Altair Engineering, Inc. makes no representation or warranties with respect to the contents of this manual or the associated software and especially disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Altair reserves the right to revise this publication and make changes from time to time to its contents without obligation to notify anyone of such revisions or changes.

Altair Panopticon Server is offered and is to be used in accordance with a SOFTWARE LICENSE AND MAINTENANCE AGREEMENT. This agreement stipulates that this software be used only in the computer system designated in that agreement. The agreement further stipulates that the customer shall not copy or alter, or permit others to copy or alter, the software or related materials in whole or in part, in any media for any purpose, except to make an archive (back-up) copy or to make a copy as an essential step in the use of the software with the customer's computer.

Altair Engineering, Inc. hereby grants the buyer the right to reprint this documentation for internal uses only. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, for any other purposes, without the prior written permission of Altair Engineering, Inc.

Altair Panopticon Server Installation, Configuration and Troubleshooting Guide v17.4.0
Copyright © 2019 by Altair Engineering, Inc.
All rights reserved. Printed in the U.S.A.
Unpublished - Rights reserved under the copyright law of the United States.
Altair Panopticon Designer, Panopticon Designer, and Altair Panopticon Server are trademarks of Altair Engineering, Inc. Other products mentioned herein may be trademarks or registered trademarks of their respective owners in the United States or other countries.

For U.S. Government End Users, the software is a “Commercial Item(s),” as that term is defined at 48 C.F.R. Section 2.101, consisting of “Commercial Computer Software” and “Commercial Computer Software Documentation,” as such terms are used in 48 C.F.R. Section 12.212 or 48 C.F.R. Section 227.7202, as applicable. Consistent with 48 C.F.R. Section 12.212 or 48 C.F.R. Sections 227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the Altair Engineering, Inc. Software License and Maintenance Agreement.

SALES INFORMATION
US: + 1.800.445.3311
International: + 1.978.441.2200

Sales Email
US: sales@datawatch.com
Europe: sales_euro@datawatch.com
Asia Pacific: sales_apac@datawatch.com

SUPPORT CONTACT INFORMATION
Customer Portal: https://support.datawatch.com
Email: support@datawatch.com
US: +1 800.988.4739
International: +1 978.275.8350
Table of Contents

[1] Introduction ..................................................................................................................... 1

Overview .............................................................................................................................. 1
System Requirements .............................................................................................................. 2
  System hardware Requirements ....................................................................................... 4

[2] Setup ............................................................................................................................. 5

Installation Packages ........................................................................................................... 5
  Panopticon Server Deployment Model ............................................................................... 6
  Server Platforms .............................................................................................................. 7
Environment Promotion Options .......................................................................................... 7
Installation .......................................................................................................................... 8
  Setting Up Panopticon Server on Windows ....................................................................... 8
  Setting Up Panopticon Server on Linux .......................................................................... 10
Licensing .................................................................................................................................. 12
  Using HyperWorks Units License in the Panopticon Server .............................................. 12
  Configuration Properties .................................................................................................. 14
Proxy ...................................................................................................................................... 14
  Apache HTTP Server ......................................................................................................... 15
Load Balancer ...................................................................................................................... 16
  Apache HTTP Server ......................................................................................................... 16
Multiple Instances ............................................................................................................... 17
Backup .................................................................................................................................. 18
Data Access & Caching ......................................................................................................... 18
Upgrade ............................................................................................................................... 20

[3] Additional or Optional Steps ....................................................................................... 22

Replacing Parameter Values with HTTP Headers ................................................................. 22


Usage in SSL Enabled Environments .................................................................................. 25
  Enabling SSL for the Panopticon Server ......................................................................... 25
  Defining a TrustStore ......................................................................................................... 26


Introduction .......................................................................................................................... 28
  Token ................................................................................................................................ 28
Tomcat Realm ....................................................................................................................... 30
  Tomcat User Base ............................................................................................................. 30
LDAP .................................................................................................................................... 33

- Adding Administrators ......................................................................................... 42
- Adding Administrator Group .................................................................................... 47
- Secure Access ........................................................................................................ 48
  - Creating Folders .................................................................................................. 50
  - Creating Sub-folders ........................................................................................... 54
  - Updating Folder or Sub-folder Properties ............................................................. 56
  - Data Level Secure Access .................................................................................... 57

[7] Using Altair Panopticon Server ......................................................................... 58

- Accessing Workbooks ............................................................................................. 58
  - Viewing Workbook History and Republishing ....................................................... 59
- Parameter Value Passing Into the Web Client ......................................................... 61
  - Special Symbols to Pass Parameter Values Into the HTML5 Client ...................... 63
- Uploading Workbooks ............................................................................................. 63
- Exporting to Excel (TSV-format) of Visualizations .................................................. 66
- Toggling Between a Visualization and a Table ....................................................... 67
- Adding, Swapping, or Removing Columns in a Breakdown and Pivot Points ........... 71
- Toggling Between Rows and Columns of a Cross Tab ............................................. 75
- Ad Hoc PDF Report Generation ............................................................................. 76
- Bookmarking .......................................................................................................... 78
  - Context Menu ...................................................................................................... 80

[8] Connectivity and Integration ............................................................................. 82

- Third Party Software Dependency Installation ....................................................... 82
- Database .................................................................................................................. 86
  - Using the Database Connector ............................................................................ 86
  - JDBC driver installation ....................................................................................... 87
  - JNDI Connection Details ...................................................................................... 87
  - Common Databases and their JNDI Configurations .............................................. 88
- R and Python Transform Support ........................................................................... 88
  - R Integration ....................................................................................................... 88
  - Python Integration ............................................................................................... 90
- Load Custom Data Plugins ..................................................................................... 93

[9] Caching ................................................................................................................ 94
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Queryable Cache Configuration</td>
<td>94</td>
</tr>
<tr>
<td>Enabling the Queryable Data Extract</td>
<td>95</td>
</tr>
<tr>
<td>Creating Global Data Extracts</td>
<td>95</td>
</tr>
<tr>
<td>Creating Global Data Extract from InfluxDB</td>
<td>97</td>
</tr>
<tr>
<td>Creating Global Data Extract from JDBC Database</td>
<td>99</td>
</tr>
<tr>
<td>Creating Global Data Extract from a JSON Data File</td>
<td>100</td>
</tr>
<tr>
<td>Creating Global Data Extract from Kx kdb+</td>
<td>103</td>
</tr>
<tr>
<td>Creating Global Data Extract from MS Excel File</td>
<td>104</td>
</tr>
<tr>
<td>Creating Global Data Extract from OneTick Cloud</td>
<td>104</td>
</tr>
<tr>
<td>Creating Global Data Extract from a Text Data File</td>
<td>106</td>
</tr>
<tr>
<td>Creating Global Data Extract from an XML File</td>
<td>109</td>
</tr>
<tr>
<td>Other Data Extracts Operations</td>
<td>112</td>
</tr>
<tr>
<td>View Plugin Subscriptions</td>
<td>113</td>
</tr>
<tr>
<td>View Cache Usage</td>
<td>115</td>
</tr>
<tr>
<td>[10] REST Interface</td>
<td>118</td>
</tr>
<tr>
<td>API</td>
<td>118</td>
</tr>
<tr>
<td>Export Data</td>
<td>118</td>
</tr>
<tr>
<td>CSV</td>
<td>118</td>
</tr>
<tr>
<td>PDF</td>
<td>119</td>
</tr>
<tr>
<td>Excel Workbook</td>
<td>122</td>
</tr>
<tr>
<td>Dashboard Image</td>
<td>123</td>
</tr>
<tr>
<td>Visualization Image</td>
<td>125</td>
</tr>
<tr>
<td>Email Data</td>
<td>127</td>
</tr>
<tr>
<td>PDF</td>
<td>127</td>
</tr>
<tr>
<td>Image</td>
<td>128</td>
</tr>
<tr>
<td>Setting Up Alerts on the Web Client</td>
<td>132</td>
</tr>
<tr>
<td>Sorting Alerts</td>
<td>135</td>
</tr>
<tr>
<td>Searching for Alerts</td>
<td>135</td>
</tr>
<tr>
<td>Enabling Alerts on the Alerts Page</td>
<td>136</td>
</tr>
<tr>
<td>Modifying Alert Settings</td>
<td>136</td>
</tr>
<tr>
<td>Deleting Alerts</td>
<td>138</td>
</tr>
<tr>
<td>Enabling Alerts</td>
<td>140</td>
</tr>
<tr>
<td>Viewing and Managing Alerts for Non-Administrator users</td>
<td>142</td>
</tr>
<tr>
<td>Sample E-mail Alerts</td>
<td>143</td>
</tr>
<tr>
<td>Sample Web Client Alerts</td>
<td>144</td>
</tr>
<tr>
<td>[12] Logging/Monitoring</td>
<td>148</td>
</tr>
<tr>
<td>View Logs</td>
<td>148</td>
</tr>
<tr>
<td>Set File Logging Level</td>
<td>150</td>
</tr>
<tr>
<td>Server Logging</td>
<td>151</td>
</tr>
<tr>
<td>Configuring Server Logs</td>
<td>152</td>
</tr>
<tr>
<td>Configuring Apache Tomcat Logs</td>
<td>152</td>
</tr>
</tbody>
</table>
Using java.util.logging (Default) ................................................................. 154
Audit Logging .................................................................................................. 157
Server Monitoring ......................................................................................... 158
Web Portal Integration ................................................................................... 159

[13] Scheduling Tasks ....................................................................................... 160

Create Task to Send PDF File via Email .......................................................... 160
Create Task to Send HTML File via Email ....................................................... 164
Create Task to Extract Data .............................................................................. 167
Create Task to Send CSV Data via Email .......................................................... 168
Modify a Scheduled Task .................................................................................. 171
Create a Duplicate of a Scheduled Task ............................................................ 171


Clearcache ...................................................................................................... 173
Plugins ............................................................................................................. 174
Publish ............................................................................................................. 174
   Publishing a Workbook to a Panopticon Server .......................................... 174
   Publishing a Workbook to a Folder .............................................................. 175
   Publishing a Workbook Folder to a Panopticon Server ................................ 175
Request ........................................................................................................... 176
Version ............................................................................................................ 177
Help ................................................................................................................. 177
Upgrade ......................................................................................................... 177
Schemify ......................................................................................................... 177
Addvariableids ............................................................................................... 178
ExportDataSource ........................................................................................... 178

[15] Troubleshooting ........................................................................................ 180

Resolving Installation Issues ............................................................................ 180
   Server Log .................................................................................................. 180
Log Variables Displayed as “Null” ................................................................... 181
No Appropriate Protocol Error When Publishing Splunk Data on Panopticon Server .......... 181
Timeout Exception When Loading Data ............................................................ 182
Pie Charts and Shapes Not Displaying Correctly in Chrome ............................... 183
Session Tokens Not Working in Chrome ............................................................ 184

[16] Known Issues ............................................................................................ 185

Out of Memory Exception .................................................................................. 185
Socket Leakage on Windows 2008 R2 / Windows 7 ......................................... 186

[Appendix] ...................................................................................................... 187

Properties: Panopticon ....................................................................................... 187
[1] Introduction

Overview

The Panopticon Server supports the following data connectors:

- Files & URLs: Altair Monarch, HTML, JSON, Altair Monarch Server – Content, MS Excel, SVG, Text, XML
- Database (via JDBC & JNDI)
- General connectivity: SAP Business Objects Universe, Google Analytics, Elasticsearch 5.x, Elasticsearch 6.x, Google Analytics Streaming, JMX, Livy Spark, Panopticon Data Extract, SAP Sybase ESP, Thomson Reuters TREP-RT, Stream Simulator, Zema
- Big Data: XMLA, XMLA On-Demand, Apache Cassandra, IBM Cloudant, Server-side Cloudant, Splunk
- Tick Database: Kx kdb+, Kx kdb+ Discovery, OneMarketData OneTick, OneTick Cloud, InfluxDB
- Predictive: Python, R
- Messaging: JMS (e.g., Apache ActiveMQ), 60East AMPS, AMPS Discovery, Apache Kafka, Kafka Publisher, RabbitMQ, Solace, WebSocket, MQTT, Google Cloud Pub/Sub
- CEP: Kx kdb+tick, Kx kdb+tick Discovery, OneTick CEP, Tibco Streambase CEP, Tibco StreamBase LiveView DataMart

In addition, Panopticon Server:

- Requires that published workbooks using the Database connector to include JNDI connection settings.
- Does not provide example workbooks. These must be published from a connected Panopticon Designer.

NOTES

- 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

The Panopticon Server is supported on these operating systems:

- LINUX (Red Hat)
- LINUX (SUSE)
- Windows 7 (64-bit) – For Development Environments Only
- Windows 10 (64-bit) – For Development Environments Only
- Windows Server 2012 (64-bit)
- Ubuntu Server

The Panopticon Server also requires:

- Java 8 JRE 64-bit
- Apache Tomcat 9.0.1x

If using the Database connector, you must separately have installed JNDI/JDBC drivers for your data repository (databases and other sources) for Panopticon Designer to connect to your data.

NOTES

- The Panopticon Server requires administrative privileges during installation. Administrative privileges are not required after installation is complete.
  For optimal performance, WebSocket is required to be enabled on Tomcat.
- The Panopticon Server does not support Tomcat 7.x, Tomcat 8.0.x, or Tomcat 8.5.x.
- Tomcat 8 uses Non-blocking IO (NIO) threads by default which works very well with WebSockets. The Tomcat 7 Server, on the other hand, uses the Blocking IO.
  For example:
  
  ```
  ```
Enabling WebSocket on Apache Tomcat

There are several Apache Tomcat (Tomcat) specific configuration options for WebSocket.

- When sending WebSocket messages in blocking mode, the default write timeout is 20000 milliseconds (20 seconds). This can be changed by setting the property `org.apache.tomcat.websocket.BLOCKING_SEND_TIMEOUT` in the user properties collection attached to the WebSocket session. The value assigned to this property should be a `Long` and represents the timeout to use in milliseconds. For an infinite timeout, use `-1`.

- If the application does not define a `MessageHandler.Partial` for incoming binary messages, any incoming binary messages must be buffered so the entire message can be delivered in a single call to the registered `MessageHandler.Whole` for binary messages. The default buffer size for binary messages is 8192 bytes. This can be changed for a web application by setting the servlet context initialization parameter `org.apache.tomcat.websocket.binaryBufferSize` to the desired value in bytes.

- If the application does not define a `MessageHandler.Partial` for incoming text messages, any incoming text messages must be buffered so the entire message can be delivered in a single call to the registered `MessageHandler.Whole` for text messages. The default buffer size for text messages is 8192 bytes. This can be changed for a web application by setting the servlet context initialization parameter `org.apache.tomcat.websocket.textBufferSize` to the desired value in bytes.

- The Java WebSocket specification 1.0 does not permit programmatic deployment after the first endpoint has started a WebSocket handshake. By default, Tomcat continues to permit additional programmatic deployment. This behavior is controlled by the `org.apache.tomcat.websocket.noAddAfterHandshake` servlet context initialization parameter. The default may be changed by setting the `org.apache.tomcat.websocket.STRICT_SPEC_COMPLIANCE` system property to `true` but any explicit setting on the servlet context will always take priority.

- When using the WebSocket client to connect to server endpoints, the timeout for IO operations while establishing the connection is controlled by the `userProperties` of the provided `javax.websocket.ClientEndpointConfig`. The property is `org.apache.tomcat.websocket.IO_TIMEOUT_MS` and is the timeout as a `String` in milliseconds. The default is 5000 (5 seconds).

- When using the WebSocket client to connect to secure server endpoints, the client SSL configuration is controlled by the `userProperties` of the provided `javax.websocket.ClientEndpointConfig`. The following user properties are supported:
  - `org.apache.tomcat.websocket.SSL_CONTEXT`
  - `org.apache.tomcat.websocket.SSL_PROTOCOLS`
  - `org.apache.tomcat.websocket.SSL_TRUSTSTORE`
  - `org.apache.tomcat.websocket.SSL_TRUSTSTORE_PWD`

  The default truststore password is `changeit`.

- If the `org.apache.tomcat.websocket.SSL_CONTEXT` property is set, then the `org.apache.tomcat.websocket.SSL_TRUSTSTORE` and `org.apache.tomcat.websocket.SSL_TRUSTSTORE_PWD` properties will be ignored.

---

When using the WebSocket client to connect to server endpoints, the number of HTTP redirects that the client will follow is controlled by the `userProperties` of the provided `javax.websocket.ClientEndpointConfig`. The property is `org.apache.tomcat.websocket.MAX_REDIRECTIONS`. The default value is 20. Redirection support can be disabled by configuring a value of zero.

**SYSTEM HARDWARE REQUIREMENTS**

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

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

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

**Large Scale Deployment**
- 8 x Quad Core CPU Or Equivalent (Hyper Threaded to 64 Cores/Threads)
- 64GB RAM
- 4GB Disk (Available)
- In Memory Caching limited to available Server RAM
[2] Setup

Installation Packages

Altair Panopticon Designer (or Panopticon Designer) and Altair Panopticon Server (or Panopticon Server) include three separate installation packages:

- Altair Panopticon Designer 32-bit
- Altair Panopticon Designer 64-bit
- Altair Panopticon Server

The Panopticon Designer program can be used on its own without being connected to the Panopticon Server component. Panopticon Designer allows users to design and use new Designer workbooks and dashboards and then publish to the web.

Using the Panopticon Server is the best way to make Designer workbooks and dashboards available to large numbers of users.

NOTES

- For optimal scalability and user experience, the Panopticon Server is recommended to be installed on a Server environment. Although basic functionality works on desktops (i.e., Windows 7, 10, etc.) those should be for demo purposes only.
- For optimal performance, it is highly recommended to use Panopticon Designer and Panopticon Server with the same version.

Both components are licensed to provide data connectivity and visualization options.

A typical installation scenario is to provide authors with copies of Panopticon Designer. They develop new workbooks and dashboards based on user requirements and then publish them to the Panopticon Server. Most business users then access the system over the web with no local software installation needed.
NOTES

• Beginning with version 16.4, instead of performing Front-end Server (datawatchvdd-frontend.war) and Back-end Server (datawatchvdd-backend.war) deployment, you need to install a single Panopticon Server package (panopticon.war). Refer to the sections below for more information.

• This document describes how to install the Panopticon Server.

If you need to upgrade your previously installed Panopticon Server, proceed to page 20.

PANOPTICON SERVER DEPLOYMENT MODEL

Full scale Panopticon Server deployment

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

The Panopticon Server exposes web services from both a SOAP interface and a REST interface. These interfaces are used by the Designer and 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 Server security model, such as authentication and configuration of directories in Tomcat.

Furthermore, the Panopticon Server is capable of the following:

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

Refer to the sections below for more information.

**SERVER PLATFORMS**

The Panopticon Designer and the Panopticon Server consist of multiple components such as the following:

<table>
<thead>
<tr>
<th>Components</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designer</td>
<td>The application used to create, manage, and view workbooks.</td>
</tr>
<tr>
<td>Web Client</td>
<td>The Web client is a thin version of the Panopticon Designer and provides the functionality to view a workbook in the browser.</td>
</tr>
<tr>
<td>Panopticon Server</td>
<td>Responsible for managing all the published workbooks and all the resources that go with them. It is also responsible for authentication, authorization, data connections, and transformations.</td>
</tr>
</tbody>
</table>

Refer to [Installation](#) for more information.

**Environment Promotion Options**

You may need to set up multiple environments for Altair Visual Data Discovery. 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
Panopticon Designer supports two environment promotion options:

- Copy the files within the Server’s **vizserverdata** folder between the environments
- Manually publish each workbook to each environment

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 Data Source Names (DSNs). These abstract the location of the data repository from the workbook. The use of DSNs allow each environment to specify the appropriate environment data repository, while the workbooks simply access the repository specified in the DSN.

When copying files between Panopticon Server’s AppData folders, the server cache must also be cleared. To do this, access the Panopticon Server, login, and select the **Settings** tab, then click the **Clear Cache** button.

**Installation**

*NOTES*

If you need to upgrade your previously installed Panopticon Server, proceed to page 20.

**SETTING UP PANOPTICON SERVER ON WINDOWS**

**Steps:**

1. **Extract the contents of the** DatawatchVisualizationServerWAR_<version number>.zip file to a new location.
   
   This zip file will contain the following folder and files:
   - pcli-java folder
   - panopticon.war
   - panopticon.xml
   - tomcat-users_example.xml
   - Panopticon Designer Quick Start
   - Panopticon Server Installation and Troubleshooting Guide
2. **Create the** AppData folder (i.e., **vizserverdata**) and ensure that the user account **Local Service** running Tomcat has read/write and execute permissions to this folder.
   
   Example: c:\vizserverdata
3. **Copy the extracted** panopticon.xml **file into the Tomcat config folder** (\Apache Software Foundation\Tomcat 9.0\conf\Catalina\localhost). This file contains the following information:

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

**NOTES**

Update this file if the path of the environment variable DatawatchVDDAppData is different from c:\vizserverdata or the data folder created in step 2.

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

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

Or alternatively edit the existing tomcat-users.xml file and add the entry:

```xml
<user username="designer" password="designer" roles="user"/>
```

For more complex authentication and user directory options, see section [5] Authentication.

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

7. **Start Tomcat to deploy the .war file.**

   The panopticon folder is extracted in the Tomcat webapps folder:

   ![File Explorer](image)

   - Name: docs
   - Date modified: 11/12/2018 5:22 PM
   - Type: File folder
   - Size: 0
   - Name: host-manager
   - Date modified: 11/12/2018 5:22 PM
   - Type: File folder
   - Name: manager
   - Date modified: 11/12/2018 5:22 PM
   - Type: File folder
   - Name: panopticon
   - Date modified: 18/12/2018 11:10 AM
   - Type: File folder
   - Name: ROOT
   - Date modified: 11/12/2018 5:22 PM
   - Type: File folder
   - Name: panopticon.war
   - Date modified: 18/12/2018 7:27 AM
   - Type: WAR File
   - Size: 104,648 KB

   Also, the CacheData, DataTables, GlobalCaches, Token, and Workbooks folders are generated in the vizserverdata folder along with the panopticon.properties file.
The Panopticon Server will rename the DatawatchBackendServer.properties file to Panopticon.properties when starting the server for the first time.

8. In the Tomcat Properties, set the Initial Memory Pool to 1Gb, and the Maximum Memory Pool to at least 16Gb (unless for a development server, where 8Gb may be appropriate).

9. Specify the license type that will be used. Refer to Licensing for more information.

10. You should now be able to log on to the Panopticon Server using the following:

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

    For example:

    http://localhost:8080/panopticon

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

**SETTING UP PANOPTICON SERVER ON LINUX**

Steps:

1. 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.

2. In the Tomcat config folder (\Apache Software Foundation\Tomcat 9.0\conf\Catalina\localhost) create the panopticon.xml file with the following information:
<?xml version="1.0" encoding="UTF-8"?>
<Context antiJARLocking="true" path="/panopticon">
  <Environment name="DatawatchVDDAppData"
    override="false"
    type="java.lang.String"
    value="/usr/share/vizserverdata" />
</Context>

NOTES

Update this file if the path of the environment variable DatawatchVDDAppData is different from /usr/share/vizserverdata or the data folder created in step 1.

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

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

    Or alternatively edit the existing tomcat-users.xml file and add the entry:

    <user username="designer" password="designer" roles="user"/>

    For more complex authentication and user directory options, see section [5] Authentication.

5. Start Tomcat to deploy the panopticon.war file.

    Also, the CacheData, DataTables, GlobalCaches, Token, and Workbooks folders are generated in the /usr/share/vizserverdata folder along with the panopticon.properties file.

NOTES

The Panopticon Server will rename the DatawatchBackendServer.properties file to Panopticon.properties when starting the server for the first time.

6. In the Tomcat Properties, set the Initial Memory Pool to 1Gb, and the Maximum Memory Pool to at least 16Gb (unless for a development server, where 8Gb may be appropriate).

7. You can also opt to install Java data connector’s dependencies.

8. Specify the license type that will be used. Refer to Licensing for more information.

9. You should now be able to log on to the Panopticon Server 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.

**Licensing**

Licensing within the Panopticon Server supports two license types:

- a volume-based XML file (named DatawatchLicense.xml) which is used to store all license information for a specific customer
  
  The Panopticon Server allows the systems administrator to copy the license file to the designated AppData folder (i.e., c:\vizserverdata).

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

The license file type you will use is delivered separately from the installation packages. Also, license files are required for both the Panopticon Designer and Panopticon Server components.

**NOTES**

If you install Panopticon Designer for stand-alone use, it will request activation through entering a separate license file (volume-based or HyperWorks Units) when using it for the first time. If you install Panopticon Designer for use with the Panopticon Server component, the Panopticon Designer software will retrieve its license automatically from the Panopticon Server.

**USING HYPERWORKS UNITS LICENSE IN THE PANOPTICON SERVER**

Before using the HyperWorks Units license type in the Panopticon Server, it is required to configure certain properties in the Panopticon.properties file located in the Appdata folder or c:\vizserverdata:

<table>
<thead>
<tr>
<th>Property</th>
<th>Service authentication level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>authentication.required</td>
</tr>
<tr>
<td>Description</td>
<td>The property that will make the authentication required. It will force the user to login in order to use any of the services provided by the server. Must be set to true.</td>
</tr>
<tr>
<td>Default Value</td>
<td>false</td>
</tr>
<tr>
<td>Property</td>
<td>Licensing</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>license.hwu.operating.system</td>
<td>The operating system where the Panopticon Server is installed. Possible values are: WIN_X86, WIN_X64, MAC, LINUX_X64, or LINUX_ARM64</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> If the Java bitness (e.g., 32-bit) is different from the operating system (e.g., 64-bit), it is recommended to add the Java bitness in this property (e.g., WIN_X86).</td>
</tr>
<tr>
<td>property</td>
<td>License</td>
</tr>
<tr>
<td>attribute</td>
<td>license.hwu.uri</td>
</tr>
<tr>
<td>description</td>
<td>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 used, they should be separated by ‘;’.</td>
</tr>
<tr>
<td>notes</td>
<td>Multiple License Servers are not supported when the Panopticon Server is on a Linux machine.</td>
</tr>
<tr>
<td></td>
<td>If value is not set in the panopticon.properties, the environment variable ALTAIR_LICENSE_PATH serves as the backup path and will be used.</td>
</tr>
<tr>
<td>property</td>
<td>Licensing</td>
</tr>
<tr>
<td>attribute</td>
<td>license.hwu.version</td>
</tr>
<tr>
<td>description</td>
<td>Value must match the license version found in the HyperWorks Units license file.</td>
</tr>
<tr>
<td>property</td>
<td>Licensing</td>
</tr>
<tr>
<td>attribute</td>
<td>license.mode</td>
</tr>
<tr>
<td>description</td>
<td>The license mode. Possible values are: FILE or HWU. Must be set to HWU.</td>
</tr>
</tbody>
</table>

For example:

```plaintext
authentication.required=true
license.hwu.operating.system=WIN_X64
license.hwu.uri=6200@192.168.5.51;6200@192.168.5.52
license.hwu.version=19.0
license.mode=HWU
```
NOTES

Alerts and scheduled tasks are leveled towards each other. Regardless of the number of alerts or scheduled tasks a user creates, only two HyperWorks Units licenses will be checked out.

These units are separate from the units that are checked out for a user of the server. For example, if a user is logged on to the server (two units) and starts an alert (two units), a total of four units are checked out. If the user then starts two more alerts and a scheduled task, the total number of checked out units will still be four. If the user logs out without shutting off any alerts, two units will remain checked out.

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 the Panopticon Server. There are a variety of proxies available. One of the most commonly used proxies is Apache HTTP Server with the proxy module. Refer to the section below on how to setup an Apache HTTP Server with Proxy functionality.
APACHE HTTP SERVER

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

1. Download the Apache HTTP Server from the official webpage: 
https://httpd.apache.org/download.cgi

2. Unzip and copy the files to a folder.

3. Configure the proxy by opening the httpd.conf file in the conf folder.

4. Update the SRVROOT variable. The value must be updated to the file location of the Apache HTTP server.

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

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

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

6. Configure the logic for the proxy and how requests should be passed. There are three different configuration setups. These are required for the requests to the Panopticon server to be passed corrected.

```
<VirtualHost *:10080>
  ServerAdmin webmaster@localhost
  ProxyPreserveHost On
  ProxyPass /panopticon http://localhost:8080/panopticon
  ProxyPassReverse /panopticon http://localhost:8080/panopticon
  ProxyPass /panopticon/event http://localhost:8080/panopticon/event
  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.
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 the Panopticon Server has on the load balancer is that it supports persistence or stickiness. This means that the proxy will establish a user session and ensure that the user continues to use the same computer resource.

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

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

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

APACHE HTTP SERVER

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

1. Download the Apache HTTP Server from the official webpage: [https://httpd.apache.org/download.cgi](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.
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.

```plaintext
<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 the software (Panopticon Designer & Panopticon Server) 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.
To deploy multiple Designers, a zip can be retrieved onto a local drive with execute permissions, unblocked and unzipped. The **ExDesigner.exe** can then be executed to start.

In the case of regression testing multiple separate Panopticon Designer versions can be run in parallel and configured to connect and publish to matching Panopticon Server instances.

## Backup

The Panopticon Server 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 & Caching

Panopticon assumes in general that data is never at rest. Data is assumed to be too big to simply load all into memory, and is either:

- Subscribed against
- Polled (Automatically refreshed on a defined period)

This means either:

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

Consequently, for direct access Panopticon Designer 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 the queryable data extract.
Consequently:

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

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

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

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

1. Stop Tomcat.
2. Moving forward, delete the existing `webapps\panopticon.war` file.
   
   If your previous Panopticon Server is from version 16.3.0 and below, delete:
   
   • `webapps\datawatchvdd-frontend.war`
   
   • `webapps\datawatchvdd-backend.war`

3. Moving forward, delete the deployed application: `webapps\panopticon`
   
   If your previous Panopticon Server is from version 16.3.0 and below, delete:
   
   • `webapps\datawatchvdd-frontend`
   
   • `webapps\datawatchvdd-backend`

4. Moving forward, delete the cache from the working folder (for example):
   
   `work\Catalina\localhost\panopticon`
   
   If your previous Panopticon Server is from version 16.3.0 and below, delete:
   
   • `work\Catalina\localhost\datawatchvdd-frontend`
   
   • `work\Catalina\localhost\datawatchvdd-backend`

5. Delete the `DefaultSettings.xml` file located in the Appdata folder (i.e., `c:\vizserverdata`).

6. If your previous Panopticon Server is from version 16.3.0 and below, rename `datawatchvdd-backend.xml` to `panopticon.xml` (located in `\Apache Software Foundation\Tomcat 9.0\conf\Catalina\localhost`). Also, delete `datawatchvdd-frontend.xml`. Otherwise, proceed to step 7.

7. Deploy the new `panopticon.war` file by copying it to the Tomcat `webapps` folder.

8. Ensure that the following properties from the previous Front-end Server properties file (`DatawatchFrontendServer.properties`) are transferred to the latest properties file (`Panopticon.properties`), located in the Appdata folder:

   • `email.host`
   
   • `email.port`
   
   • `email.security.mode`
   
   • `export.image.width`
   
   • `export.image.height`
   
   • `client.data.load.transport`
   
   • `client.webgl.enabled`
For example:

EmailHost= smtp.office365.com
EmailPort=587
EmailSecurityMode=TLS
EmailImageWidth=1024
EmailImageHeight=768
DataLoadTransport=Websocket
WebGlEnabled=true

[3] Additional or Optional Steps

Replacing Parameter Values with HTTP Headers

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

- **Incoming Parameters** are parameters sent to the Panopticon Server 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. 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 value 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 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:

<table>
<thead>
<tr>
<th>Property</th>
<th>Request parameter mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td><code>request.operation.parameters.mapping.required</code></td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
<tr>
<td>Default Value</td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Request parameter mapping</td>
</tr>
<tr>
<td>Attribute</td>
<td><code>request.operation.parameters.mapping.optional</code></td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
<tr>
<td>Default Value</td>
<td></td>
</tr>
</tbody>
</table>
### Attribute

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>request.operation.parameters.mapping.entry.delimiter</code></td>
<td>The delimiter that separates the configuration entries. This property will only affect incoming parameters.</td>
<td><code>, (Comma)</code></td>
</tr>
</tbody>
</table>

### Property

**request parameter mapping**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>request.operation.parameters.mapping.value.delimiter</code></td>
<td>The delimiter that separates the parameter name and the Header name. This property will only affect incoming parameters.</td>
<td><code>: (Colon)</code></td>
</tr>
</tbody>
</table>

The following properties can be configured to update outgoing parameters:

<table>
<thead>
<tr>
<th>Property</th>
<th>Response parameter mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td><code>response.operation.parameters.mapping.required</code></td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
<tr>
<td>Default Value</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Response parameter mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td><code>response.operation.parameters.mapping.optional</code></td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
<tr>
<td>Default Value</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Response parameter mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td><code>response.operation.parameters.mapping.entry.delimiter</code></td>
</tr>
<tr>
<td>Description</td>
<td>The delimiter that separates the configuration entries. This property will only affect incoming parameters.</td>
</tr>
<tr>
<td>Default Value</td>
<td><code>, (Comma)</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Response parameter mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td><code>response.operation.parameters.mapping.value.delimiter</code></td>
</tr>
<tr>
<td>Description</td>
<td>The delimiter that separates the parameter name and the Header name. This property will only affect incoming parameters.</td>
</tr>
<tr>
<td>Default Value</td>
<td><code>: (Colon)</code></td>
</tr>
</tbody>
</table>

### Example

This section describes how incoming parameters are replaced with Header values. For example, the Panopticon Server is required to update parameters `uid` and `uname`. 
The request will fail if the required Headers are not present in the incoming request.

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

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

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

Configurations:

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

Usage in SSL Enabled Environments

ENABLING SSL FOR THE PANOPTICON SERVER

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

Follow the steps below to configure SSL for the Panopticon Server.

Steps:

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

2. Create a keyStore file to store the private key and self-signed certificate used to identify the server:

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

   NOTES

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

3. Add an SSL HTTP/1.1 Connector entry in $CATALINA_BASE/conf/server.xml
4. Disable unencrypted server access by commenting out the default HTTP connector for port 8080.

```xml
-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 --

<Connector
    port="8443" maxThreads="200"
    scheme="https" secure="true" SSLEnabled="true"
    keystoreFile="conf/keystore.jks"
    keystorePass="keystorepassword"
    clientAuth="false" sslProtocol="TLS"/>

-->

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

   https://localhost:8443/panopticon

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


DEFINING A TRUSTSTORE

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

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

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

Steps:

1. Create a new keyStore called truststore:
   
   keytool -genkey -alias truststore -keyalg RSA -keystore truststore.jks

2. Export a certificate from a keyStore:
   
   keytool -export -keystore keystore.jks -alias myalias -file cert.cer

3. Import the certificate into the trustStore:
   
   keytool -import -trustcacerts -alias myalias -file cert.cer -keystore
truststore.jks

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

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

- **On Windows, setenv.bat:**
  ```
  set JAVA_OPTS=-Djavax.net.ssl.trustStore="C:/location/to/truststore/truststore.jks"
  ```

- **On Linux, setenv.sh:**
  ```
  export JAVA_OPTS="\$JAVA_OPTS -Djavax.net.ssl.trustStore='/location/to/truststore/truststore.jks'";
  ```

Introduction

The Panopticon Server 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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication.role</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>authentication.required</td>
<td>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.</td>
<td>false</td>
</tr>
<tr>
<td>authentication.type</td>
<td>The type of authentication that should be used when authenticating the user. The property allows the following values: BASIC, FILTER, HEADER, SAML, WINDOWS.</td>
<td>BASIC</td>
</tr>
<tr>
<td>authentication.domain</td>
<td>The default domain information for user authentication.</td>
<td></td>
</tr>
</tbody>
</table>

The web user interface supports all of the authentication mechanisms that are listed in this chapter. However, the Panopticon Designer only supports certain authentication mechanisms such as listed below:

- Tomcat Realm
- LDAP
- Active Directory
- Windows

Refer to the sections below for more information.

TOKEN

A web token is used when the user has successfully logged into the Panopticon Server when using one of the following authentication types: BASIC, SAML, or WINDOWS. The token is used to identify the user and represent the user’s ongoing session. This is done to prevent user credentials being sent between the user and server more than necessary.
The token is returned from the Panopticon Server in the form of a cookie when the user has been authenticated. By default (false), the cookie will be stored in the browser as https cookie and is accessible to the JavaScript.

The token can be configured differently to suit your needs and requirement. The token can be configured to be valid at a certain amount of time, if it can refresh itself, if it should be persistent or if it should only last for a user session (while the browser is still open), and/or it can be stored as an HttpOnly cookie. All this can be configured in the `panopticon.properties`. The table below lists all available token properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication.token.persistent</td>
<td>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.</td>
<td>PERSISTENT</td>
</tr>
<tr>
<td>authentication.token.refreshable</td>
<td>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.</td>
<td>true</td>
</tr>
<tr>
<td>authentication.token.refreshable.scope</td>
<td>This property determines who can refresh a token: ALL or CLIENT. ALL means that both the Web client and the internal subscriptions can refresh a token for a user. This ensures that an internal subscription will always be authenticated. CLIENT means that only the Web client can refresh the token. This prohibits the server from refreshing a token for an internal subscription. Therefore, it is beneficial in ensuring a user interaction is required to refresh the token. This is recommended when there are more security constraints and a short lifespan on the token.</td>
<td>ALL</td>
</tr>
<tr>
<td>authentication.token.secret</td>
<td>The secret is used to sign the token. The secret will be auto-generated when the server starts for the first time. <strong>NOTE:</strong> This value should be kept a secret.</td>
<td>Auto-generated</td>
</tr>
<tr>
<td>authentication.token.validity.seconds</td>
<td>The number of seconds that the token should be valid.</td>
<td>604800</td>
</tr>
<tr>
<td>authentication.token.cookie.httponly</td>
<td>This property determines how the browser will treat the cookie. If set to true, the cookie will be stored in the browser as an HttpOnly cookie and will not be available to the JavaScript. If set to false (default), the cookie will be stored in the browser as a regular cookie.</td>
<td>false</td>
</tr>
</tbody>
</table>
Tomcat Realm

The Panopticon Server 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.

The Panopticon Server 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:

```
authenticatıon.type=BASIC
```

NOTES

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">
</Realm>
```

TOMCAT USER BASE

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

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

The following Realm should added in the server.xml file in the Apache Tomcat conf folder:

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

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

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

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

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

By adding these two lines you have achieved the following:

- Created a new role named user
- Created a new user with username designer and password designer
- Assigned the newly created user the role user

NOTES

- Authentication towards a Tomcat Realm (i.e., Tomcat users, LDAP, AD) in Tomcat 8.5.28 is not supported. This has been supported in all the previous and the succeeding versions.
- A sample tomcat-users_example.xml is provided in the DatawatchVisualizationServerWAR_<version number>.zip file.

Tomcat Memory Configuration

NOTES

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. You can either create a file named setenv.bat (for Windows) or setenv.sh (for Linux).
3. Place them in the Tomcat bin folder.
4. Set the minimum and maximum heap size with the JVM \texttt{-Xms} and \texttt{-Xmx} parameters. A minimum of 1 GB is recommended. For example:

**For Windows:**

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

**For Linux:**

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

**NOTES**

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. Set the maximum limit of the PermGen (Permanent Generation) memory space to a size larger than the default. The default value (64 MB) is not enough for enterprise applications. 256 MB is recommended in most cases. You can set the PermGen maximum limit with the following JVM parameter:

\texttt{-XX:MaxPermSize=256m}

**NOTES**

If the PermGen space maximum is too low, OutOfMemoryError: PermGen space errors may occur.

6. Save the file.

7. Restart the Tomcat service to apply the increase in the heap size and the new Permgen maximum limit.

Another option in setting the heap size is through the System Variables. Follow these steps:

1. Stop Tomcat.

2. Go to System Environment variables. (Right-click \textbf{Computer} > \textbf{Properties} > \textbf{Advanced System Parameters} > \textbf{Environment Variables}.) The \textit{System Properties} dialog displays.

3. Select the \textbf{Advanced} tab and click \textbf{Environment Variables}.

4. Click \textbf{New} under the \textbf{System Variables} section.

5. Define the following:

   - Variable Name: CATALINA_OPTS
- Variable value: -Xms512m -Xmx2g

6. Click OK then OK again.
7. Restart the Tomcat service.

**LDAP**

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

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

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

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

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

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

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

```xml
<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.

**NOTES**

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

**ACTIVE DIRECTORY**

The Panopticon Server can be configured to authenticate towards an Active Directory server. The Panopticon Server 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.

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

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.

Windows Authentication

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

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

- `authentication.type=WINDOWS`

NOTES

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

SAML

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

The Panopticon Server will redirect the user back to the IdP upon a logout request. The IdP logout service should then invalidate the SAML token.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication.saml.assertion.roles</td>
<td>User attribute for roles configured in the IdP.</td>
</tr>
<tr>
<td>authentication.saml.assertion.username</td>
<td>User attribute for username configured in the IdP.</td>
</tr>
<tr>
<td>authentication.saml.assertionconsumerurl</td>
<td>The URL to the Panopticon assertion consumer service. \URL: [Protocol]://[Host]:[Port]/[Context]/serve\r\n</td>
</tr>
<tr>
<td>authentication.saml.callback.url</td>
<td>Relay state.</td>
</tr>
<tr>
<td>authentication.saml.certificate.name</td>
<td>The name of the certificate used to validate signature.</td>
</tr>
<tr>
<td>authentication.saml.certificate.password</td>
<td>The password of the certificate used to validate signature.</td>
</tr>
<tr>
<td>authentication.saml.challenge.required</td>
<td>Determines whether the IdP-first authentication with SAML is enabled or not. To enable, set this property to false.</td>
</tr>
<tr>
<td>authentication.saml.identityprovider.logouturl</td>
<td>The URL to the IdP logout service.</td>
</tr>
<tr>
<td>authentication.saml.identityprovider.url</td>
<td>The URL to the IdP login service.</td>
</tr>
<tr>
<td>authentication.saml.keystore.file</td>
<td>The location of the Keystore file that contains the certificate.</td>
</tr>
<tr>
<td>authentication.saml.keystore.password</td>
<td>The password to the Keystore file.</td>
</tr>
<tr>
<td>authentication.saml.redirect</td>
<td>Redirects the user back to the Panopticon Server URL. This is mainly used with a proxy. In which case, the Panopticon Server does not know the endpoint which the user is going towards to, and therefore cannot redirect the user back to the Overview page. This can be left blank.</td>
</tr>
<tr>
<td>authentication.saml.serviceprovider.id</td>
<td>The ID of the service provider configured in the IdP.</td>
</tr>
</tbody>
</table>

**OAuth 2.0**

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

Note that OAuth 2.0 does not normally provide support on how to authenticate the user, the Panopticon Server will only know if the user is authorized or not. To authenticate the user, Panopticon Server can be configured to use a REST service to extract the user identity with an access token retrieved from the OAuth 2.0 provider. In addition to the standard OAuth 2.0 configurations, the server includes properties (i.e., authentication.oauth2.*) that are specifically used to extract the user details.
- `authentication.type=OAUTH2`

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>authentication.oauth2.client.id</code></td>
<td>The ID of the OAuth 2.0 client.</td>
</tr>
<tr>
<td><code>authentication.oauth2.client.secret</code></td>
<td>The secret used by the OAuth 2.0 client.</td>
</tr>
<tr>
<td><code>authentication.oauth2.identity.attribute.username</code></td>
<td>The attribute that will be extracted from the identity response and used as the username.</td>
</tr>
<tr>
<td><code>authentication.oauth2.identity.url</code></td>
<td>The URL to the REST service that provides details about the authenticated user.</td>
</tr>
<tr>
<td><code>authentication.oauth2.login.callback.url</code></td>
<td>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 the Panopticon Server.</td>
</tr>
<tr>
<td><code>authentication.oauth2.login.response.type</code></td>
<td>The response type. The only response type that is currently supported is <strong>CODE</strong>. The value can also be left blank.</td>
</tr>
<tr>
<td><code>authentication.oauth2.login.scope</code></td>
<td>The requested scope. The field can be left blank.</td>
</tr>
<tr>
<td><code>authentication.oauth2.login.state</code></td>
<td>The requested state. The field can be left blank.</td>
</tr>
<tr>
<td><code>authentication.oauth2.login.url</code></td>
<td>The URL to the OAuth 2.0 login resource.</td>
</tr>
<tr>
<td><code>authentication.oauth2.logout.url</code></td>
<td>The URL to the OAuth 2.0 logout resource. This field can be left blank.</td>
</tr>
<tr>
<td><code>authentication.oauth2.redirect</code></td>
<td>Redirects the user back to the Panopticon Server URL. This is mainly used with a proxy. In which case, the Panopticon Server does not know the endpoint which the user is going towards to, and therefore cannot redirect the user back to the Overview page. This can be left blank.</td>
</tr>
<tr>
<td><code>authentication.oauth2.token.method</code></td>
<td>The method on how the token should be retrieved. Supported values are <strong>QUERY</strong>, <strong>BODY</strong>, and <strong>HEADER</strong>.</td>
</tr>
<tr>
<td><code>authentication.oauth2.token.url</code></td>
<td>The URL to the OAuth 2.0 token resource.</td>
</tr>
</tbody>
</table>
EXAMPLE

authentication.oauth2.client.id=ClientId
authentication.oauth2.client.secret=ClientSecret
authentication.oauth2.identity.attribute.username=name
authentication.oauth2.identity.url=https://oauth2/me
authentication.oauth2.login.response.type=CODE
authentication.oauth2.login.scope=
authentication.oauth2.login.state=
authentication.oauth2.logout.url=https://oauth2/authorize
authentication.oauth2.redirect=
authentication.oauth2.token.method=QUERY
authentication.oauth2.token.url=https://oauth2/access_token
authentication.type=OAUTH2

Filter

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

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

NOTES

When there is no need for the Login button, it can be hidden. This configuration is also made in the panopticon.properties by setting the global.authentication.button.hidden=true
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 the Panopticon Server instead of the original incoming HTTP request.

   ```java
   import org.apache.catalina.realm.GenericPrincipal;
   import org.apache.catalina.users.MemoryUser;
   import javax.servlet.http.HttpServletRequest;
   import javax.servlet.http.HttpServletRequestWrapper;
   import java.security.Principal;
   
   public class FilterRequestWrapper extends HttpServletRequestWrapper {
       private final GenericPrincipal principal;
       
       public FilterRequestWrapper(final HttpServletRequest request, final GenericPrincipal principal) {
           super(request);
           this.principal = principal;
       }
       
       @Override
       public Principal getUserPrincipal() {
           return principal;
       }
       
       @Override
       public boolean isUserInRole(final String role) {
           if (principal != null) {
               return principal.hasRole(role);
           }
           return super.isUserInRole(role);
       }
   }
   ```

2. Create a custom filter. The filter will create a new Principal which includes both the credentials and the groups/roles for the user.
   
   In this example, the class `GenericPrincipal` contains username, password, and groups. The Panopticon Server 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 the Panopticon Server.

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

public class ExampleFilter implements Filter{
    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
    }

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

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

    @Override
    public void destroy() {
    }
}
```

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

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

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

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

# Header

It is possible to use a web-facing Panopticon Server behind a proxy server that will handle the authentication of users. The proxy server forwards the name of the user and roles to the Panopticon Server 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 the Panopticon Server with regards to the validity of the username. The authorization on workbooks and administration will work as usual.

To activate the Header authentication, add or update the following properties in the panopticon.properties file:

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

Adding Administrators

You can assign users that were defined in Tomcat User Base Realm, LDAP, or Active Directory, as administrators in the System page.

Steps:

1. Log on to the Panopticon Server using your authentication credentials (i.e., designer/designer).

   ![Login to Panopticon Server](image)

   The protected version of the server displays the tabs where you can perform administrator roles.

2. Click the System tab.

3. Enter a user (e.g., John) in the Users box, this enables the Add button.
4. Click **Add**.

Since you logged on as **designer**, it now has no administrator privileges. A message box is displayed.

![Message Box]

Click **OK**.

The other tabs are no longer displayed for **designer**.

![Screenshot]

**John** now has administrator rights while the rest of the users in **tomcat-users.xml** will only be able to open workbooks.
NOTES

There are three tabs available for non-Administrator users:

- Workbook
  Lists available folders and uploaded or published workbooks that can be opened and analyzed.

- Alerts
  Refer to [Alerting] for more information on setting up and managing alerts.

- Profile
  Where the e-mail of the user or group who will receive the alerts are defined.

5. Log out as designer and log on as John. It is now added in the list.
### Administrators

<table>
<thead>
<tr>
<th>Users</th>
<th>Groups</th>
</tr>
</thead>
</table>

### Logging

File logging level: **FINEST**

### Server Information

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Windows 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java version</td>
<td>1.8.0_171</td>
</tr>
<tr>
<td>Java vendor</td>
<td>Oracle Corporation</td>
</tr>
<tr>
<td>Tomcat</td>
<td>Apache Tomcat 9.0.13</td>
</tr>
<tr>
<td>Tomcat version</td>
<td>9.0.13.0</td>
</tr>
<tr>
<td>Total memory (MB)</td>
<td>2654</td>
</tr>
<tr>
<td>Max memory (MB)</td>
<td>7607</td>
</tr>
<tr>
<td>Free memory (MB)</td>
<td>490</td>
</tr>
</tbody>
</table>
Also, the Administrators.txt file is added in the AppData folder (i.e., c:\vizserverdata). John is added in the list.

6. You may opt to delete an administrator by clicking . A confirmation message displays.
7. Click **Yes**.

**NOTES**

If for example, the only administrator in the list is deleted (e.g., **John**), then all of the users defined in Tomcat will have administrator privileges again.

---

**Adding Administrator Group**

You can assign roles or groups that were defined in **Tomcat User Base Realm**, **LDAP**, or **Active Directory**, as administrator groups on the **System** tab.

**NOTES**

It is recommended to have administrator users already added on the **System** tab before adding administrator groups.

For example:

![User List](image)

**Steps:**

1. Log on as an administrator (e.g., **John**) and click **System** tab.
2. Enter a group name in the **Groups** box (e.g., **Administrators**), this enables the **Add** button.

![Group List](image)

3. Click **Add**. The group is added in the list.

![Updated User List](image)

Also, the **AdministratorGroups.txt** file is added in the **AppData** folder (i.e., c:\vizserverdata).
This means that all of the users that are assigned

4. You may opt to delete an administrator group by clicking \( \times \). A confirmation message displays.

5. Click **Yes**.

## Secure Access

Panopticon workbooks that are published to the Panopticon Server can be secured so that:

- Only specified authenticated users can access the workbook.
- Only specified authenticated users can see restricted access within the workbook.

The workbook is secured at the publication stage by specifying the users and groups that should be granted or denied access, which are defined in the group and user directories configured on the Server.
NOTES

Beginning with version 16.1.0, new workbooks must be published to a folder or sub-folder 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 Server versions.

A folder access restriction file (GroupAccessRestrictions.xml) is applied to all the published workbooks in a folder.

Note that there can be sub-folders or subgroups.

This is beneficial in maintaining and assigning access restrictions especially if there are numerous workbooks that should have the same authentication or in implementing to workbooks that should only be shared and accessed within a certain company’s department (e.g., Finance, IT, Business, etc.)
In addition, the following operations can be defined in the GroupAccessRestrictions.xml:

- Allowed read groups or users
- Denied read groups or users

Proceed to the sections below for more information.

**CREATING FOLDERS**

**Steps:**

1. On the **Workbooks** tab, click the button and select **Create Folder** in the context menu.

   ![Create Folder button](image)

   The **Create Folder** dialog displays.

   ![Create Folder dialog](image)

   Enter a **Folder Name**.

2. Proceed to defining the Authorization to **Allowed** or **Denied** groups and users.

3. Click **Create**.

   The new folder is displayed on the expanded Folder hierarchy list and on the Root Folders/Workbooks list.
NOTES

Folders and sub-folders can be deleted as long as they do not contain published workbooks.

Adding Groups and Users with Allowed Authorization

Steps:
1. Under the Allowed section, click the Add link beside Groups.

2. Enter the name of the Group and click Add.
   The new group is added in the list.
3. Proceed to adding users that are included in the Group by clicking the Add link beside the Users section.

4. Enter the user’s name and click Add.

5. Click Create to save the changes.
NOTES

Users must first login to the Panopticon Server before viewing the workbook.

Once authenticated the username will be displayed on the top right, and all Panopticon workbooks and folders that the user has permission to access will be displayed.

Adding Groups and Users with Denied Access

Steps:

1. Under the *Denied* section, click the **Add** link beside *Groups*.

2. Enter the name of the Group and click **Add**.

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

3. You can opt to proceed to adding users that will not have access to the workbooks in this folder by clicking the **Add** Link beside *Users*.
4. Enter the name of the User and click **Add**.

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

5. Click **Create** to save the changes.

**CREATING SUB-FOLDERS**

**Steps:**

1. To create sub-folders, you can either click a folder:
   - on the expanded *Folder* hierarchy list
   - on the Root workbooks/folders list
The *Folder* page is displayed.

Click the topmost folder to go back to the root folder/workbooks page.

2. Click the button and select *Create Folder* in the context menu.

Refer to [Creating Folders](#) for the steps in creating the sub-folders. 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 sub-folder is added.
3. You can also opt to delete a sub-folder by clicking \( \times \) as long as it does not contain published workbooks.

UPDATING FOLDER OR SUB-FOLDER PROPERTIES

Steps:

1. To update folder properties, click a folder or a sub-folder.

2. Click the \( \text{Configure Folder} \) button and select \textbf{Folder Properties} in the context menu.
The corresponding *Properties* dialog displays.

3. Make the necessary changes such as new folder name, add or delete Users and Groups.
4. Click **Update** to save the changes.

**DATA LEVEL SECURE ACCESS**

In this case the data being displayed is filtered to a particular authenticated user. Data is filtered through the use of the special parameter **userid**. This **userid** parameter is replaced at run time by the authenticated user id in lower case.

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

- Connection Details to Data Sources
- Filter constraints on data queries (e.g., SQL WHERE Clauses)
[7] Using Altair Panopticon Server

Accessing Workbooks

The default web page for the Panopticon Server lists available folders and uploaded or published workbooks. The workbooks include their titles and thumbnail images.

Enter text in the *Filter Workbooks* box to filter the workbook thumbnails.
When a workbook is opened in the Web client, other folders are generated in the AppData folder (i.e., c:\vizserverdata) such as Archive, Bookmarks, and Data aside from CacheData, DataTables, GlobalCaches, and Workbooks.

**VIEWING WORKBOOK HISTORY AND REPUBLISHING**

Aside from opening workbooks, analysts can also perform the following:

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

Steps:

1. On the **Workbooks** tab, click the **History** button of a workbook.

   The **History of Workbook <Name>** dialog displays:
You can sort the archival list either through the Date or User name 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. Click on an archived workbook in the list.

Then click Republish. A notification message displays.

3. Click Yes. A notification message displays.
4. Click **OK**.

5. You may also opt to rename an archived workbook by entering a new one in the *New Name* box and follow steps 2 to 4 to republish it.

## Parameter Value Passing Into the Web Client

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

For example:

```json
/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 in 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:

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.

<table>
<thead>
<tr>
<th>Character</th>
<th>Double Encode Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;&lt;&quot;</td>
<td>&quot;%253C&quot;</td>
</tr>
<tr>
<td>&quot;/&quot;</td>
<td>&quot;%252F&quot;</td>
</tr>
<tr>
<td>&quot;&gt;&quot;</td>
<td>&quot;%253E&quot;</td>
</tr>
</tbody>
</table>

Uploading Workbooks

Administrators can upload and publish workbooks to the currently selected Folder in the Panopticon Server.

Steps:

1. On the **Workbooks** tab, click the button and select Upload Workbook in the context menu.

   ![Upload Workbook Button]

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

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

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

   You will be notified once the workbook is uploaded.
The workbook is added and displayed.

**NOTES**

- 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 Server 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.
Exporting to Excel (TSV-format) of Visualizations

NOTES

This feature is only available for static visualizations.

Open a static visualization in the Panopticon Server.

Click the Export to Excel icon of a visualization to download a CSV spreadsheet corresponding to the existing Copy Data functionality.
Toggling Between a Visualization and a Table

NOTES

This feature is only available for static visualizations.

Open a static visualization in the Panopticon Server.

Click the **Toggle Display Mode** icon of a visualization. It will be replaced with a Table visualization.
The Table details display the same breakdowns of the original visualization and all the visualization detail variables as visible members of the Table.

NOTES

The Table will default to displaying zebra stripes.

When the **Toggle Display Mode** icon is clicked again, the Table will toggle back to the original visualization.
NOTES

- Closing and opening the dashboard will revert to the original visualization.
- Changing dashboard tabs will revert to the original visualization.
- Applying filters on the dashboard will not cause the Table to be toggled back to a visualization but will display the filtered view of the Table. For example: Industry = Basic Materials and Telecommunications
Consequently, toggling back will then display the filtered view of the visualization. The example below will only display Basic Materials and Telecommunications.
Adding, Swapping, or Removing Columns in a Breakdown and Pivot Points

This feature supports the ability to easily add or remove columns in a breakdown and pivot points.

In the breakdown of a visualization, right-click on a row or column and select either:

- Add Column

Select other columns in the list to add more hierarchical levels in the breakdown.
Remove Column
To limit the hierarchical levels in the breakdown. If there is only one column left, the Remove Column option is disabled.
You can also swap columns by selecting and dragging them to the preferred hierarchy level.

From: **Industry > Supersector > Company**

To: **Supersector > Industry > Company**
Toggling Between Rows and Columns of a Cross Tab

**NOTES**

This feature supports the easy swapping between rows to columns, and vice versa, in the pivot points of a cross tab.

In a visualization that is cross tabbed, right-click on row or column and select **Swap Columns and Rows** on the context menu.

The rows and columns will be swapped in the Columns or Rows section of the visualization.
You can opt to revert to the original columns of the Columns and Rows by selecting the **Swap Columns and Rows** on the context menu.

## Ad Hoc PDF Report Generation

PDF reports of a workbook can be produced on an on-demand basis by clicking on the PDF icon to the right of the dashboard tabs.

Clicking this icon displays the *PDF Settings* dialog:
This defines which dashboards to output into the PDF, and for visualizations that include scroll bars (commonly visual tables, and horizon graphs), whether to either:

- Output all the data items, producing multiple pages in the resulting PDF report
- Hide scrollbars

### NOTES

The *Create Multiple Pages to Show All Data Items* check box is only available when it is set to `true` in `panopticon.properties`:

```
client.pdf.multiplepages.enabled=true
```

To produce a single page for the current dashboard or a single page for each selected dashboard, set this to `false` and restart the Tomcat service to disable this property.

### NOTES

Ad hoc PDF generation in the Web client using Google Chrome (or other browsers) may be hindered by the AdBlock Extension.

To remove the AdBlock Extension in Google Chrome, perform these steps:

1. Click the **Chrome Menu** icon on the browser toolbar.
2. Highlight **Tools**, then click **Extensions** from the sub-menu.
3. Click **Remove** in the AdBlock Plus entry (e.g., uBlockOrigin).
4. Click **Remove** in the confirmation message that displays.
Bookmarking

Bookmarks are saved configurations of a dashboard and workbook. A bookmark can be added by authenticating and clicking on the **Bookmarks Panel** icon.

Bookmarks do not save data, but do save:

- selected dashboard
- selected parameters
- selected filters
- selected breakdowns, hierarchies, visible depth, and drill level
- selected variables (Size, Color, X, Y, etc).

Therefore, although the underlying data may change, a specific view of that data can be specified and bookmarked for future usage.

Bookmarks can be added and are available to all authenticated users of the workbook.

Bookmarks also generate a unique URL, which can be sent to another individual with access, allowing them to see the same view of the selected dashboard/workbook.

Click **New**.
Enter the Bookmark Title and click Add.

Bookmarks can also be renamed and updated.

To rename a bookmark, click Edit.

Enter the new Bookmark Title and click Rename.

You can also delete a bookmark by clicking X. A message box displays.
Click **OK**.

To update a bookmark, click **Update**. A message displays.

Click **OK**.

**CONTEXT MENU**

Visualizations provide a right-click Context Menu.

- Set Region, Industry
- News on Company
- Reuters Stock Quote
- Filter Include
- Filter Exclude
- Filter Clear
- Drill Down
- Drill Up
- Drill to Top
- Copy Data
- Export Raw Data
- Zoom
- Zoom out
- Show Shelves

On computers supporting a right mouse button, hold your cursor over the visualization and right-click and you will see the following options:

- Actions
- Auto-Parameterization
- Set Snapshot Here (for Time series visualizations)
- Visual Filtering – Filter Include, Filter Exclude, Filter Clear
- Hierarchical drill down – Drill Down, Drill Up, Drill to Top
- Copy Image
- Copy Data
- Export Raw Data
- Show Shelves
- Zoom

For tablets, which are touch-enabled, touching and holding will display the context menu.

Refer to the *Designer & Analyst User Guide* for more information on each option.
[8] Connectivity and Integration

Third Party Software Dependency Installation

As with the Panopticon Designer, some data connectors require additional third-party software installation to be enabled. This typically requires adding JAR files to the Lib folder of the Tomcat installation and restarting Tomcat.

Common additions include:

- **JDBC Drivers**
- **Streambase CEP**
  
  This 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 requires the following JARs 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 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

- **OneMarketData OneTick / OneTick CEP**
  
  This 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.

**NOTES**
- 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.

**JMX**
Use the following java options to enable JMX monitoring for 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)
NOTES

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

Advanced Message Processing System (AMPS)
The latest version for AMPS can be downloaded from the 60East Technologies official website: http://www.crankuptheamps.com/amps/

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

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

NOTES

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

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

- 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

**Thomson Reuters (TREP-RT)**

Manually copy the rfa.jar from the RFA distribution libraries (it can be found in RFA distribution under rfa\8.0.1.E3.all.rrg\Libs) 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

In addition, if the local dictionary is used, you need to copy the following files to the same WEB-INF\lib folder specified above:

- enumtype.def
- RDMFieldDictionary

These files are also included in the RFA distribution.

**Solace**

Manually copy the following dependency files from [http://dev.solace.com/tech/java-api/](http://dev.solace.com/tech/java-api/):

- commons-logging-1.2.jar
- org.osgi.annotation-6.0.0.jar
- sol-jcsmp-10.2.0.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
Database

There are two ways of connecting to a SQL database from the Altair Panopticon Server.

a. Use the Listed Data Connector for the specific Database (if available)
   Includes: Amazon RedShift, Cloudera Impala, DB2, Greenplum, Hadoop Hive, MongoDB, MySQL, Oracle, PostgreSQL, Salesforce, Spark SQL, SQL Server, Sybase IQ, Teradata, Vertica

b. Use the Database connection, via JDBC & JNDI
   This requires:
   1. Addition of the JDBC JAR(s) for the required Database into Tomcat/Lib
   2. Update of the server configuration file: panopticon.xml to include the new JNDI resource name
   3. Update of Workbook in Designer to reference this new JNDI resource name, prepended with: java:comp/env/, leaving the Username and Password fields blank.
      Example:
      JNDI Resource Name: jdbc/Postgre_Tickets
      In Designer: java:comp/env/jdbc/Postgre_Tickets

USING THE DATABASE CONNECTOR

For any database that you use with Panopticon Designer and Panopticon Server, it is necessary that there are drivers available that let you connect both from a .NET application like Panopticon Designer, and from a Java application like Panopticon Server:

- You would need to install the ODBC driver or OLEDB provider for the database on the system running Panopticon Designer, matching the bitness (e.g., 64-bit) of the Designer installation.
- You would need to install the JDBC driver for the database on the system running Panopticon Server.
JDBC DRIVER INSTALLATION

Install the relevant JDBC driver(s) on the system where you are running Tomcat and Panopticon Server. 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 the Panopticon Server configuration file `panopticon.xml`.

Each connection has the following structure:

```xml
<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

<table>
<thead>
<tr>
<th>DATABASE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
</table>
| 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=[DatabaseName]"/ |
| Sybase ASE   | Using jconn4.jar  
  • driverClassName="com.sybase.jdbc4.jdbc.SybDriver"
  • url="jdbc:sysbase:Tds:[HostName]:5000/[DatabaseName]"/ |
| PostgreSQL   | Using postgresql-9.4.1208.jar  
  • driverClassName="org.postgresql.Driver"
  • url="jdbc:postgresql://[HostName]:5432/[DatabaseName]"/ |
| MySQL        | Using mysql-connector-java-5.1.38-bin.jar  
  • driverClassName="com.mysql.jdbc.Driver"
  • url="jdbc:mysql://[HostName]:3306/[DatabaseName]"/ |

R and Python Transform Support

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

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

R INTEGRATION

To enable R connectivity:

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

![Image of Rserve interface]

Note: Connectivity by default is over Port 6311.

To enable authentication across the Rserve TCP/IP link

- create a password file (pwdfile.pwd)
  
  Each line of the file should have the user and then the password.
  
  Example:
  
  user1 password1
  user2 password2

- Create a configuration file with following parameters (rconfig.conf)
  
  auth required
  pwdfile [path of password file]
  
  Example:
  
  remote enable
  auth required
  port 6311
  pwdfile C:\\RIntegration\\pwdfile.pwd

- load the created configuration file (the default Rserve configuration file is still loaded, but its settings have lower priority) and run Rserve:
  
  Rserve(args="--RS-conf [path of configuration file]")
  
  Example:
  
  Rserve(args="--RS-conf C:\\RIntegration\\rconfig.conf")
PYTHON INTEGRATION

To enable Python connectivity:

1. Download and install either Python 2.7, or Python 3.5
   For ease of deployment we suggest the Anaconda Python distribution.
   [https://www.anaconda.com/distribution/](https://www.anaconda.com/distribution/)
2. Add Python installation folder to your PATH environment variable.
3. Download Pyro 4.41 (NOT any later version).
   [https://pypi.python.org/pypi/Pyro4/4.41](https://pypi.python.org/pypi/Pyro4/4.41)
   For Windows, download the Wheel: Pyro4-4.41-py2.py3-none-any.whl
4. From the command prompt, navigate to the Python scripts folder, and run:
   Pip install [Path to Wheel]\Pyro4-4.41-py2.py3-none-any.whl
   Pyro is now installed into the Python site packages.
5. Copy the start_Python_connectivity.bat and pyro.py files from the installed
   Panopticon Designer folder (both can be found in C:\Program Files\Datawatch
   Desktop\Designer) to a new location.
6. Run the start_Python_connectivity.bat file.
   Only step 6 needs to be repeated when Python connectivity is required.

NOTES

To uninstall prior versions of Pyro:

    pip uninstall Pyro4

The default password to secure connectivity is set to: password

To change this:

Edit start_Python_connectivity.bat, editing the line:

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

Edit pyro.py, editing the line:

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

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

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

```
@echo on
start python -m Pyro4.naming -n "[Required HostName]" -k "password"
ping localhost
cd "%~dp0"
python pyro.py "[Required HostName]"
pause
```
To enable the Pickle serialization, modify the `configuration.py` file located in `..\Anaconda2\Lib\site-packages\Pyro4` to specify the serialization to be used.

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

---

92 Altair Panopticon Server 17.4.0 Installation & Troubleshooting Guide
Load Custom Data Plugins

The Panopticon server 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.

**NOTES**

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.
[9] Caching

The Panopticon Server supports three levels of caching:
- Queryable Data Extract
- Recordset Cache
- Subscription 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 are only terminated on Server instance restart.

The Recordset cache, is a traditional key value cache and stores query results from prior data requests to underlying data repositories. The query results are keyed against:
- Originating workbook
- Originating data table
- Supplied parameter values
- On Demand Queries

And each resultset is cached with a time to live (TTL) defined in the original data table.

The Resultset cache does not store data extracts that can then be further queried, it simply stores query results to minimize underlying data repository load, by minimizing query duplication.

Server Queryable Cache Configuration

The Queryable cache stores data extracts from underlying databases. These data extracts can then be further queried by Panopticon. This approach is ideal if the underlying repository is limited in functionality (e.g., a flat CSV file), or slow to query. As a data extract is made, the data is not live, which may limit the usage.

Cache refresh can be manually initiated or can be scheduled using the separate URLs generated for each workbook specific or global cache. Cache refresh by default deletes the existing data.
Enabling the Queryable Data Extract

To enable the Queryable data extract on the Panopticon Server:

- The Panopticon Server can be configured to use Queryable data extract by updating the `panopticon.properties` file. The following property value has to be updated accordingly in order to configure the Panopticon server to use Queryable data extract:

  ```
cache.plugin.id=BinaryTableFile-Cache
  ```

- The Panopticon Server can also be configured to use Queryable data extract by updating the Panopticon Server configuration file (`panopticon.xml`). Please be aware that the configured value in the property file (mentioned above) will be prioritized and used if a value is provided. The configured value in the Panopticon Server configuration file will only be applied if no cache plugin id is provided in the Panopticon Server property file. Apply the following configuration in the Server configuration file in order to enable Queryable data extract on the Panopticon Server:

  ```
<Environment
    name="CachePlugin"
    override="false"
    type="java.lang.String"
    value="BinaryTableFile-Cache" />

Example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Context antiJARLocking="true" path="/panopticon">
  <Environment name="DatawatchVDDAppData"
    override="false"
    type="java.lang.String"
    value="c:\vizserverdata"/>
  <Environment name="CachePlugin"
    override="false"
    type="java.lang.String"
    value="BinaryTableFile-Cache" />
</Context>
```}

Creating Global Data Extracts

One of the methods in accessing data is by retrieving only the required results into memory, by querying on demand, pushing aggregation and filtering tasks to underlying big data repositories, or queryable data extracts.

This is commonly known as a ROLAP implementation, where the product is dynamically writing data queries to the underlying data repository and retrieving aggregated and filtered datasets. Given the on-demand nature of this method it is more suitable to exploratory data analysis but requires dynamic query generation.

Beginning with version 17.3, creating global data extracts can be directly done on the Panopticon Server. These server-based global data extracts can then be retrieved and used in the Panopticon Designer through the Panopticon Data Extract connector.
Steps:

1. On the **Extracts** tab, click **New Data Extract**.

   The **Extracts** tab displays with the following sections:

<table>
<thead>
<tr>
<th>SECTION/PANEL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract Name</td>
<td>Name of the data extract. Click the button to go back to the Data Extracts listing page.</td>
</tr>
<tr>
<td>Connector drop-down list</td>
<td>Includes the non-streaming connectors to extract data from.</td>
</tr>
<tr>
<td>Save</td>
<td>Saves the changes made on the <strong>Extracts</strong> tab.</td>
</tr>
</tbody>
</table>
2. Enter the *Name* of the data extract. This should be unique and should only contain letters (a to Z), numbers (0 to 9), and underscores.

3. Click ✓ or press **Enter** to apply the name.

4. Select any of the following non-streaming connectors:
   - InfluxDB
   - JDBC Database
   - JSON
   - Kx kdb+
   - MS Excel
   - OneTick Cloud
   - Text
   - XML

   The tab page changes depending on the selected connector. Refer to the sections below for more information.

**CREATING GLOBAL DATA EXTRACT FROM INFLUXDB**

The InfluxDB connector allows for the retrieval of a JSON data set from the InfluxDB. The database communicates over HTTP(S) where you can define a query in the URL to return the desired data.

**Steps:**

1. In the *New Data Extract* page, select **Influx DB** in the *Connector* drop-down list.
2. Enter the following information:

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>InfluxDB host address.</td>
</tr>
<tr>
<td>Port</td>
<td>InfluxDB host port. Default is <strong>8086</strong>.</td>
</tr>
<tr>
<td>User Name</td>
<td>The user name that will be used to connect to the InfluxDB service.</td>
</tr>
<tr>
<td>Password</td>
<td>The password to connect to the InfluxDB service.</td>
</tr>
<tr>
<td>Database</td>
<td>The name of the database that will communicate over the HTTP(S).</td>
</tr>
</tbody>
</table>

3. Enter an SQL-like query language into the *Query* box.

4. Click 📄. The new data extract is added in the *Data Extracts* list.
CREATING GLOBAL DATA EXTRACT FROM JDBC DATABASE

Steps:

1. In the New Data Extract page, select JDBC in the Connector drop-down list.

2. You can either select:
   - JNDI Name
     Enter the JNDI resource name to be used, then the Username and Password.
   - URL
     Enter the URL specific to the database’s JDBC driver, the Driver Class Name specific to the driver, and the Username and Password.

3. Select the appropriate SQL Dialect in the drop-down list to be able to generate the correct SQL for the required data repository.
   You can select any of the following SQL dialects: AnsiSQL, Access/Excel, MySQL, Oracle, SQL Server, Sybase IQ/ASA, Sybase ASE, Netezza, Vertica, SQLite, HadoopHive, KxQ, DB2, PostgreSQL, Impala, Redshift, Informix, Teradata, dBase, SparkSQL.
   Default is **AnsiSQL**.

4. Enter the Timeout. Default is **60**.

5. Enter an SQL-like query language into the Query box.
6. Click . The new data extract is added in the *Data Extracts* list.

**CREATING GLOBAL DATA EXTRACT FROM A JSON DATA FILE**

The JSON connector allows the retrieval and processing of JSON files, either from a disk, a Text, or from a defined URL.

**Steps:**

1. In the *New Data Extract* page, select **JSON** in the *Connector* drop-down list.

2. Select the JSON file source:
   - **File**
     Then enter the JSON File Path.

   - **Text**
     Then enter the text block to be parsed.
• Web URL

The dialog changes slightly to allow specification of the following:

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path</td>
<td>The absolute path including the http where the JSON file is located.</td>
</tr>
<tr>
<td>Headers</td>
<td>• Headers are separated by a comma</td>
</tr>
<tr>
<td></td>
<td>• Each Header is entered as <strong>Name = Value</strong>, where Name and Value can be</td>
</tr>
<tr>
<td></td>
<td>enclosed in double quotes to allow inclusion of any character except for</td>
</tr>
<tr>
<td></td>
<td>double quotes</td>
</tr>
<tr>
<td></td>
<td>• <strong>Name</strong> and <strong>Value</strong> can also be left unquoted, in which case they may</td>
</tr>
<tr>
<td></td>
<td>not include comma or equals characters</td>
</tr>
<tr>
<td>Content Encoding</td>
<td>Select the <strong>Content Encoding</strong> with the HTTP Header: <strong>None</strong>, <strong>GZip</strong>,</td>
</tr>
<tr>
<td></td>
<td><strong>Deflate</strong>, or <strong>GZip and Deflate</strong></td>
</tr>
</tbody>
</table>
User Name | The user name that will be used to connect to the JSON service.
---|---
Password | The password to connect to the JSON service.
Http Method | Select the HTTP Method to map any of the following operations to HTTP requests
| GET – retrieve information
| POST – create or update an entity
| PUT – replace an existing entity
| DELETE – remove a request
Timeout | The length of time to wait for the server response (10 to 300). Default is 10.
Request Body | The Request Body for HTTP POST.
Content Type | The required Content Type. Default is application/x-www-form-urlencoded

3. Enter the **Record Path** (e.g., myroot.items.item).

4. Click + to add columns and specify their properties:

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The column name of the source schema.</td>
</tr>
<tr>
<td>Json Path</td>
<td>The Json Path of the source schema.</td>
</tr>
<tr>
<td>Type</td>
<td>The data type of the column. Can be a Text, Numeric, or Time</td>
</tr>
<tr>
<td>Date Format</td>
<td>The format when the data type is Time.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Determines whether the message field should be processed.</td>
</tr>
</tbody>
</table>

To delete a column, check its ✅ or all the column entries, check the topmost ✅, then click −.

5. Instead of manually adding columns, you can opt to click 📋 to the fetch the schema based on the connection details. This populates the list of columns with the data type found from inspecting the first ‘n’ rows of the input data source.

6. Click 📋. The new data extract is added in the Data Extracts list.
CREATING GLOBAL DATA EXTRACT FROM KX KDB+

The Kx kdb+ connector allows connection to the Kx kdb+ databases on a polled basis.

Steps:
1. In the New Data Extract page, select Kdb+ in the Connector drop-down list.

2. Enter the following properties:

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Kx kdb+ host address.</td>
</tr>
<tr>
<td>Port</td>
<td>Kx kdb+ host port. Default is 5001.</td>
</tr>
<tr>
<td>User Name</td>
<td>The user name that will be used to connect to Kx kdb+.</td>
</tr>
<tr>
<td>Password</td>
<td>The user name that will be used to connect to Kx kdb+.</td>
</tr>
<tr>
<td>Timeout</td>
<td>30</td>
</tr>
</tbody>
</table>

3. Enter the Timeout. Default is 30.
4. Enter an SQL-like query language into the Query box.
   If a parameter has been defined, the SQL entry can refer to it.
5. Click . The new data extract is added in the Data Extracts list.

CREATING GLOBAL DATA EXTRACT FROM MS EXCEL FILE

Used for retrieving data from MS Excel workbooks or spreadsheets, where for each selected sheet, the first row contains the field/column names, and subsequent rows contain the data.

Steps:
1. In the New Data Extract page, select MS Excel in the Connector drop-down list.

2. Enter the Excel File Path of the source file.
   Example: E:\ bidoffertrade.xls
3. Click Fetch Sheets. This will populate the Select Sheet drop-down list box.
4. Select the required sheet.
5. Click . The new data source is added in the Data Extracts list.

CREATING GLOBAL DATA EXTRACT FROM ONE Tick CLOUD

The OneTick Cloud connector allows access to historic market data with no software dependencies by using the OneTick Cloud and their web API.

Steps:
1. In the New Data Extract page, select OneTick Cloud in the Connector drop-down list.
2. Enter the OneTick Cloud WebAPI URL into the Query URL box with the following form:

```plaintext
http://<host>/omdwebapi/rest/?params={"context":"DEFAULT","query_type":"otq","otq":"1/12/otq/71b50459-8431-48dc-829f","s":"20150305130802","e":"20150305140805","timezone":"America/New_York","response":"csv","compression":"gzip"}
```

Where:
- `s, e, timezone` – the start and end time of the query `YYYYMMDDhhmmss` form. The timezone used to interpret this value is taken from the timezone parameter.
- response – the supported response format is csv.
- compression – if available, this option enables gzip compression of the results stream. Large data should always be pulled with compression on.

3. Enter the Username (email) and Password to execute the query and retrieve the data. Note that the Username is case sensitive.

4. Enter the time window Start Date and End Date.

5. Enter the Symbol List. This value filters the query output with matching symbols.
   To make it work, ensure to include Symbol in the Query URL. Consequently, the data will be filtered out for the input (Symbols) provided in the Symbol List field.

6. Enter the Symbol Pattern. This value filters the query output with the data for all the symbols with matching pattern.
To make it work, ensure to include Symbol_Pattern in the Query URL. Consequently, the data will be filtered (for all the Symbols) with matching pattern provided in the Symbol Pattern field.

7. Click +. A new column entry displays. Enter or select the following properties:

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The column name of the source schema.</td>
</tr>
<tr>
<td>Column Index</td>
<td>The column index controls the position of a column. Must be $\geq 0$.</td>
</tr>
<tr>
<td>Type</td>
<td>The data type of the column. Can be a Text, Numeric, or Time.</td>
</tr>
<tr>
<td>Date Format</td>
<td>The format when the data type is Time.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Determines whether the message should be processed.</td>
</tr>
</tbody>
</table>

To delete a column, check its $\square$ or all the column entries, check the topmost $\square$, then click $\mathbb{X}$. 

8. Instead of manually adding columns, you can opt to click $\square$ to fetch the schema based on the connection details. This populates the list of columns with the data type found from inspecting the first ‘n’ rows of the input data source.

9. The time zone of input parameters and output data is by default, unchanged. Changing the time zone is supported by using the Timezone drop-down list box based on the assumption that data are stored in UTC time and outputs are presented in the selected time zone.

10. Click $\mathbb{D}$. The new data extract is added in the Data Extracts list.

**CREATING GLOBAL DATA EXTRACT FROM A TEXT DATA FILE**

The Text connector allows the retrieval and processing of delimited Text files (such as CSV, TSV, and so on), either from a disk or from a defined URL.

**Steps:**

1. In the New Data Extract page, select Text in the Connector drop-down list.
2. Select the text file source:
   
   - Text
     
     Enter the text block to be parsed.
     
     ![Text block interface]

   - File
     
     Ensure to enter the Text File Path of the source file.
     
     ![File selection interface]

   - WebURL
The dialog changes slightly to allow specification of the following:

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path</td>
<td>The absolute path including the HTTP where the Text file is located.</td>
</tr>
</tbody>
</table>
| Headers        | - Headers are separated by a comma  
- Each Header is entered as **Name = Value**, where **Name** and **Value** can be enclosed in double quotes to allow inclusion of any character except for double quotes  
- **Name** and **Value** can also be left unquoted, in which case they may not include comma or equals characters |
| Content Encoding | Select the **Content Encoding** with the HTTP Header: **None**, **GZip**, **Deflate**, or **GZip and Deflate**                                      |
| User Name      | The user name that will be used to connect to the Text service.                                                                                  |
| Password       | The password to connect to the Text service.                                                                                                       |
| Http Method    | Select the **HTTP Method** to map any of the following operations to HTTP requests  
|                | - GET – retrieve information  
- POST – create or update an entity  
- PUT – replace an existing entity  
- DELETE – remove a request |
### Timeout

The length of time to wait for the server response (10 to 300). Default is 10.

### Request Body

The Request Body for HTTP POST.

### Content Type

The required Content Type. Default is application/x-www-form-urlencoded

The standard settings controlling how the text file is parsed, is listed.

These include:

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skip First N Rows</td>
<td>Specifies the number of rows that will be skipped.</td>
</tr>
<tr>
<td>Data Type Discovery</td>
<td>Specifies how many rows from the text file should be used when automatically determining the data types of the resulting columns.</td>
</tr>
<tr>
<td>Text Qualifier</td>
<td>Specifies if fields are enclosed by text qualifiers, and if present to ignore any column delimiters within these text qualifiers.</td>
</tr>
<tr>
<td>Column Delimiter</td>
<td>Specifies the column delimiter to be used when parsing the text file.</td>
</tr>
<tr>
<td>First Row Headings</td>
<td>Determines if the first row should specify the retrieved column headings, and not be used in data discovery.</td>
</tr>
</tbody>
</table>

3. Click +. A new column entry displays. Enter or select the following properties:

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The column name of the source schema.</td>
</tr>
<tr>
<td>Column Index</td>
<td>The column index controls the position of a column. Must be &gt;= 0.</td>
</tr>
<tr>
<td>Type</td>
<td>The data type of the column. Can be a Text, Numeric, or Time</td>
</tr>
<tr>
<td>Date Format</td>
<td>The format when the data type is Time.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Determines whether the message should be processed.</td>
</tr>
</tbody>
</table>

To delete a column, check its ✗ or all the column entries, check the topmost ✗, then click −.

4. Instead of manually adding columns, you can opt to click 🔄 to the fetch the schema based on the connection details. This populates the list of columns with the data type found from inspecting the first ’n’ rows of the input data source.

5. Click 🔄. The new data extract is added in the Data Extracts list.

### CREATING GLOBAL DATA EXTRACT FROM AN XML FILE

The XML connector allows the retrieval and processing of XML files, either from a disk, a Text, or from a defined URL.
Steps:

1. In the *New Data Extract* page, select **Xml** in the *Connector* drop-down list.

![XMLExtract](image)

2. Select the XML file source:
   - **File**
     
     Ensure to enter the **XML File Path** of the source file.

     ![XML File Source](image)

   - **Text**
     
     Then enter the text block to be parsed.

     ![XML File Source](image)
- **Web URL**

  The dialog changes to allow specification of the following:

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path</td>
<td>The absolute path including the http where the Xml file is located.</td>
</tr>
<tr>
<td>Headers</td>
<td>- Headers are separated by a comma</td>
</tr>
<tr>
<td></td>
<td>- Each Header is entered as <strong>Name = Value</strong>, where Name and Value can be</td>
</tr>
<tr>
<td></td>
<td>enclosed in double quotes to allow inclusion of any character except for</td>
</tr>
<tr>
<td></td>
<td>double quotes</td>
</tr>
<tr>
<td></td>
<td>- Name and Value can also be left unquoted, in which case they may not</td>
</tr>
<tr>
<td></td>
<td>include comma or equals characters</td>
</tr>
<tr>
<td>Content Encoding</td>
<td>Select the <em>Content Encoding</em> with the HTTP Header: <strong>None, GZip, Deflate, or</strong></td>
</tr>
<tr>
<td></td>
<td><strong>GZip and Deflate</strong></td>
</tr>
<tr>
<td>User Name</td>
<td>The user name that will be used to connect to the Xml service.</td>
</tr>
<tr>
<td>Password</td>
<td>The password to connect to the Xml service.</td>
</tr>
<tr>
<td>Http Method</td>
<td>Select the HTTP Method to map any of the following operations to HTTP requests</td>
</tr>
<tr>
<td></td>
<td>- GET – retrieve information</td>
</tr>
<tr>
<td></td>
<td>- POST – create or update an entity</td>
</tr>
<tr>
<td></td>
<td>- PUT – replace an existing entity</td>
</tr>
<tr>
<td></td>
<td>- DELETE – remove a request</td>
</tr>
</tbody>
</table>
3. Enter the Record XPath (e.g., //myroot/items/item).

4. Click +. A new column entry displays. Enter or select the following properties:

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The column name of the source schema.</td>
</tr>
<tr>
<td>XPath</td>
<td>The XPath of the source schema.</td>
</tr>
<tr>
<td>Type</td>
<td>The data type of the column. Can be a Text, Numeric, or Time</td>
</tr>
<tr>
<td>Date Format</td>
<td>The format when the data type is Time.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Determines whether the message should be processed.</td>
</tr>
</tbody>
</table>

To delete a column, check its ☑ or all the column entries, check the topmost ☑, then click -.

5. Instead of manually adding columns, you can opt to click to the fetch the schema based on the connection details. This populates the list of columns with the data type found from inspecting the first ‘n’ rows of the input data source.

6. Click . The new data extract is added in the Data Extracts list.

OTHER DATA EXTRACTS OPERATIONS

On the Extracts tab, you can also perform the following:
- View workbook usage
- Refresh data
- Delete a data extract
- Modify data extract definition

Viewing Workbook Usage

On the Extracts tab, you can view the workbooks that currently use the data extract.

Steps:

1. Click the icon of a global data extract.
   The list of workbooks that currently use the data extract displays.
2. Click OK.
Refreshing Data

On the **Extracts** tab, click the **Refresh** icon of a global data extract.

Deleting Global Data Extracts

Steps:
1. Click the **X** icon of a global data extract.
   A confirmation message displays.
2. Click **Yes** to delete.

Modifying Global Data Extract Properties

Steps:
1. On the **Extracts** tab, click the link of a global data extract you want to modify.
   The corresponding data extract page is displayed.
2. Make the necessary changes then click **Save** to save them.

View Plugin Subscriptions

View and manage all of the currently running real-time Plugin subscriptions.

Including the following information:
- Data Source with an installed plugin
- Workbook Name
- Data Table Name
You can also opt to do any of the following:

- Click **Cancel All** to cancel all of the subscriptions

- Cancel a Plugin Subscription by clicking
  A notification message displays.
  Click **Yes** to cancel.

- Move to other pages
View Cache Usage

View the caches currently in use on the Server.

Including the following information:

- Cache Rendering Type
- Workbook Name
- Data Table Name
- Cache Usage Count
- Date Created
- Date Last Used
- Cache Usage Hits
- The Time to Live
- Row Count of the cached dataset
You can also opt to do the following:

- Clear Cache and refresh page
- Display Server-rendered Cache
- Display Client-rendered Cache
- Refresh the Cache Usage list
- Move to other pages
- Delete a Cache Usage instance in the list

Click the **Refresh** button to refresh the list.

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

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

![Altair Panopticon Visualization](image)

<table>
<thead>
<tr>
<th>Type</th>
<th>Workbook</th>
<th>Datatable</th>
<th>#Caches</th>
<th>Created</th>
<th>Last Used</th>
<th>#Hits</th>
<th>TTL</th>
<th>#Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Rendered Cache</td>
<td>How To Layout</td>
<td>8bf9f6b5-2095-423b-8b33-6246998e8b64</td>
<td>1</td>
<td>2019-03-21 16:05:13</td>
<td>2019-03-21 16:05:16</td>
<td>13</td>
<td>784</td>
<td>1750</td>
</tr>
<tr>
<td>Client Rendered Cache</td>
<td>How To Auto Parameterize</td>
<td>By Region</td>
<td>1</td>
<td>2019-03-21 16:04:52</td>
<td>2019-03-21 16:04:57</td>
<td>1</td>
<td>763</td>
<td>584</td>
</tr>
<tr>
<td>Client Rendered Cache</td>
<td>How To Auto Parameterize</td>
<td>Summary</td>
<td>1</td>
<td>2019-03-21 16:04:52</td>
<td>2019-03-21 16:04:52</td>
<td>2</td>
<td>763</td>
<td>1750</td>
</tr>
<tr>
<td>Client Rendered Cache</td>
<td>How To Actions</td>
<td>Equity Portfolio</td>
<td>1</td>
<td>2019-03-21 16:04:41</td>
<td>2019-03-21 16:04:46</td>
<td>11</td>
<td>751</td>
<td>1750</td>
</tr>
<tr>
<td>Client Rendered Cache</td>
<td>How To Auto Parameterize</td>
<td>By Industry</td>
<td>1</td>
<td>2019-03-21 16:04:52</td>
<td>2019-03-21 16:04:52</td>
<td>0</td>
<td>763</td>
<td>373</td>
</tr>
<tr>
<td>Client Rendered Cache</td>
<td>How To Auto Parameterize</td>
<td>By Region And Industry</td>
<td>1</td>
<td>2019-03-21 16:04:52</td>
<td>2019-03-21 16:04:52</td>
<td>0</td>
<td>763</td>
<td>138</td>
</tr>
<tr>
<td>Client Rendered Cache</td>
<td>How To Actions</td>
<td>Filtered Equity Universe</td>
<td>1</td>
<td>2019-03-21 16:04:46</td>
<td>2019-03-21 16:04:46</td>
<td>11</td>
<td>759</td>
<td>138</td>
</tr>
<tr>
<td>Client Rendered Cache</td>
<td>How To Drill</td>
<td>Static</td>
<td>1</td>
<td>2019-03-21 16:05:02</td>
<td>2019-03-21 16:05:05</td>
<td>12</td>
<td>773</td>
<td>1750</td>
</tr>
</tbody>
</table>

Click the **Delete Cache Usage** icon of a cache usage in the list.
[10] REST Interface

API

The Panopticon Server provides an interactable REST API that allows you to test the API from the browser. The REST API is powered by Swagger and is defined in the OpenAPI specification (formerly called the Swagger Specification). The API is accessed by going to a specific server URL in your browser. To use the REST API, it needs to be enabled first. The REST API is enabled by updating the following property to true in the Panopticon.properties:

documentation.enabled=true

Updating the property file requires the Panopticon Server to be rebooted. Once the Panopticon Server has been rebooted, the REST API can be accessed from the browser. Use the following address to view and interact the documentation:

http://[server]/[path]/server/doc/rest

The OpenAPI specification can be accessed on the following address:

http://[server]/[path]/server/doc/rest/api

Example: http://localhost:8080/panopticon/server/doc/rest/api

Export Data

CSV

The Panopticon Server provides the functionality to export data from a visualization to a CSV file.

Use the following URL to download the CSV file from the Server:

URL: http://[server]/[path]/server/rest/media/data/dashboard/part

Each URL has the following properties:

- Mandatory arguments
  - Workbook – Workbook name without an extension.
  - Dashboard – Dashboard name in the workbook.
  - Part – The visualization part ID

The following examples show how to export the data of a visualization from a local Server. For these examples, we have used the example workbook How To Actions.
Export data as a CSV file

- **Syntax:**
  
  ```
  http://[server]/[path]/server/rest/media/data/dashboard/part?workbook={Workbook name}&dashboard={Dashboard name}&part={Visualization part id}
  ```

- **Example:**
  
  ```
  http://localhost:8080/panopticon/server/rest/media/data/dashboard/part?workbook=How+To+Actions&dashboard=Data+Entry&part=visualization.Treemap1
  ```

Dashboard Parameters

The CSV file can be generated based on the workbook data table parameters. The parameter and its values can be specified to determine the context of the exported data.

**Syntax:**

```
http://[server]/[path]/server/rest/media/data/dashboard/part?workbook={Workbook name}&dashboard={Dashboard name}&part={Visualization part id}&{dashboardParameterName1=value1}&{dashboardParameterName2=value2}
```

Adding `Region=Europe` and `Industry=Financials` parameters

**Example:**

```
http://localhost:8080/panopticon/server/rest/media/data/dashboard/part?workbook=How+To+Actions&dashboard=Scatter+of+Filtered+Universe&part=visualization.ScatterPlot1&Region=Europe&Industry=Financials
```

Adding `Region=Asia Pacific`, or `Region=Europe` and `Industry=Financials` parameters produces a CSV file that is focused on Asia Pacific & European Financials. In this case the Region parameter is repeated for each of the supplied regions.

**Example:**

```
http://localhost:8080/panopticon/server/rest/media/data/dashboard/part?workbook=How+To+Actions&dashboard=Scatter+of+Filtered+Universe&part=visualization.ScatterPlot1&Region=Asia+Pacific&Region=Europe&Industry=Financials
```

PDF

The Panopticon Server provides the functionality to generate and download PDFs. Use the following URL to download PDFs from the server:

- **URL:** `http://[server]/panopticon/server/rest/media/pdf`

The URL can be accessed through scheduled batch tasks to retrieve and process generated PDFs. (e.g., email to predefined mailing list).

Each URL has the following properties:

- Mandatory arguments
  - **Workbook** – Workbook name without an extension.

- Optional arguments
  - **Dashboard** – Dashboard name in the workbook.
• **HideScrollbars** – Show/Hide the visualization scrollbar in the PDF. Possible values are true/false. The default value is **true**.

• **EnablePagination** – Enable pagination in the PDF. Possible values are true/false. The default value is **true**.

The following examples show how to export a PDF from a local Server. For these examples, we have used the example workbook **How To Actions**.

- **Generate PDF report of the entire workbook**
  - **Syntax:**
    
    http://[server]/[path]/server/rest/media/pdf?workbook={Workbook name}
  
  - **Example:**
    
    http://localhost:8080/panopticon/server/rest/media/pdf?workbook=How+To+Actions

- **Generate PDF report of the entire workbook in a folder**
  - **Syntax:**
    
    http://[server]/[path]/server/rest/media/pdf?workbook={Folder name%5CWorkbook name}
  
  - **Example:**
    
    http://localhost:8080/panopticon/server/rest/media/pdf?workbook=my+folder%5CHow+To+Actions

- **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 = TXlVc2VybmFtZTpoYXBhY2hlcmN3b3Jk


The PDF generator supports the following authentication mechanisms:

- BASIC
- LDAP
- Filter authentication
- Header authentication
- Windows authentication
EXCEL WORKBOOK

The Panopticon Server 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 the Panopticon Server:

- 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

**DASHBOARD IMAGE**

The Panopticon Server provides the functionality to export a workbook dashboard as a PNG image.

Use the following URL to download the images from the Server:

- **URL:** http://[server]/panopticon/server/rest/media/image/dashboard
Each URL has the following properties:

- **Mandatory arguments**
  - `workbook` – Workbook name without an extension.
  - `dashboard` – Dashboard name in the workbook.

- **Optional arguments**
  - `hideScrollbars` – Show/Hide the visualization scrollbar in the image. Possible values are true/false. The default value is **true**.
  - `width` – The width of the exported image. The default value is **1024px**.
  - `height` – The height of the exported image. The default value is **768px**.

The following examples show how to export an image from a local Server. For these examples, we have used the example workbook **How To Actions**.

- **Generate and export dashboard image**
  - **Syntax:**
    
    http://[server]/[path]/server/rest/media/image/dashboard?workbook={Workbook name}&dashboard={Dashboard name}
  
  - **Example:**
    
    http://localhost:8080/panopticon/server/rest/media/image/dashboard?workbook=How+To+Actions&dashboard=Data+Entry

- **Hide scrollbars from visualizations**
  - **Syntax:**
    
    http://[server]/[path/server/rest/media/image/dashboard?workbook={Workbook name}&dashboard={Dashboard name}&hideScrollbars={true/false}
  
  - **Example:**
    
    http://localhost:8080/panopticon/server/rest/media/image/dashboard?workbook=How+To+Actions&dashboard=Data+Entry&hideScrollbars=true

- **Set height and width for image**
  - **Syntax:**
    
    http://[server]/[path]/server/rest/media/image/dashboard?workbook={Workbook name}&dashboard={Dashboard name}&width={value}&height={value}
  
  - **Example:**
    
    http://localhost:8080/panopticon/server/rest/media/image/dashboard?workbook=How+To+Actions&dashboard=Data+Entry&width=512&height=384

**Dashboard Parameters**

The dashboard image 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 dashboard image.

**Syntax:**

http://[server]/[path]/server/rest/media/image/dashboard?workbook={Workbook name}
Adding Region=Europe and Industry=Financials parameters

Example:
http://localhost:8080/panopticon/server/rest/media/image/dashboard?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 a dashboard image 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/image/dashboard?workbook=How+To+Actions&dashboard=Scatter+of+Filtered+Universe&Region=Asia+Pacific&Region=Europe&Industry=Financials

VISUALIZATION IMAGE

The Panopticon Server provides the functionality to export a workbook dashboard visualization part as a PNG image.

Use the following URL to download the images from the Server:

- **URL:** http://[server]/[path]/server/rest/media/image/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

- **Optional arguments**
  - **hideScrollbars** – Show/Hide the visualization scrollbar in the image. Possible values are true/false. The default value is true.
  - **width** – The width of the exported image. The default value is 1024px.
  - **height** – The height of the exported image. The default value is 768px.

The following examples show how to export an image from a local Server. For these examples, we have used the example workbook **How To Actions**.

- **Generate and export Dashboard image**
  - **Syntax:**
    - [server]/[path]/server/rest/media/image/dashboard/part?workbook={Workbook name}&dashboard={Dashboard name}&part={Visualization part id}
  - **Example:**
- **Hide scrollbars from visualizations**
  - **Syntax:**
    ```
    http://[server]/[path]/server/rest/media/image/dashboard/part?workbook={Workbook name}&dashboard={Dashboard name}&part={Visualization part id}&hideScrollbars={true/false}
    ```
  - **Example:**
    ```
    ```

- **Set height and width for image**
  - **Syntax:**
    ```
    http://[server]/[path]/server/rest/media/image/dashboard/part?workbook={Workbook name}&dashboard={Dashboard name}&part={Visualization part id}&width={value}&height={value}
    ```
  - **Example:**
    ```
    ```

**Dashboard Parameters**

The visualization part image 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 Visualization part image.

**Syntax:**
```
http://[server]/[path]/server/rest/media/image/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/image/dashboard/part?workbook=How+To+Actions&dashboard=Scatter+of+Filtered+Universe&part=visualization.ScatterPlot1&Region=Europe&Region=Europe&Region=Financials
```
Email Data

PDF

The Panopticon Server provides the functionality to generate and e-mail 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 e-mail 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.

Server Configurations

The Panopticon Server needs to be configured with a valid email Server information in order to be able to send emails. The following values need to be configured in the panopticon.properties file in the AppData folder (e.g., c:\vizserverdata):

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>mail.host</td>
<td>the email Host address</td>
</tr>
<tr>
<td>mail.port</td>
<td>the email Host port</td>
</tr>
<tr>
<td>mail.security.mode</td>
<td>the security mode used when sending emails. Possible values are: NONE, SSL, or TLS. The value NONE will be used if there was no value configured for the property.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workbook</td>
<td>How to Actions</td>
</tr>
<tr>
<td>Dashboard Name</td>
<td>Scatter of Filtered Universe</td>
</tr>
<tr>
<td>Recipients (To)</td>
<td><a href="mailto:to-mail1@mail.com">to-mail1@mail.com</a>, <a href="mailto:to-mail2@mail.com">to-mail2@mail.com</a></td>
</tr>
<tr>
<td>Sender</td>
<td><a href="mailto:from-mail@mail.com">from-mail@mail.com</a></td>
</tr>
<tr>
<td>Password</td>
<td>password</td>
</tr>
<tr>
<td>Subject</td>
<td>Altair PDF Generator</td>
</tr>
<tr>
<td>Body Message</td>
<td>Hello. This is an auto-generated PDF.</td>
</tr>
</tbody>
</table>

As an example:

Panopticon configuration *(panopticon.properties)*:

```properties
mail.host=smtp.server.com
mail.port=587
mail.security.mode=TLS
```

URL:

http://localhost:8080/panopticon/server/rest/media/pdf/email?workbook=How+To+Actions&dashboard=Scatter+of+Filtered+Universe

Body:

```javascript
{
   "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**

The Panopticon Server provides the functionality to generate and e-mail 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 e-mail dashboard images. Hyperlinks can redirect to a workbook and a dashboard in the server.
NOTES

In cases when you schedule the e-mailing 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://localhost:8080/panopticon

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.

Server Configurations

The Panopticon Server needs to be configured with a valid email Server information in order to be able to send emails. The following values need to be configured in the panopticon.properties file in the AppData folder (e.g., c:\vizserverdata):

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>mail.host</td>
<td>the email Host address</td>
</tr>
<tr>
<td>mail.port</td>
<td>the email Host port</td>
</tr>
<tr>
<td>mail.security.mode</td>
<td>the security mode used when sending emails. Possible values are: NONE, SSL, or TLS. The value NONE will be used if there was no value configured for the property.</td>
</tr>
<tr>
<td>server.host</td>
<td>the server endpoint address. For example:</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workbook</td>
<td>How to Actions</td>
</tr>
<tr>
<td>Dashboard Name</td>
<td>Scatter of Filtered Universe</td>
</tr>
<tr>
<td>Recipients (To)</td>
<td><a href="mailto:to-mail1@mail.com">to-mail1@mail.com</a>, <a href="mailto:to-mail2@mail.com">to-mail2@mail.com</a></td>
</tr>
<tr>
<td>Sender</td>
<td><a href="mailto:from-mail@mail.com">from-mail@mail.com</a></td>
</tr>
<tr>
<td>Password</td>
<td>password</td>
</tr>
<tr>
<td>Subject</td>
<td>Altair Image Generator</td>
</tr>
<tr>
<td>Body Message</td>
<td>Hello. This email contains dashboard images.</td>
</tr>
</tbody>
</table>

As an example:

**Panopticon configuration** (**panoption.properties**):

```properties
mail.host=smtp.server.com
mail.port=587
mail.security.mode=TLS
```

**URL:**

```
http://localhost:8080/panopticon/server/rest/media/image/dashboard/email?workbook=How+To+Actions&dashboard=Scatter+of+Filtered+Universe
```

**Body:**

```javascript
{
    "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"
}
```
Alerting

The Alerts function allows to notify you when the data in a visualization has met your predefined settings.

Before setting the Alerts on the Web client, configure the following properties in the **Panopticon.properties** file.

<table>
<thead>
<tr>
<th>Property</th>
<th>Alert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>alert.creation.only.by.administrators</td>
</tr>
<tr>
<td>Description</td>
<td>Enable or disable whether only the Administrators can create alerts.</td>
</tr>
<tr>
<td>Default Value</td>
<td><strong>false</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Alert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>alert.email.address</td>
</tr>
<tr>
<td>Description</td>
<td>The email address where the alert will be sent from.</td>
</tr>
<tr>
<td>Default Value</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Alert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>alert.email.password</td>
</tr>
<tr>
<td>Description</td>
<td>The email password.</td>
</tr>
<tr>
<td>Default Value</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>email.host</td>
</tr>
<tr>
<td>Description</td>
<td>The host name used by the email server.</td>
</tr>
<tr>
<td>Default Value</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>email.port</td>
</tr>
<tr>
<td>Description</td>
<td>The port number used by the email server.</td>
</tr>
<tr>
<td>Default Value</td>
<td></td>
</tr>
</tbody>
</table>

Save the updated file and restart Tomcat.
Setting Up Alerts on the Web Client

Alerts can be defined against:

- Streaming data sources (including CEP Engines and message queues)
- Periodically refreshed data sources (like Oracle, SAP Sybase, SQL Server, and so on)

Alert definition can be done by right-clicking on a streaming numeric 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.

**NOTES**

Before setting up the visualization alert, enter the e-mail of the user or group who will receive the alert on the **Profile** tab:

Steps:

1. Open a workbook on the Web client and right-click on a streaming numeric 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.
2. Enter or select the following properties:

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the alert.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the alert.</td>
</tr>
<tr>
<td>Condition</td>
<td>Allows setting the Upper (&lt;= or Lower (&gt;=) Limit of all the available numeric variables in the visualization.</td>
</tr>
<tr>
<td>For the Last</td>
<td>Checks if a value has reached the limit on the set Date/Time unit:</td>
</tr>
<tr>
<td></td>
<td>• second(s)</td>
</tr>
<tr>
<td></td>
<td>• minute(s)</td>
</tr>
<tr>
<td></td>
<td>• hour(s)</td>
</tr>
<tr>
<td></td>
<td>• day(s)</td>
</tr>
<tr>
<td>Breakdown</td>
<td>Current breakdown of the visualization.</td>
</tr>
<tr>
<td>Parameters</td>
<td>Available parameters in the visualization.</td>
</tr>
<tr>
<td>Action Limit</td>
<td>The maximum number of times an alert will be sent on the set Date/Time unit:</td>
</tr>
<tr>
<td></td>
<td>• second(s)</td>
</tr>
<tr>
<td></td>
<td>• minute(s)</td>
</tr>
<tr>
<td></td>
<td>• hour(s)</td>
</tr>
<tr>
<td></td>
<td>• day(s)</td>
</tr>
<tr>
<td>Send E-mail</td>
<td>Determines when an alert e-mail will be sent:</td>
</tr>
<tr>
<td></td>
<td>• on enter</td>
</tr>
<tr>
<td></td>
<td>• on leave</td>
</tr>
<tr>
<td></td>
<td>• on enter/leave</td>
</tr>
<tr>
<td>Include</td>
<td>Determines whether the image of the visualization or dashboard will be included in the alert e-mail.</td>
</tr>
</tbody>
</table>

3. Tap the Activated slider to turn it on.

4. Click OK. The new alert is added on the Alerts page.

You can then opt to perform any of the following operations:
• Sort alerts
• Search for alerts
• Enable an alert
• Delete Alerts

SORTING ALERTS

By default, the list of alerts is sorted by Title in an ascending order. You can modify the sorting of the list by clicking the or button of the Title, Workbook, Dashboard, Created By, Creation Time, Enabled, Status, Triggered, or Triggered Today columns. The icon beside the column that was used for the sorting will indicate if it was in an ascending or descending order.

SEARCHING FOR ALERTS

To search for a particular alert, enter it in the Search box.

You can also enter one of more characters into the Filter Applications box and the suggested list of alerts that matched the entries will be displayed.
ENABLING ALERTS ON THE ALERTS PAGE

NOTES

Before enabling alerts, ensure that the sender e-mail settings are set in the Panopticon.properties file. Otherwise, you will get this error message:

Alerts can not be enabled:

- Sender email settings not set in Panopticon properties file. Please contact server administrator.

Tap the Enabled slider to turn it on.

Enabling alerts can also be performed on a visualization’s Alerts panel.

Other Alerts Operations can be modified, enabled, and deleted in the workbook where it was set.

MODIFYING ALERT SETTINGS

Steps:

1. Open a workbook with an alert and click on the Alerts icon.
The Alerts panel displays with the list of alerts.

2. Click an alert to modify.

The Alerts dialog displays.

3. Make the necessary changes then click **OK** to save them.
DELETING ALERTS

Alerts can be deleted either on:

- the Alerts panel
- an Alerts dialog
- the Alerts tab

Deleting Alerts on the Alerts Panel:

1. Open a workbook with an alert and click on the Alerts icon. The Alerts panel displays with the list of alerts.

2. Check the box of an alert and click the Delete icon. You can also check several boxes to delete multiple alerts.

Deleting Alerts on an Alerts Dialog:

1. Open a workbook with an alert and click on the Alerts icon. The Alerts panel displays with the list of alerts.
2. Click an alert. The Alerts dialog displays.

3. Click the Delete icon.

Deleting Alerts on the Alerts tab:

1. Go to the Alerts tab.

   The Alerts tab displays the list of alerts.
3. Click the **X** of an alert to delete.

A confirmation message displays.

4. Click **YES**.

**ENABLING ALERTS**

Alerts can be enabled either on:

- the Alerts panel
- an Alerts dialog

**Enabling Alerts on the Alerts Panel:**

1. Open a workbook with an alert and click on the **Alerts** icon. The **Alerts** panel displays with the list of alerts.
5. Tap the *Activated* slider to turn it on.

**Enabling Alerts on an Alerts Dialog:**

1. Open a workbook with an alert and click on the **Alerts** ! icon. The *Alerts* panel displays with the list of alerts.

2. Click an alert. The *Alerts* dialog displays.
3. Tap the *Activated* slider to turn it on and click **OK**.

**VIEWING AND MANAGING ALERTS FOR NON-ADMINISTRATOR USERS**

There are three tabs that are available for non-Administrator users:

<table>
<thead>
<tr>
<th>Alert for Visual &gt; Order Map</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Sum(fills)</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Condition</strong></td>
<td>Limit</td>
</tr>
<tr>
<td>WeightedMean(orderduration,usdumfilledvalue)</td>
<td>&lt;=</td>
</tr>
<tr>
<td>Sum(usdumfilledvalue)</td>
<td>&lt;=</td>
</tr>
<tr>
<td>Sum(orderdurationminute)</td>
<td>&lt;=</td>
</tr>
<tr>
<td>Sum(fills)</td>
<td>&lt;=</td>
</tr>
<tr>
<td><strong>For the last</strong></td>
<td>30 second(s)</td>
</tr>
<tr>
<td><strong>Breakdown</strong></td>
<td>algotype,Opportunistic,algoname,Liquidity Driver,orderid,ordernum</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Action Limit</strong></td>
<td>max 1 per hour(s)</td>
</tr>
<tr>
<td><strong>Send E-mail</strong></td>
<td>on enter/leave</td>
</tr>
</tbody>
</table>

Click on the **Alerts** tab to view and manage the available alerts.
Also perform any of the following operations:

- Sort alerts
- Search for alerts
- Enable an alert
- Delete alerts

On the **Profile** tab, enter the e-mail of the user or group who will receive the alert.

## Sample E-mail Alerts

An alert is generated when the alert set state changes from **Off** to **On** and recorded in the alert history.

An alert is only issued by email if the alert has not already been sent in the last 'n' minutes as defined in the **Alerts** dialog.

When an alert is issued, an email is sent to the defined email address.

The email includes:

- Link to the workbook or dashboard
- Condition and limit value
- Breakdown
- Name of the visualization where the alert was set
- PNG image of the visualization or dashboard

All items that do not match the criteria are excluded from the display.
Sample Web Client Alerts

When an alert is triggered, aside from the e-mail notifications, a visual indication or pop-up in active Web clients will draw attention to the alerting visualization or dashboard.

In the example below, an alert initially displays highlighted in blue:
The alert eventually fades away and the pop-up screen fills up with the four latest triggered alerts.
The pop-up stays on screen until it is closed by clicking the button.

Saved alert notifications can be opened on the Notifications panel by clicking the icon.
Clicking on a notification highlights the item in the workbook that triggered the alert.

Click the \( \times \) button to delete a notification or click \( \square \) to delete all of the notifications.
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. 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 300 rows of the selected log level or higher are fetched.

3. You can also click any of the following buttons:
   - ![Pause] to pause the logging, it changes to
   - ![Resume] to resume the logging
SET FILE LOGGING LEVEL

On the **System** tab, the level that is logged to file can be set.

Steps:
1. The current set level (e.g., **FINEST**) is displayed. To change, click the drop-down list and select another log level.
The new log level is written in the **Panopticon.properties** file:

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

## Server Logging

Logging occurs both:

- Within the platform
- In the underlying web / application server
- In the underlying OS

Panopticon Server logging is stored in the Tomcat Logs folder in a dedicated Altair log. 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 **Finest**, which additionally records:

- Data Plugin (Visualization & 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.

The Panopticon Server logging and auditing capabilities are greater than the Panopticon Designer in that they also include Java JMX counters for usage & load monitoring, and additional logging around secured access to workbooks.

CONFIGURING SERVER LOGS

The Panopticon Server is preconfigured with recommended logging settings for performance. All of the logging will be directed to a file prefixed by datawatch in the Tomcat log folder (C:\Program Files\Apache Software Foundation\Tomcat 9.0\logs). 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

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 API

Calls 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(…)
- 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)³

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.

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.:

```java
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:
Example `logging.properties` for the servlet-examples web application to be placed in WEB-INF/classes inside the web application:
Audit Logging

The Panopticon Server can also produce audit logs. All of the audit logging will be directed to a file prefixed 'datawatch-audit' in the Tomcat log folder (C:\Program Files\Apache Software Foundation\Tomcat 9.0\logs). The audit logs can be configured just like the regular logs produced by the Panopticon Server. Refer to Configuring Panopticon Server Logs for more information on how to configure logs.

Panopticon Server 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:

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>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).</td>
</tr>
<tr>
<td>Log Level</td>
<td>The severity of the log level.</td>
</tr>
<tr>
<td>Username</td>
<td>The username of the user that executed the action. The username will be ANONYMOUS if the user is not authenticated.</td>
</tr>
<tr>
<td>IP-address</td>
<td>The user’s IP address.</td>
</tr>
<tr>
<td>Action</td>
<td>Detailed message about the executed action.</td>
</tr>
</tbody>
</table>

Audit logs use comma (,) as a delimiter to separate these values.
Server Monitoring

The Panopticon Server 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:

1. **Download and install Java Development Kit (JDK)**
   

2. **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`

3. **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.**

4. **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

Designer workbooks can be embedded into existing portals with minimal effort. An iframe folder example in webapps\panopticon\api is included in the release, which details how to include the HTML client inside an iframe.

This folder includes the following files:

- CSS for styling and animations
- JavaScript for logic and control
- Help page with the post message example and the list of actions that HTML5 Client supports
  - `getWorkbooks`
  - `getDashboards`
  - `getSelectedWorkbook`
  - `getSelectedDashboardParameters`
  - `getBookmarks`
  - `addDashboardChangedListener`
  - `addParametersChangedListener`
  - `setWorkbook`
  - `setDashboard`
  - `setBookmark`
  - `setParameters`

The custom page simply needs to implement the embedded Web client as the source of the iframe tag. For example:

```html
<div class="right-column">
  <iframe id="datawatch" src="[url_to_web_client]" name="datawatch"></iframe>
</div>
```

You can then access the JavaScript API through the iframe ID reference, where you can for instance, navigate to another dashboard or workbook.
[13] Scheduling Tasks

Panopticon Server supports scheduling of tasks such as:

- Sending of a PDF file via Email
- Sending of a HTML file via Email
- Extracting Data
- Sending of a CSV Data via Email

Create Task to Send PDF File via Email

Panopticon Server provides the functionality to create tasks to generate and e-mail PDF files.

Steps:

1. Select the **Scheduler tab**.

2. 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
5. Select the task **Type:** Email PDF.

6. Upon selecting Email PDF, the Scheduler page changes to allow specification of the following:

| Create Task |  |
|-------------|--
| **Name**    | Email PDF Task |
| **Activated** | On |
| **Trigger** | Period |
| **Interval (sec)** | 30 |
| **Type**    | Email PDF |
| **Description** |  |
| **Workbook Name** |  |
| **Dashboards & Parts** | All Dashboards |
| **Parameter Values** | A=1,B=2... |
| **Hide Scrollbars** | Off |
| **Enable Pagination** | Off |
| **Sender Email** | example@domain.com |
| **Sender Password** |  |
| **To** | example@domain.com, example@domain.com... |
| **CC** | example@domain.com, example@domain.com... |
| **BCC** | example@domain.com, example@domain.com... |
| **Subject** |  |
| **Body** |  |

7. Enter the **Description** of the task.
8. Select the *Workbook Name* in the drop-down list. These are the published workbooks available in the *Workbooks* page.

9. Select other dashboards or parts to include in the PDF by checking their corresponding boxes in the *Dashboards & Parts* drop-down list.

10. 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

   **NOTES**

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

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

11. Enter the e-mail address of the sender in the *Sender Email* box. Then enter the *Sender Password*.

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

14. Click Create. The new task is added in the list.

A task displays the following columns: Name, Status, Type, Trigger, Created By, Created, and Last Run

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

Tasks can also be:
   - manually started
     Instead of waiting for the set Period interval or CRON Expression, you can manually execute the task by clicking . A confirmation message displays. Click YES.
   - modified or duplicated
Create Task to Send HTML File via Email

Panopticon Server provides the functionality to create tasks to generate and email HTML files. In addition, hyperlinks can also be used in email dashboard images. Hyperlinks can redirect to a workbook and a dashboard in the server.

**NOTES**

In cases when you schedule the e-mailing 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://localhost:8080/panopticon
```

Refer to the Email Data: Image section for more information.

**Steps:**

1. Select the **Scheduler** tab.
2. 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
   - **CRON** then enter the **CRON Expression**
5. Select the task **Type**: **Email HTML**.

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

6. Enter the **Description** of the task.
7. Select the **Workbook Name** in the drop-down list. These are the published workbooks available in the **Workbooks** page.
8. Select other dashboards or parts to include in the HTML file by checking their corresponding boxes in the **Dashboards & Parts** drop-down list.
9. 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 HTML file
     Such as Parameter=Value, and are comma separated. For example:
     Region=Europe, Industry=Financials

   **NOTES**

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

   - check the **Hide Scrollbars** box
   - check the **Hyperlinks** box. This makes the HTML file in the e-mail will be clickable.

10. Enter the **Width** and **Height** of the HTML file. Default values are **1024** and **768**, respectively.

11. Enter the e-mail address of the sender in the **Sender Email** box. Then enter the **Sender Password**.

12. Enter the e-mail address of the recipient in the **To** field.

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

14. Click Create. The new task is added in the list.

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

You can modify the sorting of the list by clicking the or button of any of these columns. The icon beside the column that was used for the sorting will indicate if it was in an ascending or descending order.
Create Task to Extract Data

Tasks can be created to reload workbook or global extracts.

Steps:

1. Select the **Scheduler** tab.
2. 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
   - *CRON* then enter the **CRON Expression**
5. Select the task **Type**: **Extract Data**.
6. Enter the **Description** of the task.
7. 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.
8. Click **Create**. The new task is added in the list.
A task displays the following columns: Name, Status, Type, Trigger, Created By, Created, and Last Run.

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

Create Task to Send CSV Data via Email

Panopticon Server provides the functionality to create tasks to generate and e-mail CSV data from a workbook, dashboard, or visualization.

Steps:
1. Select the Scheduler tab.
2. 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
   - CRON then enter the CRON Expression
5. Select the task Type: Email CSV Data.
6. Upon selecting Email CSV Data, the Scheduler page changes to allow specification of the following:
7. Enter the *Description* of the task.

8. Select the *Workbook Name* in the drop-down list. These are the published workbooks available in the *Workbooks* page.

9. Select other dashboards or parts where to source the CSV data from, by checking their corresponding boxes in the *Dashboards & Parts* drop-down list.
10. 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

NOTES
The following Date 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.

11. Enter the e-mail address of the sender in the Sender Email box. Then enter the Sender Password.
12. Enter the e-mail address of the recipient in the To field.
13. You can opt to enter the following:
   - the CC and/or BCC recipients of the e-mail separated by a comma
   - the mail message subject to be used in the e-mail notifications in the Subject field
   - the content of the email in the Body box
14. Click Create. The new task is added in the list.
Modify a Scheduled Task

Steps:

1. In the **Scheduler** tab, click the link of a task to modify. The properties of the task are displayed.
2. Apply the desired changes. For the *Email HTML* or *Email PDF* type, ensure the *Sender Password* is provided.
3. Click **Update**.

Create a Duplicate of a Scheduled Task

Steps:

1. In the **Scheduler** tab, click the link of a task to make a duplicate copy. The properties of the task are displayed.
2. Apply the desired changes, making sure to enter a new name.
3. For the *Email HTML* or *Email PDF* type, ensure the *Sender Password* is provided.
4. Click **Create as New**. A duplicate of the task is created and added in the *Scheduled Tasks* list.
The Panopticon Server 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\

### NOTES

Ensure not to replace existing files when copying from ...\WEB-INF\lib.

This supports the following:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clearcache</td>
<td>Clears the cache on a Panopticon Server.</td>
</tr>
<tr>
<td>plugins</td>
<td>Troubleshoot the plugins that this program utilizes.</td>
</tr>
<tr>
<td>publish</td>
<td>Publishes a workbook to a server or folder.</td>
</tr>
<tr>
<td>request</td>
<td>Requests workbooks and data from the server.</td>
</tr>
<tr>
<td>version</td>
<td>Prints program (and optionally server) version and exits.</td>
</tr>
<tr>
<td>help</td>
<td>Use 'help &lt;command&gt;' to get help on a specific command.</td>
</tr>
<tr>
<td>upgrade</td>
<td>Upgrades specified workbook to the newest version.</td>
</tr>
<tr>
<td>schemify</td>
<td>Updates workbook data tables with missing schema information.</td>
</tr>
<tr>
<td>addvariableids</td>
<td>Updates the Details variable for each visualization in a workbook with unique identifiers to enable sorting on details in the Web client.</td>
</tr>
<tr>
<td>exportdatasource</td>
<td>Export workbook data sources.</td>
</tr>
</tbody>
</table>

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 on the Panopticon Server.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-w, --workbook</td>
<td>Workbook name. Syntax example: <code>java -jar pcli.jar clearcache -w &quot;workbook&quot; -u &quot;http://username:password@host:port/app_name/&quot;</code></td>
</tr>
<tr>
<td>-d, --datatable</td>
<td>Datatable name. Syntax example: <code>java -jar pcli.jar clearcache -w &quot;workbook&quot; -d &quot;datatable&quot; -u &quot;http://username:password@host:port /app_name/&quot;</code></td>
</tr>
<tr>
<td>-u, --url</td>
<td>URL to the Panopticon Server, syntax: <code>java -jar pcli.jar clearcache -u http://username:password@host:port/app_name/</code></td>
</tr>
</tbody>
</table>

Command example: `java -jar pcli.jar clearcache -w "How To Actions" -d "StocksTimeSeriesFilteredTimeParameters" -u "http://username:password123@localhost:8080/panopticon/PanopticonServer.svc"`
Plugins

Troubleshoot the plugins that this program utilizes.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-v, --verbose</td>
<td>Print all information normally traced by the plugin manager.</td>
</tr>
</tbody>
</table>

Publish

You can either publish a workbook to a Panopticon Server or to a specific folder.

PUBLISHING A WORKBOOK TO A PANOPTICON SERVER

Publishes a workbook to a Panopticon Server.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f, --force</td>
<td>Overwrite existing workbook on server. Syntax example: java -jar pcl.jar publish -w &quot;workbook&quot; -d -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot; -n &quot;name&quot; -f</td>
</tr>
<tr>
<td>-w, --workbook</td>
<td>The workbook file to publish. Syntax example: java -jar pcl.jar publish -w &quot;workbook&quot; -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot;</td>
</tr>
<tr>
<td>-d, --dataFiles</td>
<td>Find and upload data files used by workbook.</td>
</tr>
<tr>
<td>-u, --url</td>
<td>URL to the Panopticon Server, syntax: java -jar pcl.jar publish -w &quot;workbook&quot; -d -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot;</td>
</tr>
<tr>
<td>-n, --name</td>
<td>Publish workbook with a different name. Syntax example: java -jar pcl.jar publish -w &quot;workbook&quot; -d -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot; -n &quot;name&quot;</td>
</tr>
<tr>
<td>--local</td>
<td>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.</td>
</tr>
<tr>
<td>-cp</td>
<td>Java classpath. Syntax example: java -cp pcl.jar; [plugin dir]/* com.panopticon.dashboards.pcli.Pcli publish -w &quot;workbook&quot; -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot;</td>
</tr>
</tbody>
</table>

Command example: java -cp pcl.jar;lib/* com.panopticon.dashboards.pcli.Pcli publish -w "How To Actions.exw" -d -u "http://username:password123@localhost:8080/panopticon/PanopticonServer.svc" -n "Published by pcl" -f
PUBLISHING A WORKBOOK TO A FOLDER

Publishes a workbook to a specific folder.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-w, --workbook</td>
<td>The workbook file to publish. Syntax example: java -jar pcl.jar publish -w &quot;workbook&quot; -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot;</td>
</tr>
<tr>
<td>-u, --url</td>
<td>URL to the Panopticon Server, syntax: java -jar pcl.jar publish -w &quot;workbook&quot; -d -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot;</td>
</tr>
<tr>
<td>NOTE: The username in the -u command must have permission to the folder. Just being in the list of Administrators is not enough.</td>
<td></td>
</tr>
<tr>
<td>-n, --name</td>
<td>Publish workbook to a folder on the server with a different name. Syntax example: java -jar pcl.jar publish -w &quot;workbook&quot; -d -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot; -n &quot;folder\name&quot;</td>
</tr>
<tr>
<td>-d, --dataFiles</td>
<td>Find and upload data files used by workbook.</td>
</tr>
</tbody>
</table>

Command example: java -jar pcl.jar publish -w "E:\Temp\How to Actons.exw" -u "http://username:password123@localhost:8080/panopticon/PanopticonServer.svc" -n "test\How to Actions.exw"

PUBLISHING A WORKBOOK FOLDER TO A PANOPTICON SERVER

Publishes a workbook folder to a Panopticon Server.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-tf, --targetFolder</td>
<td>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 pcl.jar publish -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot; -wf &quot;folder containing workbooks&quot;-tf &quot;server folder name&quot; -r</td>
</tr>
<tr>
<td>-r, --root</td>
<td>Publish workbooks to the ROOT folder. This is only applicable with -wf Syntax example: java -jar pcl.jar publish -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot; -wf &quot;folder containing workbooks&quot;-tf &quot;server folder name&quot; -r &quot;default or root folder&quot;</td>
</tr>
<tr>
<td>-u, --url</td>
<td>URL to the Panopticon Server, syntax: java -jar pcl.jar publish -w &quot;workbook&quot; -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot;</td>
</tr>
<tr>
<td>-wf, --workbookFolder</td>
<td>The workbook folder from which workbooks will be picked to publish. Use -w to publish single workbook. Syntax example: java -jar pcl.jar publish -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot; -wf &quot;folder containing workbooks&quot;-tf &quot;server folder name&quot; -r</td>
</tr>
</tbody>
</table>
**Command example:** java -cp pcl1.jar publish
"http://username:password123@localhost:8080/panopticon/PanopticonServer.svc" -wf "C:\Serverdata\Data" -tf "c:\Streamsdata\Data" -r

---

## Request

Requests workbooks and data from the Server.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>--xml</td>
<td>Path to requests XML, see ExampleRequests.xml for syntax. Syntax example: java -jar pcl1.jar request -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot; -- xml &quot;request.xml&quot;</td>
</tr>
<tr>
<td>-u, --url</td>
<td>URL to server, syntax: java -jar pcl1.jar request -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot;</td>
</tr>
</tbody>
</table>

**Command example:** java -jar pcl1.jar request -u "http://username:password123@localhost:8080/panopticon/PanopticonServer.svc"
Version

Prints program (and optionally server) version and exits.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-u, --url</td>
<td>URL to the Panopticon Server, syntax: java -jar pcli.jar version -u &quot;<a href="http://username:password@host:port/app_name/">http://username:password@host:port/app_name/</a>&quot;</td>
</tr>
</tbody>
</table>

Command example: java -jar pcli.jar version -u "http://username:password123@localhost:8080/panopticon/PanopticonServer.svc"

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.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-w, --workbook</td>
<td>Workbook path to upgrade. Syntax example: java -jar pcli.jar upgrade -w workbook.exw</td>
</tr>
<tr>
<td>-o, --output</td>
<td>Output workbook path. Syntax example: java -jar pcli.jar upgrade -w workbook.exw -o workbook1.exw</td>
</tr>
</tbody>
</table>

Schemify

Updates workbook data tables with missing schema information.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>--dd, --data-directory</td>
<td>Data directory path. Syntax example: java -jar pcli.jar schemify -wd &quot;workbook directory&quot; -od &quot;output directory&quot; -dd &quot;C:\Users\Public\Documents\Datawatch Desktop\Data&quot;</td>
</tr>
<tr>
<td>-D</td>
<td>Default parameter.</td>
</tr>
</tbody>
</table>
### Addvariableids

Updates the Details variables for each visualization in a workbook with unique identifiers to enable sorting on Details in the Web client.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-od, --output-directory</td>
<td>Output directory path.</td>
</tr>
<tr>
<td>-w, --workbook</td>
<td>Workbook to upgrade.</td>
</tr>
<tr>
<td>-l, --license-file</td>
<td>License file path.</td>
</tr>
<tr>
<td>-wd, --workbook-directory</td>
<td>Directory of workbooks to upgrade.</td>
</tr>
<tr>
<td>-o, --output</td>
<td>Output path.</td>
</tr>
</tbody>
</table>

Command example: java -jar pcl.jar addvariableids -l "E:\projects\Dashboards .NET\DatawatchLicense.xml" -w "E:\workbooks\details_sorting\sortproblem.exw" -o "E:\workbooks\details_sorting\sortproblem_fixed.exw"

### ExportDataSource

Export workbook data source.
<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-dd, --data-directory</td>
<td>Data directory path.</td>
</tr>
<tr>
<td>-od, --output-directory</td>
<td>Output directory path.</td>
</tr>
<tr>
<td>-w, --workbook</td>
<td>Export data sources of workbook.</td>
</tr>
<tr>
<td>-l, --license-file</td>
<td>License file path.</td>
</tr>
<tr>
<td>-wd, --workbook-directory</td>
<td>Directory of workbooks.</td>
</tr>
</tbody>
</table>

Command example: java -jar pcl.jar exportdatasource -l "E:\projects\Dashboards .NET\DatawatchLicense.xml" -w "E:\workbooks\exportdb.exw" -dd "E:\Serverdata\export" -od "E:\Streamsdata\export" -wd "E:\workbooks"
[15] Troubleshooting

Resolving Installation Issues

Issues are investigated and resolved through investigation and controlled reproduction. A number of known issues are included in the next section and predominately relate to problematic installations of the Panopticon Server.

If you experience an unknown issue, send complete details to: support@datawatch.com

Be sure to send this important information to Altair Support in the event of a problem.

SERVER LOG

The Panopticon Server log files are located in the [tomcat_home]\logs folder:

C:\Program Files\Apache Software Foundation\[Tomcat_Home]\logs\*.log

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

   **NOTES**

   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.
NOTES

Refer to Configuring Server Logs for more information. When sending your issue, include your Panopticon Designer workbook & associated data sources if the issue is specific to a particular workbook.

Log Variables Displayed as “Null”

When using Tomcat 8, edit conf\logging.properties file to display the package, class, or method names in the log file instead of having a “null” value.

Steps:
1. Stop the Tomcat service.
2. Open the file conf\logging.properties.
3. Change all instances of AsyncFileHandler to FileHandler.
4. Save the updated file, and restart Tomcat.

No Appropriate Protocol Error When Publishing Splunk Data on Panopticon Server

The Altair log written into the Panopticon Server 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.
Timeout Exception When Loading Data

This timeout issue is caused by the AWS connection taking a longer time in returning data from the plugin to the visualization.

Follow instructions below to increase the service timeout threshold.

Steps:

1. **Open the** `EXDesigner.exe.config` **file (located in C:\Program Files\Datawatch Desktop\Designer).**

2. **Add** `closeTimeout="00:59:00" openTimeout="00:59:00" receiveTimeout="00:59:00" sendTimeout="00:59:00"` **for the following Binding attributes:**
   - `<binding name="BasicHttpBinding_IPanopticonServer_Anonymous">`
   - `<binding name="BasicHttpBinding_IPanopticonServer_Authenticated">`
   - `<binding name="BasicHttpBinding_IPanopticonServer_SSL_Anonymous">`
   - `<binding name="BasicHttpBinding_IPanopticonServer_SSL_Authenticated">`  

   For example:
   ```xml
   <binding name="BasicHttpBinding_IPanopticonServer_Anonymous"
           closeTimeout="00:59:00" openTimeout="00:59:00"
           receiveTimeout="00:59:00" sendTimeout="00:59:00">
   ```

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.
[16] Known Issues

OUT OF MEMORY EXCEPTION

If the data is too big, an out of memory exception may occur. You may increase the memory for the Panopticon Server in Tomcat.

To resolve out of memory exception:

1. Stop the Tomcat service.

   The Apache Tomcat Properties dialog is displayed.

3. Most users who are running Java are likely doing it on Linux. However, if you are using Windows, select the Java tab as shown below.
4. Set the Initial memory pool to **1GB**.

5. Set the Maximum memory pool to **16GB**.

6. Click **OK**.

7. Restart the Tomcat service to apply the changes.

## SOCKET LEAKAGE ON WINDOWS 2008 R2 / WINDOWS 7

Socket leakage may occur when you:

- have a multiprocessor computer running on Windows 2008 R2 or Windows 7
- are running an application such as Panopticon that creates loopback sockets on the computer

You can resolve this problem by downloading and installing the hotfix provided by Microsoft:

https://support.microsoft.com/en-us/help/2577795/kernel-sockets-leak-on-a-multiprocessor-computer-that-is-running-windows-server-2008-r2-or-windows-7
## Properties: Panopticon

The `panopticon.properties` file located in the DatawatchVDDAppdata folder (i.e., `c:\vizserverdata`), contains majority of properties for controlling the configuration of the Panopticon Server. The following properties can be overridden by updating the file.

<table>
<thead>
<tr>
<th>Property</th>
<th>Attribute</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache</td>
<td>cache.purge.condition</td>
<td>Defines the condition for when the cache will be purged. Allowed values: NONE, MEMORY</td>
<td>MEMORY</td>
</tr>
<tr>
<td>Cache</td>
<td>cache.purge.condition.memory.threshold</td>
<td>Defines a percentual memory threshold for cache purging, when the cache.purge.condition = MEMORY.</td>
<td>80</td>
</tr>
<tr>
<td>Cache</td>
<td>cache.purge.enabled</td>
<td>Enables scheduled cache purging.</td>
<td>true</td>
</tr>
<tr>
<td>Cache</td>
<td>cache.purge.rate</td>
<td>Defines a fixed rate, in milliseconds, for cache purging.</td>
<td>10000</td>
</tr>
<tr>
<td>Cache</td>
<td>cache.serialization.size</td>
<td>Data table processor cache size in items.</td>
<td>100</td>
</tr>
<tr>
<td>Cache</td>
<td>cache.serialization.type</td>
<td>Data table processor cache type. Allowed values: NONE, IN_MEMORY</td>
<td>IN_MEMORY</td>
</tr>
<tr>
<td>Property</td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>cache.service.enabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Enables and disable the service cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td>true</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>cache.service.getresourcedworkbook.queue.size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Defines the cache size in items for the workbook service.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>cache.service.type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>The service cache mechanism being used. Allowed values: MEMORY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td>IN_MEMORY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>cache.service.getdata.queue.size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>The max size for the cache queue for the service GetData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>cache.service.getondemanddata.queue.size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>The max size for the cache queue for the service GetOnDemandData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>cache.service.getformatteddata.queue.size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>The max size for the cache queue for the service GetFormattedData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>cache.service.getdatatabledata.queue.size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>The max size for the cache queue for the service GetDatatableData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>cache.schedule.clear.enabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Enable the cache clearing schedule. This is scheduling the clear cache operation which will remove all the expired cache entries.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td>true</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>cache.schedule.clear.rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Set an interval for how often the clear cache scheduler will clear the expired cache entries</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td><strong>600000</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>cache.plugin.id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>The id of the cache plugin that will be used. Possible values: BinaryTableFile-Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td><strong>BinaryTableFile-Cache</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>cache.data.plugin.connection.aggregate.size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>The size of the On-demand cache queue.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td><strong>100</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>cache.data.plugin.connection.raw.size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>The size of the raw non On-demand cache queue.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td><strong>100</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>cache.data.plugin.connection.type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>The service cache mechanism being used. Allowed values: IN_MEMORY, NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td><strong>IN_MEMORY</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>cache.diskwrite.enabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Enables the creation of the temporary cache file (.btf), along with the .meta and .source files. If set to false, .btf files will not be written on the disk system. Instead, data will just be held in memory.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td><strong>true</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>cache.temporary.enabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Enables temporary caching where the data in the workbook visualizations, when opened in the web client, should be loaded from the cache source (.btf, .meta, and .source files that were created in the server's CacheData folder). This helps minimize the amount of times the underlying data source is requested. Also, when set to true, temporary cache files are removed when the server is closed, which also minimizes the load on the disk system.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td><strong>false</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Client Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>client.cache.control.age.max</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Controls the cache-control max-age header for static content.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td><strong>31536000</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cache.datatable.processor.type</td>
<td>The service cache mechanism being used. Allowed values: IN_MEMORY, NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cache.datatable.processor.usecacheonfailure</td>
<td>Enables the cache fallback functionality on failure. This means that the Server will use expired cache even on failure.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cache.datatable.processor.size</td>
<td>The size of the cache queue.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cache.data.plugin.processor.type</td>
<td>The type of cache that is going to be used for the data plugin. Allowed values: IN_MEMORY, NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cache.data.plugin.processor.usecacheonfailure</td>
<td>Enables the cache fallback functionality on failure. This means that the server will use expired cache even on failure.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cache.data.plugin.processor.size</td>
<td>The size of the cache queue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>error.default.message</td>
<td>Defines a generic error message override.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>authentication.role</td>
<td>The authentication role.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

190  Altair Panopticon Server 17.4.0 Installation & Troubleshooting Guide
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Default Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication.required</td>
<td>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.</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Authentication.type</td>
<td>The type of the authentication mechanism that will be used on the Server.</td>
<td>BASIC</td>
<td>authentication.type=SAML</td>
</tr>
<tr>
<td>Authentication.domain</td>
<td>The default domain information for user authentication.</td>
<td></td>
<td>authentication.domain=company.com</td>
</tr>
<tr>
<td>Authentication.timeout.callback</td>
<td>The timeout (in milliseconds) for the user between initiated login and callback. The default value is five minutes.</td>
<td>300000</td>
<td>authentication.timeout.callback=300000</td>
</tr>
<tr>
<td>Authentication.saml.serviceprovider.id</td>
<td>SAML Service provider id.</td>
<td></td>
<td>authentication.saml.serviceprovider.id=DwchFrontLocal8080</td>
</tr>
<tr>
<td>Authentication.saml.identityprovider.url</td>
<td>SAML Identity provider URL.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Attribute</td>
<td>Description</td>
<td>Default Value</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>----------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Authentication: SAML</td>
<td>authentication.saml.callback.url</td>
<td>SAML Callback URL</td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>authentication.saml.assertion.username</td>
<td>SAML assertion username.</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>authentication.saml.assertion.roles</td>
<td>SAML assertion roles.</td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td>authentication.saml.assertion.roles</td>
<td>SAML assertion roles.</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>authentication.saml.certificate.name</td>
<td>SAML certificate name.</td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>authentication.saml.certificate.password</td>
<td>SAML certificate password.</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>authentication.saml.identityprovider.logout.url</td>
<td>URL of the IdP logout service.</td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td>authentication.saml.identityprovider.logout.url</td>
<td>URL of the IdP logout service.</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>authentication.saml.keystore.file</td>
<td>SAML keystore file.</td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>authentication.saml.keystore.file=D:/SAML/mykeystore.jks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Authentication: SAML</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>authentication.saml.keystore.password</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>SAML keystore password.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>authentication.saml.keystore.password=samlpassword</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Authentication: SAML</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>authentication.saml.challenge.required</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>This property determines whether the IdP-first authentication with SAML is enabled or not. To enable, set this property to <code>false</code>.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td><code>true</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Authentication: Token</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>authentication.token.cookie</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>The name of the cookie used to store the authentication cookie. Must be unique for each server instance on the host.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Authentication: Token</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>authentication.token.domain</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Specifies the token cookie domain.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Authentication: Token</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>authentication.token.persistence</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>PERSISTENT</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Authentication: Token</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>authentication.token.refreshable</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>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 <code>true</code>. The token will expire and invalidate the user session if the property is set to <code>false</code>.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td><code>true</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Authentication: Token</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>authentication.token.refreshable.scope</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>This property determines who can refresh a token: <code>ALL</code> or <code>CLIENT</code>.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td>Property</td>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>ALL</strong></td>
<td>Authentication: Token</td>
<td>authentication.token.secret</td>
<td>The secret is used to sign the token. The secret will be auto-generated when the server starts for the first time. <strong>NOTE: This value should be kept a secret.</strong></td>
</tr>
<tr>
<td><strong>Auto-generated</strong></td>
<td>Authentication: Token</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>604800</strong></td>
<td>Authentication: Token</td>
<td>authentication.token.validity.seconds</td>
<td>The number of seconds that the token should be valid.</td>
</tr>
<tr>
<td><strong>false</strong></td>
<td>Authentication: Token</td>
<td>authentication.token.cookie.httponly</td>
<td>This property determines how the browser will treat the cookie. If set to <strong>true</strong>, the cookie will be stored in the browser as a HttpOnly cookie and will not be available to the JavaScript. If set to <strong>false</strong> (default), the cookie will be stored in the browser as https and will be accessible to the JavaScript.</td>
</tr>
<tr>
<td><strong>false</strong></td>
<td>Authentication: Token</td>
<td>authentication.token.cookie.secure</td>
<td>The property determines how the browser will treat the cookie depending on the security of the connection. If set to <strong>true</strong>, when the browser receives a secure cookie (HttpOnly cookie), you will not be able to transmit it unless the connection is secure.</td>
</tr>
<tr>
<td><strong>true</strong></td>
<td>SOAP</td>
<td>soap.enabled</td>
<td>Enable or disable the SOAP interface</td>
</tr>
<tr>
<td><strong>false</strong></td>
<td>WebSocket</td>
<td>websocket.enabled</td>
<td>Enable or disable the WebSocket interface</td>
</tr>
<tr>
<td><strong>false</strong></td>
<td>Authentication: Header</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
<td>Default Value</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>authentication.header.username</td>
<td>The name of the header that contains the username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>authentication.header.roles</td>
<td>The name of the header that contains all the roles.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>authentication.header.role.delimiter</td>
<td>The delimiter used to separate the roles. Example: role1, role2,role3</td>
<td>, (Comma)</td>
<td></td>
</tr>
<tr>
<td>subscription.limitation.action</td>
<td>Controls the behavior when the subscription.limitation.limit is reached. Allowed values: EXCEPTION, PURGE</td>
<td>EXCEPTION</td>
<td></td>
</tr>
<tr>
<td>subscription.limitation.enabled</td>
<td>Enables limitation of subscriptions.</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>subscription.limitation.limit</td>
<td>Defines a subscription limit.</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>subscription.maximum.failure</td>
<td>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.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>subscription.orphan.purge.enabled</td>
<td>Enables or disables the purge inactive data subscription functionality. This property should only be used when using Client-side rendering mode. All data subscriptions which are not being used by any visualization subscriptions will be terminated.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td>false</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Subscription</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>subscription.orphan.purge.rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>The frequency of how often the purge feature will be executed. The value is in milliseconds.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td><strong>10000</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Subscription</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>subscription.parameter.user.force</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>By enabling this, the Server will always add a user id parameter to all incoming data requests. This can be used if you want to pass the user id to the plugins without having to add the userid parameter to the workbook or data table.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>false</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Subscription</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>subscription.purge.condition</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Defines the condition for when subscriptions will be purged. Allowed values: NONE, MEMORY</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>MEMORY</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Subscription</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>subscription.purge.condition.memory.threshold</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Defines a percentual memory threshold for subscription purging, when the subscription.purge.condition = MEMORY.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td><strong>80</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Subscription</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>subscription.purge.enabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Enables subscription purging.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>true</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Subscription</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>subscription.purge.post.restart</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Option to re-start active subscriptions after purge. Only valid when subscription.purge.scope = ALL</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Subscription</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>subscription.purge.rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Defines a fixed rate, in milliseconds. for subscription purging.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td><strong>10000</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Subscription</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>subscription.purge.scope</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Defines the scope of subscriptions to purge. Allowed values: NON_PERSISTENT_ORPHANS, ALL.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td>NON_PERSISTENT_ORPHANS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Subscription</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>subscription.compression.enabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Enable or disable compression and encoding of subscription broadcast messages.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>true</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Property** Request parameter mapping

<table>
<thead>
<tr>
<th><strong>Attribute</strong></th>
<th>request.operation.parameters.mapping.required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Property** Request parameter mapping

<table>
<thead>
<tr>
<th><strong>Attribute</strong></th>
<th>request.operation.parameters.mapping.value.delimiter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The delimiter that separates the parameter name and the Header name. This property will only affect incoming parameters.</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>: (Colon)</td>
</tr>
</tbody>
</table>

**Property** Request parameter mapping

<table>
<thead>
<tr>
<th><strong>Attribute</strong></th>
<th>request.operation.parameters.mapping.entry.delimiter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The delimiter that separates the configuration entries. This property will only affect incoming parameters.</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>, (Comma)</td>
</tr>
</tbody>
</table>

**Property** Request parameter mapping

<table>
<thead>
<tr>
<th><strong>Attribute</strong></th>
<th>request.operation.parameters.mapping.optional</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Property** Response parameter mapping

<table>
<thead>
<tr>
<th><strong>Attribute</strong></th>
<th>response.operation.parameters.mapping.required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Property** Response parameter mapping

<table>
<thead>
<tr>
<th><strong>Attribute</strong></th>
<th>response.operation.parameters.mapping.value.delimiter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The delimiter that separates the parameter name and the Header name. This property will only affect incoming parameters.</td>
</tr>
</tbody>
</table>
| Default Value | REST
|--------------|--------------------------------------------------|
| Attribute    | rest.enabled
| Description  | Enable or disable the REST interface.
| Default Value| true

| Default Value | REST
|--------------|--------------------------------------------------|
| Attribute    | rest.response.error.stacktrace.included
| Description  | Include the error stack trace in REST responses.
| Default Value| false

| Default Value | REST Documentation
|--------------|--------------------------------------------------|
| Attribute    | documentation.enabled
| Description  | Enable or disable the OpenAPI Specification documentation for the REST interface.
| Default Value| false

| Default Value | Client
|--------------|--------------------------------------------------|
| Attribute    | client.webgl.enabled
| Description  | Enable or disable WebGL for the Web client.
| Default Value| true

| Default Value | Client
|--------------|--------------------------------------------------|
| Attribute    | client.data.load.transport
| Description  | Configure the transportation protocol for loading data from the Web client. Possible values: WEBSOCKET, LONG_POLLING.
| Default Value| WEBSOCKET

| Default Value | Client
|--------------|--------------------------------------------------|
| Attribute    | client.pdf.multiplepages.enabled
| Description  | Enable or disable the creation of multiple pages to show all data items in the resulting PDF output.
| Default Value| true

| Default Value | Email
|--------------|--------------------------------------------------|
| Attribute    | email.port
| Description  | The port number used by the email server.
| Default Value| 587

| Default Value | Email
|--------------|--------------------------------------------------|
| Attribute    | email.host
| Description  | The host name used by the email server.
| Default Value| localhost
<table>
<thead>
<tr>
<th>Property</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>server.host</td>
</tr>
<tr>
<td>Description</td>
<td>The server endpoint address. For example: server.host=<a href="http://localhost:8080/panopticon">http://localhost:8080/panopticon</a></td>
</tr>
<tr>
<td>Default Value</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>email.security.mode</td>
</tr>
<tr>
<td>Description</td>
<td>The security mode used when sending emails. Possible values: NONE, SSL, TLS.</td>
</tr>
<tr>
<td>Default Value</td>
<td>NONE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Image export</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>export.image.width</td>
</tr>
<tr>
<td>Description</td>
<td>The default width for an exported image.</td>
</tr>
<tr>
<td>Default Value</td>
<td>1024</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Image export</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>export.image.height</td>
</tr>
<tr>
<td>Description</td>
<td>The default height for an exported image.</td>
</tr>
<tr>
<td>Default Value</td>
<td>768</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>global.authentication.button.hidden</td>
</tr>
<tr>
<td>Description</td>
<td>Hide or show the login button for the Web client.</td>
</tr>
<tr>
<td>Default Value</td>
<td>False</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Log level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>logger.level.file</td>
</tr>
<tr>
<td>Description</td>
<td>Controls the level that is logged to file.</td>
</tr>
<tr>
<td>Default Value</td>
<td>INFO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Authorization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>statistics.authorization.level</td>
</tr>
<tr>
<td>Description</td>
<td>Allows users to set the authorization level for the statistics and diagnostic REST services. Possible values: ANONYMOUS, AUTHENTICATED, ADMINISTRATOR, DISABLED</td>
</tr>
<tr>
<td>Default Value</td>
<td>ADMINISTRATOR</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Licensing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>license.hwu.operating.system</td>
</tr>
<tr>
<td>Description</td>
<td>The operating system where the Panopticon Server is installed. Possible values are: WIN_X86, WIN_X64, MAC, LINUX_X64, or LINUX_ARM64. <strong>NOTE:</strong> If the Java bitness (e.g., 32-bit) is different from the operating system (e.g., 64-bit), it is recommended to add the Java bitness in this property (e.g., WIN_X86).</td>
</tr>
</tbody>
</table>

NOTE: If the Java bitness (e.g., 32-bit) is different from the operating system (e.g., 64-bit), it is recommended to add the Java bitness in this property (e.g., WIN_X86).
### Default Value

<table>
<thead>
<tr>
<th>Property</th>
<th>Licensing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>license.hwu.uri</td>
</tr>
</tbody>
</table>

**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 used, they should be separated by ';'.

**NOTES:**

- Multiple License Servers are not supported when the Panopticon Server is on a Linux machine.
- If value is not set in the panopticon.properties, the environment variable ALTAIR_LICENSE_PATH serves as the backup path and will be used.

### Default Value

<table>
<thead>
<tr>
<th>Property</th>
<th>Licensing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>license.hwu.version</td>
</tr>
</tbody>
</table>

**Description**

Value must match the license version found in the HyperWorks Units license file.

**Default Value**

**19.0**

### Default Value

<table>
<thead>
<tr>
<th>Property</th>
<th>Licensing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>license.mode</td>
</tr>
</tbody>
</table>

**Description**

The license mode. Possible values are: **FILE** or **HWU**. To use the HyperWorks Units license, set this property to HWU.

**Default Value**

**FILE**