to display their corresponding settings.

Most of these actions share the same settings as below:

Part   ^

Foreground   ☐ _____

Background   ☐ _____

Font   ▾

                            B | I

Border   ☐ _____

Padding   _____   ⌜⌟

Border Radius   _____

Margin   0   ⌜⌟

Button   ^

Foreground   ☐ _____

Background   ☐ _____

Font   ▾

                            B | I

You may opt to modify the colors of the following properties:

Property	Description
Foreground	Foreground color of the action, button, or slider.
Background	Background color of the action, button, or slider.

7.1. Follow step 1.1 to define the colors of the actions.

7.2. Select the action and button's *Font.*

7.3. Specify the action and button's *Font Size.*

7.4. Specify whether **Bold** and **Italic**.

> **NOTE**   For the action form, **Bold** is selected by default.

7.5. Specify the action's border color and size.

7.6. Specify the *Padding* of the actions.

7.6.1. To modify the *Top*, *Right*, *Left*, and *Bottom* padding values, click   .

The page updates to display the following fields:

Padding

Top	Right

Left	Bottom

7.6.2. Set the desired padding values. If the values are not the same, **Mixed** is displayed in the *Padding* field.

7.7. Specify the *Border Radius*. When set to **0px**, the border is displayed as a sharp corner. Setting to higher values makes the border more rounded.

7.8. Specify the *Margin* of the actions.

7.8.1. To define the *Top, Right, Left*, and *Bottom* margin values, click   .

Margin   0

Top	Right
0	0

Left	Bottom
0	0

7.8.2. Set the desired margin values. If the values are not the same, **Mixed** is displayed in the *Margin* field.

8. Proceed to the **Custom Styles** tab to specify the custom styles of the theme.

Defining the Custom Style Settings of a Theme

Published custom style configuration of a part can be modified in the **Custom Styles** tab and can be applied to other parts.

Steps:

1. Click **Custom Styles** tab. The available published custom styles and properties are displayed.

2. See Define Default Styles to specify the settings depending on the custom style part.
3. Proceed to the **Color Palettes** tab to define the color palettes of the theme.

Defining the Color Palettes Settings of a Theme

When you define the settings of the color palettes, you can manage, import, or export Text, Sequential, and Diverging color palettes.

Steps:

1. To select the *Diverging*, *Sequential*, and *Text* color palettes to use within the workbooks, click the **Color Palettes** tab.

2. Select the checkboxes of the provided color palettes that will be included for each category.

3. Click the radio button of the preferred *Default* color palette for each category.

4. To upload color palettes, click **Import Palettes** . The *Upload Color Palette* dialog displays.

Upload color palette ✕

Choose color palettes file to upload Drag file here

☐ Replace color palettes

Upload   Cancel
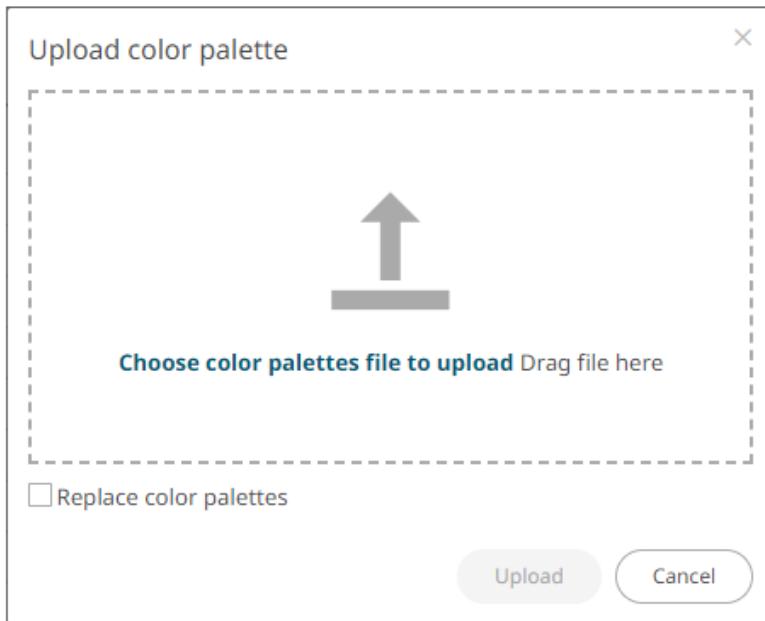
5. To export color palettes, you can either:

- Click **Export All Palettes** for all color palettes. The `ColorPalettes.excp` file is exported.

- Click the **Export Palette** icon of a color palette. The `<ColorPaletteName>.excp` file is exported.

You can move the exported file to the desired location.

6. To replace the color palettes, select the *Replace Color Palettes* checkbox.

7. Click **Upload** .

A notification displays once the color palettes file is uploaded.

Click **Cancel** to close the dialog. The uploaded color palette is added in the list.

8. Proceed to the **General Colors** tab to specify the general colors of the theme.

Defining the General Color Settings of a Theme

You can specify new general colors or duplicate or remove them.

Steps:

1. To set the general colors to be used for visualizations, click the **General Colors** tab.

By default, the new *General Colors* is named **GeneralColorsLight**.

← StocksTheme

Default Styles   Custom Styles   Color Palettes   General Colors   Editor   Shape Palettes   Dashboard Templates

General Colors

GeneralColorsLight

GeneralColorsLight

| Title | GeneralColorsLight |
| Set default | |

General Colors

Major Grid Color		#d0d0d0
Minor Grid Color		#f1f1f1
Missing Color		#c0c0c0
Fore Color		#808080
Zebra Stripe Color		#fbfbfb
Snapshot Color		#d0d0d0
Border Color		#808080
Back Color		#ffffff
Selection Color		#808080
Focus Color		#808080
Axis Color		#d0d0d0

2.   Click **Duplicate**   to make a duplicate copy of the new general colors.

← StocksTheme

Default Styles   Custom Styles   Color Palettes   **General Colors**   Editor   Shape Palettes   Dashboard Templates

General Colors

GeneralColorsLight

GeneralColorsLight 1

GeneralColorsLight 1

Title	GeneralColorsLight 1

Set default

General Colors

Major Grid Color		#d0d0d0
Minor Grid Color		#f1f1f1
Missing Color		#c0c0c0
Fore Color		#808080
Zebra Stripe Color		#fbfbfb
Snapshot Color		#d0d0d0
Border Color		#808080
Back Color		#ffffff
Selection Color		#808080
Focus Color		#808080
Axis Color		#d0d0d0

3. You can enter a new name and click ✓ . **Set Default** is turned off and the **Remove** icon is now available.

4. Tap the **Set Default** slider to turn it on and the **Remove** icon is no longer available.

StocksTheme

Default Styles   Custom Styles   Color Palettes   **General Colors**   Editor   Shape Palettes   Dashboard Templates

General Colors

GeneralColorsLight

GeneralColorTheme

GeneralColorTheme

Title	GeneralColorTheme

Set default

General Colors

Major Grid Color		#d0d0d0
Minor Grid Color		#f1f1f1
Missing Color		#c0c0c0
Fore Color		#808080
Zebra Stripe Color		#fbfbfb
Snapshot Color		#d0d0d0
Border Color		#808080
Back Color		#ffffff
Selection Color		#808080
Focus Color		#808080
Axis Color		#d0d0d0

5. Click any of the color boxes to display the *Color* dialog.

#d0d0d0

208	208	208	
R	G	B	
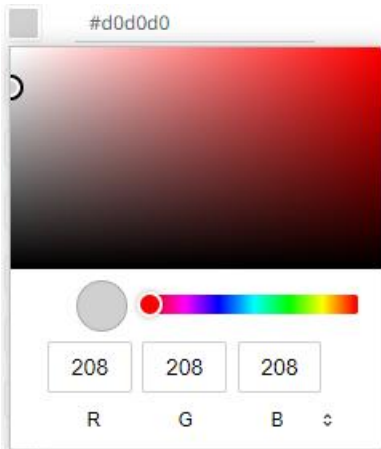
Select or specify the new general colors: AxisColor, BackColor, BorderColor, FocusColor, ForeColor, MajorGridColor, MinorGridColor, MissingColor, SelectionColor. SnapshotColor,ZebraStripeColor.

Or enter the corresponding Hex color code.

6. Repeat steps 2 to 5 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.

7. Select any of the general colors and tap the **Set Default** slider to make it the default.

8. Select any of the general colors that is not set as the default and click **Delete** 🗑 to remove.

9. Proceed to the **Editor** tab to specify the editor style of the **Dark** theme.

Defining the Editor Style Settings of a Theme

You can define the editor style settings of a dark theme.

Steps:

1. To set the *Foreground*, *Background*, *Primary*, *On Primary*, and *Secondary* colors for the editor style of the **Dark** theme, click the **Editor** tab.

← StocksTheme

Default Styles	Custom Styles	Color Palettes	General Colors	Editor	Shape Palettes	Dashboard Templates

Foreground	◼	#4D4D4D
Background	◻	#FFFFFF
Primary	◼	#005776
On Primary	◻	#FFFFFF
Secondary	◼	#2DCCD3

2. Click on any of the color boxes to display the *Color* dialog and select or enter the preferred color.

3. Proceed to the **Shape Palettes** tab to specify the shape palettes of the theme.

Defining the Shape Palettes of a Theme

When you define the shape palettes of a theme, you specify the settings of shape palettes and add, upload, download, duplicate, or remove them.

Steps:

1. To set the shape palettes that can be used with the workbook theme, click the **Shape Palette** tab.

← StocksTheme

Default Styles   Custom Styles   Color Palettes   General Colors   Editor   **Shape Palettes**   Dashboard Templates

Shape Palettes   +   ⬆

Default Shape Palette   ⬇ 🗐
● ■ ♦ ▲ ▼ ○ □ ◇ △ ▽

Arial   ⬇ 🗐 🗑
A B C D E F G H I J

Default Shape Palette

Title   Default Shape Palette

Default Palette   🔵

Add Shape   +

🗑 ●	🗑 ■	🗑 ♦	🗑 ▲	🗑 ▼
🗑 ○	🗑 □	🗑 ◇	🗑 △	🗑 ▽
🗑 ✕	🗑 +	🗑 ✳	🗑 ⊠	🗑 ⊞
🗑 ⊗	🗑 ⊕	🗑 ⧂	🗑 ⬙	

Default Shape   ● ⌄

NOTE	For more information in how to create, upload, download, modify, duplicate, or delete shape palettes, refer to the sections below.

2. Proceed to the **Dashboard Templates** tab to specify the dashboard templates of the theme.

Defining the Dashboard Templates of a Theme

Default dashboard templates are provided in Panopticon. You can modify the name or delete default and new dashboard templates.

Steps:

1. To modify the dashboard templates that can be used with the workbook theme, click the **Dashboard Templates** tab.

← StocksTheme

Default Styles   Custom Styles   Color Palettes   General Colors   Editor   Shape Palettes   **Dashboard Templates**

Dashboard Templates		Blank	
Blank	🗑	Title	Blank
Single	🗑		
Single + Filter	🗑		
Single + Time Filter	🗑		
Single + Filters	🗑		
Two Columns	🗑		
Two Columns + Filter	🗑		
2x2 Grid	🗑		
2x2 Grid + Filter	🗑		
Cards	🗑		

2. Click on a dashboard template, then you can either:

   • modify the *Title*, or

   • click 🗑 to delete.

3. Click **Save** 💾 Save to save the new theme.

4. Clicking the ← displays the **Themes** tab page with the new theme added in the list.

		Name ↑	Last Modified	Last Modified By
☐		Dark		
☐		Dark2023		
☐		Light		
☐		Light2023		
☐		StocksTheme	Mar 22, 2023 2:51 PM	admin
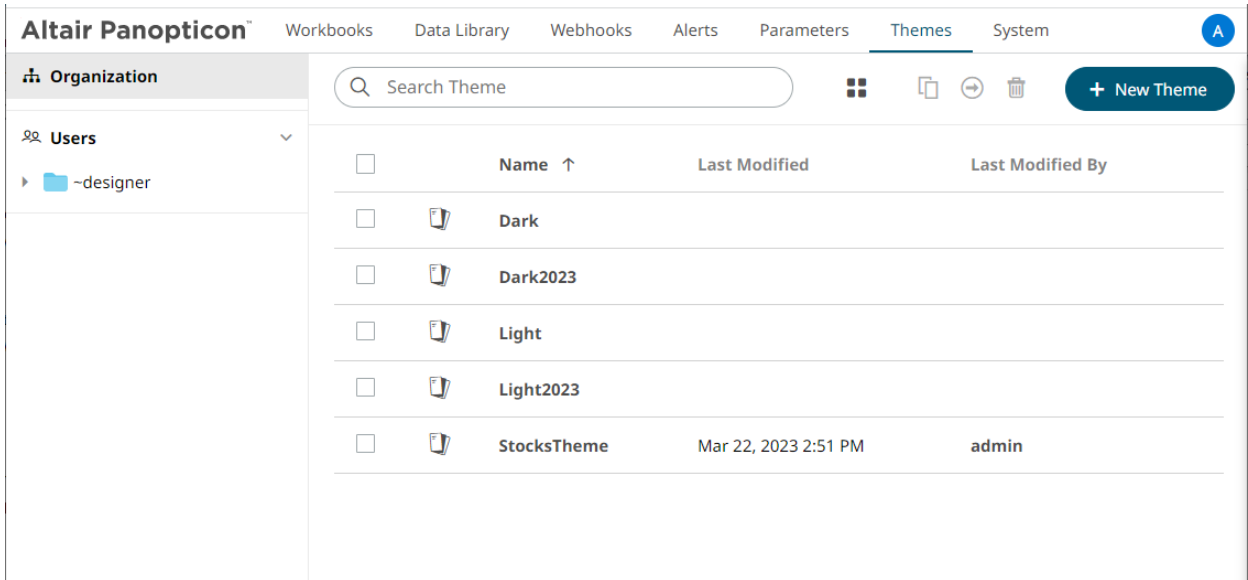
NOTE	Unlike the default **Dark, Dark2023, Light,** and **Light2023** themes, new themes can be deleted.

THEMES TOOLBAR AND CONTEXT MENU

Moving, copying, and removing themes can either be done using:
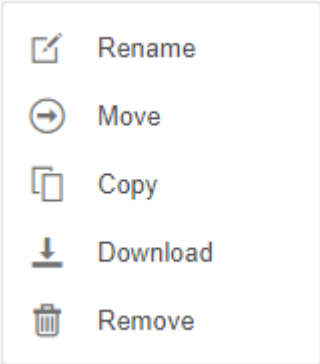
❑   Context menu

Theme Context Menu
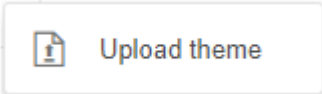
Theme Folder Context Menu

❑   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**.
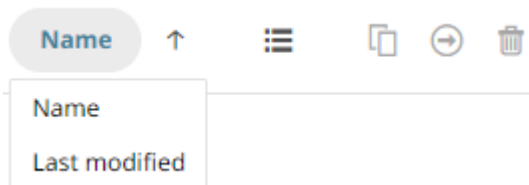
- Name
- Last Modified

Then click the *Sort Order*:

- ↑   Ascending

- ↓   Descending

❑ click on the **Name**, **Last Modified**, or **Last Modified By** column header of the *List View.*
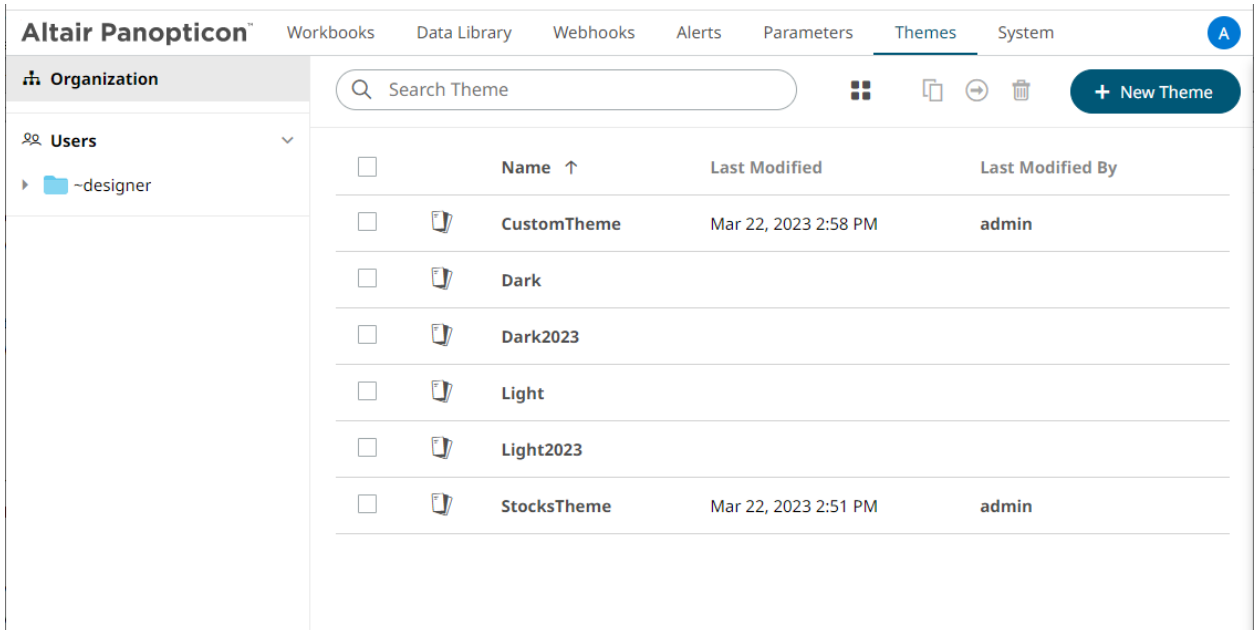
Then click the *Sort Order*:

- ↑ Ascending

- ↓ Descending

Renaming a Theme

Steps:

1. Right-click on a theme then select **Rename** on the context menu.
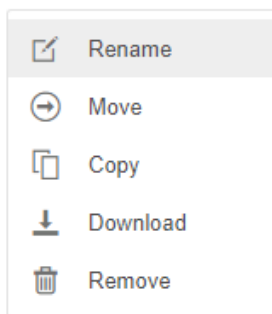
The *Rename Theme* dialog displays.

Rename Theme

StocksTheme

Rename   Cancel
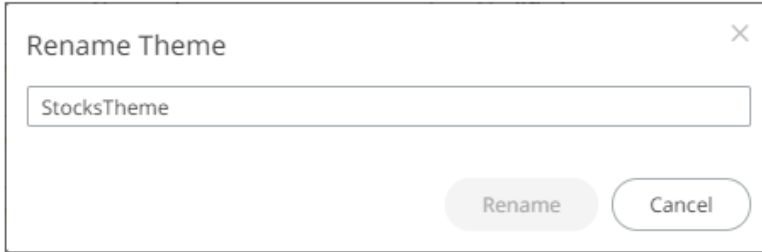
2. Enter a new name then click **Rename** .
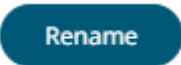
Moving Themes

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.

Move Theme

Select folder to move 'selected themes' to:

Organization   *Current folder*

▶ Orders

Users

   ~designer

Move   Cancel

2. Click **Move** .
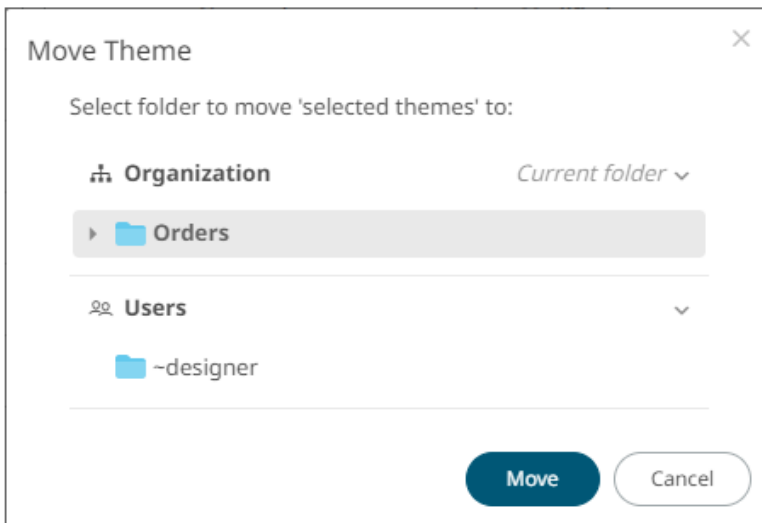
    The themes are moved and displayed on the selected folder.

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] 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] .
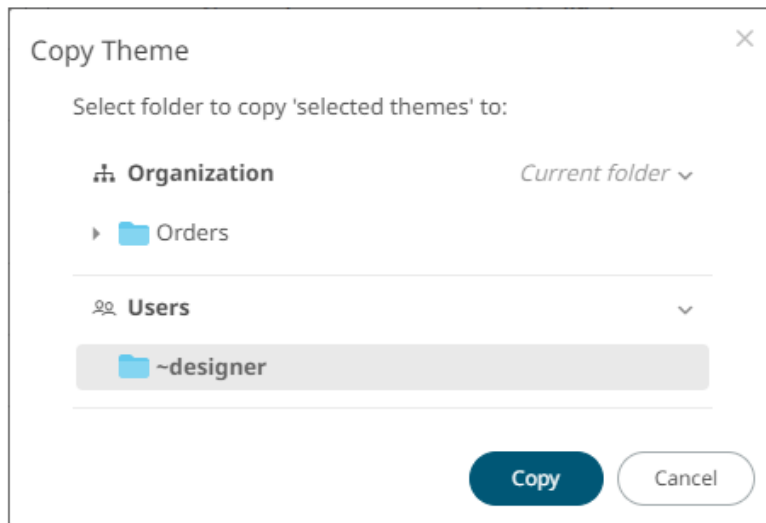
   The themes are copied and displayed on the selected folder.

Downloading Themes

On the *List* or *Grid* view, right-click on a theme and selected **Download** on the context menu to download a copy.
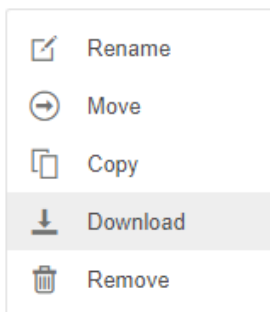
You can copy this file to the desired location.

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.
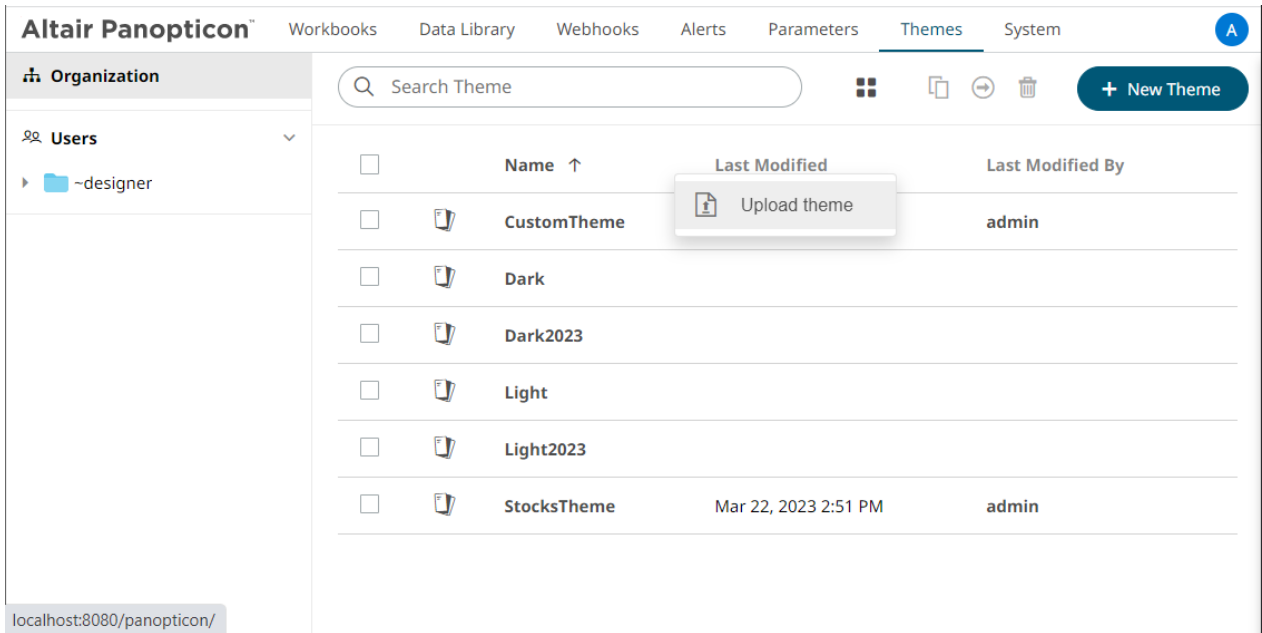
The *Upload Theme* dialog displays.
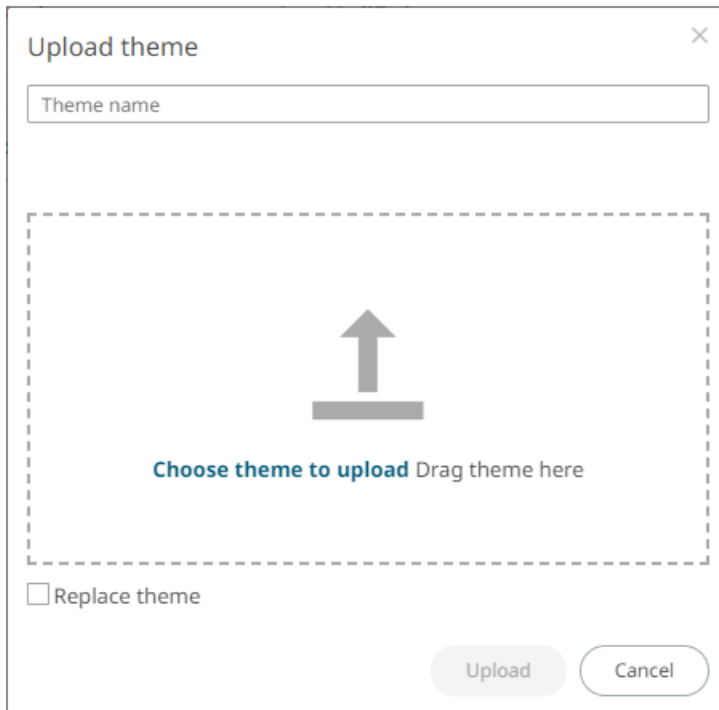
2. To upload a workbook theme, either:

- 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.
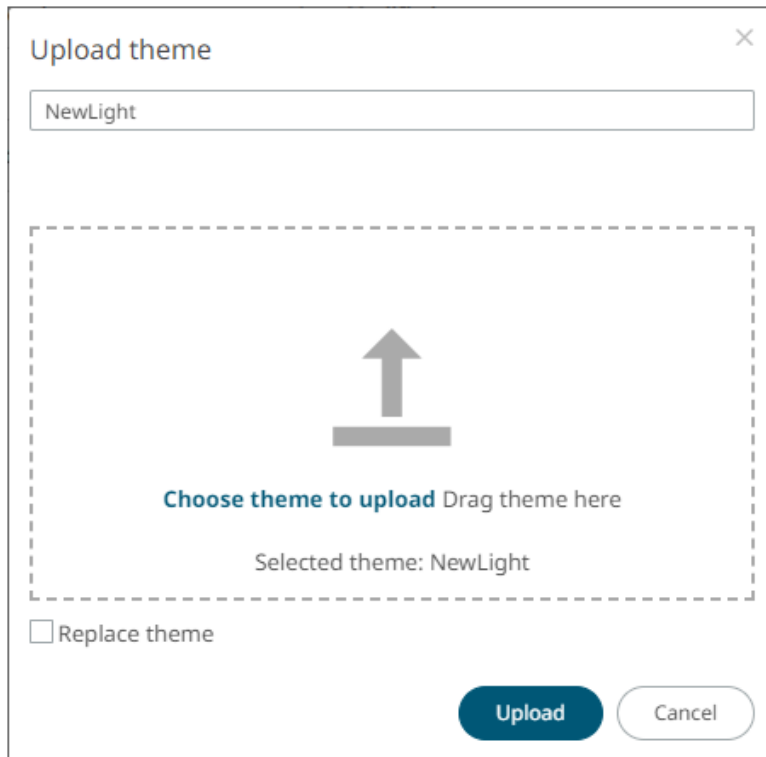
You can opt to rename the uploaded workbook theme.

3. To replace the workbook theme, check the *Replace Theme* box.

4. Click **Upload** .

A notification displays once the file is uploaded.
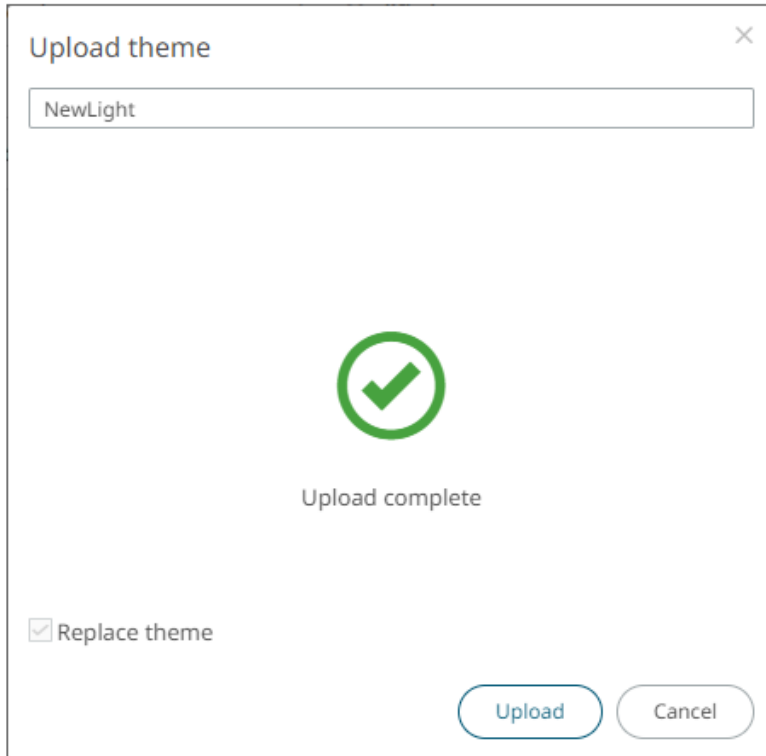
The uploaded theme is added in the *Theme* list.

Deleting Themes

The default themes (**Dark, Dark2023, Light**, and **Light2023**) cannot be removed.

Steps:

1. Right-click on one or two themes then either:
   - select **Remove** on the context menu, or
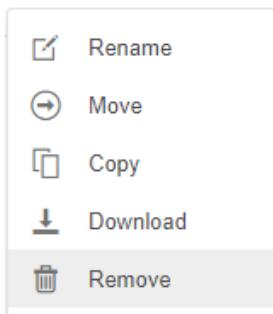
   - click the **Remove** 🗑 icon on the toolbar.

   A notification message displays.

Are you sure you want to remove theme Dark 1?

Yes   No

2. Click **Yes**.
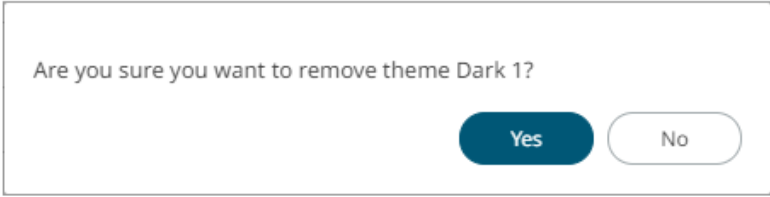
COLOR PALETTES

The single, sign, text, sequential, and diverging color palettes that is used in text or numeric color variables in visualizations can be created, imported, exported, modified, duplicated, or deleted in the **Color Palettes** tab of a *Theme* page.

Import Palettes   **Export All Palettes**

Single   +

Include	Name
☑	Light Blue
☑	Light Gray
☑	Light Green
☑	Light Orange
☑	Light Red
☑	Medium Blue
☑	Medium Gray
☑	Medium Green
☑	Medium Orange
☑	Medium Red

Sign   +

Include	Name
☑	Light Orange-Blue
☑	Light Orange-Green
☑	Light Red-Blue
☑	Light Red-Green
☑	Medium Orange-Blue
☑	Medium Orange-Green
☑	Medium Red-Blue
☑	Medium Red-Green
☑	Red-Gray

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	◯	✎	⧉	🗑	⬇

Diverging   +

Include	Name					
☐	Brown-Gray-Petrol	○	✎	⧉	🗑	⬆
☑	Brown-White-Petrol	○	✎	⧉	🗑	⬆
☐	Orange-Gray-Blue	○	✎	⧉	🗑	⬆
☐	Orange-Gray-Green	○	✎	⧉	🗑	⬆
☑	Orange-White-Blue	○	✎	⧉	🗑	⬆
☑	Orange-White-Green	○	✎	⧉	🗑	⬆
☑	Purple-White-Turquoise	○	✎	⧉	🗑	⬆
☐	Red-Black-Blue	○	✎	⧉	🗑	⬆
☐	Red-Black-Green	○	✎	⧉	🗑	⬆
☐	Red-Gray-Blue	○	✎	⧉	🗑	⬆
☐	Red-Gray-Green	○	✎	⧉	🗑	⬆
☑	Red-White-Blue	◉	✎	⧉	🗑	⬆
☐	Red-White-Blue-Print	○	✎	⧉	🗑	⬆
☑	Red-White-Green	○	✎	⧉	🗑	⬆
☐	Red-White-Green-Print	○	✎	⧉	🗑	⬆
☑	Red-Yellow-Green	○	✎	⧉	🗑	⬆
☐	Red-Yellow-Green-Print	○	✎	⧉	🗑	⬆

NOTE	Creating, modifying, duplicating, or deleting color palettes can also be done inside a workbook in *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.

Importing a Color Palette

Users can upload their own color palettes.

Steps:

1. On the *Color Palettes* pane, click [Import Palettes] .

    The *Upload Color Palette* dialog displays.
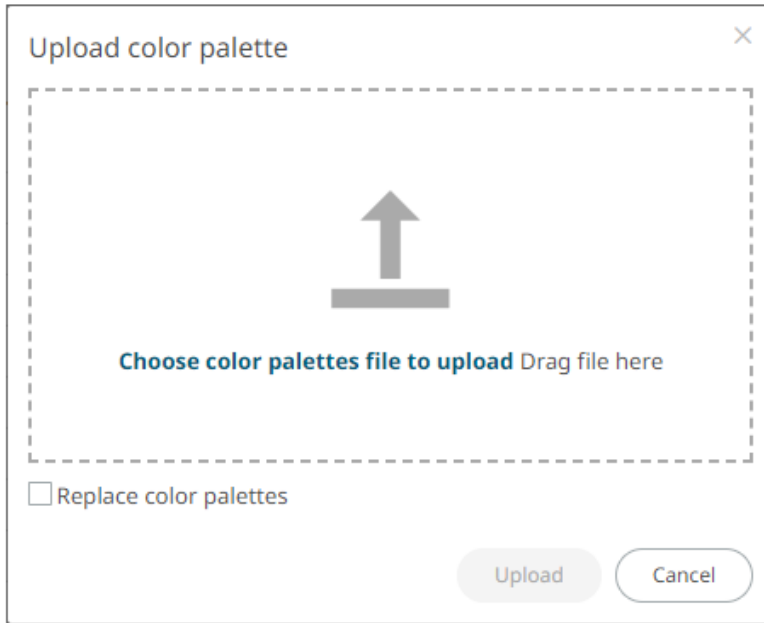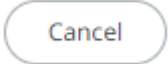
2. To upload a color palette, either:

   - 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.

3. To replace the color palettes, check the *Replace Color Palettes* box.

4. Click .

   A notification displays once the color palettes file is uploaded.

   Click to close the dialog. The uploaded color palette is added in the list.

Exporting Color Palette

You can either:

❑ Click to export all color palettes. The `ColorPalettes.excp` file is exported.

❑ Click the **Export Palette** icon to export a color palette. The `<ColorPaletteName>.excp` file is exported.

You can move the downloaded file to the desired location.

Creating a New Single Color Palette

These are the single colors that will be shared in a workbook for:

❑ records in Table and Record visualizations for the background, text, or shape

❑ visual members in Combination visualizations for the background or text

Light and medium single color palettes are provided in Panopticon Real Time, but you can also add new ones.

Steps:

1. On the *Single* section, click the **Add Palette** ╋ icon.

   The *New Single Palette* dialog displays.

New Single Palette	✕
Title	New Single Palette
Palette	▪ #4682b4
	Cancel OK

2. Enter the *Title* then click ✓ .

3. Click the **Color** box to display the *Color* dialog and set the palette color or enter the Hex color code.

4. Click **Ok** .

   The new single color palette is added in the list (e.g., **Medium Yellow**). Note that it is already included and can be modified, duplicated, and deleted.

Single				╋
Include	**Name**			
✓	Light Blue	○ ✎ ⧉ 🗑		
✓	Light Gray	○ ✎ ⧉ 🗑		
✓	Light Green	○ ✎ ⧉ 🗑		
✓	Light Orange	○ ✎ ⧉ 🗑		
✓	Light Red	○ ✎ ⧉ 🗑		
✓	Medium Blue	◉ ✎ ⧉ 🗑		
✓	Medium Gray	○ ✎ ⧉ 🗑		
✓	Medium Green	○ ✎ ⧉ 🗑		
✓	Medium Orange	○ ✎ ⧉ 🗑		
✓	Medium Red	○ ✎ ⧉ 🗑		
✓	Medium Yellow	○ ✎ ⧉ 🗑		
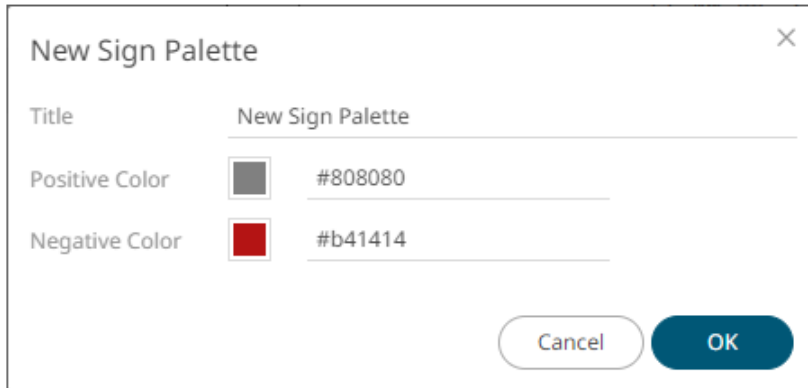
Creating a New Sign Color Palette

The Sign color palette is used to signify the positive or negative values in numeric visual members.

Steps:

1. On the *Sign* section, click the **Add Palette** ✛ icon.

    The *New Sign Palette* dialog displays.

    New Sign Palette

Title	New Sign Palette
Positive Color	⬛ #808080
Negative Color	🟥 #b41414

    Cancel   OK

2. Enter the *Title* then click ✓ .

3. To set the *Positive Color* (default is **Gray**) and the *Negative Color* (default is **Red**), click the **Color** box to display the *Color* dialog and select the palette color or enter the Hex color code.

4. Click **Ok** .

    The new Sign color palette is added in the list (e.g., **Red-Green**). Note that it is already included and can be modified, duplicated, and deleted.

    Sign   ✛

Include	Name				
☑	Light Orange-Blue	○	✏	⧉	🗑
☑	Light Orange-Green	○	✏	⧉	🗑
☑	Light Red-Blue	○	✏	⧉	🗑
☑	Light Red-Green	○	✏	⧉	🗑
☑	Medium Orange-Blue	○	✏	⧉	🗑
☑	Medium Orange-Green	○	✏	⧉	🗑
☑	Medium Red-Blue	○	✏	⧉	🗑
☑	Medium Red-Green	○	✏	⧉	🗑
☑	Red-Gray	◉	✏	⧉	🗑
☑	Red-Green	○	✏	⧉	🗑

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:
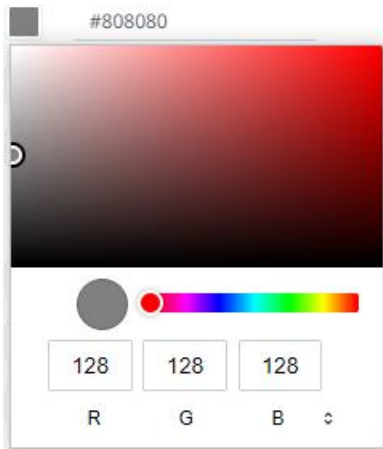
- ◆ select the color, or
- ◆ click ⬛ to enter the values

  for RGB
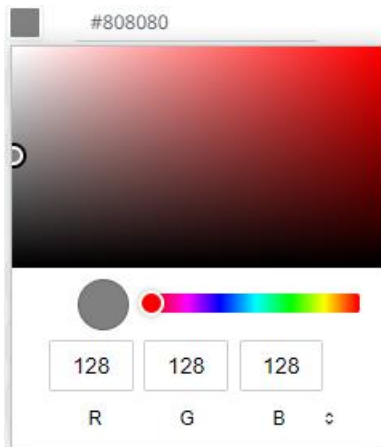
  for HSL
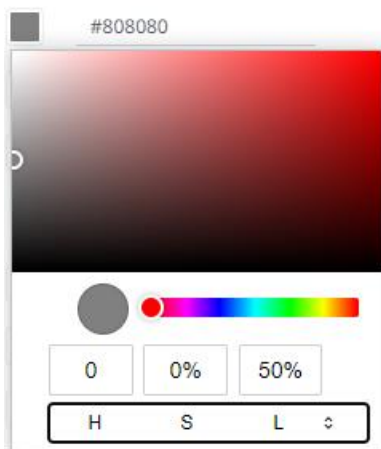
  for the Hex color code
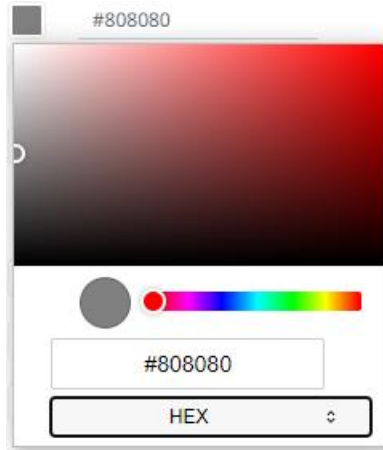
- or enter the *Hex* color code
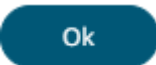
5. Click .

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
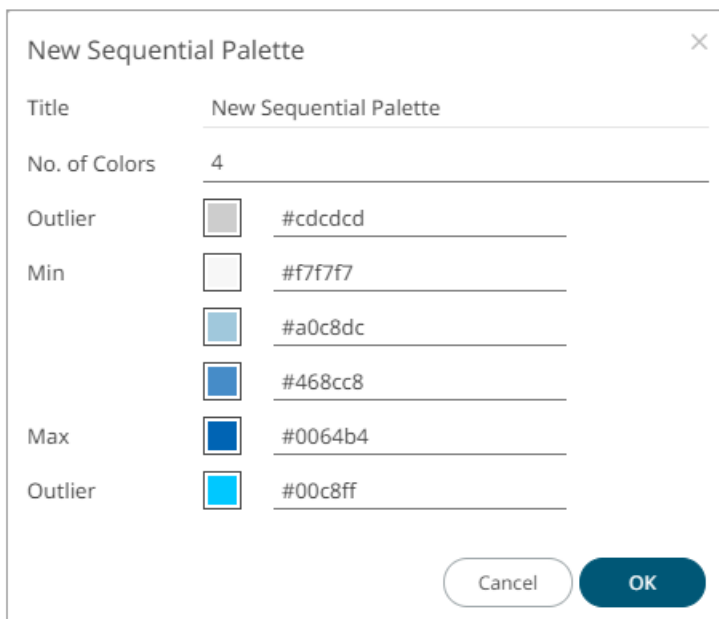
Diverging Palettes use a three-color gradient between a minimum, middle and a maximum value. Numeric columns containing both positive and negative values default to the Diverging Palette with the **Red White Blue** color palette selected.

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.

New Sequential Palette		✕
Title	New Sequential Palette	
No. of Colors	4	
Outlier	⬜	#cdcdcd
Min	⬜	#f7f7f7
	🟦	#a0c8dc
	🟦	#468cc8
Max	🟦	#0064b4
Outlier	🟦	#00c8ff

   Cancel   OK

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**).

Sequential +

Include	Name				
☑	Gray	○	✎	⎘	🗑
☑	Green-Red	○	✎	⎘	🗑
☑	Purple-Orange	○	✎	⎘	🗑
☑	White-Blue	⦿	✎	⎘	🗑
☐	White-Blue-Print	○	✎	⎘	🗑
☑	White-Green	○	✎	⎘	🗑
☑	White-Orange	○	✎	⎘	🗑
☑	White-Red	○	✎	⎘	🗑
☐	White-Red-Print	○	✎	⎘	🗑
☑	Yellow-Red	○	✎	⎘	🗑

To create a new diverging numeric color palette:

1. On the *Diverging* section, click the **New** ✚ icon.

   The *New Diverging Palette* dialog displays.

   New Diverging Palette ✕

Title	New Diverging Palette
No. of Colors	7
Outlier	⬛ #ff6400
Min	⬛ #b41414
	⬛ #e13232
	⬛ #f7aa9b
Mid	⬜ #f7f7f7
	⬛ #a0c8dc
	⬛ #468cc8
Max	⬛ #0064b4
Outlier	⬛ #00c8ff

   Cancel   OK

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**).

Include	Name				
☐	Brown-Gray-Petrol	○	✎	⎘	🗑
☑	Brown-White-Petrol	○	✎	⎘	🗑
☐	Orange-Gray-Blue	○	✎	⎘	🗑
☐	Orange-Gray-Green	○	✎	⎘	🗑
☑	Orange-White-Blue	○	✎	⎘	🗑
☑	Orange-White-Green	○	✎	⎘	🗑
☑	Purple-White-Turquoise	○	✎	⎘	🗑
☐	Red-Black-Blue	○	✎	⎘	🗑
☐	Red-Black-Green	○	✎	⎘	🗑
☐	Red-Gray-Blue	○	✎	⎘	🗑
☐	Red-Gray-Green	○	✎	⎘	🗑
☑	Red-White-Blue	◉	✎	⎘	🗑
☐	Red-White-Blue-Print	○	✎	⎘	🗑
☑	Red-White-Green	○	✎	⎘	🗑
☐	Red-White-Green-Print	○	✎	⎘	🗑
☑	Red-Yellow-Green	○	✎	⎘	🗑
☐	Red-Yellow-Green-Print	○	✎	⎘	🗑
☑	Yellow-White-Red	○	✎	⎘	🗑

Diverging   **+**

Modifying Color Palettes
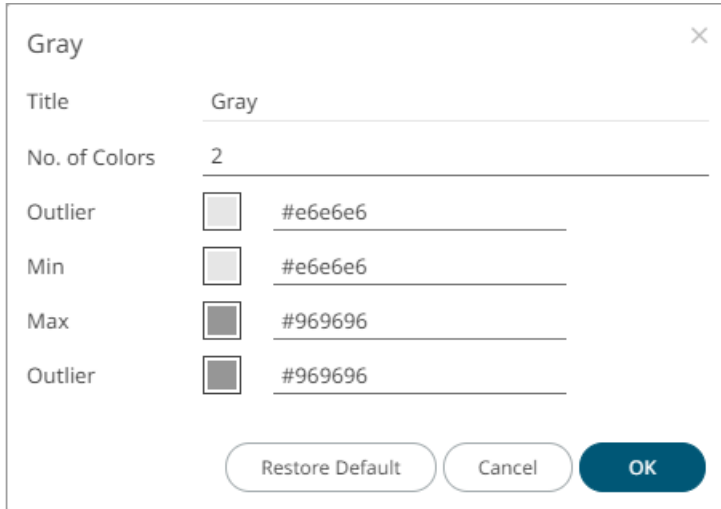
Any of the included or checked color palettes can be modified.

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.

Gray	✕
Title	Gray
No. of Colors	2
Outlier	☐ #e6e6e6
Min	☐ #e6e6e6
Max	◼ #969696
Outlier	◼ #969696

   Restore Default   Cancel   OK

2. Modify the *Title, Number of Colors*, and assigned colors.

3. Click **Ok** to commit the changes or Restore Default 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**).

Text   +

Include	Name				
☑	Coffee Bean	○	✎	⧉	🗑
☑	Fourteen Colors	○	✎	⧉	🗑
☑	Panopticon BI	○	✎	⧉	🗑
☑	Seven Light Colors	○	✎	⧉	🗑
☑	Seven Light Colors 1	○	✎	⧉	🗑
☑	Seven Standard Colors	○	✎	⧉	🗑
☑	Sixteen Colors	○	✎	⧉	🗑
☑	Spectral	○	✎	⧉	🗑
☑	Sunshine	○	✎	⧉	🗑
☑	Twenty Eight Colors	⦿	✎	⧉	🗑
☐	Twenty Eight Colors Print	○	✎	⧉	🗑
☑	Vintage	○	✎	⧉	🗑

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.

← Light

| Default Styles | Custom Styles | Color Palettes | General Colors | Editor | Shape Palettes | Dashboard Templates |

Shape Palettes   + ⬆

Default Shape Palette   ⬇ ⧉
●■◆▲▼○□◇△▽

Arial   ⬇ ⧉ 🗑
A B C D E F G H I J

Default Shape Palette

Title   Default Shape Palette

Default Palette   ⬤─

Add Shape   +

🗑 ●	🗑 ■	🗑 ◆	🗑 ▲	🗑 ▼
🗑 ○	🗑 □	🗑 ◇	🗑 △	🗑 ▽
🗑 ✕	🗑 +	🗑 ✳	🗑 ⊠	🗑 ⊞
🗑 ⊗	🗑 ⊕	🗑 ⬙	🗑 ⟡	🗑 │
🗑 —				

Default Shape   ⬤ ⌄

| NOTE | Panopticon is shipped with two shape palettes (**Default Shape Palette** and **Arial**). |

Creating a New Shape Palette

Steps:

1. Click **Add Palette** ╋ .

   A new shape palette displays (i.e., **ShapePalette.0**).

← Light

Default Styles Custom Styles Color Palettes General Colors Editor **Shape Palettes** Dashboard Templates

Shape Palettes + ⬆

Default Shape Palette ⬇ 🗐
● ■ ◆ ▲ ▼ ○ □ ◇ △ ▽

Arial ⬇ 🗐 🗑
A B C D E F G H I J

ShapePalette.0 ⬇ 🗐 🗑

Default Shape Palette

Title   Default Shape Palette

Default Palette   🔵⬤

Add Shape   +

🗑 ●   🗑 ■   🗑 ◆   🗑 ▲   🗑 ▼

🗑 ○   🗑 □   🗑 ◇   🗑 △   🗑 ▽

🗑 ✕   🗑 ＋   🗑 ✳   🗑 ⊠   🗑 ⊞

🗑 ⊗   🗑 ⊕   🗑 ◈   🗑 ⬦   🗑 |

🗑 —

Default Shape   ⬤ ⌄

2. Click *ShapePalette.<Number>*.

The page changes to allow the definition of the new shape palette.

Default Styles Custom Styles Color Palettes General Colors Editor **Shape Palettes** Dashboard Templates

Shape Palettes + ⬆

Default Shape Palette ⬇ 🗐
● ■ ◆ ▲ ▼ ○ □ ◇ △ ▽

Arial ⬇ 🗐 🗑
A B C D E F G H I J

ShapePalette.0 ⬇ 🗐 🗑

ShapePalette.0

Title   ShapePalette.0

Default Palette   ⚪

Add Shape   +

Default Shape   ⌄

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 cannot 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.
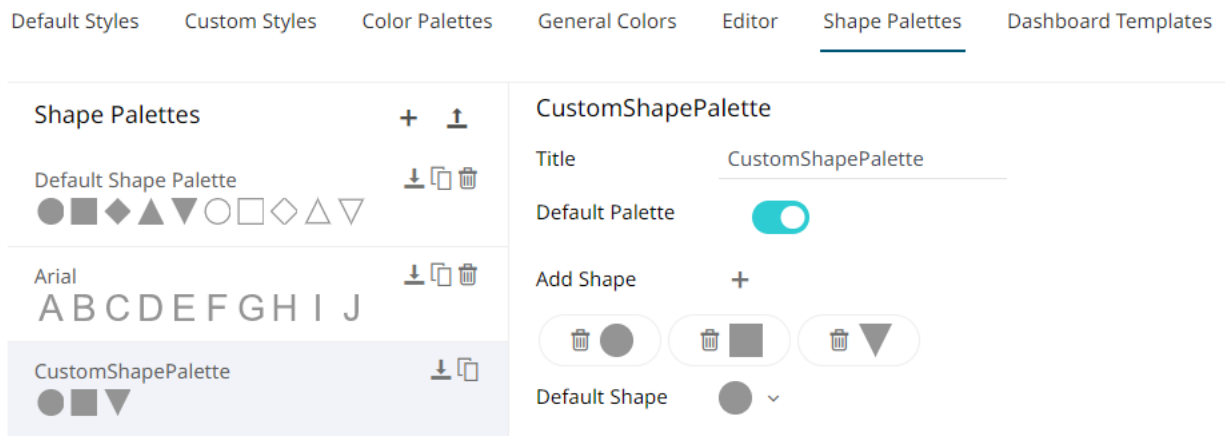
To delete a shape, click its corresponding **Delete** 🗑 icon.

6. Select the *Default Shape* in the drop-down list.

7. Click the **Save** 🖫 Save .

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.
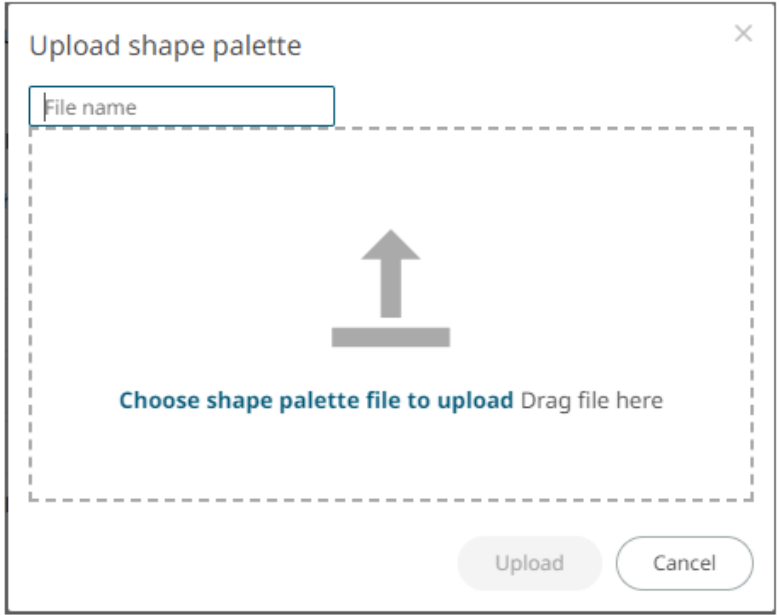
2. To upload a shape palette, either:

   - 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.
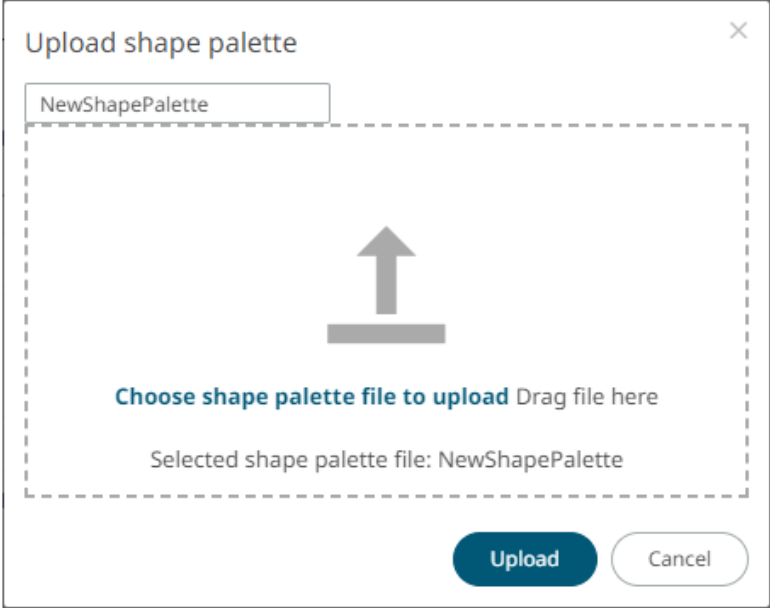
   You can opt to rename the uploaded shape palette.

3. Click **Upload** .

   A notification displays once the file is uploaded.
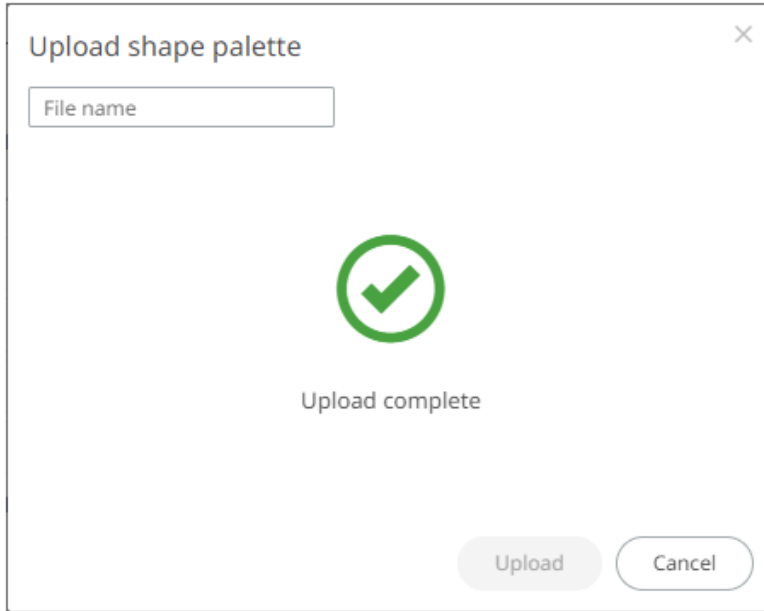
Click to close the dialog. The uploaded shape palette is added in the list.

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 delete 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** 🖫 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 Legacy connector in the given workbook/directory of workbooks.
mockdata	Updates workbook data tables with mock data based on the stored schema.
encrypt	Allows encryption of either a single text or an entire `.properties` file.
decrypt	Allows decryption of either a single text input or an encrypted `.properties` file.

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\Panopticon\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\Panopticon\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 Legacy 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`

Encrypt

Allows encryption of either a single text or an entire `.properties` file.

Option	Description
-t, --text	Text to encrypt.
-p, --properties	Input `Panopticon.properties` file.
-o, --output	Output property file.
-f, --filter	Property filter regex.

Command examples:

❏   .properties file

```
java -jar pcli.jar encrypt -p
/url/share/vizserverdata/Panopticon.properties -o Define.properties -f
.*password
```

❏   text string

```
java -jar pcli.jar encrypt -t passwordName
```

Decrypt

Allows decryption of either a single text input or an encrypted `.properties` file.

Option	Description
-t, --text	Text to encrypt.
-p, --properties	Input `Panopticon.properties` file.

Command examples:

❏   .properties file

```
java -jar pcli.jar decrypt -p /usr/share/vizserverdata/Define.properties
```

❏   text string

```
java -jar pcli.jar decrypt -t a7DUF0EONaFBAqNI2W4NoA==
```

[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 exposes services through a REST API. You can use this for scripting and automation, and other tasks like review query statistics and monitor performance.

NOTE	You can use PCLI for some common tasks like upload a workbook and example workbooks to view server performance too.

There are two API groups: the public API which is being built out starting in version 2022.1, and the legacy API. Going forward, new services will only be added to the public API, and old services may migrate there. Other than that, the main differences are:

The public API	The legacy API
• Is officially supported by Altair	• Is "unsupported" in the sense that we cannot guarantee that an endpoint will stay unchanged or even remain between releases
• Will evolve predictably in the future	
• Is designed specifically for REST	• Was designed when the server had both REST and SOAP APIs, so is a bit cumbersome from a REST perspective
• Has endpoints that begin with `/api`, e.g., **http://localhost:8080/panopticon/api/user/data/profile**	• Has endpoints that begin with `/server/rest`
• Has documentation in OpenAPI 3 (see https://openapis.org) at `/v3/api-docs/public`, e.g., **http://localhost:8080/panopticon/v3/api-docs/public**	• Has documentation in OpenAPI 2 (see https://swagger.io/specification/v2/) at `/v2/api-docs`

• Has a Swagger UI (see https://swagger.io/tools/swagger-ui/) at /swagger-ui.html, e.g., **http://localhost:8080/panopticon/swagger-ui.html**	• Has a Swagger UI at `/swagger-ui.html`, but you need to select the **legacy** definition in the top bar

NOTE	The API documentation endpoints and Swagger UI are disabled by default. You need to set `documentation.enabled=true` in `Panopticon.properties` (and restart the server) to use them. The REST endpoints and services themselves are always enabled. You should never enable the documentation on a production server.

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?workbo
ok=How+To+Actions&dashboard=Scatter+of+Filtered+Universe&part=visualization.S
catterPlot1&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?workbo
ok=How+To+Actions&dashboard=Scatter+of+Filtered+Universe&part=visualization.S
catterPlot1&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+fold
       er%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

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

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 <u>schedule the emailing of dashboard images</u> 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: <u>`server.host`</u>`=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?work
book=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.

DATA LOG ACCESS IN DASHBOARDS

The `subscription.data_log.always_on` server property has a **false** default value. When set to **true**, the data log is always passed from server to client if the user is a Designer or Admin on the server. Previously, the data log would only be passed for workbooks in design mode.

The data log will be passed also when the data request fails. The "**Invalid Configuration**" message shown in the visualization will show a "**Data Log**" button, which will display the relevant logs and error message.

NOTE	The actual passing of runtime exception is currently implemented in the Kdb+ connector only.

The benefit of running a server with `subscription.data.log.always_on=true` is that, the data log is more easily accessed and can be viewed both as success and failure. The data log can also be viewed without having **Write** permissions on the folder where the workbook is used, which is helpful when connection failures need to be examined in production environments where you have restrictions on workbook editing.

NOTE	Viewer users are not able to view the Data Log, only Designers and Admins.

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

NOTE	Set the `documentation.enabled` property in [Panopticon.properties](#) to **true** to view the documentation**.**

Panopticon workbooks can be embedded into existing portals with minimal effort. Open `[tomcat]/panopticon/panopticon-api-doc/index.html` to view the documentation.
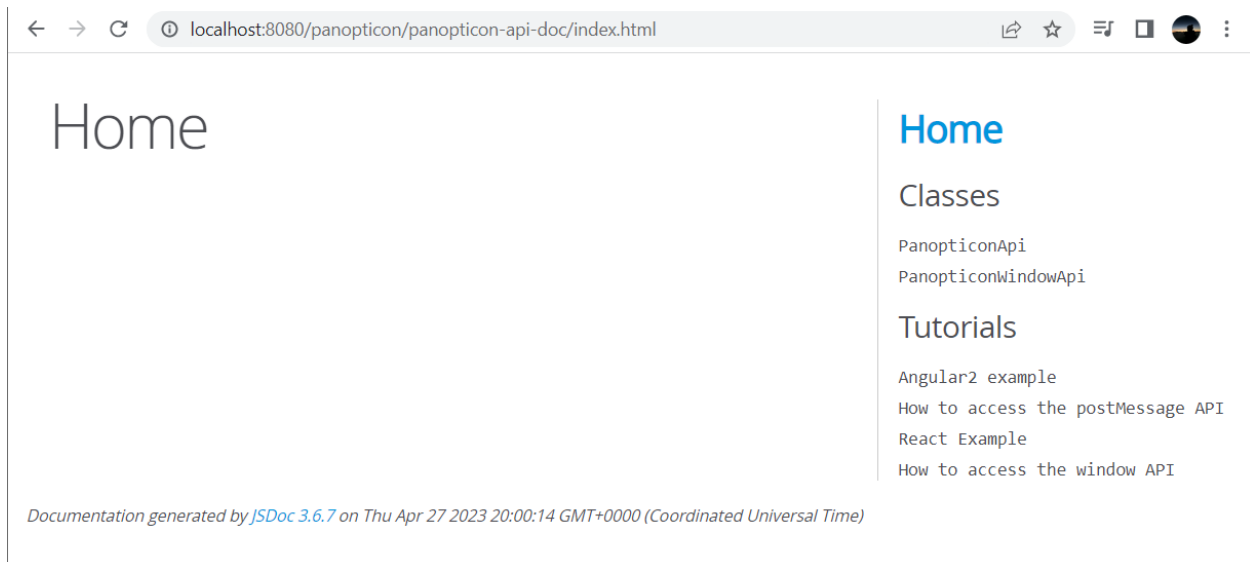
SETTING THE SERVER METRICS PUBLISHER

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

❑ Long polling, WebSocket, and total number of connections

❑ CPU loading percentage

❑ Maximum, size, and used Heap Bytes

❑ Subscription alerts, users, and total

❑ Number of parallel data loading and live threads

❑ Average data load time or refresh rate

You can configure the following properties in the Panopticon.properties file located in the AppData folder or C:\vizserverdata:

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 specified value, the oldest value is removed to make room for the newest one (FIFO). Only used if the metrics.publisher.type is set to **MEMORY**.
Default Value	**100**
Property	Server Metrics
Attribute	metrics.publisher.configuration
Description	Specifies the id for which metric publisher configuration to use.
Default Value	
Property	Server Metrics
Attribute	metrics.publisher.type
Description	Specifies the current metric publisher that is used. Available values are **NONE, MEMORY, FILE, EMAIL, INFLUX_DB, JDBC, KAFKA, KDB, MQTT, REST, TEXT**.
Default Value	**MEMORY**

SETTING THE LOGGING LEVEL

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

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

[19] TROUBLESHOOTING

RESOLVING INSTALLATION ISSUES

Issues are investigated and resolved through investigation and controlled reproduction. Several 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:
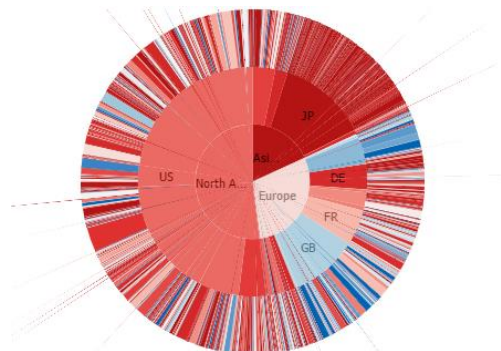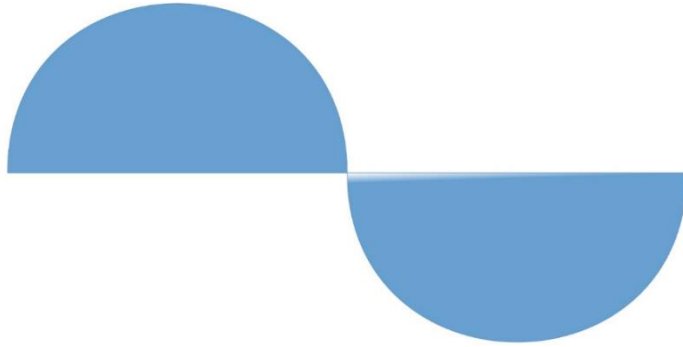
To resolve this issue, follow the steps below to disable Hardware Acceleration in Chrome:

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.

   System

   ☑ Continue running background apps when Google Chrome is closed

   ☐ Use hardware acceleration when available (requires Chrome restart)

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.

MANAGED ALTAIR UNITS LICENSE SSL ERROR

If you encounter the following issue when using Managed Altair Units license:

"SSL_ERROR_SSL
error:14007086:SSL routines:CONNECT_CR_CERT:certificate verify failed
unable to get local issuer certificate"
Detail: SSL/TLS handshake failed

Follow the steps below to resolve this error:

1. Make sure you have installed all updates and are using the latest version of the product. The latest version is always available from the [Marketplace](#).

2. Work with your IT department to create an exception in your proxy for the traffic going to our servers:

   - https://client.hhwu.altair.com

   - https://auth.hhwu.altair.com

   - https://auth.login.solidthinking.com

   - https://auth.admin.altairone.com

   - https://alas.admin.altairone.com
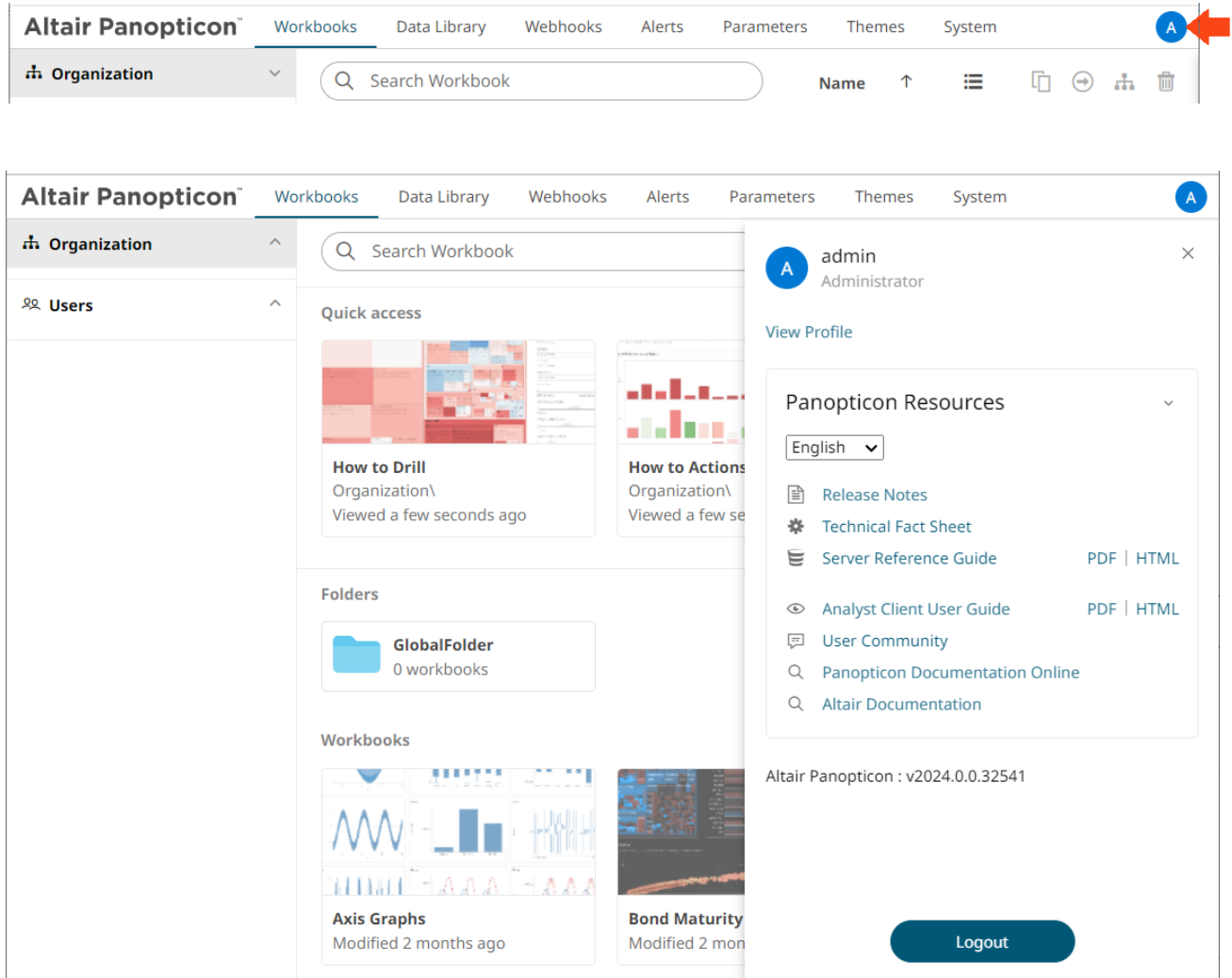
[20] KNOWN ISSUES

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 the user profile icon 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. |
| Altair Documentation | Link to product documentation in the Altair Community page. |

[APPENDIX]

PROPERTIES: PANOPTICON

The majority of configuration options for the server are set in the `Panopticon.properties` file in the `<appdata>` directory (e.g., `C:\vizserverdata\`). If this file does not exist when the server starts, it will create it with all default values. When the server starts after an upgrade, it may add new properties and remove deprecated ones.

You can optionally move sensitive properties like passwords and URLs from this file, where they are stored in clear text, into a file named `Secret.properties` in the same directory. The `Secret.properties` file stores values encrypted, and you can manage it with [PCLI](#). A property can only be defined in one of these files at a time.

The following properties can be set in the property files:

Property	Access
Attribute	`access.administrator.groups`
Description	The role that is mapped to the administrator group.
Default Value	**admin**
Property	Access
Attribute	`access.administrator.users`
Description	Normally administrator access should be handled with the `access.adminstrator.groups` mapping, but for scenarios where the authentication cannot provide roles or you want to make exceptions for specific users, you can list individual usernames in this property. Any user listed here will get administrator access, regardless of their roles. Separate multiple users with the `access.list.delimiter`.
Default Value	
Property	Access
Attribute	`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.designer.users`
Description	Normally designer access should be handled with the `access.designer.groups` mapping, but for scenarios where the authentication cannot provide roles or you want to make exceptions for specific users, you can list individual usernames in this property. Any user listed here will get designer access, regardless of their roles. Separate multiple users with the `access.list.delimiter`.
Default Value	
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	Access
Attribute	`access.viewer.users`
Description	Normally viewer access should be handled with the `access.viewer.groups` mapping, but for scenarios where the authentication cannot provide roles or you want to make exceptions for specific users, you can list individual usernames in this property. Any user listed here will get viewer access, regardless of their roles.

	Separate multiple users with the `access.list.delimiter`.
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	Alert
Attribute	`alert.event.history.days`
Description	Days to keep closed alert events before deletion. **NOTE:** Open alert events are never deleted.
Default Value	**7**
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`
Description	The name of the header that contains all the roles.
Default Value	
Property	Authentication: Header
Attribute	`authentication.header.rolesdynamic`

Description	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:

KEY	VALUE
HEADER_ROLES	designer, watcher
HEADER_COMPANY	industrials, consumers

	Then the roles to create the authentication token will be the following:
	• designer
	• watcher
	• financials
	• role_for_company_industrials
	role_for_company_consumers
Default Value	
Property	Authentication: Header
Attribute	`authentication.header.username`
Description	The name of the header that contains the username
Default Value	
Property	Authentication: Header
Attribute	`authentication.header.validate.token`
Description	If set to **true**, the authentication will validate the token. If set to **false**, the authentication of every request will be based on headers.
Default Value	
Property	Authentication: Logout
Attribute	`authentication.logout.redirect.url`
Description	Takes a URL as a parameter. Clicking the logout button redirects the user to the specified URL.
	If this property is not set, user will be returned to the start page of Panopticon.
Default Value	
Property	Authentication: OAuth 2.0
Attribute	`authentication.oauth2.client.id`
Description	The ID of the OAuth 2.0 client.
Default Value	

Property	Authentication: OAuth 2.0
Attribute	`authentication.oauth2.client.secret`
Description	The secret used by the OAuth 2.0 client.
Default Value	

Property	Authentication: OAuth 2.0
Attribute	`authentication.oauth2.identity.attribute.roles`
Description	The attribute that will be extracted from the *identity response* and used as the role.
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=des`
	`igner,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.method`
Description	The method on how the access token is passed along in the identity request. Supported values are **QUERY**, **BODY**, and **HEADER**.
Default Value	**QUERY**

Property	Authentication: OAuth 2.0
Attribute	`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.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	Bookmark Administration
Attribute	`bookmark.show_shared`

Description	Allows private bookmarking. If set to **true**, all of the users will be able to view each other's bookmarks. If set to **false**, bookmarks will only be viewed by the one who created them.
Default Value	**true**
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. 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 File Path Upload
Attribute	`client.filepath.upload.disabled`
Description	If set to **true**, the **Upload File** option will not be available.
Default Value	**false**
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	Altair AI Hub Connector Endpoint Listing
Attribute	`connector.aihub.client.id`
Description	Application (client) ID assigned at the Altair AI Hub.
Default Value	
Property	Altair AI Hub Connector Endpoint Listing
Attribute	`connector.aihub.client.secret`
Description	Your application's client secret created at the Altair AI Hub.
Default Value	
Property	Altair AI Hub Connector Endpoint Listing
Attribute	`connector.aihub.grant_type`
Description	The grant type to be used to make token request.
Default Value	**refresh_token**
Property	Altair AI Hub Connector Endpoint Listing
Attribute	`connector.aihub.url`
Description	The root URL to Altair AI Hub, where token and endpoint request are sent.
Default Value	
Property	AMPS Connector Custom Authenticator
Attribute	`connector.amps.authenticators`
Description	This property is required when a custom authenticator is needed for AMPS connection. A custom authenticator needs be implemented as java .JAR file. The property excepts a JSON object, where key is fully qualified name of the Authenticator Java class, and values are list of constructor parameter names, e.g., `"{"com.panopticon.examples.amps.AMPSClientAuthenticator":["User","Shared Key"]}"`
Default Value	
Property	Connector Column Generator

Attribute	`connector.column_generator.max_columns`
Description	Controls the maximum number of columns generated by the text- based connectors/parsers. Currently only used by JSON connector and parser.
Default Value	**100**
Property	Connector File Path
Attribute	`connector.common.filepath.link.disabled`
Description	If set to **true**, the *Link to File* option will not be available.
Default Value	**false**
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	OneDrive connector
Attribute	`connector.onedrive.client.id`
Description	Application (client) ID assigned by the Azure administration portal.
Default Value	
Property	OneDrive connector
Attribute	`connector.onedrive.client.secret`
Description	Your application's client secret created in the application registration portal.
Default Value	
Property	OneDrive connector
Attribute	`connector.onedrive.root.url`
Description	The Microsoft Graph API root URL.
Default Value	**https\://graph.microsoft.com/v1.0/me/drive/root**
Property	OneDrive connector

Attribute	`connector.onedrive.tenant.id`
Description	Controls who can sign into the application. The allowed values are: • **common** - for both Microsoft accounts and work or school accounts • **organizations** - for work or school accounts only • **consumers** - for Microsoft accounts only • **tenant** - identifiers such as the tenant ID or domain name
Default Value	
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	Data Store
Attribute	`datastore.connection.schema`
Description	Name of the database schema to be used for creating or managing objects inside database.
Default Value	**dbo**
Property	Data Store
Attribute	`datastore.type`
Description	Controls which data store connector should be used. Valid values are **MonetDB**", **MSSQLServer** and **PostgreSQL.**
Default Value	**MonetDB**
Property	Data Store
Attribute	`datastore.connection.jndi`
Description	JNDI resource name for the connection e.g., **jdbc/MyDB**. More details on how to configure JNDI is at JNDI Connection Details section.
Default Value	
Property	Data Store
Attribute	`datastore.connection.url`

Description	JDBC connection URL for the database e.g., **jdbc:monetdb://localhost:49153/PanopticonDataStore** This property value is discarded If `datastore.connection.jndiproperty` is set.
Default Value	
Property	Data Store
Attribute	`datastore.connection.driverclassname`
Description	Fully qualified Java class name of the JDBC driver used for the connection.
Default Value	**org.monetdb.jdbc.MonetDriver**
Property	Data Store
Attribute	`datastore.connection.username`
Description	Username for the connection. Only required when using connection URL.
Default Value	
Property	Data Store
Attribute	`datastore.connection.password`
Description	Password for the connection. Only required when using connection URL.
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. **NOTE:** When using a Gmail account, you must use an app password to authenticate with the Gmail SMTP server. See Sign in with app passwords for more information.
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	Export File Path
Attribute	`export.filesystem_copy.path`
Description	If set to a directory, whether existing or not, then: • Operations that email CSV, Excel, HTML, Image, PDF, or ZIP files (i.e., scheduled tasks) will also export a copy of the file to the specified path • Operations that generate and return PDF, CSV, or ZIP files (e.g., **Create PDF Report** toolbar button on a dashboard) will also export a copy of the file to the specified path If not set (blank), the server works the same way as before.
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 Managed or Local Altair Units licensing. Set to **true** if you wish to use managed licensing.
Default Value	**false**
Property	Licensing
Attribute	`license.hwu.hosted.authorization.password`
Description	Password to the Altair One account.
Default Value	
Property	Licensing
Attribute	`license.hwu.hosted.authorization.token`
Description	An authorization token generated through the Altair One admin portal. Used to authorize a machine to the managed Altair Units system.
Default Value	
Property	Licensing
Attribute	`license.hwu.hosted.authorization.username`
Description	Username to the Altair One account.

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.use_client_timezone`
Description	Determines how the ALJDK should process the timezone details. If set to **true**, the ALJDK will process the timezone details sent by Panopticon client to the Panopticon server. If set to **false**, the Panopticon server timezone is used.
Default Value	**true**
Property	Licensing
Attribute	`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	**INFO**
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 an 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	Statistics
Attribute	`statistics.accumulated.enabled`
Description	By default, the server accumulates statistics from every run into files in `<appdata>/Statistics/`, e.g., `WorkbookStatistics_Accumulated.json`. You can delete these files if you are not interested in this information, or you can set this property to **false** to disable the accumulation completely.
Default Value	**true**
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 need 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 is scheduled to refresh each X seconds (based of the refresh period of the data table) using the TaskScheduled built into 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.data_log.always_on`
Description	When set to **true**, the data log is always passed from server to client if the user is a Designer or Admin on the server. Previously, the data log would only be passed for workbooks in design mode.
	The data log will be passed also when the data request fails. The "**Invalid Configuration**" message shown in the visualization will show a "**Data Log**" button, which will display the relevant logs and error message.
	NOTE: The actual passing of runtime exception is currently implemented in the Kdb+ connector only.
	The benefit of running a server with `subscription.data.log.always_on=true` is that, the data log is more easily accessed and can be viewed both as success and failure. The data log can also be

	viewed without having **Write** permissions on the folder where the workbook is used, which is helpful when connection failures need to be examined in production environments where you have restrictions on workbook editing.
	NOTE: Viewer users are not able to view the Data Log, only Designers and Admins.
Default Value	**false**
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**

ABOUT PANOPTICON

For more information on Panopticon and other resources, go to https://www.altair.com/panopticon.