XRay

User's Guide

Version 1.6

Document Number: XRUG1.09

Document Title: XRay User's Guide
meshIQ Document Number: XRUG1.09
Document Release Date: November 2023

Published by:
Research & Development
meshIQ
88 Sunnyside Blvd, Suite 101
Plainview, NY 11803

Table of Contents

History of this Document

Table 1. Document History		
Release Date	**Doc Number**	**Summary**
February 2021	XRUG 1.6	Update figures 2.4.2.2-A (Open Dashboard) and Figure 3.2.5-C (Add Fields for Schema). Add figure 3.2.5-D (Alias Names). Update section 3.1.1.4. Update section 2.5 (add links to viewlet chart type sections). Add sections 2.5.4.15 (Clustering), 2.5.4.16 (Correlation), 2.5.4.17 (Feature Suggestion, 2.5.4.18 (Forecast) and 2.5.4.19 (Expected). Add sections 2.5.11 - AutoPilot Integration, and 2.5.4.20 – Images.
April 2022	XRUG 1.07	Add information on logo image size limit in section 3.1.1.1 (Logos). Add support note to section 2.5.4.20 (Images). Add sharing info to Images section (2.5.4.20). Added Admin sections for Views and jKQL Scripts (XRay version 1.4).
March 2023	XRUG 1.08	XRay version 1.5: Social Integration (Slack); Redesigned Data Import wizard; Support Having Filter in Group By....; Volume passwords; Image Viewlets (Annotated Business Flows) — partial documentation; meshIQ branding. Forthcoming: Image Viewlets, continued
June 2023	XRUG 1.09	XRay version 1.6: Admin settings revamp, including updates to setup of repositories and tokens (such as personal token expiration and access token quotas); cloning setup items; Machine Learning model definition and training wizard; Data Import enhancements (data retention and CSV file encoding).

Chapter 1: About XRay

1.1 Introduction

For IT Ops professionals, members of the DevOps group, and developers, the ultimate advantage would be to know everything as it happens in their business – and everything that could happen. To know all and see all with complete vision is the competitive ideal: operations managers armed with real-time analytics, detecting performance problems before delays arise; the company discovering trends the moment they form.

If there is a function in modern technology that offers anything close to this ideal of omniscience, it's providing real-time analytics to prevent problems (or at least their impact) and forensics to resolve the problems you can't prevent.

Such awareness is difficult to attain, and it is often impossible for companies to know in advance what events need to be analyzed and when that analysis must happen. IT must store and analyze everything, or risk missing evidence of operational lags, risks, or rising customer trends. For security compliance reasons alone, enterprises are required to maintain good logs, store them securely for at least one year, and review them daily.

An important consideration is whether staff are looking at the data with the right degree of resolution. A common strategy is to restrict support staff to low-resolution viewlets, so they can isolate a problem to a specific infrastructure tier and then pass the problem to a specialist, who starts diagnosing the problem all over again with a different tool. While this approach works, it is time-consuming, expensive, and disruptive. A more productive approach would be to allow application support to use a high-resolution analysis, equipping them with the forensic tools to both diagnose a problem and immediately begin its resolution.

To answer business-centric questions and provide guidance for decision-makers, XRay combines:

Analytics using advanced predictive anomaly detection and machine learning algorithms for problem prevention across apps, messaging, logs, mobile, and the IoT.
Insight into applications, including payment processing, trade compliance, order tracking, healthcare claims processing, compliance, machine data, and more.
Visibility across the IBM stack (MQ, IIB, DP, MFT), Java, mobile, and the newer open-source technologies such as Kafka, STORM, Spark, MQTT, log files, Python, REST, and much more.
Multi-tenancy with private data repositories available on premises or in SaaS.
Lambda architecture with grids for real-time, in-memory analytics as well as historical analytics, data replication, and time-to-live for all streaming data.
End-to-end business transaction tracking that spans technologies, tiers, and organizations.
Intuitive, easy-to-use data visualizations and dashboards.

These capabilities fuse seamlessly across dynamic IT environments, from mobile to mainframe. They provide the broad array of analytic and decision-support capabilities needed by developers, IT admins, and business analysts to satisfy real-time operations intelligence and APM needs.

1.2 Key Benefits

Key benefits are insight, visibility, prediction, and machine learning that is easy-to-use to:

Improve service to customers and reduce operational risk – using machine learning analytics.

Highly scalable with self-service access, without need for data scientists – using flexible web-based UI's and natural language for ease of use and a powerful Lambda architecture with microservices for scalability.

Reduce support costs – via Docker deployment, open-source data collectors and ease of use.

1.3 Activities, Events and Snapshots Concept Overview

The XRay data model consists of the following items:

Events: Actions or occurrences recognized by software that may be handled by the software. Event is the smallest item, which can be measured by time.

Activities: A collection of related tracking events (TrackingEvent) and other sub-activities. Relation is established via a grouping specified by a developer or set of correlators (across thread, application boundaries). Activities may have a set of user-defined properties which are grouped into property snapshots (PropertySnapshot).

Sets: Named collections of Activities that meet specific criteria.

Snapshots: A collection of properties with category, name, and a time stamp associated with when the snapshot is taken. Activities may have one or more property snapshots.

Dictionaries: Generic, free-form items that can have user-defined properties.

Sources: Represent origins of Events and Activities. They can be references generically as Sources, or by the specific class of source:

GeoLocation; DataCenter; Network; Device; Address; Server; Process; Runtime; AppServer; Application; SourceUser; VirtualSource

The concept of activities, events and snapshots are displayed in the following figure.

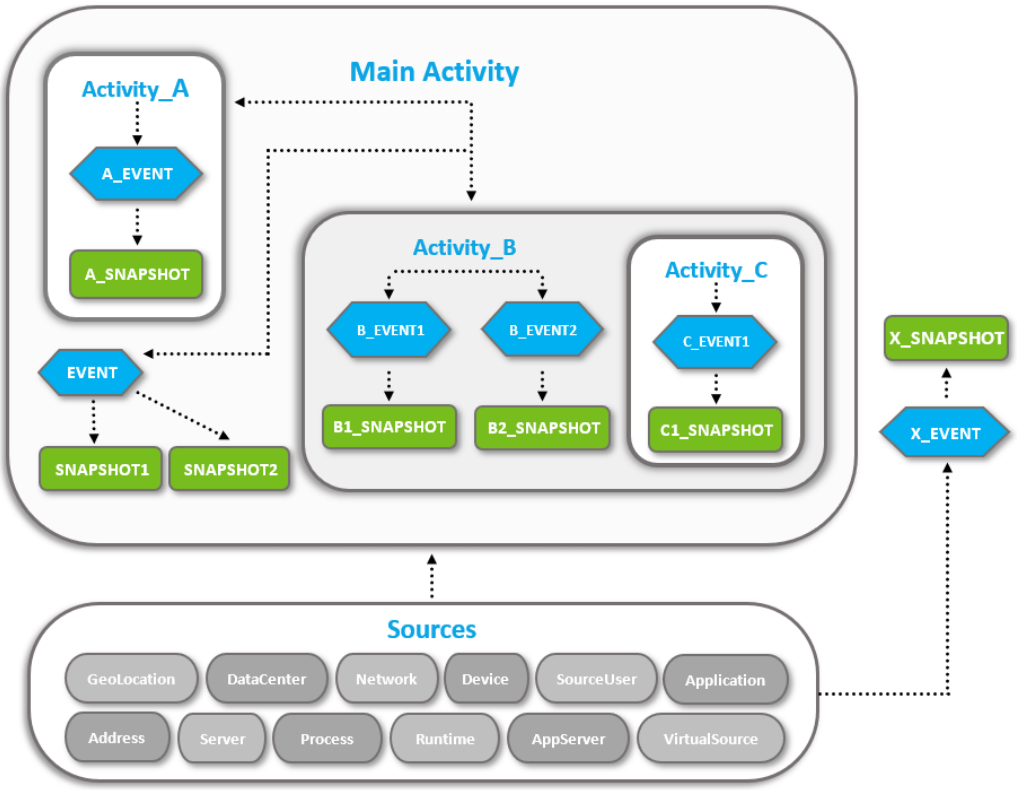


Figure 1.3-A. Activities, Events and Snapshots Concept

1.3.1 Example: My Workday

See *Figure 1.3.1-A* below for an example. The main activity is "My Workday" which is the highest (super) activity. It is the outcome of smaller activities, events, and snapshots.

For example, "My Workday" consists of the following smaller activities:

- I come to work
- I go to lunch
- I have a Skype call

These smaller activities are outcomes of related events. For example, "I come to work," consists of the following related events:

- I open the office door with my key card
- I say hello to my colleague
- I turn on my computer

Events not related to the main activity can exist; they occur on their own without any parent activity. For example, "a bird hit a window." It happened during your workday, and you saw it in your office, but it is not related to your workday (the main activity).

The events can have snapshots. Snapshots are collections of event data aspects which can be measured statistically. For example:

- The amount of time it took to enter the office with my key card
- The eye color of my colleague

- The type of computer I use

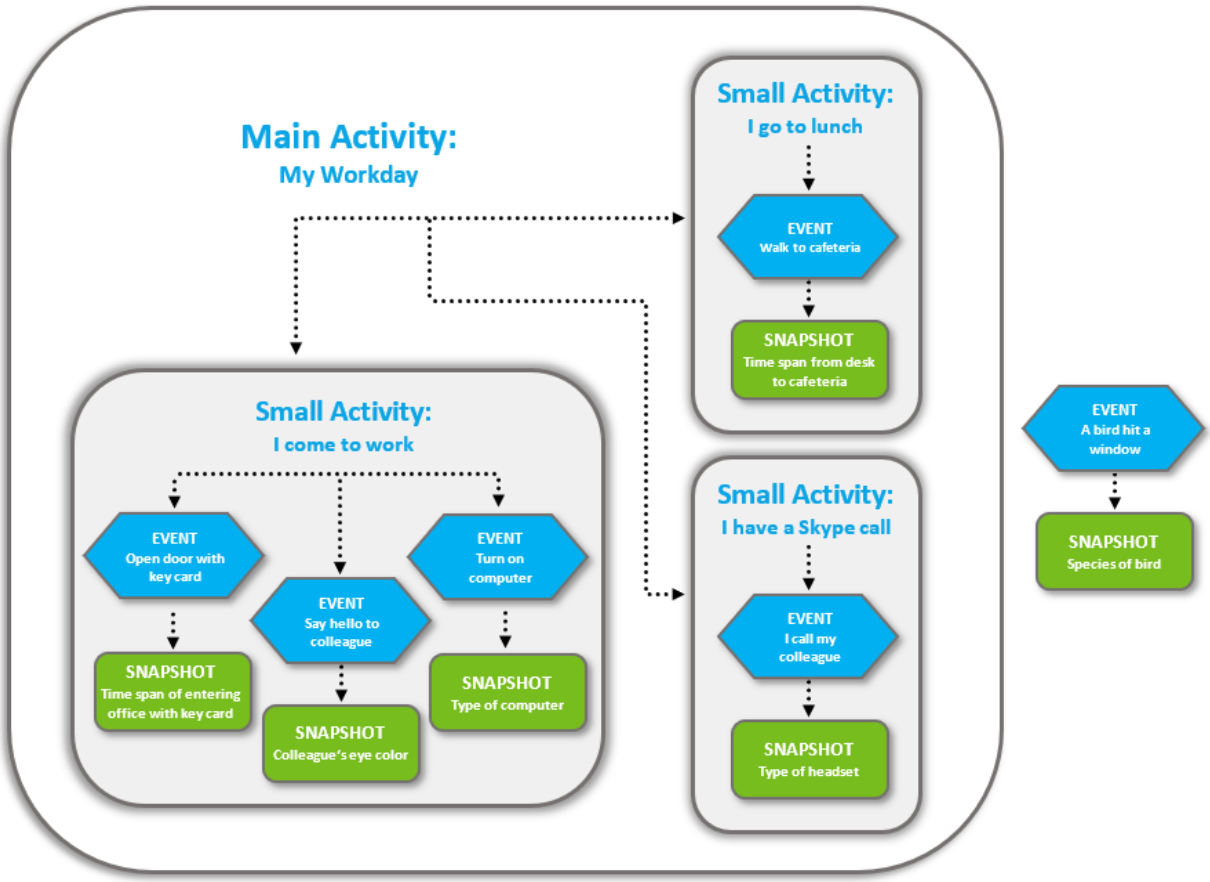


Figure 1.3.1-A. Example: My Workday

1.3.2 Example: Grocery Shopping

You are buying products in the supermarket (this is the activity). The event would be the process of scanning the product barcodes. The snapshots are all related to the event of scanning. Examples are the time it took to scan the products, package color, product weight.

Activity: You are buying products in the supermarket

Event: The process of scanning product barcodes

Snapshots: The time it took to scan, product package colors, product weight

1.4 Data

Users can either stream their data or upload a file (*Section 2.2.1*) to use XRay to analyze and present their data.

1.5 jKQL

jKQL is an English-like query and stream processing language for analyzing machine data in flight (Fast Data) and at rest. It defines the syntax of statements used for manipulating data in the XRay Data Model. It enables the user to search, filter, group, and count data. It is designed to be used by both the business user and the data scientist. Use jKQL to

analyze anomalies, behavior, flows, relationships, and patterns in time-series data as it relates to your business. (See *Chapter 5: Using jKQL*).

Chapter 2: Using XRay

XRay puts your data (streaming or imported from a file) in a repository and displays it as a collection of customized viewlets grouped into one or more dashboards depending on your needs.

2.1 Accessing XRay

1. Open your internet browser.
2. Go to the URL address provided by your System Administrator *and press **Enter**. The* XRay *login dialog box is displayed.*

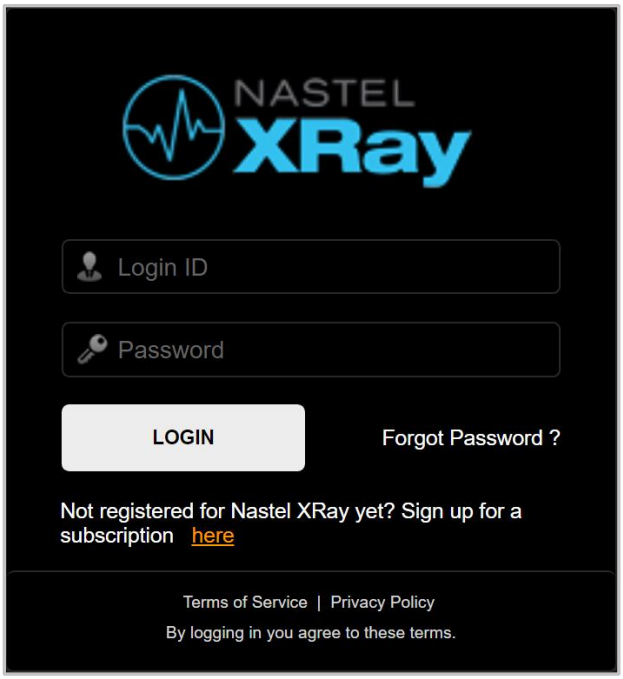


Figure 2.1-A. Login Dialog Box

NOTE

The login dialog box may display your company logo instead of the XRay logo. Please see Branding.

3. Enter your **Login ID** and **Password** and click **LOGIN**. If you would like to reset your password, click **Forgot Password**.

NOTE

Please note that the *Forgot Password* page is specified by going to the **Main Menu** and selecting **Admin Settings** > **Branding** > **Login Page** > **Forgot Password URL** (see *Branding*).

4. By default, the Landing Page is displayed. Please see the next section, 2.2, Landing Page, for more information.

2.2 Landing Page

The Landing Page is used as an initial screen for XRay novices, providing guidance on what the solution provides as well as an easy-to-use wizard for importing data. Experienced users can skip this screen and go directly to their dashboards by clicking **Go to Dashboard**.

The landing page can be accessed at any time by clicking the **Main Menu** and selecting **Landing Page** (*Figure 2.3.6-A*).

The *Landing Page* provides three options:

- **Analyze Your Data** (*Section 2.2.1*): import your data
- **Explore a Demo** (*Section 2.2.2*): sample walk-throughs
- **Go to Dashboard** (*Section 2.2.3*): view your dashboard

Figure 2.2-A. Landing Page

2.2.1 Analyze Your Data: Import a File

To import your data, select **Analyze Your Data** from the Landing Page (*Figure 2.2-A*). The following file formats are supported:

- .xls, .xlsx
- Apache log
- .csv
- Custom (with configuration)

NOTE

From the dashboard, users can also import data by clicking the green **Import Data** button located on the top right of the screen. Please note that this button does not appear for sample repositories.

Perform the following to import your file:

1. Click **Choose File** to select a file to import.

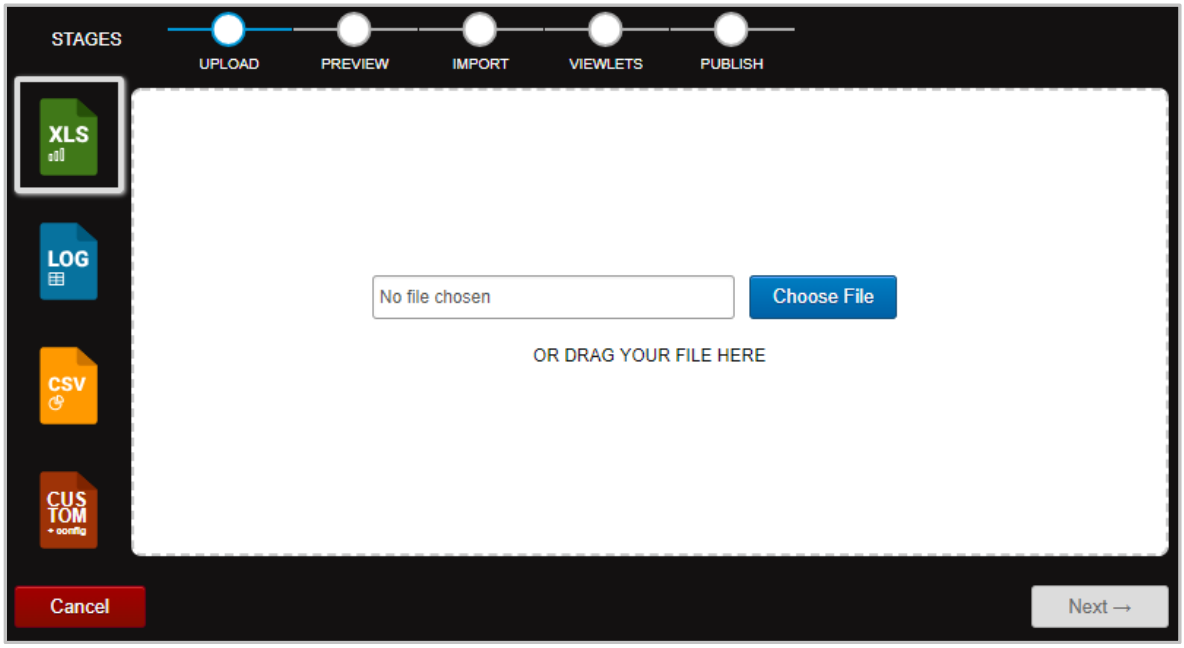


Figure 2.2.1-A. Import File

2. If you have selected the wrong file, you can update it by clicking **Change** and selecting another file. Click **Next**.

Figure 2.2.1-B. Change Your File Selection

3. Analyze the preview. If your file does not look correct, click **Advanced** to change processing options.

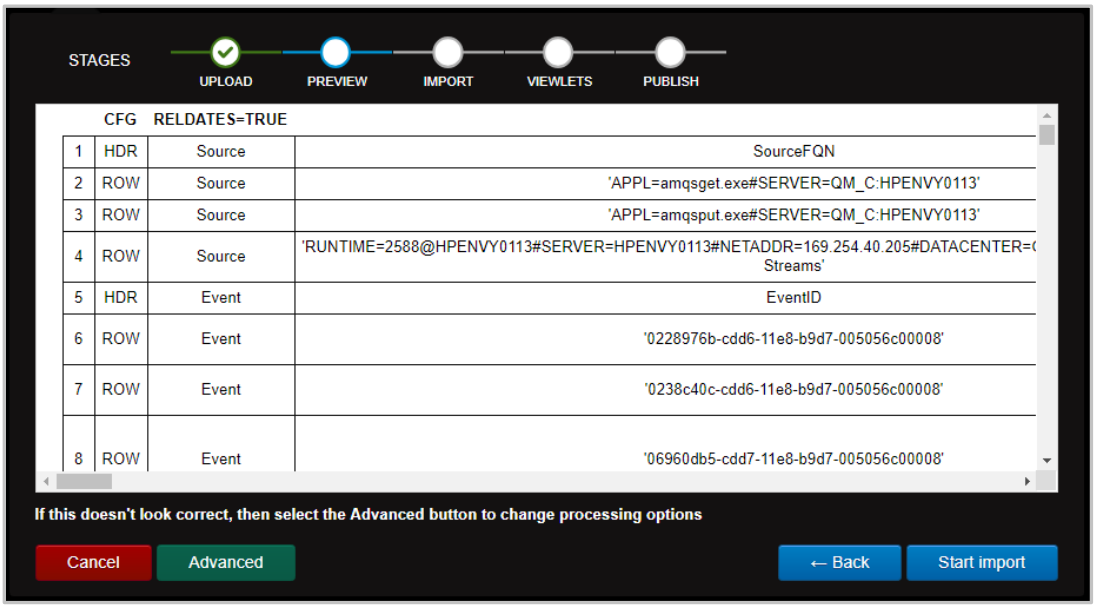


Figure 2.2.1-C. Preview Imported File

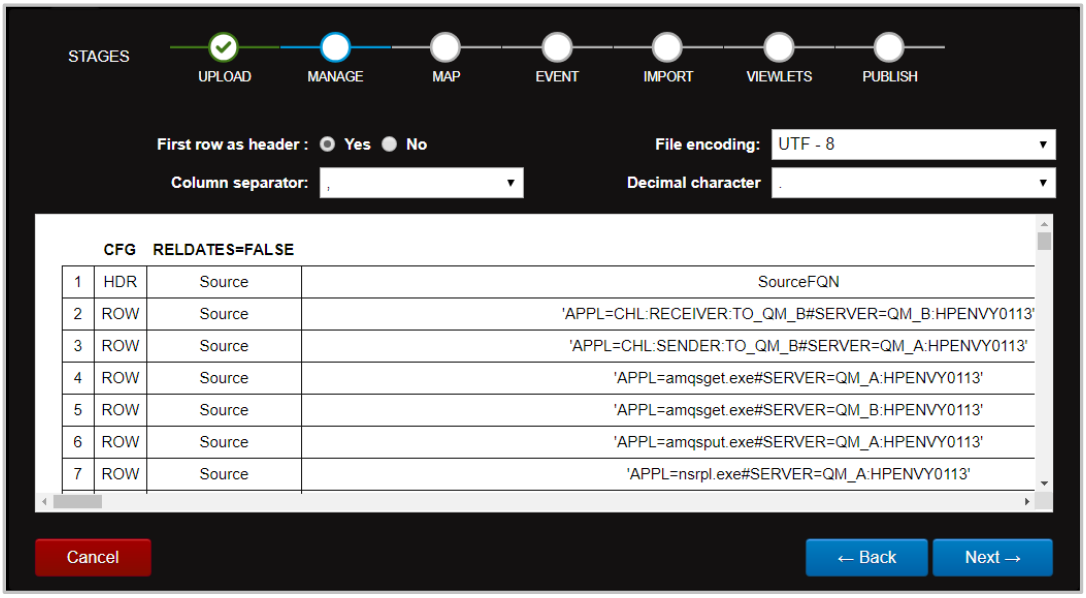


Figure 2.2.1-D. Advanced Options for Imported File

4. The following options are available:

   - **First row as header** – Specify if the first row is a header row.
   - **File encoding** – Select encoding type.
   - **Column separator** – From the drop-down menu select the column delimiter: comma, semicolon, or tab.
   - **Decimal character** – Specify the decimal number delimiter: period or comma.

   After you have made your selections, click **Next**.

5. This screen allows you to:

- Change the name of column headers. You can select from the drop-down list or type over the existing header name.
- Map data imported into XRay to an existing field within the XRay data model. Alternatively, by selecting the option "new property," this can be used to import custom data and label it with a name that has relevance to the user.
- Remove a column.

Click **Next** to continue.

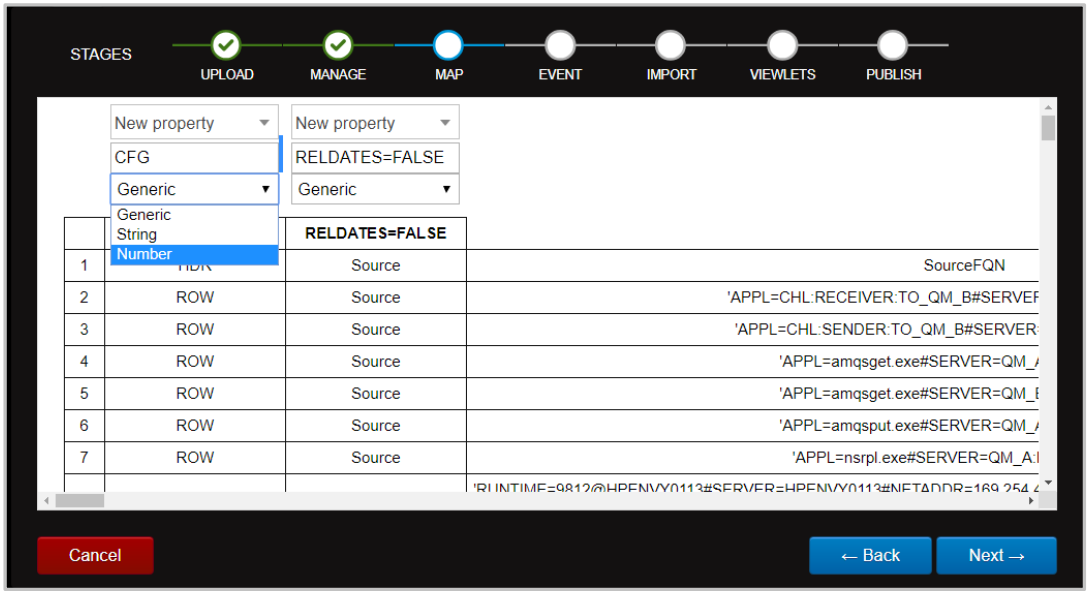


Figure 2.2.1-E. Additional Advanced Options for Imported File

6. The fields on the following screen are optional. Populate them to make the data more detailed.
   - **Application name:** Enter the application name.
   - **Server name:** Enter the name of the server.
   - **Network address:** Enter the data source IP address.
   - **Data center:** Specify the name of the data center.
   - **Geo address** – Click **Use current location** to populate the field with the latitude and longitude of your current location.

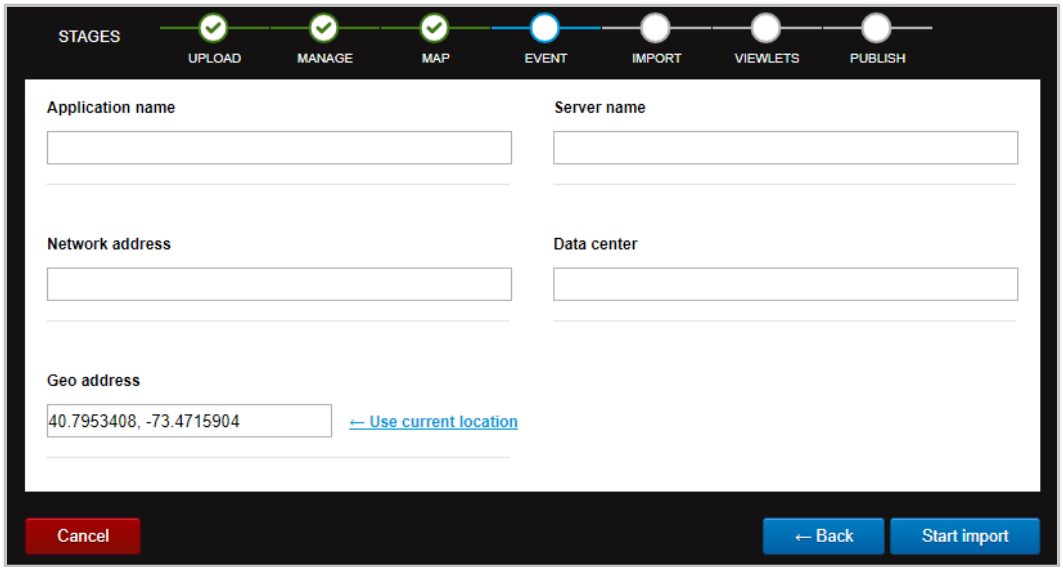


Figure 2.2.1-F. More Advanced Options for Imported File

Click **Start import** to import your file.

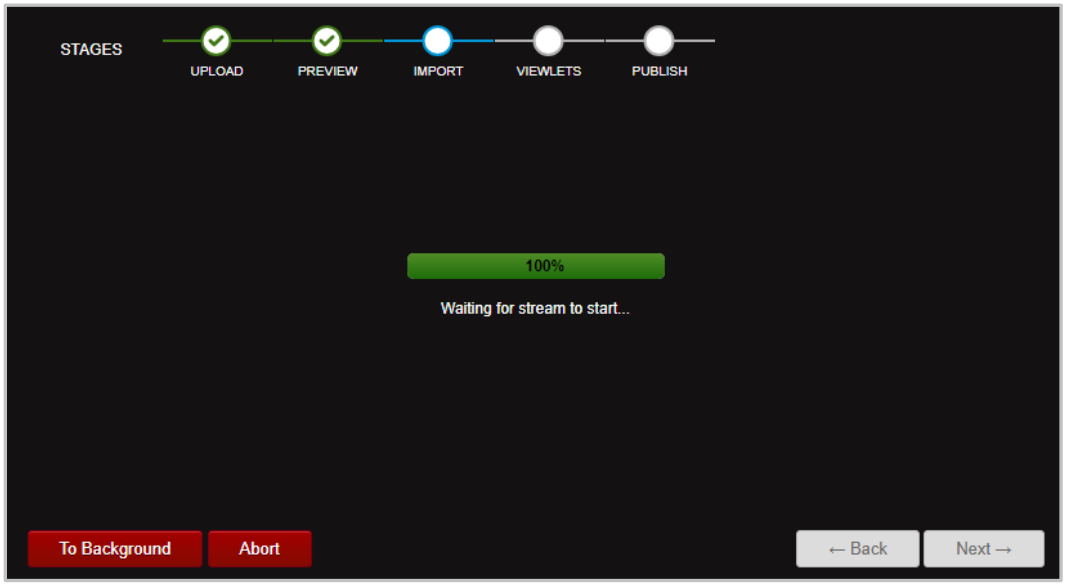


Figure 2.2.1-G. Import Process

The import process can be run in the background by clicking **To Background**. To cancel the import, click **Abort**. A confirmation dialog box will appear when the import is finished. Click **Next**.

If you selected to run the import process in the background, you can view the imported data by going to **Main Menu** > **Import / Export** > **Data** where you can create viewlets and specify a dashboard (see *Section* 2.6.1, Import Data into a Dataset, for more information). Otherwise, continue to step 7.

7. A summary of the data will be displayed on the following screen. Click next to select viewlets or click **Finish** to load default viewlets.

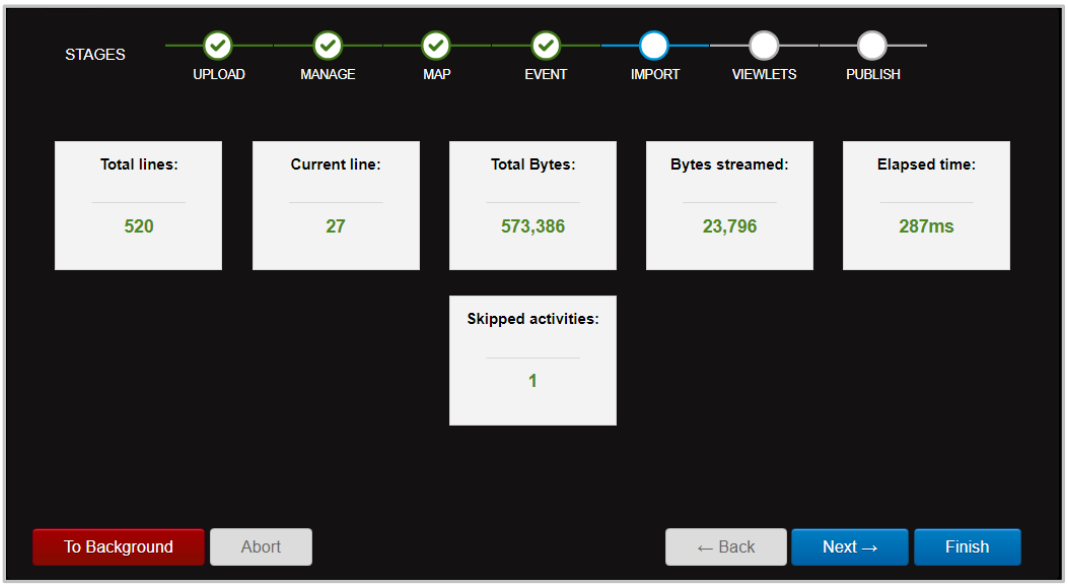


Figure 2.2.1-H. Imported File Summary

8. A default set of viewlets is provided. By default, all viewlets are selected (selected viewlets appear in blue). Simply click a viewlet to unselect it (unselected viewlets appear in full color). Click **Next** to add the selected viewlets to a dashboard.

Figure 2.2.1-I. Viewlets for Imported File

9. Add your viewlets to an existing dashboard or create a new dashboard.
   - **Add to Dashboard** section– Add your viewlets to an existing dashboard by selecting a dashboard from the **Select dashboard** drop-down list.
   - **Create New Dashboard** section – Add your viewlets to a new dashboard. Enter a name for the new dashboard and select one, two, or three columns.

- **RUM** (Real User Monitoring): illustrates how to determine the root cause of poor end-user experience. (Go to *https://www.youtube.com/watch?v=OuYvkRix6iM* to watch a brief use-case demonstration.)
- **Order Tracking:** illustrates how to trace the flow of an order from order placement through verification, payment, shipping, and more.
- **IoT**: illustrates the Internet of Things (IoT) as used in athletics, specifically basketball.
- **DevOps**: illustrates how to analyze the Build and Deploy processes.

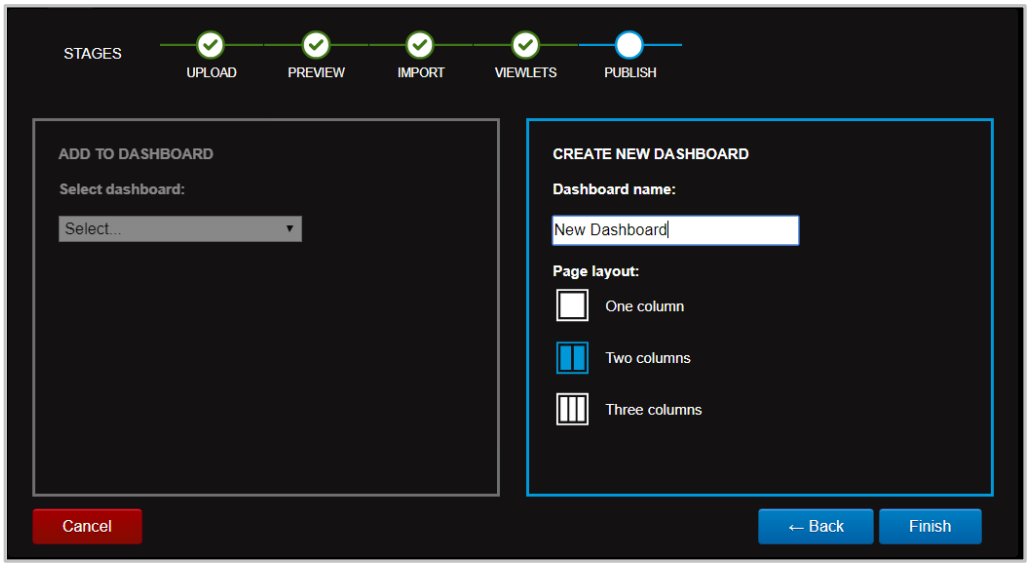


Figure 2.2.1-J. Add Imported File Viewlets to Dashboard

Click **Finish** to display your imported data in your dashboard.

NOTE

See *Section 2.6.1, Data*, for information on how to view and manage all data previously imported.

2.2.2 Explore a Demo

Clicking **Explore a Demo** on the *Landing Page* provides walk-throughs of the following four business issue scenarios where XRay can be used to solve a problem:

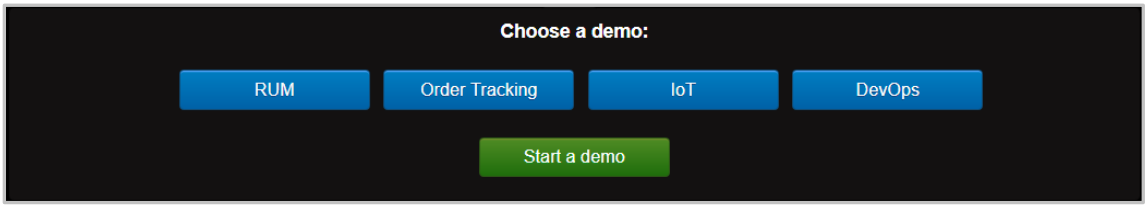


Figure 2.2.2-A. Choose a Demo

Each walk-through starts with an explanation of the problem, the solution, and the steps taken to solve the problem. To view a demo, select it and click **Start a demo**.

At the end of each demo, there is an option to load your own data into the example. Click **Load your data** and select your file (refer to *Section 2.2.1, Analyze Your Data*, for information on importing data).

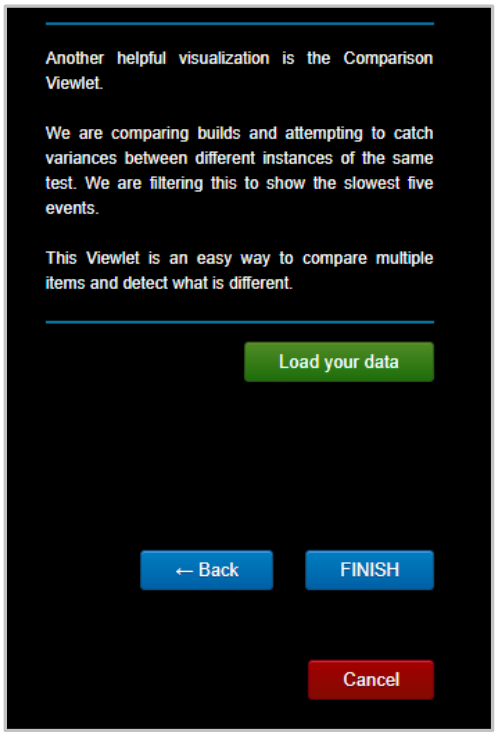


Figure 2.2.2-B. Load Your Data

The next section is a walk-through of the **Order Tracking** demo.

2.2.2.1 Order Tracking Demo

After selecting **Explore a Demo**, select **Order Tracking**. Click **Start a demo**.

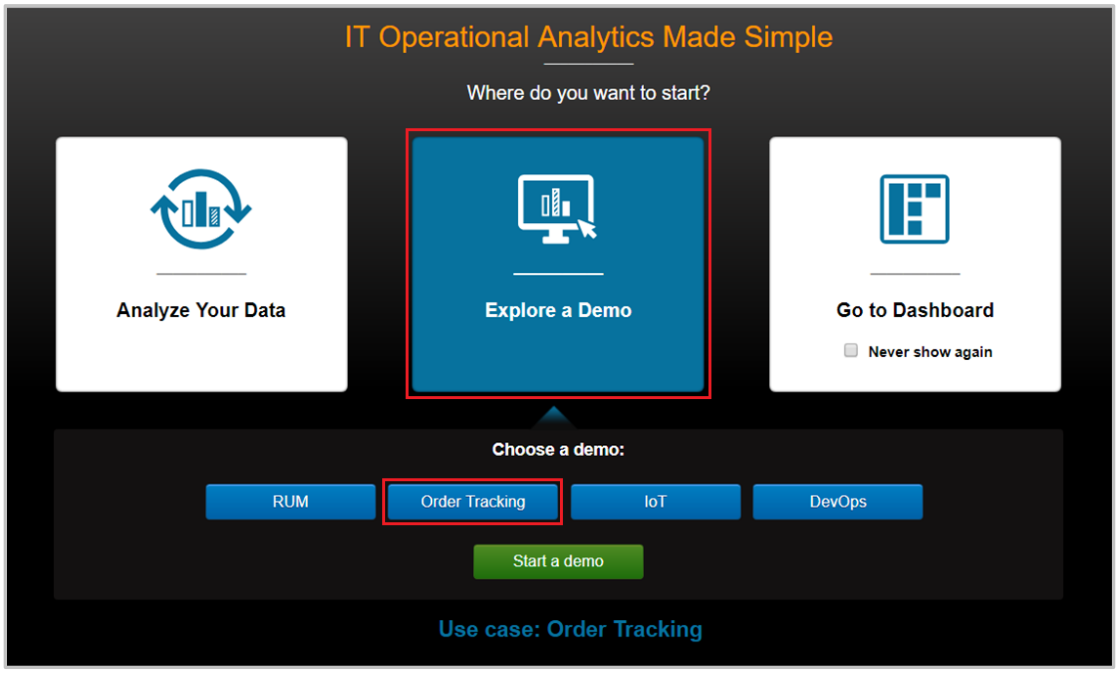


Figure 2.2.2.1-A. Start a Demo

A viewlet is displayed which shows a topology map of the business milestones. The jKQL query that produced this viewlet is shown at the top of the viewlet.

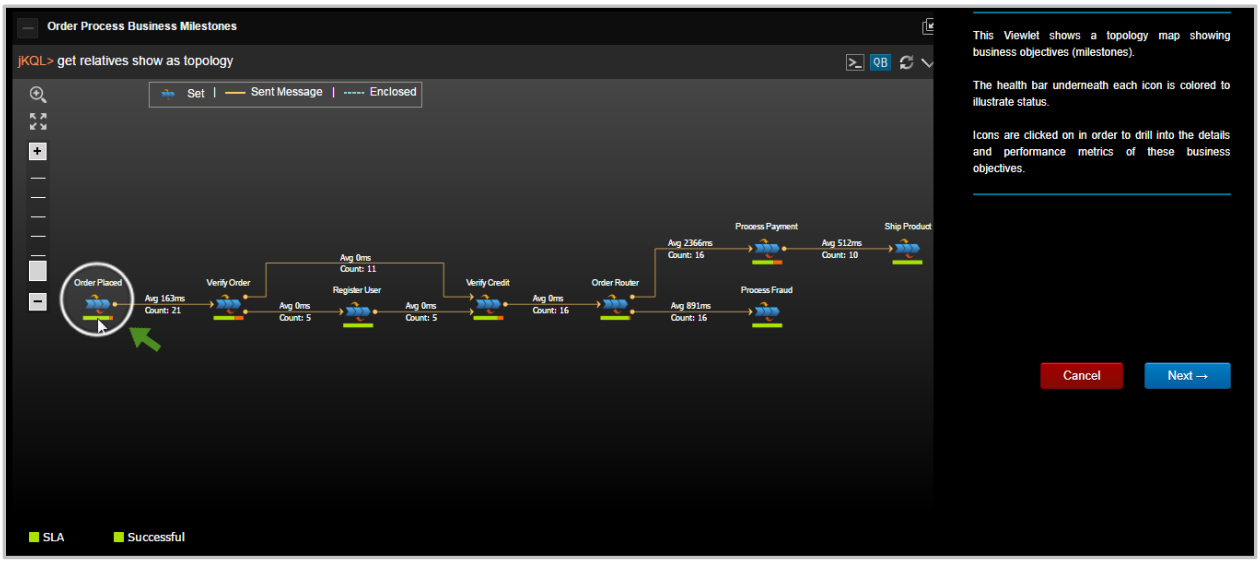


Figure 2.2.2.1-B. Order Tracking Demo – Page 1

- The health bar under each icon is color coded to reflect status (green = good, yellow = warning, red = critical).
- To drill into the details of an event, click the icon. Click **Next** to view the details of the circled milestone, **Order Placed**.
- Clicking the health bar for **Order Placed** produces a pop-up menu for drill-down into SLAs and performance metrics for transactions and activities.

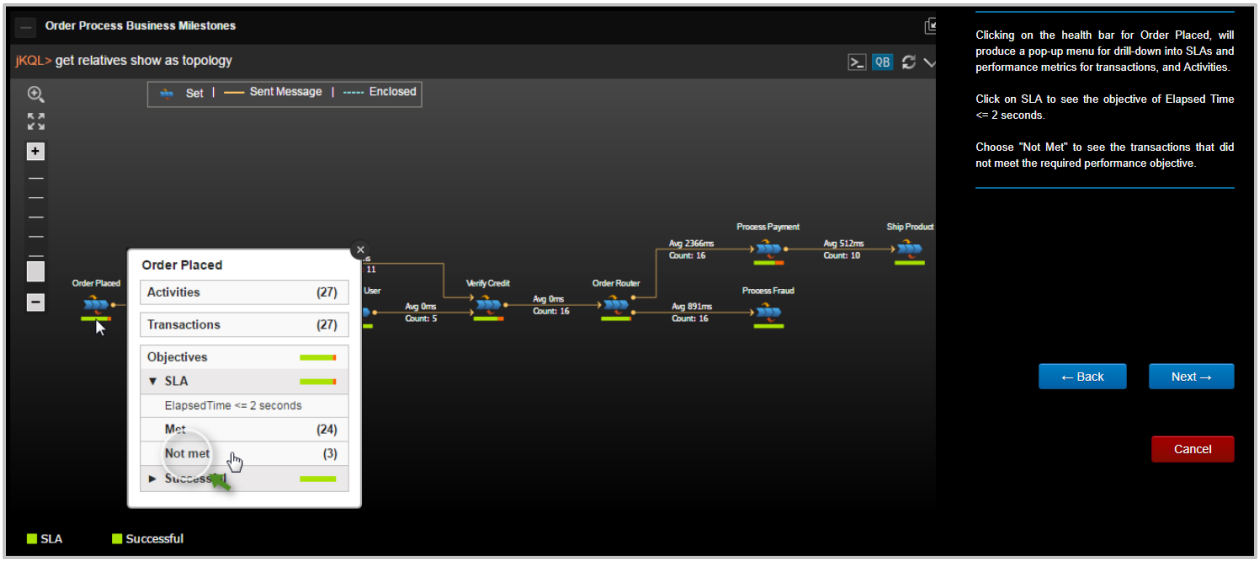


Figure 2.2.2.1-C. Order Tracking Demo – Page 2

- Click **SLA** to see the objective of **Elapsed Time <= 2 seconds**.
- Choose N**ot Met** to see the transactions that did not meet the required performance objective.
- Click **Next** to proceed.

This screen shows the open **Console** where the slow transactions are listed. In this example, a transaction was selected, and topology chosen. Click **Next** to view the topology.

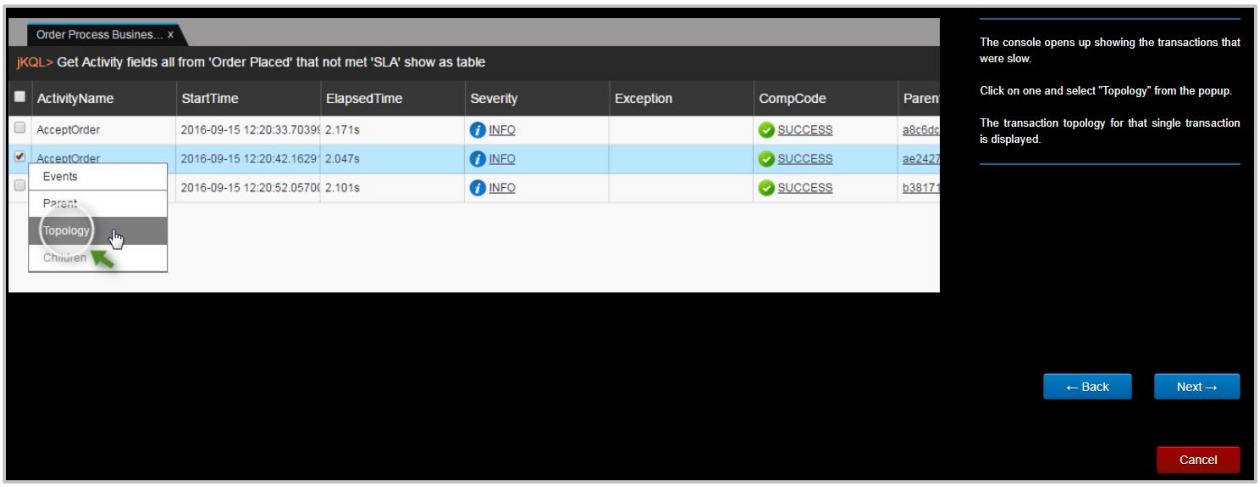


Figure 2.2.2.1-D. Order Tracking Demo – Page 3

This screen shows the topology. By clicking the various icons, you can drill down into each event to see the root cause of the problem.

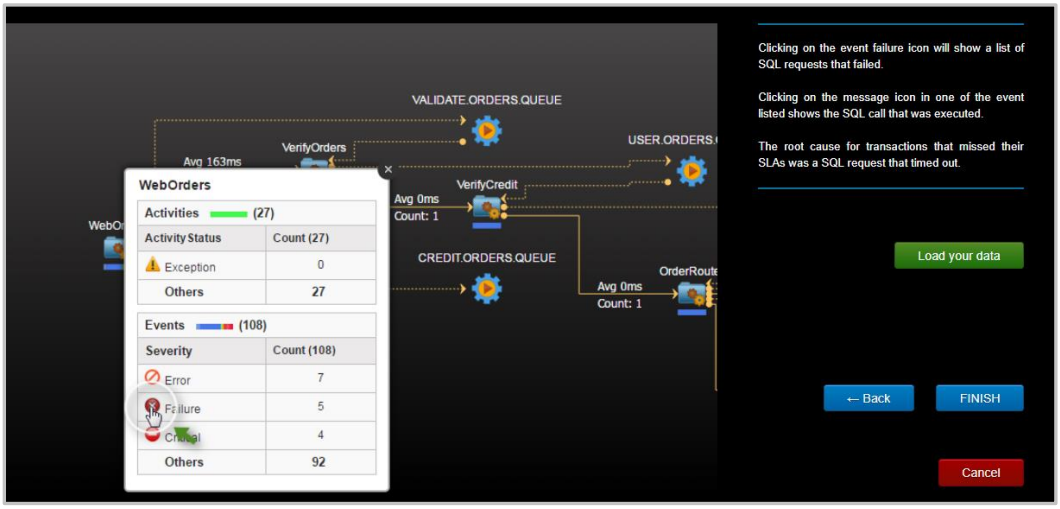


Figure 2.2.2.1-E. Order Tracking Demo – Page 4

This is the end of this demo. You can either:

- Return to the *Landing Page* by clicking **Finish**.

  OR

- Import your own data file into the example by clicking **Load your data** and following the prompts (see *Section 2.2.1, Analyze Your Data*, for more information).

2.2.3 Go to Dashboard

The **Go to Dashboard** option on the Landing Page takes you to your dashboard if you have previously created one. If you have not, you will be asked to create one (*Figure 2.4.2.1-B*).

See the next section, Disable Landing Page, for information on the **Never show again** option.

2.2.4 Disable Landing Page

The landing page can be disabled to allow users to view their dashboard immediately after logging in. Perform one of the following to disable the landing page:

- Before clicking **Go to Dashboard** from the Landing Page, enable the **Never show again** option.

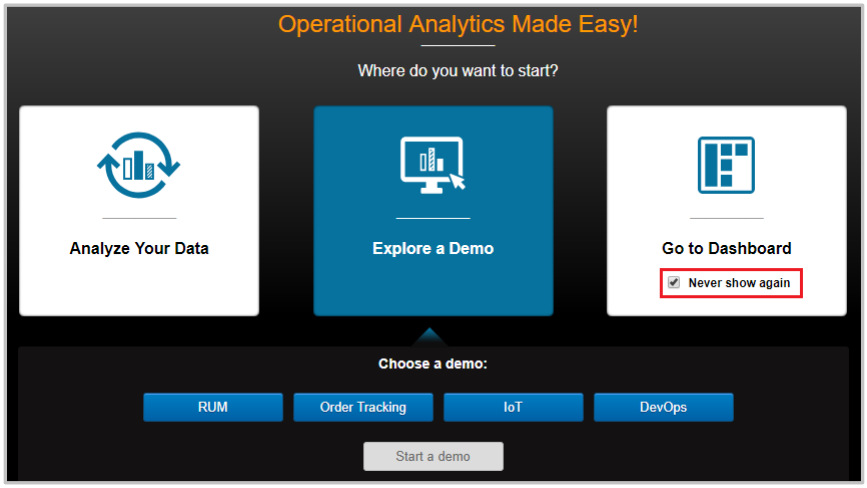


Figure 2.2.4-A. Landing Page – Never Show Again

- Or on the **Main Menu**, select **User Settings** > **Manage Settings** to open the *Manage Global Settings* dialog box. Select **Off** for **Landing page** and click **Save** (see Manage Settings).

2.3 Toolbar

The main toolbar of the screen has the following options. Use the figure below as a reference.

A-C: **Repository Quota Limits** (*Section 2.3.1.1*)

- A: **Data Points**
- B: **Stream Messages Per Day**
- C: **Stream Bytes Per Day**

D: **Repository** Drop-down (*Section 3.2.5*)

E: **Search** (*Section 2.3.4*)

F: **Current User**

G: **Help** (*Section 2.3.2*)

H: **Log Out** (*Section 2.3.3*)

I: **Default Date & Time** (*Section 3.2.5*)

J: **Main Menu** (*Section 2.3.6*)

K: **Import Data** button (*Section 2.2.1*)

L: Add **Viewlet** button (*Section 2.5.1*)

M: **Modify** button (*Section 2.5.8.1*)

Figure 2.3-A. Main Toolbar

2.3.1 Repository

To load a repository, select it from the **Repository** drop-down menu on the main toolbar as seen in the figure below. Repositories appearing under **Global Repositories** are sample repositories available to all users (see *Section 2.4.1, Sample Dashboards*, for more information).

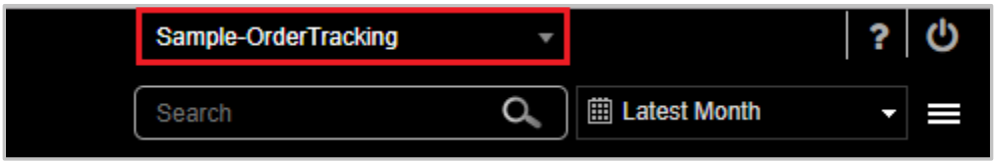


Figure 2.3.1-A. Repository

Each section within the **Repository** drop-down menu represents a different organization (organization name will be bolded) and their repositories. Use the search field to quickly search the menu.

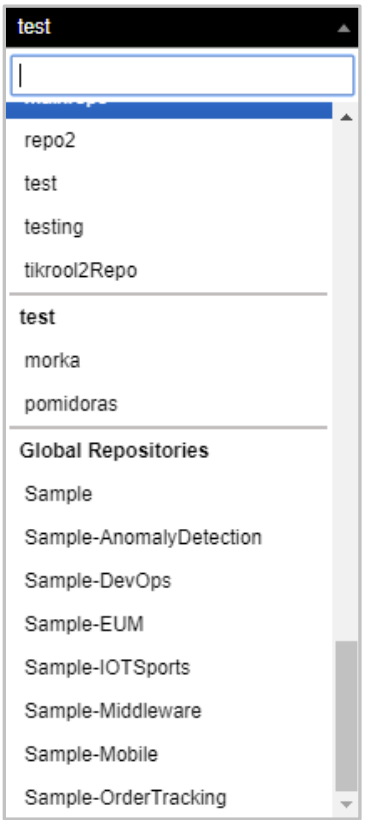


Figure 2.3.1-B. Repository Drop-down Menu

2.3.1.1 Repository Quota Limits

The **Repository Quota Limit** drop-down (immediately to the left of the **Repository** drop-down) displays your data and repository limits (dependent on your license). Click the **Repository Quota Limit** drop-down menu to view limit amounts for **Data points**, **Stream messages per day** and **Stream bytes per day**.

For limited licenses, the data point usage percentage will display within the drop-down.

Figure 2.3.1.1-A. Data Points Percentage

For unlimited licenses, **Unlimited** will display.

Figure 2.3.1.1-B. Unlimited Data Points

Clicking on the limit bar will generate a **Data Points** dashboard consisting of viewlets displaying data points of events, activities, and snapshots from the latest week (see *Section 5.4* for examples of "*Last*" and "*Latest*").

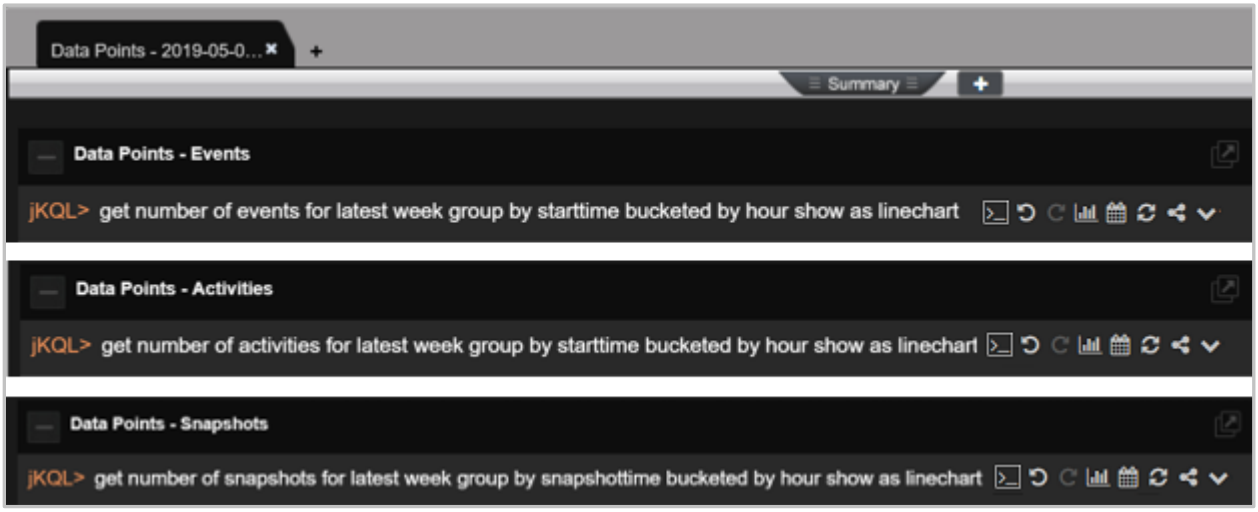


Figure 2.3.1.1-B. Data Points Dashboard

2.3.2 Help

Click the question mark icon on the toolbar (*Figure 2.3-A*) to access the help page. This link is defined in **Main Menu** > **Admin Settings** > **Branding** > **Support Url** (see Section *3.1.1.2*, *Branding*).

2.3.3 Logout

Click the **Logout** icon on the toolbar (*Figure 2.3-A*) to exit the system. Before exiting, the following dialog box appears asking if you would like to save or discard updates made.

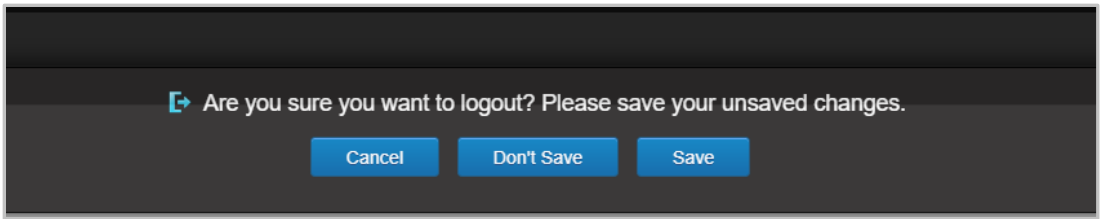


Figure 2.3.3-A. Save Changes

2.3.4 Search

At the top right of the screen there is a **Search** field used to search through event data. Within the **Search** field, enter a word or phrase relevant to event data. The search will run a jKQL query as follows:

```
jKQL> Find 'typed search word' in Events
```

The results will display in a viewlet within a new dashboard titled, **Search - *<search query>* - *<date and time>***. In the below example, **longest activity** was entered in the **Search** field.

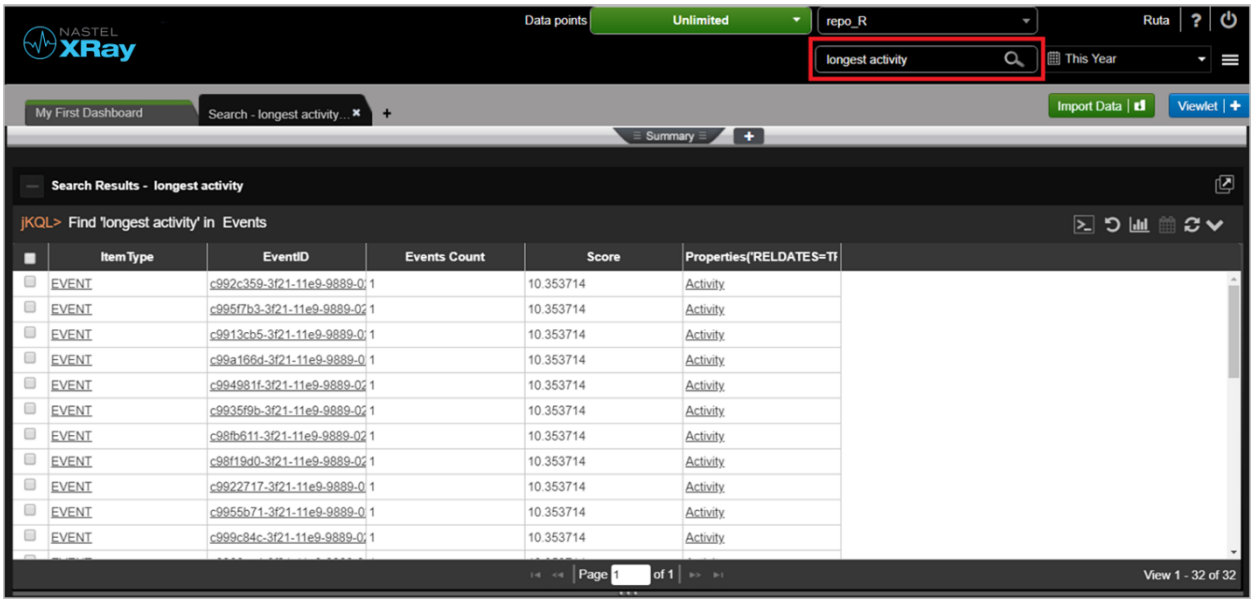


Figure 2.3.4-A. Search Field

NOTE

Please note that the Search box only searches events data. To search through activity and snapshot data, use jKQL queries (see *Chapter 5*).

2.3.5 Default Date & Time Range

Use the **Date & Time Range** option on the toolbar (*Figure 2.3.5-A*) to set the date and time for the viewlets of the selected repository. Click the drop-down menu to customize. The following are possible options:

Predefined

- This: Hour, Week, Month, Year
- Last: Hour, Week, Month, Year
- Today
- Yesterday

Custom

- Limit
  - This
  - Earliest
  - Last
  - Latest
- Value: Enter a number value (available when *This* is not selected)
- Units
  - Minute
  - Hour
  - Day
  - Week
  - Month
  - Year

Date range

- From: Enter the start date and time, or select from the scheduler with additional options.
- To: Enter the end date and time, or select from the scheduler with additional options.

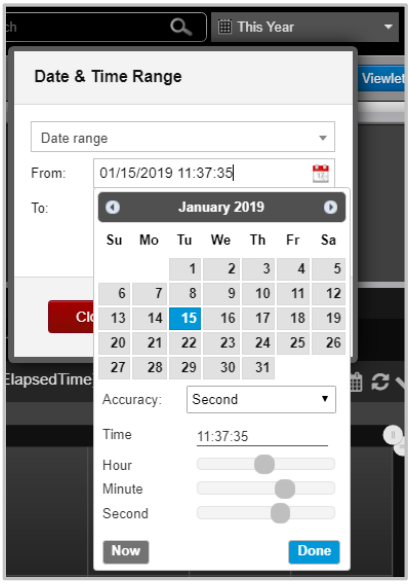


Figure 2.3.5-A. Date & Time Range

Please note that this date and time range will only be effective for the current session; when you exit and log back in, the default date and time range will be used. To set the default date and time range, go to **Main Menu** > **User Settings** > **Manage Settings** (see *Section 3.2.5, Manage Settings*).

2.3.6 Main Menu

The **Main Menu** is accessed by clicking the menu icon at the top right of the screen. This menu provides the functionality described in Table 2 below.

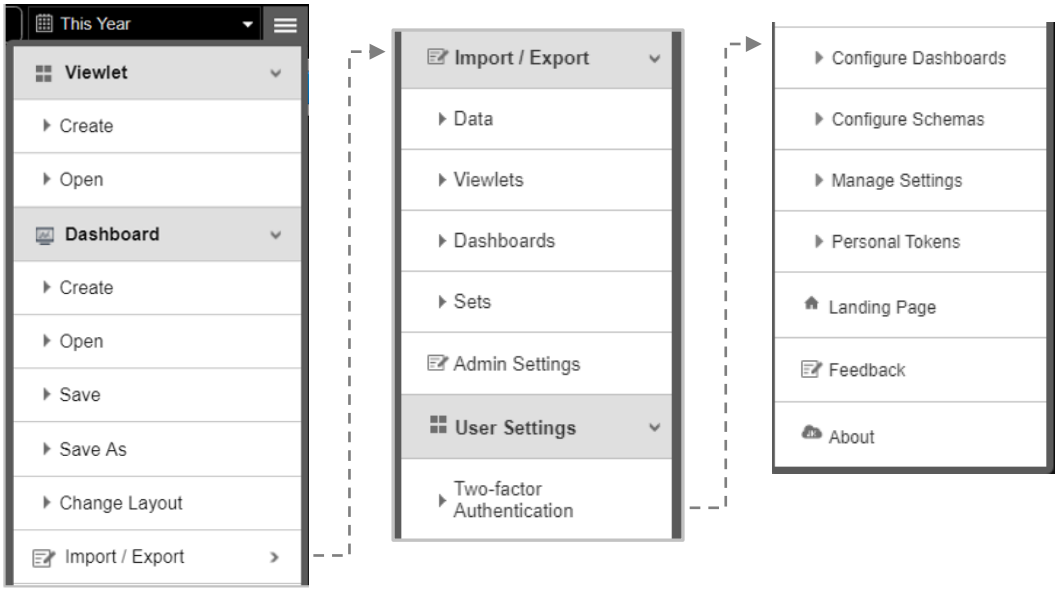


Figure 2.3.6-A. Main Menu

Table 2. Main Menu Functions	
Viewlet	Expand to access the following viewlet options: • **Create**: *Section 2.5.1* • **Open**: *Section 2.5.1.4*
Dashboard	Expand to access the following Dashboard options: • **Create**: *Section 2.4.2.1* • **Open**: *Section 2.4.2.2* • **Save**: *Section 2.4.2.5* • **Save As**: *Section 2.4.2.6* • **Change Layout**: *Section 2.4.2.8*
Import / Export	Expand to access the Import / Export sub-menu. See *Section 2.6, Import / Export*, for more information.
Admin Settings	Opens the *Admin Settings* dialog box. Please see *Section 3.1, Admin Settings*, for more information. Please note that only administrative users with repository permissions will have this option available.
User Settings	Please see *Section 3.2, User Settings*, for more information.
Landing Page	Takes you to the Landing Page *(Figure 2.2-A)*.
Feedback	Opens the page to leave feedback and ask questions. This page is defined in **Branding** > **Index Page** > **Leave Feedback**.
About	Displays the user's data point definitions and application information. Includes links for getting collectors and license information: • Click **Get Collectors** to open the page of open-source collector download links. The Get Collectors URL can be changed in **Branding** > **Index Page** > **Collectors URL**. • Click **License** to view license and upgrade information. The license URL can be changed in **Branding** > **Index Page** > **License URL**.

2.4 Dashboards

A dashboard is a collection of viewlets. Your data repository can have multiple dashboards. Each dashboard is displayed by clicking the desired dashboard tab located at the top of the screen.

Figure 2.4-A. Dashboard Tabs

A red asterisk appearing at the front of a dashboard name signifies an unsaved dashboard. To save a dashboard, right click the dashboard tab and select **Save**, or select **Dashboard** > **Save** from the main menu. If your browser crashes before saving, the dashboard will be restored upon next login.

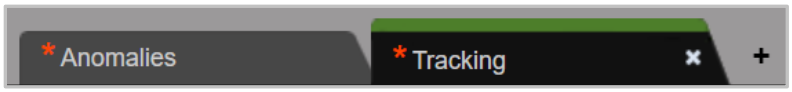


Figure 2.4-B. Unsaved Dashboards

2.4.1 Sample Dashboards

In your system sample repositories are provided. You can find these within the **Global Repositories** section of the **Repository** drop-down menu (see *Section 2.3.1*). It is recommended to review the sample repository dashboards before creating your own.

The **Sample Order Tracking** dashboard is shown in the figure below. To open this dashboard, select the **Sample-OrderTracking** repository. The individual viewlets are described in *Section 2.5.4, Viewlet Chart Types and Samples*.

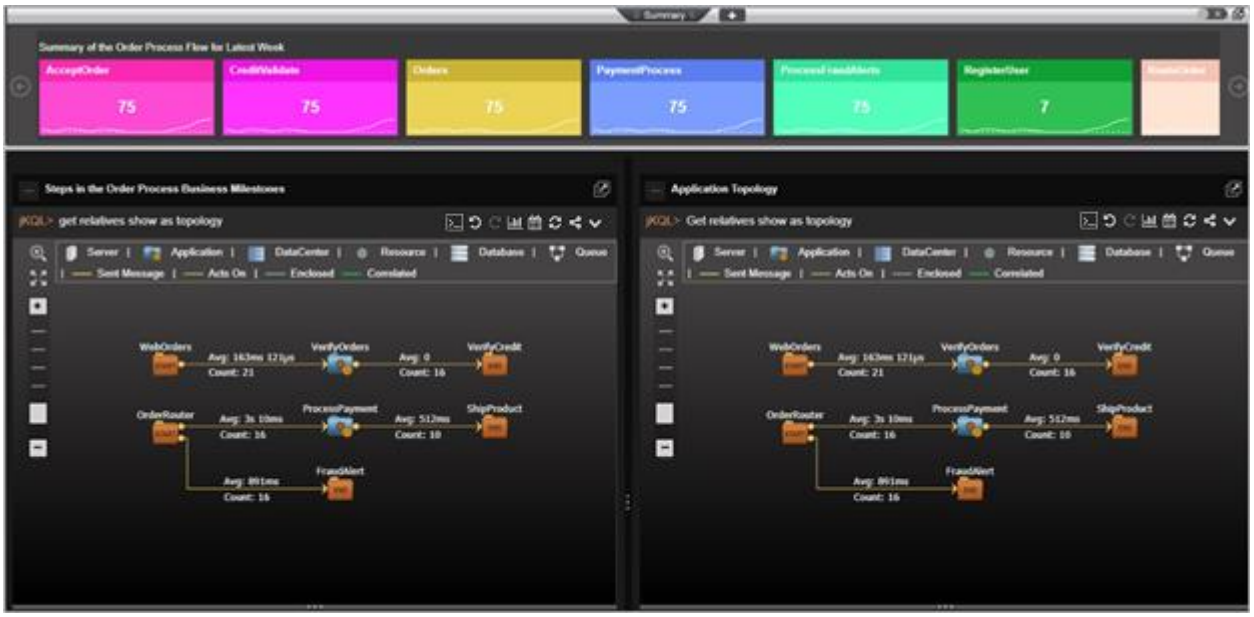


Figure 2.4.1-A. Sample Dashboard

Global Repositories have limited features. For example, if you right click on a dashboard tab of a Global Repository, **Save As** and **Set As Default** are not available in the pop-up menu because this is a "sample" repository which is read-only. Updates made in sample repositories are not saved before changing the repository or logging out. If it were a repository created by you, all functions would be available.

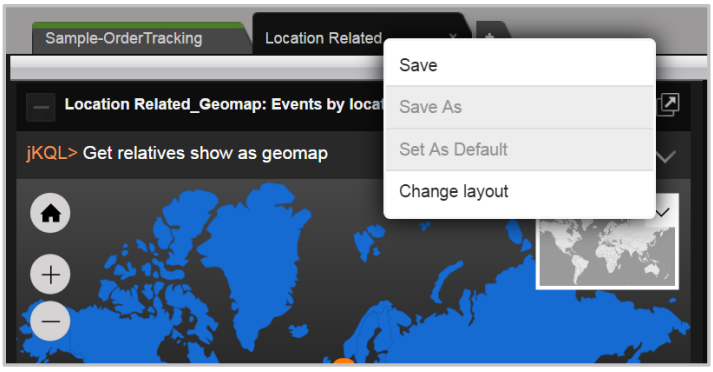


Figure 2.4.1.B. Dashboard Menu

2.4.2 Dashboard Actions

2.4.2.1 Create

Users create multiple dashboards as a way of grouping different data or analytics. While all could be on a single dashboard within the same repository, it can be more convenient to break them up by separate dashboard tabs.

After clicking **Go to Dashboard** from the **Landing Page** (*Figure 2.2-A*), the *Create new Dashboard* dialog box opens if no dashboards have been previously created.

You can also create a new dashboard by going to **Main Menu** > **Dashboard** > **Create** (*Figure 2.3.6-A*) or by clicking the plus button + immediately to the right of the existing dashboard tabs.

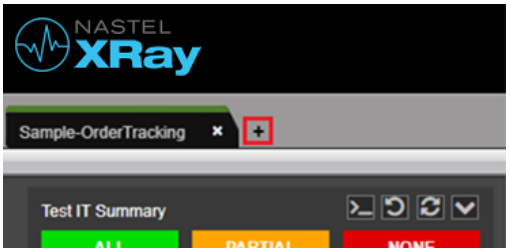


Figure 2.4.2.1-A. Create Dashboard Button

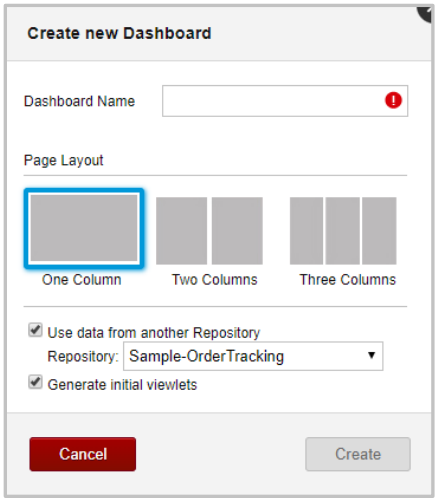


Figure 2.4.2.1-B. Create New Dashboard Dialog Box

To create your dashboard:

1. Enter a name for your dashboard.
2. Select the number of columns.
3. To make writing queries easier, enable the **Use data from another Repository** option to specify data will come from a distinct repository. Select the repository from the drop-down menu. The repositories you can select from are the ones that are available to you, including global repositories.
4. To create a set of default Viewlets, select **Generate initial viewlets**.

5. Click **Create**. Your dashboard has been added. The figure below shows a new dashboard with a set of default viewlets displayed as thumbnails. By clicking the viewlet thumbnail, the viewlet opens in the Console at the bottom of the screen.

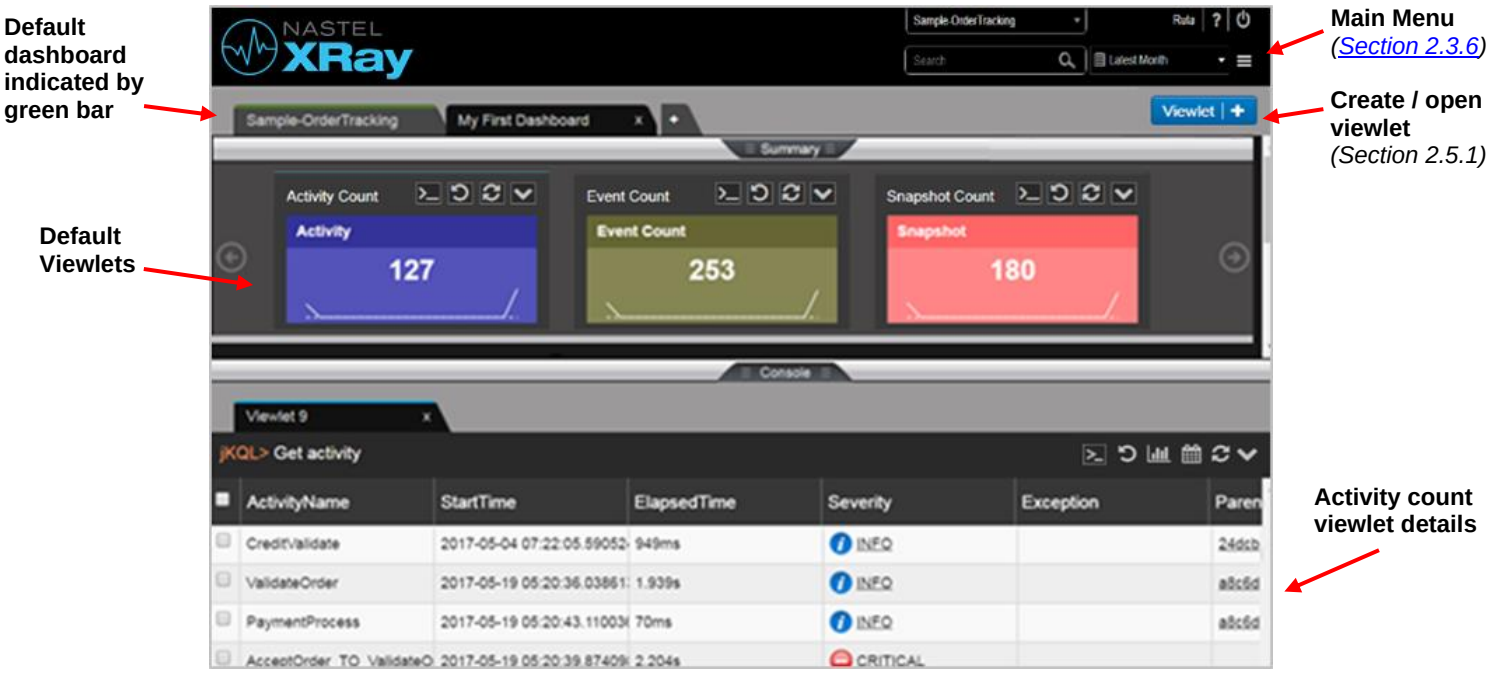


Figure 2.4.2.1-C. Default Viewlets

The upper portion of the screen above is called the Summary Panel. It contains summary viewlets which are used when counting the number of objects like events, activities, or snapshots and presenting the count in a summarized view. It can be closed and default to closed when no summaries are defined for that dashboard (See *Section, 2.5.4.10 Summary*).

2.4.2.2 Open Dashboards

To open a dashboard, go to **Main Menu** > **Dashboard** > **Open**. The *Open Dashboard* dialog box opens. If there are no additional dashboards, this option will be greyed out.

All saved dashboards will be listed in the **Dashboard Name** drop-down menu. Select the desired dashboard and click **Open**.

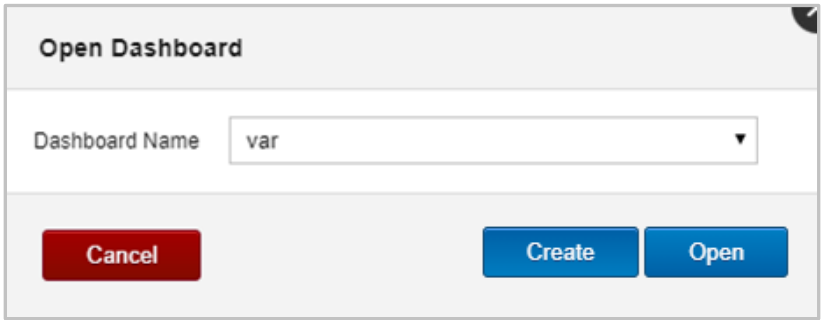


Figure 2.4.2.2-A. Open Dashboard

2.4.2.3 Menu

To display the menu of a dashboard, right click on the dashboard tab. A pop-up menu opens with the following options:

- **Assign to Teams** (*Section 2.4.2.4*)
- **Save** (*Section 2.4.2.5*)
- **Save As** (*Section 2.4.2.6*)
- **Set As Default** (*Section 2.4.2.7*)
- **Configure** (*Section 2.4.2.8*)
- **Close other tabs**: close all other tabs except for the tab you are currently viewing
- **Close tabs to the right**: close all tabs appearing to the right of the tab you are currently viewing
- **Close tabs to the left**: close all tabs appearing to the left of the tab you are currently viewing

2.4.2.4 Assign to Teams

The **Assign to Teams** option allows you to enable view and modify permissions for teams. Hover over this option to view all teams which have been added to the dashboard's repository.

NOTE

To create a team, add a user to a team and manage team repositories, go to **Main Menu** > **Admin Settings** > **Organization** > **Teams** (see Teams for more information).

After hovering over the **Assign to Teams** option, simply click the eye icon to enable viewing privileges and/or the pencil icon for modifying privileges (clicking on the pencil icon selects both options). Enabled options for the teams will appear in green. Assigning teams viewing privileges prevents the users from saving changes.

Please note that this feature is only available for creators of the dashboard (dashboard owners), users who belong to a team with modification permissions, and repository admin users.

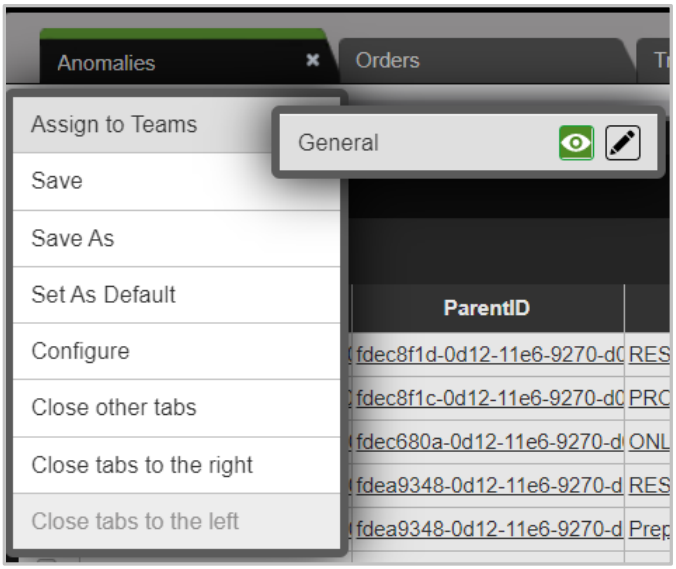


Figure 2.4.2.4-A. Assign to Teams

2.4.2.5 Save

To save a dashboard to a repository so that it appears every time you log in, either right click on the dashboard tab and select **Save** from the pop-up menu (*Figure 2.4.2.4-A*), or go to **Main Menu** > **Dashboard** > **Save** (*Figure 2.3.6-A*). A dialog box appears confirming that the dashboard has been saved.

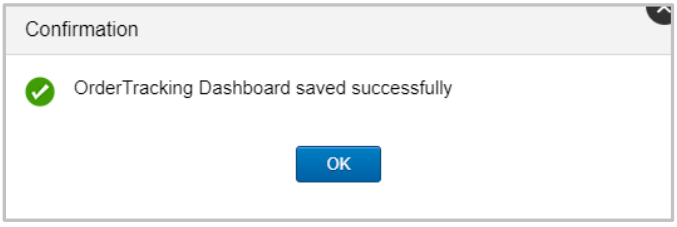


Figure 2.4.2.5-A. Dashboard Successfully Saved

2.4.2.6 Copy (Save As)

To make a copy of a dashboard, go to **Main Menu** > **Dashboard** > **Save As** or right click the dashboard tab and select **Save As** from the pop-up menu. Type in a new name for the dashboard.

2.4.2.7 Default

Your default dashboard is indicated by a green bar on the top of the tab. When a repository is loaded, the default dashboard will automatically display first.

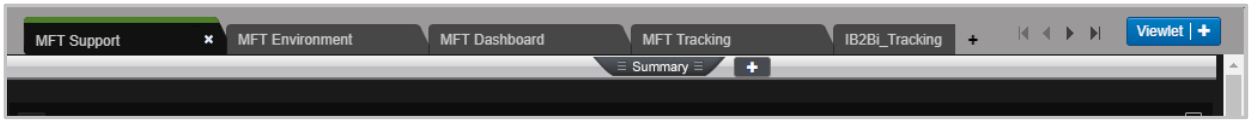


Figure 2.4.2.7-A. Dashboard Tabs

To set a dashboard as the default, right click on the tab of the dashboard and select **Set As Default** from the pop-up menu (*Figure 2.4.2.4-A*). The dashboard is now set as default and will have a green bar located at the top of its tab.

2.4.2.8 Change Layout

Users can change the layout and schema of the dashboard. Open the *Change Layout* dialog box by right clicking the dashboard tab and selecting **Configure** (*Figure 2.4.2.4-A*) from the pop-up menu, or go to **Main Menu** > **Dashboard** > **Change Layout**. The *Change Layout* dialog box opens.

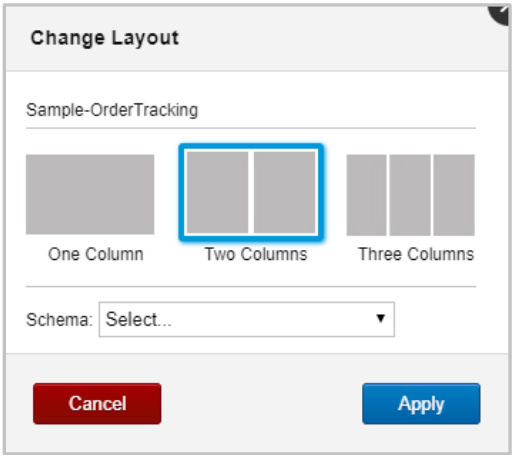


Figure 2.4.2.8-A. Change Layout

A new layout can be selected (one, two, or three columns). Select an option from the **Schema** drop-down menu. The schema will change how viewlets are displayed in your dashboard. The schema selected will control which columns are displayed and their sequence in viewlets (See *3.2.4, Configure Schemas*). Please note that schemas are not available for Global Repository dashboards (the sample repositories).

To go back to the default schema where all columns are displayed in viewlets, choose **Select** from the **Schema** drop-down menu.

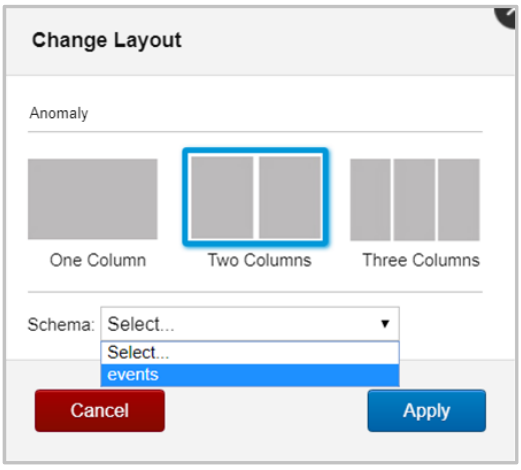


Figure 2.4.2.8-B. Change Schema

Additional dashboard customization options are available on the *Configure Dashboard* dialog screen. See *Section 3.2.4, Configure Schemas*, for more information.

2.4.2.9 Dashboard Arrangement

The order in which dashboards display can be changed. Click on the tab of the dashboard you would like to move and drag and drop it to a new position.

2.4.2.10 Close

To close a dashboard, simply click the **X** located on the right side of the dashboard tab. The **X** will appear if the dashboard is currently displayed. For dashboards not displayed, hover over the tab and the **X** will appear.

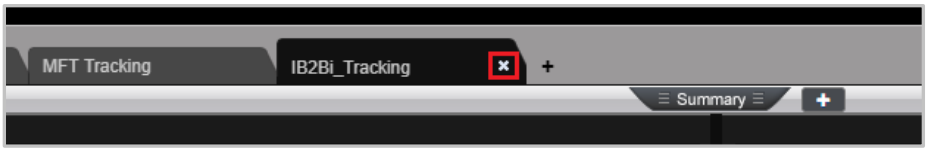


Figure 2.4.2.10-A. Delete Dashboard

Once the **X** is clicked, a confirmation dialog box will appear. If it's a global repository dashboard, the dialog box will confirm the close action (*Figure 2.4.2.10-B*). If the dashboard is in a repository created by you, the dialog box will ask to save your changes (*Figure 2.4.2.10-C*).

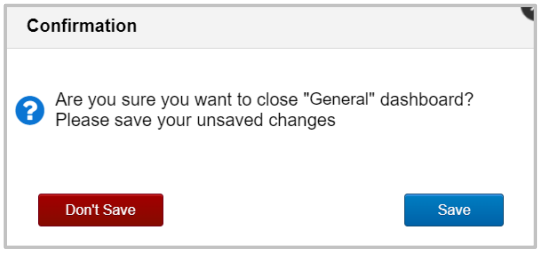


Figure 2.4.2.10-B. Close Dashboard Confirmation

2.4.2.11 Import / Export

Please see *Section 2.6.3, Dashboard,* for more information on importing and exporting dashboards.

2.4.2.12 Rename

Please see *Section 3.2.3.1, Rename*, for more information.

2.4.2.13 Delete

Please see *Section 3.2.3.3, Delete Dashboard*, for more information.

2.4.2.14 Refresh

Please see *Section 3.2.5, Manage Settings*, for information about the Auto Refresh setting on the User Settings (Manage Global Settings dialog) General tab.

2.4.3 Summary Panel

The Summary panel appears at the top of the screen. It contains summary viewlets, which can be created several ways. See *Section 2.5.4.10, Summary*, for more information about summary viewlets.

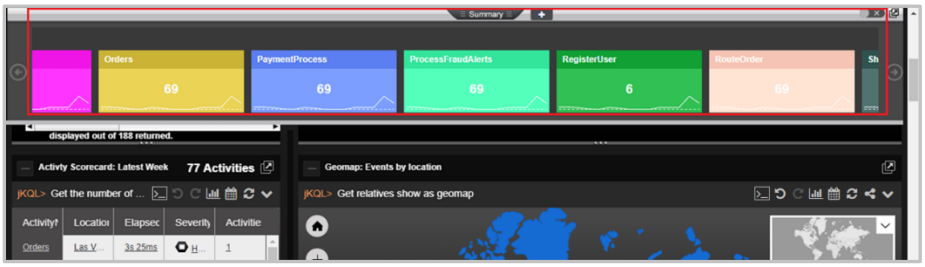


Figure 2.4.3-A. Summary Viewlets

2.4.3.1 Auto-Play Viewlets

Summary viewlets can automatically scroll for easy viewing. To implement this feature, enable **Auto-Play** located at the top right of the **Summary** panel.

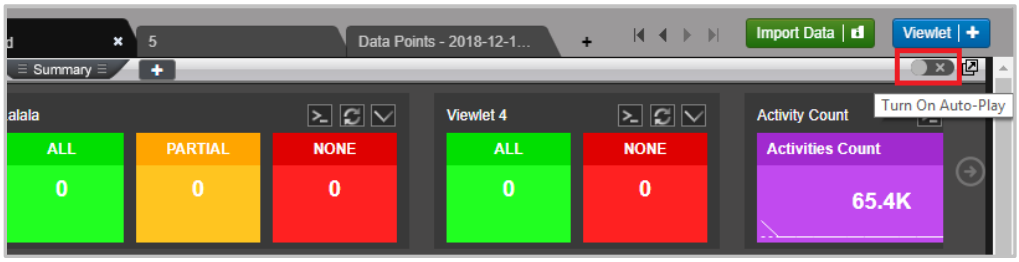


Figure 2.4.3.1-A. Turn On Auto-Play

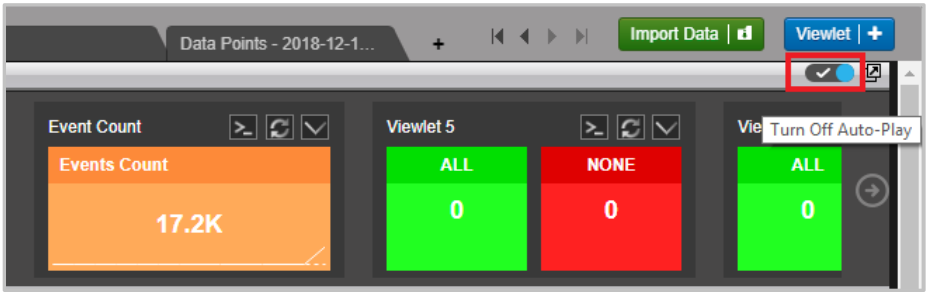


Figure 2.4.3.1-B. Turn Off Auto-Play

2.4.3.2 Scrolling Viewlets

Summary viewlets not displayed on the screen can easily be viewed by clicking on the left and right arrows at each end of the Summary panel.

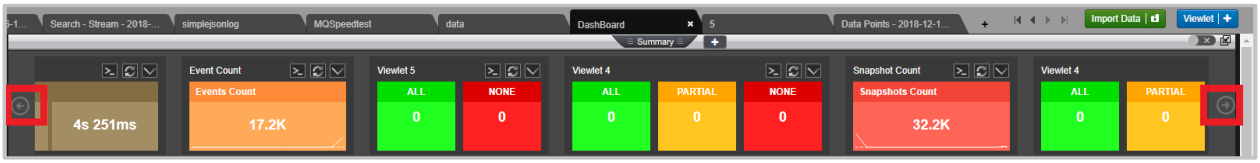


Figure 2.4.3.2-A. Scrolling Summary Viewlets

2.4.3.3 Show / Hide Section

The Summary panel can be hidden by simply clicking the **Summary** tab to collapse the section. The system can be configured to automatically have the Summary panel hidden every time you log in. Please see *Section 3.2.5, Manage Settings*, for more information.

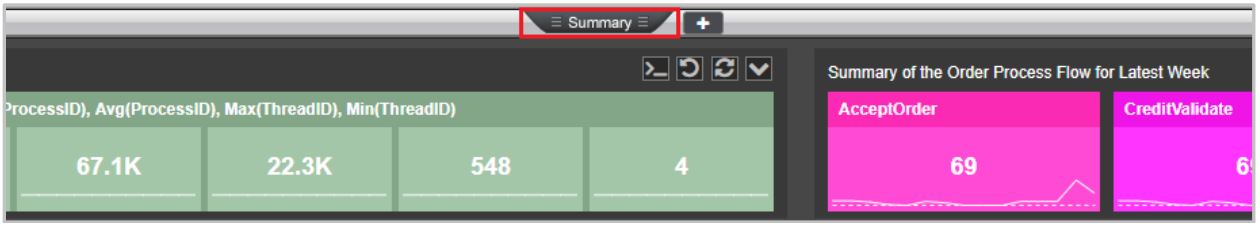


Figure 2.4.3.3-A. Show / Hide Summary Section

2.4.4 Console Panel

When users click on data records from any of the viewlets on a dashboard to view additional details, new viewlets will open in the **Console** panel allowing users to drill deeper into the data.

Within the **Console** panel, new temporary viewlets can be created (see *Section 2.5.1.3, Create Temporary Viewlet*).

Click the **Console** tab 

to display or hide this section. When no viewlets are in the **Console**, the section will collapse automatically. The viewlets in the Console panel are temporary – they will not be saved after switching repositories or logging out.

If any data is clicked within **Console** panel viewlets, the additional details will display in new tabs.

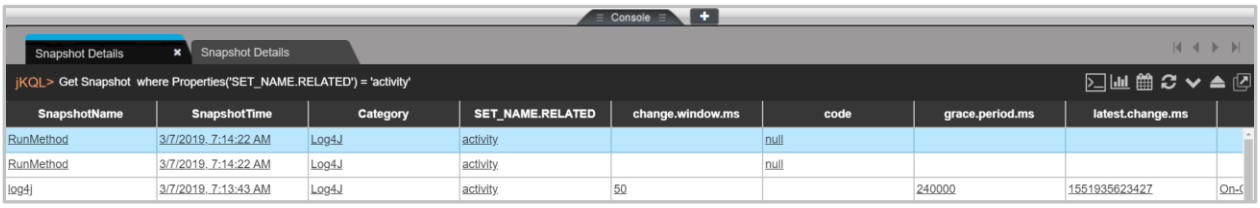


Figure 2.4.4-A. Viewing Console Viewlet Details

Console panel viewlets can be moved to the **Summary** panel so that they can be saved for future sessions. Click the **Move to dashboard** button to perform this.

Figure 2.4.4-B Console Viewlet Toolbar – Move to Dashboard

2.4.5 End User Monitoring

The XRay RUM (Real-User Monitoring) plugin (*https://github.com/Nastel/jkool-rum-plugin*) can be added to get data and monitor web pages using XRay. When the data is streamed through this plugin to XRay, the **EUM** button will appear on the main toolbar. When clicked, it produces a separate dashboard with viewlets that can be used to monitor the activity data of your webpages. Examples of these viewlets can be found in the **Sample-EUM** repository.

Figure 2.4.5-A. End User Monitoring Button

2.5 Viewlets

Viewlets display data in various chart layouts. Examples of possible viewlet layouts along with jKQL queries are provided in the sample dashboards of the Global Repositories. The chart layouts include the following:

- *Table*
- *Column*
- *Bar*
- *Line*
- *Pie*
- *Stack*
- *Geo Map*
- *Scorecard*
- *Area*
- *Summary*
- *Topology*
- *Anomaly*
- *Histogram*
- *Compare*
- *Tree*
- *Clustering*
- *Correlation*
- *Feature Suggestion*
- *Forecast*
- *Expected*

A red asterisk appearing in front of a viewlet name signifies an unsaved viewlet. Save the viewlet from the viewlet's menu (click the down arrow on the top right corner, see *2.5.7 Viewlet Menu* for more information), or save the entire dashboard (see *2.4.2.5 Save*). If your browser crashes before saving, the viewlet will be restored upon next login.

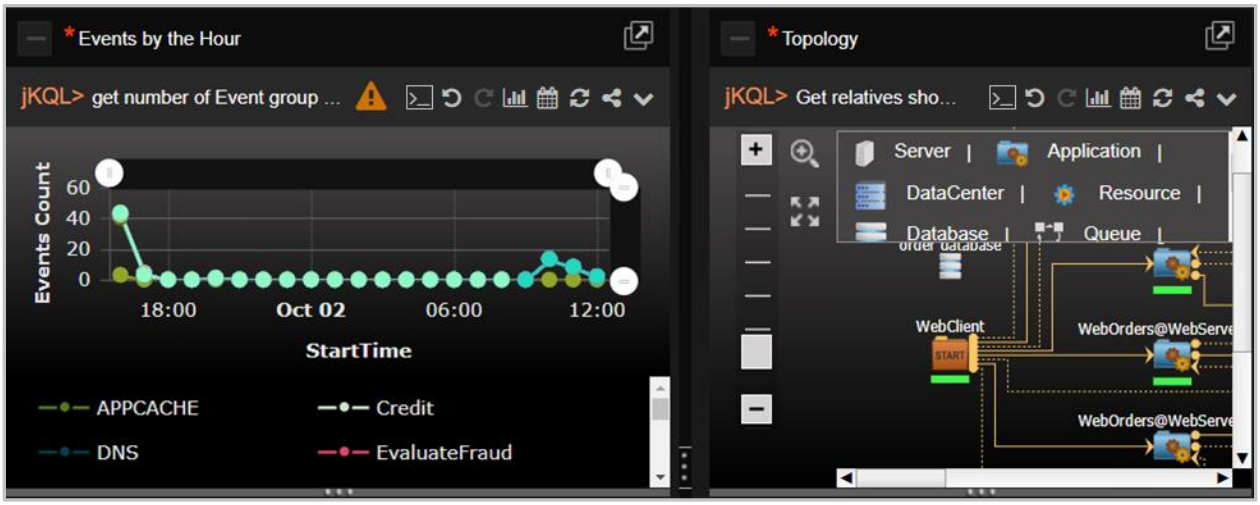


Figure 2.5-A. Viewlets

2.5.1 Create / Open Viewlets

Viewlets can be created by using forms or jKQL queries. To create viewlets, open the *Create/Open Viewlet* dialog box by clicking the **Viewlet** button at the top right of the screen (*Figure 2.5.1-A*) or by going to **Main Menu** > **Viewlet** > **Create** (see *Section 2.3.6, Main Menu*). Users that have the jKQL query interface suppressed (see Users for more information on

suppressing jKQL queries) will have the **Viewlet Form** button instead (*Figure 2.5.1-B*; *see Section 2.5.1.2 Create a Viewlet with a Form*).

NOTE

To import or export viewlets, please see *Section 2.6.2, Viewlets*, for more information.

Figure 2.5.1-A. Create Viewlet with Viewlet Button

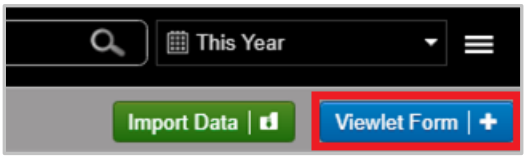


Figure 2.5.1-B. Create Viewlet with Viewlet Form Button

The *Create/Open Viewlet* dialog box opens. See sections 2.5.1.1 (Create Viewlet with a jKQL Query) and 2.5.1.2 (Create a Viewlet with a Form) below on how to add viewlets.

2.5.1.1 Create Viewlet with a jKQL Query

NOTE

Please see *Chapter 5, Using jKQL*, for information on jKQL query language.

1. After clicking the blue **Viewlet** button, select **Create Viewlet with jKQL** on the *Create/Open Viewlet* dialog box.

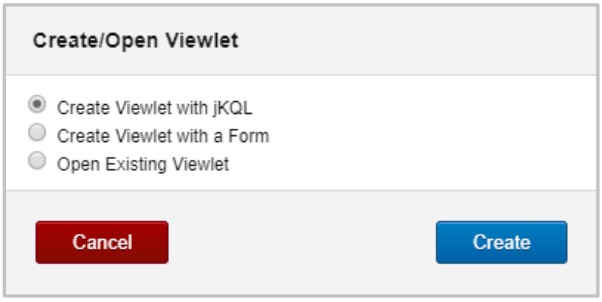


Figure 2.5.1.1-A. Create/Open Viewlet – Create Viewlet with jKQL

2. In this example, the query **Get Log** is entered. As you type, suggestions are provided in a drop-down list.

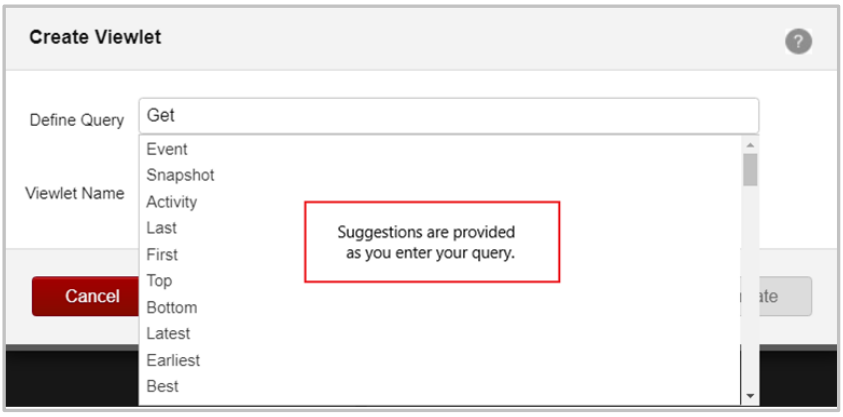


Figure 2.5.1.1-B. Enter a jKQL Query

3. Enter a name for your viewlet. In this example, **My First Query** was entered.

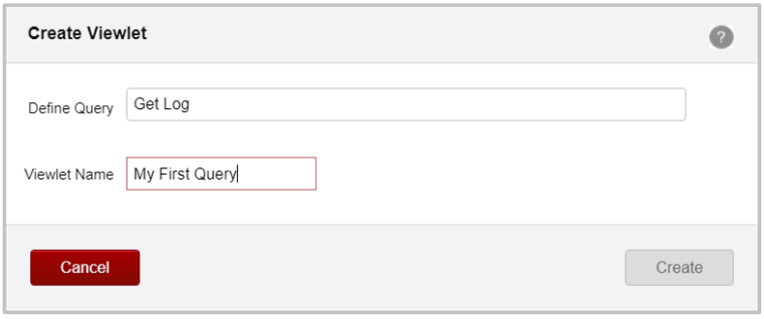


Figure 2.5.1.1-C. Name Your Viewlet

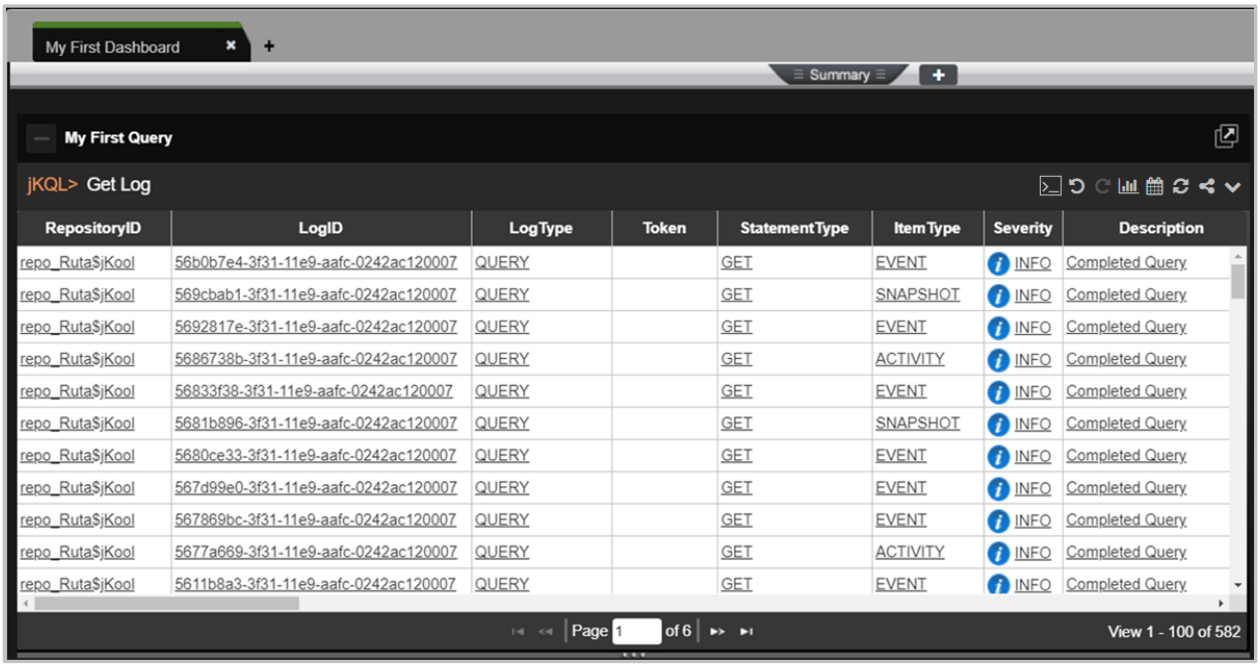


Figure 2.5.1.1-D. My First Query

4. Click **Create**. Your first viewlet is added to the dashboard.

When the GenerateDashboard and GenerateMLDashboard JKQL scripts are found, you can automatically build a dashboard containing viewlets of machine learning queries by following the instructions below:

1. Issue the query 'get active models'.
2. Right-click the results to view the pop-up menu.
3. Choose **Automate Dashboard**.

2.5.1.2 Create a Viewlet with a Form

Select **Create Viewlet with a Form** on the *Create/Open Viewlet* dialog box.

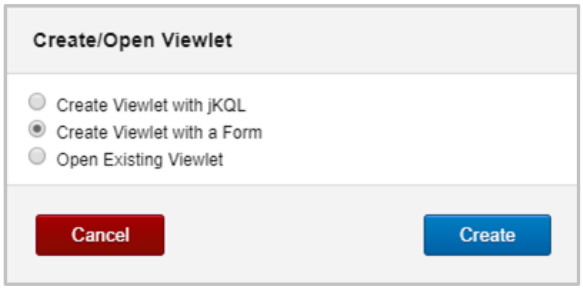


Figure 2.5.1.2-A. Create/Open Viewlet – Create Viewlet with a Form

The form view opens with all available options for the viewlet. Options are explained in the sections immediately below.

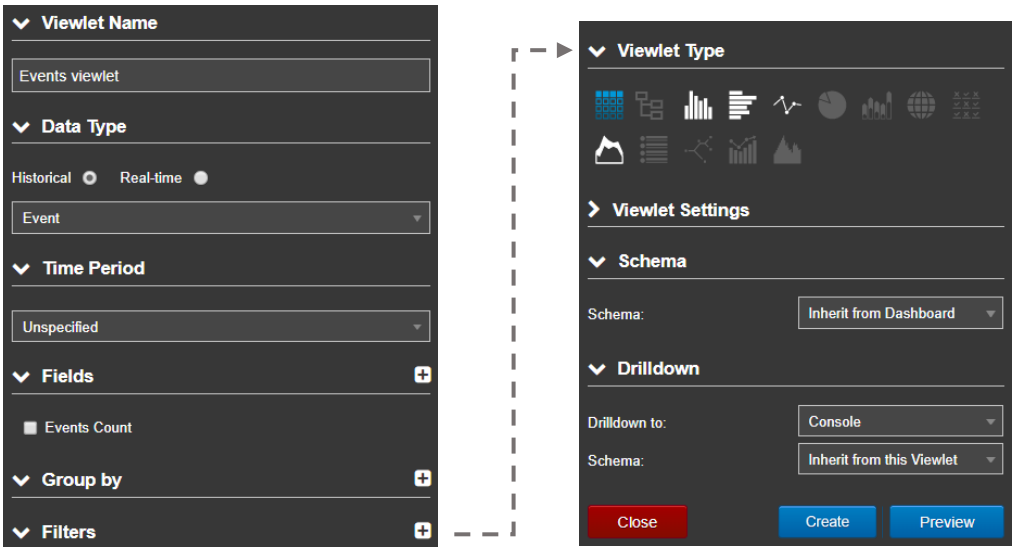


Figure 2.5.1.2-B. Form Options

2.5.1.2.1 Viewlet Name

Specify a name for the viewlet. The name must be unique; if a viewlet name already in use is entered, the field border will appear red and the **Create** button will be deactivated.

2.5.1.2.2 Data Type

Within this section, select if you would like to view **Historical** or **Real-Time** data:

- **Historical**: Select the data type from the drop-down menu. Specify the timespan you would like to view within the **Time Period** section immediately below, select from the following:
    - **Unspecified**: No time filter will be used
    - **Predefined**: Select from the predefined options.
    - **Custom**: Specify a custom time period using a value and a selected time unit.
    - **Date Range**: Enter specific start and end dates.
- **Real-Time**: Select the data type from the drop-down menu. Populate the following fields within the **Real-time Settings** section immediately below:
    - **Frequency:** The time interval in which the viewlet is refreshed with new data.
    - **Window Size:** The amount of the most recent responses from the server to report on. For example, if set to 50, the data is displayed from the latest 50 responses.

2.5.1.2.3 Fields

Within this section, specify the fields to display in the viewlet. Depending on the chart type, the **Count** option is required for certain viewlets (please see *Section 2.5.4, Viewlet Chart Types and Samples*, for more information on viewlet types). Associated required fields will be signified with a red box as seen in the figure below.

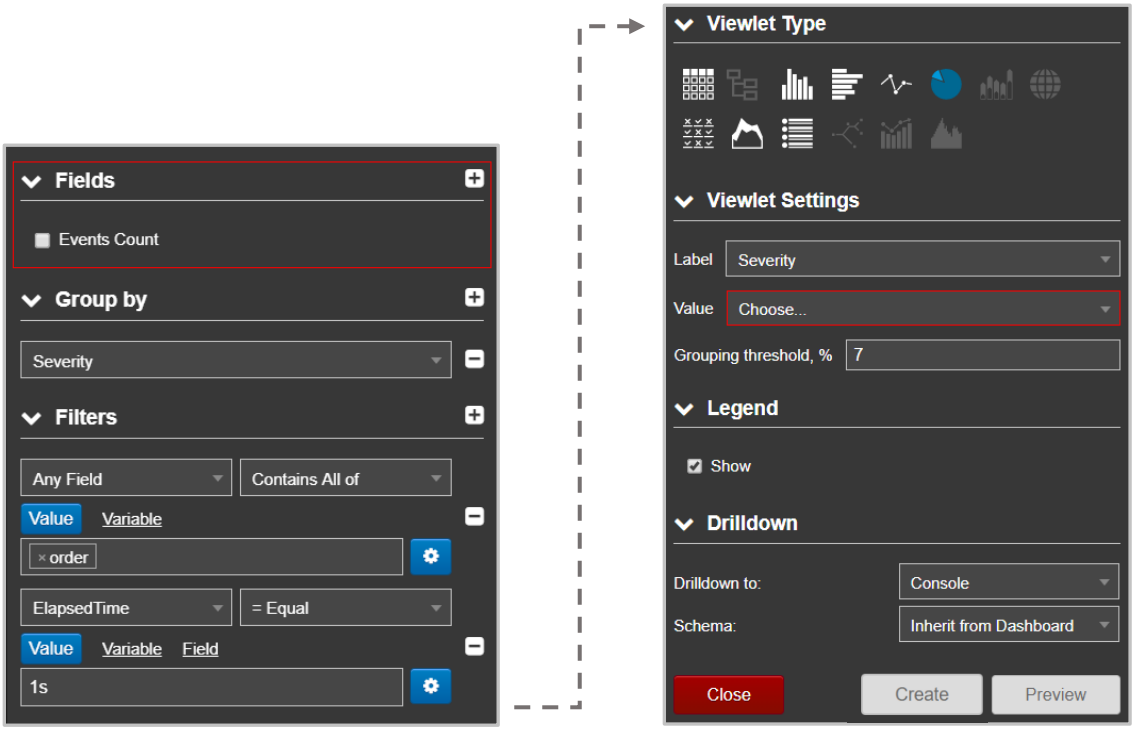


Figure 2.5.1.2-C. Field Requirements

Click the **Add** ⊞ button to add a field. From here you can select multiple fields and their associated functions. These fields and the operation outcome of the selected function will

be displayed in the viewlet. Please see *Section 4.5 Built-in Aggregate Functions* for more information on these functions.

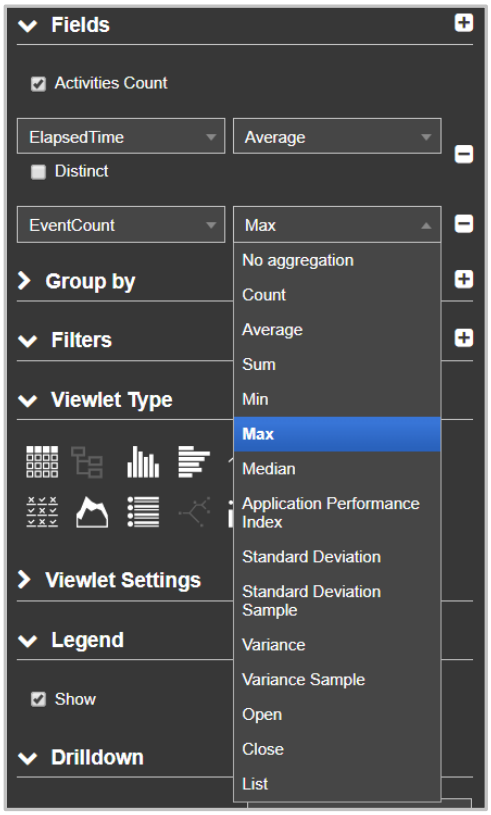


Figure 2.5.1.2-D. Add Fields

In the example below, the fields and their information are displayed in the pop-up.

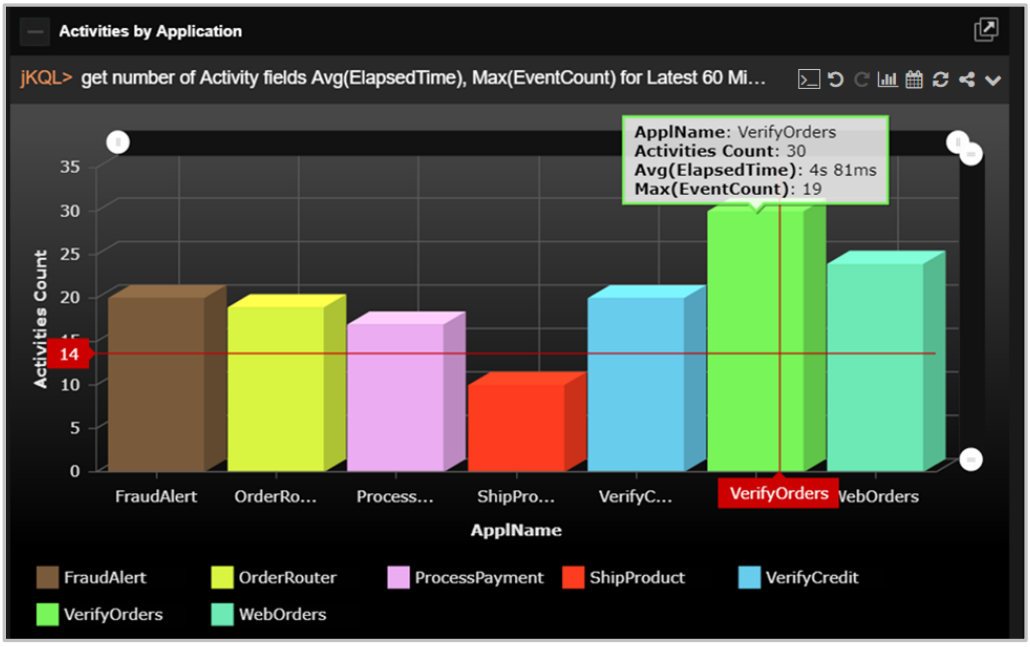


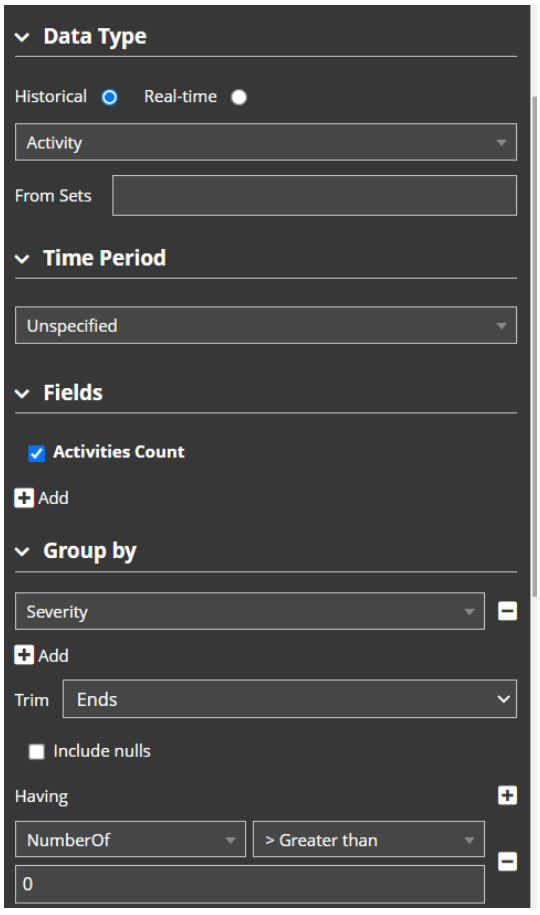
Figure 2.5.1.2-E. Fields Example

2.5.1.2.4 Group By

Within this section, select an option from the drop-down menu to use as the criteria to group data. Items with numerical elements will have a **bucket** option which allows you to specify how data should be grouped. Enable this option and select the type of bucketing. The types of bucketing are described in *Section 2.5.4.13*.

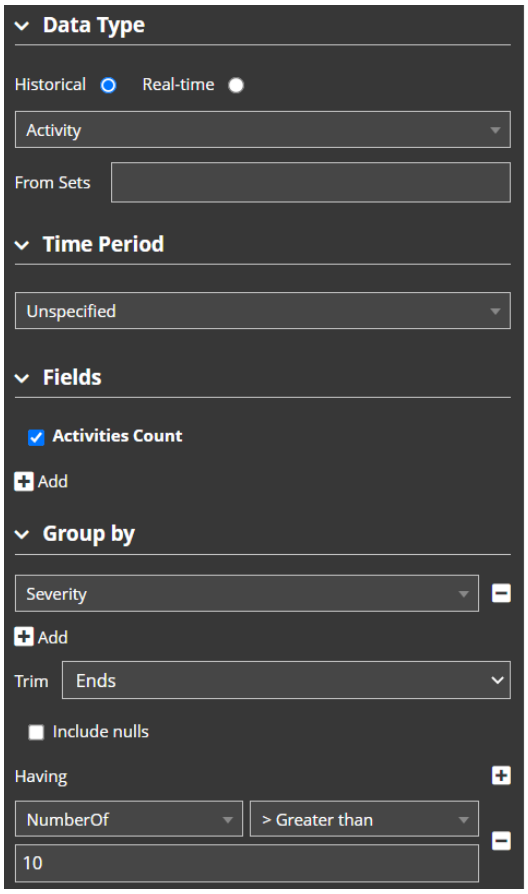
When adding or editing a viewlet's Group by option, a **Having** filter is available to apply a "number of records" filter to the column that results are to be grouped by. The Having filter applies only when a "count" of records is selected for the Data Type (using the ***[Data Type]* Count** checkbox under the Fields section). In other words, for the Activity Data Type, the **Activities Count** checkbox must be selected; for the Log Data Type, the **Logs Count** checkbox must be selected; and the same holds true for all other data types (Events, Sets, Datasets, and so on).

The Having filter helps narrow the "count" for grouped results based on the number of records in those counts. Compare the two examples below. When the Activities records are Grouped by Severity and filtered for a **NumberOf > Greater than 0**, Activities Counts for all Severity values are included:



Severity	Activities Count
HALT	16
FATAL	8
CRITICAL	23
FAILURE	4
ERROR	18

But when the same records are filtered for Having a **NumberOf > Greater than 10**, only Activities Counts for the Severity values with more than ten records are included:



Severity	Activities Count
HALT	16
CRITICAL	23
ERROR	18

2.5.1.2.5 Filters

Use the **Filters** section to add multiple filters. Click the **Add** button ⊞ to add a filter and select an operator.

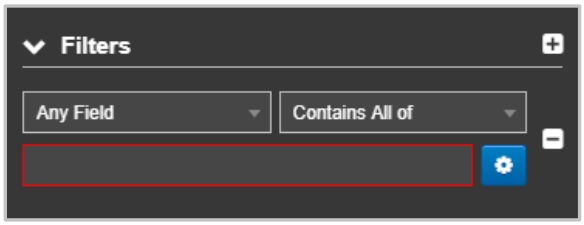


Figure 2.5.1.2-F. Filters Options

After selecting the filter and operator, populate the **Value** field or click the settings button ⚙ to filter by **Variables** or **Fields** (depending on the item type, these filtering options may not be available). Please note that one filter tab can be used at a time. For more information on filtering with variables, see *Section 2.5.8.1, Filtering with Variables*.

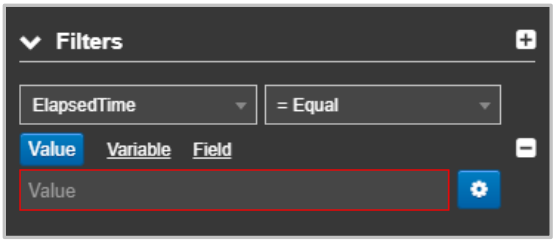


Figure 2.5.1.2-G. Filters Options

When a time-related filter is selected, the appropriate time can be set by clicking on the calendar icon  . The field value automatically populates with the current day/time, but you can change it using the time widget. Click **Done** when finished.

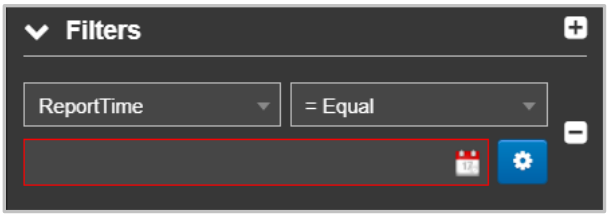


Figure 2.5.1.2-H. Filters Operators

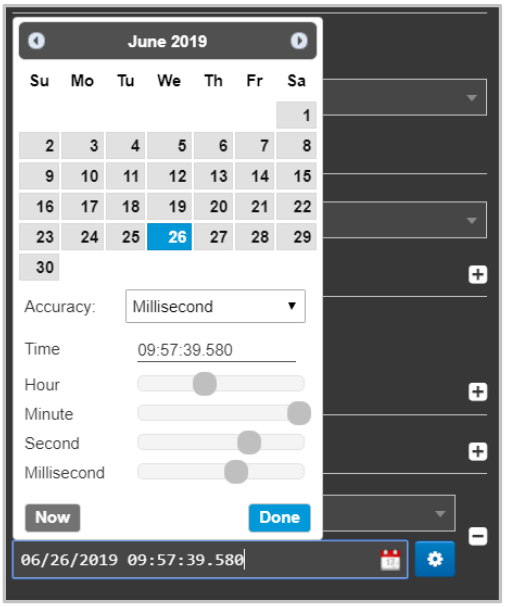


Figure 2.5.1.2-I. Time Setup

Click the **Variable** tab and then the pencil button to create or modify variables used to filter viewlets. After the pencil button is clicked, the **Create new variable** window opens (see *Section 2.5.8.1, Filtering with Variables*, for more information).

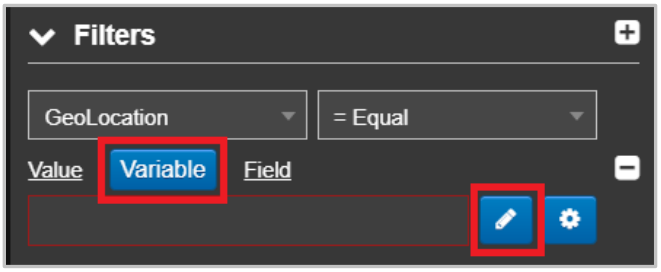


Figure 2.5.1.2-J. Variable Filtering

When you select the **Field** tab, a drop-down menu becomes available. The selected fields will be filtered using operators from this list.

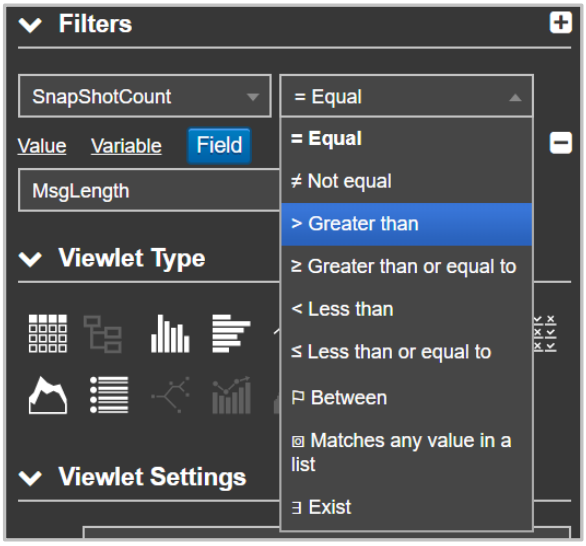


Figure 2.5.1.2-K. Filters Operators

2.5.1.2.6 Viewlet Type

Select the viewlet type within this section. Please see *Section 2.5.4, Viewlet Chart Types and Samples*, for more information on viewlet types.

2.5.1.2.7 Viewlet Settings

Modify viewlet options. Please note that not all viewlets will have this section. Select the X and Y axes values and the **Collated by** option (available for column, bar, line, stack, and area charts).

To set the axis width and label display (axis labels can display vertically or horizontally), click the **Settings** button immediately to the right of the X and Y axis fields. Specify either **Auto** or **Manual** (enter pixel value) for the width and check off the **Rotate labels** setting to rotate the Axis labels.

Enable the **Show** option within the **Legend** section to display the chart definitions.

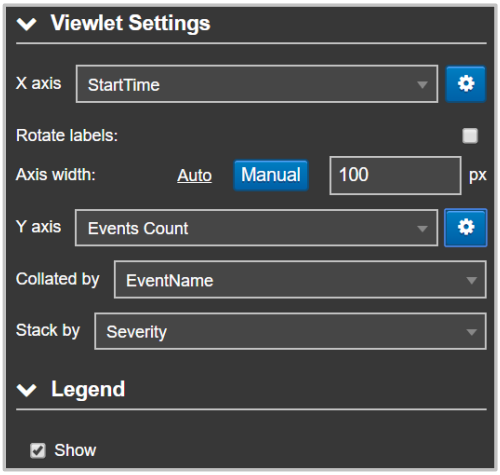


Figure 2.5.1.2-L. Bar chart Form Options

Pie charts will have the following options to specify: **Label**, **Value** and **Grouping threshold, %**.

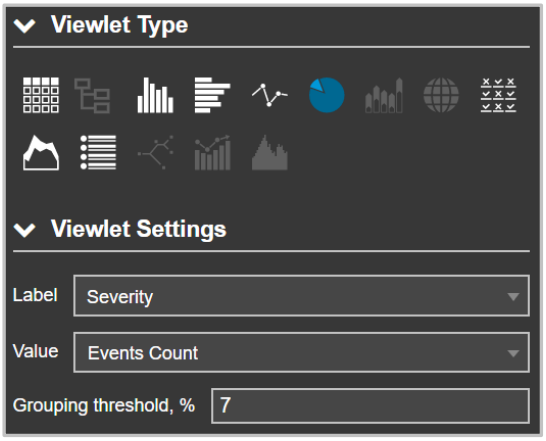


Figure 2.5.1.2-M. Pie chart Form Options

2.5.1.2.8 Drilldown

NOTE

Topology and geomap viewlets do not have the drilldown option.

The drilldown feature is a convenient way to investigate a given problem in more details. You can either drilldown to a viewlet's details within the *Console* panel, which is the default method, or to a target dashboard (including the current dashboard) that utilizes variables. The variable filter from the data selected in the originating viewlet is passed to the specified dashboard and updates that dashboard's viewlets using the variable.

The following is an example scenario of drilling down to a target dashboard:

- Dashboard #1: Contains a viewlet with data for average temperature by state. You have configured this viewlet to drilldown to Dashboard #2 using its state variable.
- Dashboard #2: Contains the following three viewlets which have a filter defined for state:
    - Temperature by city
    - Humidity by city
    - Rainfall by city

  When you select a specific state within the viewlet located in Dashboard #1, you will be brought to Dashboard #2. All three viewlets in Dashboard #2 will reflect data for the state selected from Dashboard #1's viewlet.

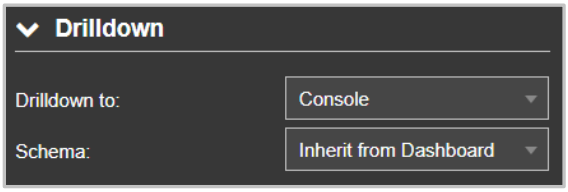


Figure 2.5.1.2-N. Drilldown Options

TIP

Watch the following video for an overview of the drilldown featured:
https://vimeo.com/383411780

The **Drilldown** section controls viewlet filtering within and across dashboards. From the **Drilldown to** dropdown setting, select one of the following options to specify how the drilldown will behave:

- **Console**: this option is enabled by default. Leave this option set if you want to open drilldown results within the *Console* panel. Within the **Schema** field, specify which schema to apply to the viewlet (the schema determines which fields are displayed within the viewlet).

  -OR-

  Dashboard: select this option to drilldown to a specific dashboard. This option should only be used with dashboards that utilize variables (see *Section 2.5.8.1, Filtering with Viewlets*, for more information) as data will be passed to them upon drilldown.
  Select the desired dashboard from the **Dashboard Name** field. The selected dashboard's associated variables will display. Select the parameter you want to pass into the variable.
  From this point forward, when you click on the configured data in the viewlet, instead of going to the *Console* panel, you will be brought to the selected dashboard instead. Please note, if the destination dashboard is closed, it will automatically open. All the viewlets will be updated to show you data for the specified variable from the originating dashboard.

  If **Self** was selected for the dashboard, when drilling down, instead of opening a new dashboard, the current dashboard will be refreshed. This is a handy way of creating a dashboard that you want to quickly refresh its viewlets with variable data. There are lots of other use cases, for example, you can have your top 10 problematic queues in MQ located at the top of the dashboard which will be dynamically updated.

2.5.1.3 Create Viewlet Using Image Template

In XRay, you can create image viewlets to display SVG images that can be designed to visually represent various aspects of your data. For example, you can use image templates to create annotated business flows. However, in this guide, we will use a simple example of a battery image to demonstrate the capabilities of this type of viewlet.

When you save an image template, you are assigning the name used to store the SVG file. Once the SVG file is saved, you can write a data query that uses the saved image template. A jKQL query that uses an image template ends with "show as image('imageTemplateName')".

To create an image viewlet:

1. Click the Viewlet button in the upper-right corner of the XRay window.
2. Choose the Create Viewlet Using Image Template option and click Create.

3. Name the viewlet.
4. Name the Image template. As stated above, the Image template name is used to store the SVG file.
5. Click **Save**.

The image viewlet editor opens. It is made up of four panels:

- jKQL Query Editor (upper left)
- Selected Element Rules Editor (upper right)
- SVG File Editor (lower left)
- Image Preview (lower right)

2.5.1.3.1 jKQL Query Editor

Use the jKQL query editor in the upper-left corner to enter a query to obtain data.

Example:

get Dataset fields all where DatasetName = 'battery' show as table

2.5.1.3.2 Selected Element Rules Editor

In the Selected Element Rules Editor panel (upper right), you can edit the rules for the selected object (as long as other elements do not use the same rule). See Editing Rules for a Selected Element.

For more information about viewing and editing all rules in an SVG file, see Rules Tab.


NOTE

Every text rule that can be applied to an element can have a "default value." Default values are used to quickly add basic functionality to an element. The default value is applied in two situations: when there is no rule in use, and whenever rule conditions have not been met. Expand the rule to see the Default Value field. Although the Default Value acts like a rule event, it doesn't need a condition to be met, or rule the to be applied. However, a Binding Rule must be applied to bind the group or element to a dataset for it to work, because the data allows the condition to be evaluated.

2.5.1.3.3 SVG File Editor Actions

In the SVG File Editor (lower left) you can edit all elements of the SVG file for the image viewlet. See SVG File Editor Tabs for more information about the six tabs in this editor.

2.5.1.3.3.1 Create objects and groups

As explained later in this document, the All HTML tab contains all HTML code associated with the SVG image; the Main tab contains the code for elements that are rendered. Therefore you can create objects in either tab.


TIP

To add more elements, you can copy existing elements in the SVG File Editor, paste the elements onto a new line, and update their coordinates.

Create objects

To create an object, enter the HTML code for the object in the All HTML or Main tab.

For example, the code for a rectangle would look like this:

<rect x="0" y="0" width="650" height="320" rx="64" ry="64" fill="transparent" />

However, most objects do not stand on their own, but are a single part of a more complex representation. Even one part of a diagram may contain several individual shapes. For this reason, it is recommended that objects be arranged in groups. Grouping elements makes it easier to apply the binding rules that will control how the objects appear, because you can apply one rule to a collection of objects. (Binding rules are covered later in this document.)

For example, the following three shapes form the exterior and interior elements of the battery image that is being used for this example:

Exterior shape (transparent rectangle with rounded corners)

<rect x="0" y="0" width="650" height="320" rx="64" ry="64" fill="transparent" />

Interior shape (offset from, and inside, the exterior shape; a rounded rectangle with no border)

```
<rect x="20" y="20" width="610" height="280" rx="56"
      ry="56" stroke="none" />
```

Vertical line representing the cathode end of the battery

```
<path d="M670,110 h10 q5, 0 5,5 v90 q0,5 -5,5 h-10 z">
```

Create groups

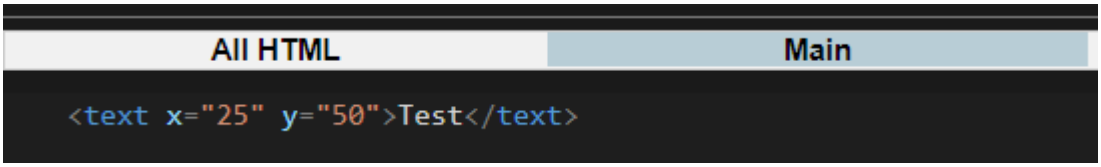
To group these elements, wrap them in the SVG element <g> tag.

```
<g data-is-main='true' >
  <rect x="0" y="0" width="650" height="320" rx="64" ry="64" fill="transparent" />
  <rect x="20" y="20" width="610" height="280" rx="56" ry="56"
    stroke="none" />
  <path d="M670,110 h10 q5, 0 5,5 v90 q0,5 -5,5 h-10 z">
</g>
```

2.5.1.3.3.2 Create text

This example shows how you would display the word "Test" as part of an SVG image:

```
<text x="25" y="50">Test</text>
```



2.5.1.3.4 SVG File Editor Tabs

2.5.1.3.4.1 All HTML Tab

The first tab (All HMTL) includes all the required HTML code associated with the SVG image. The other five tabs represent sections of the code. The tabs make navigation easier by allowing you to skip to different sections within it.

2.5.1.3.4.2 Main Tab

The Main tab allows you to focus only on what is rendered (for example, elements such as shapes or text). For example, the code below renders a depleted battery image:

```
<g data-binding-rules="bind-battery">
  <rect x="0" y="0" width="650" height="320" rx="64" ry="64" fill="transparent" />
  <rect data-class-rules="fully-charged partly-charged depleted" x="20" y="20" width="610"
height="280" rx="56" ry="56"
    stroke="none" />
  <path d="M670,110 h10 q5, 0 5,5 v90 q0,5 -5,5 h-10 z">
</g>
```

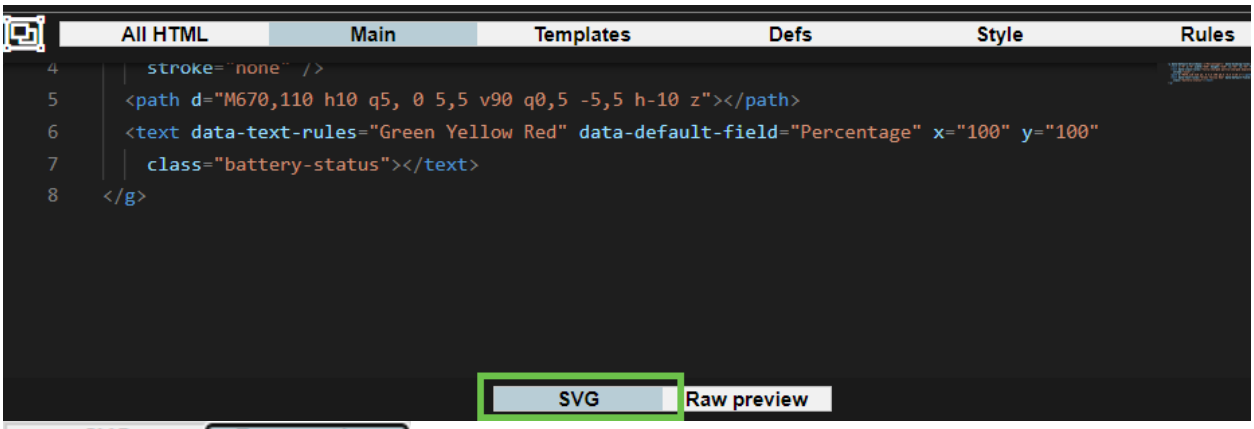
When you click a portion of the battery image in the Image Preview panel, the Main tab is shown.

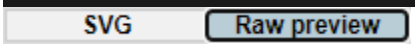
> IMPORTANT!
>
> Preview elements are modified by the browser when they are rendered. Therefore, XRay may not always be able to find the element that corresponds to the portion of the image you clicked on in the HTML text editor. If XRay cannot find the element by its id, by the closest binding rule, or by its code, you may have to find the element's code manually.

Both the Main and Templates tabs include two buttons at the bottom of the panel so you can choose how to view the content:

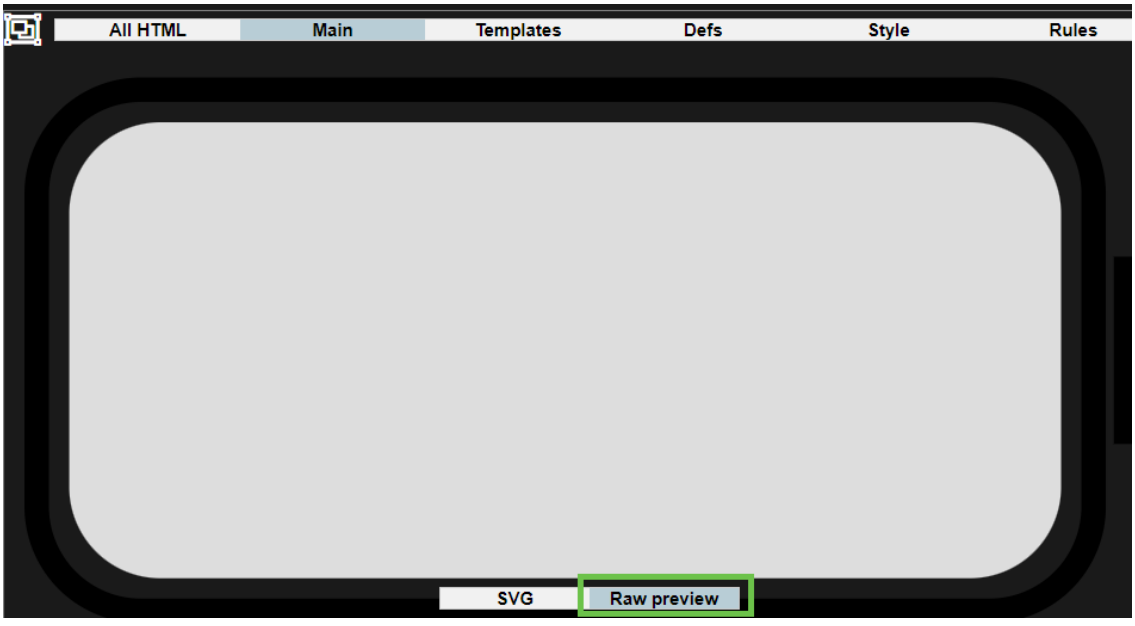
- 

Click **SVG** to view the code for the selected element.



- 

Click **Raw preview** to view the element as it would look without rules applied.



Select with Cursor Toggle

The Select with Cursor toggle icon on the Main tab allows you to select elements from the SVG File Editor panel.

	OFF. By default, the Select with Cursor toggle is off. To turn it on, click it once.
	ON. When the Select with Cursor toggle is on, you can use your cursor to select elements from the code in the Main tab. You can either click and drag your cursor across the code or, for elements of a single line of code, double-click the line of code to select the element. Once the element is selected, the Selected Element Rules Editor panel becomes available so you can add rules to the element.

2.5.1.3.4.3 Templates Tab

The Templates tab allows you to enforce consistency among elements by creating elements that can be reused or by applying the attributes of existing elements to new elements.

More information about the Templates tab is forthcoming.

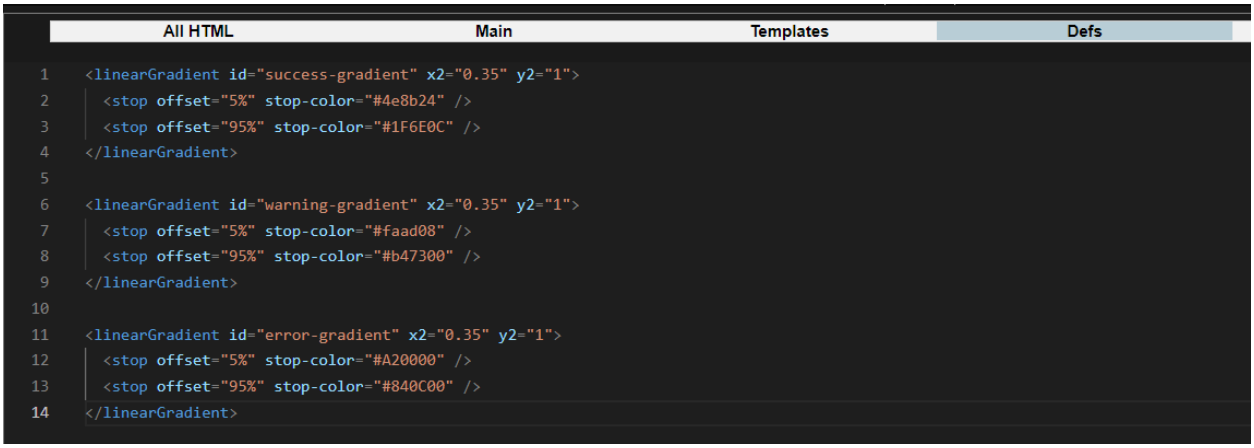
2.5.1.3.4.4 Defs Tab

The Defs tab is for the <defs> element, which can be used to store definitions or objects that will be rendered elsewhere in the code. For example, it can be used to store shapes, images, or color gradients. In the example below, the Defs tab stores the values associated with the green, yellow, and red color gradients for the battery image. These definitions are used in the fill styles that are applied to the rectangle portion of the battery image according to Class rules.

```
<linearGradient id="success-gradient" x2="0.35" y2="1">
  <stop offset="5%" stop-color="#4e8b24" />
  <stop offset="95%" stop-color="#1F6E0C" />
</linearGradient>

<linearGradient id="warning-gradient" x2="0.35" y2="1">
  <stop offset="5%" stop-color="#faad08" />
  <stop offset="95%" stop-color="#b47300" />
</linearGradient>

<linearGradient id="error-gradient" x2="0.35" y2="1">
  <stop offset="5%" stop-color="#A20000" />
  <stop offset="95%" stop-color="#840C00" />
</linearGradient>
```

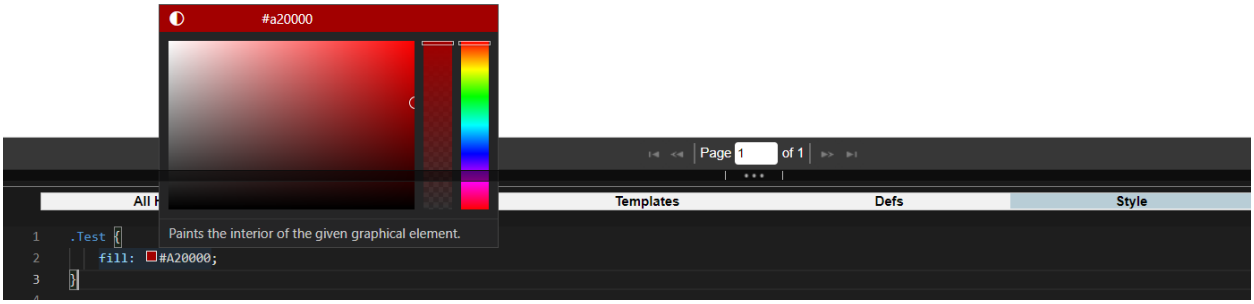


2.5.1.3.4.5 Style Tab

In the Style tab, create CSS classes to control how elements are formatted. Then you can set up Class Rules to apply these styles to elements. As a result, the formatting of elements is standardized, and it is easier to make changes to multiple elements at once. For example, a style for text might include size, font, and color. A style for a rectangle might include size, border width, and shading.

NOTE

When you create a style with a color attribute, a color palette is available to allow you to select the color, as shown in the image below.



2.5.1.3.4.6 Rules Tab

Rules are displayed on the Rules tab of the SVG File Editor. On this tab, you can view and edit all existing rules in the SVG image file, regardless of the number of elements that use each rule.

Rules for a specific element are also available in the Selected Element Rules Editor panel. However, if more than one element uses a rule, you cannot use the Selected Element Rules Editor. You must instead use the Rules tab of the SVG File Editor.

NOTE

Some rules that can be applied to an element can have a "default value." Default values are used to quickly add basic functionality to an element. The default value is applied in two situations: when there is no rule in use, and whenever rule conditions have not been met. Expand the rule to see the Default Value field. Although the Default Value acts like a rule event, it doesn't need a condition to be met, or rule the to be applied. However, a Binding Rule must be applied to bind the group or element to a dataset for it to work, because the data allows the condition to be evaluated.

Rules tab Procedures

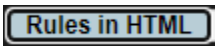
On the Rules tab, you can add, edit, and delete rules that can be applied to any element.

Add a rule

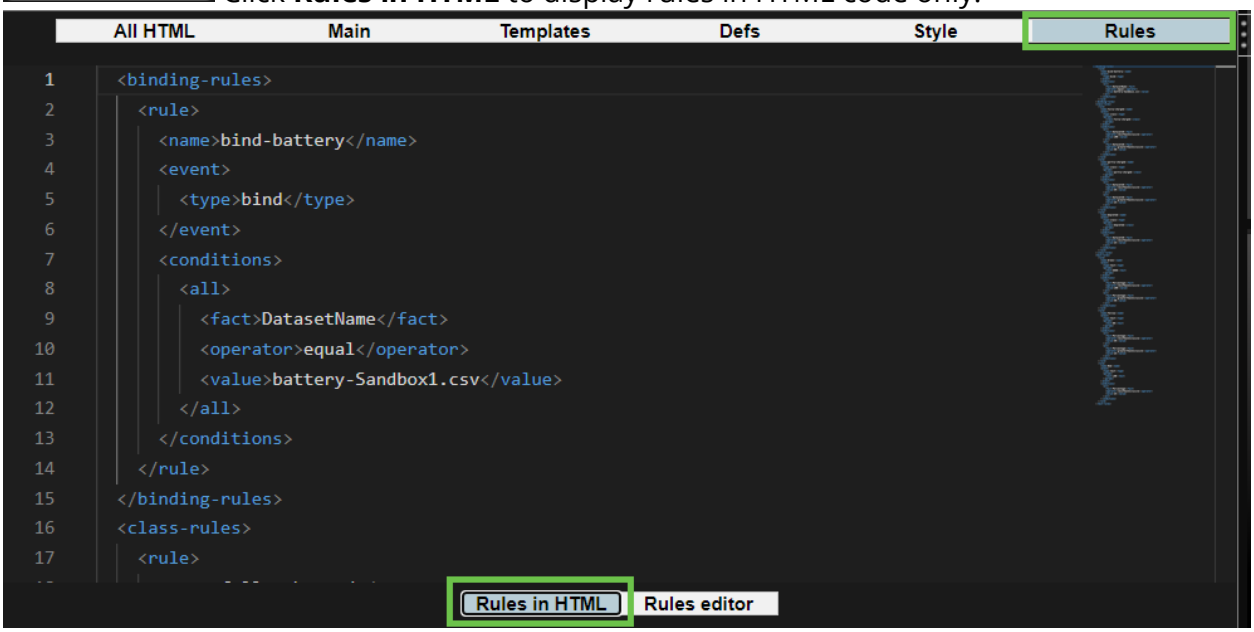
1. Click  .
2. Add a name for the rule in the **Rule name** field.
3. Set Rule conditions. In the example below, the condition is the DatasetName (DatasetName = "battery"). When the condition is met, the rule will be applied.
4. (Optional.) To preview your changes, click **Preview**.
5. Continue to make changes until you have completed edits to the image template.
6. To save changes and close the editor, click **Save** in the lower-right corner of the window.

View a rule

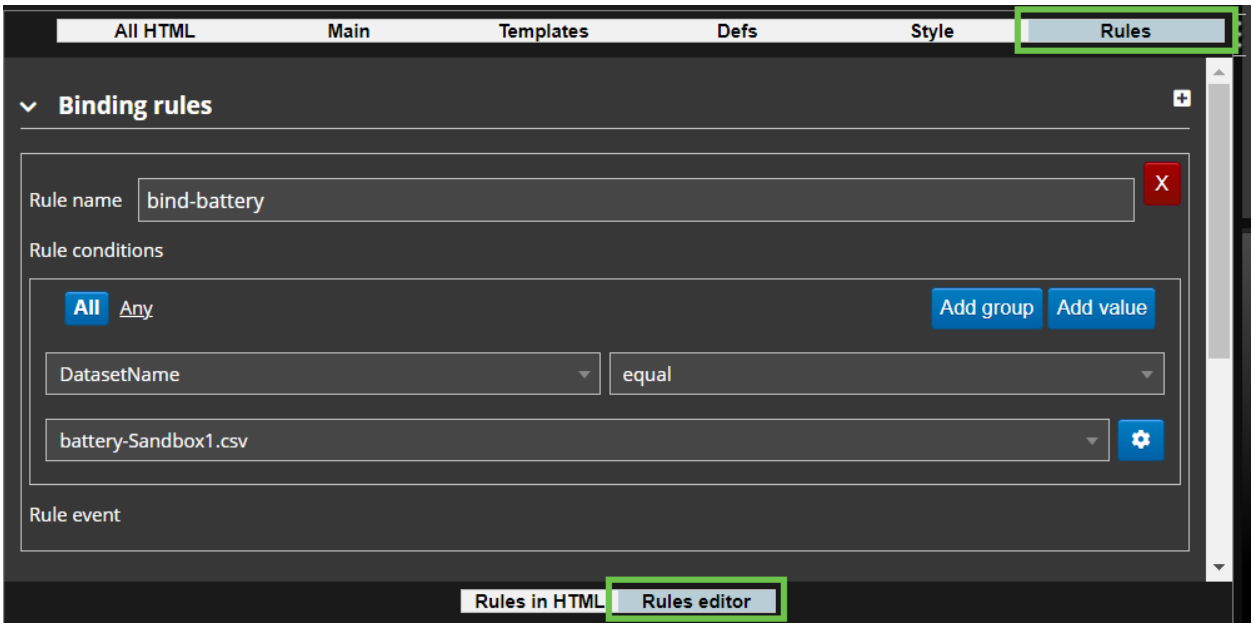
After a rule has been created, you can use the buttons at the bottom of the Rules tab to determine how you want it to be displayed:

- 

Click **Rules in HTML** to display rules in HTML code only:



- Rules editor Click **Rules editor** to display rules in the user-friendly graphical user interface format:



Edit a rule

1. On the Rules tab, Expand the type for the rule you want to edit (for example, Binding, Drilldown, or Class).
2. Make your changes to the Rule name or Rule conditions.
3. (Optional.) To preview your changes, click **Preview**.
4. To save changes, click **Save** in the lower-right corner of the window.

Delete a rule

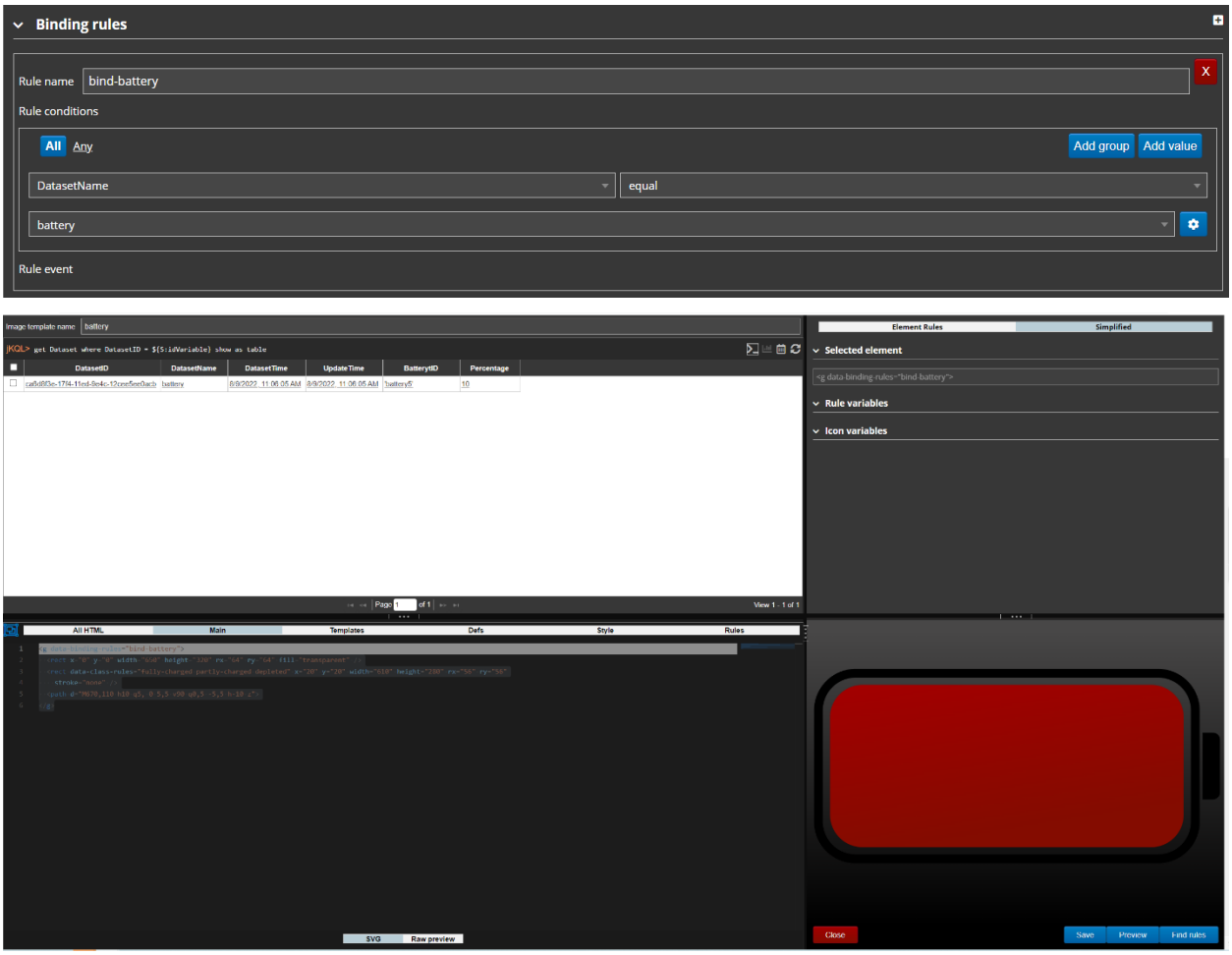
1. On the Rules tab, Expand the type for the rule you want to delete (for example, Binding, Drilldown, or Class).
2. Click X.

To learn how to view and edit rules for a specific element, see Editing Rules for a Selected Element.

Binding Rules

Binding rules bind SVG elements or groups of elements to sets of data. When elements are grouped, the binding rule for the <g> tag is applied to all grouped elements.

The Binding Rule for the battery example shows that the battery image is associated with, or bound to, the dataset called "battery."



Drilldown Rules

Drilldown rules control when happens when users interact with certain portions of an image viewlet. More information about Drilldown rules is forthcoming.

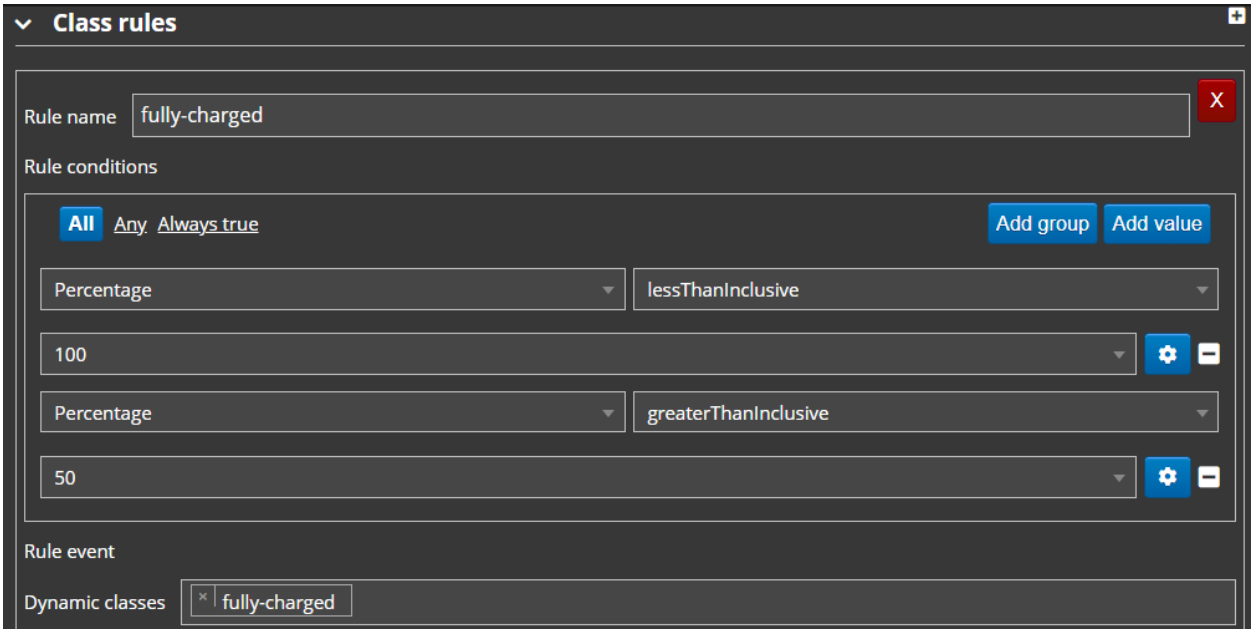
Icon Rules

Icon rules allow icons, such as icons from the data, to be displayed in image viewlets. More information about Icon rules is forthcoming.

Class Rules

You can use Class rules to apply the CSS class styles that you set up on the Style tab. For example, you can apply a Class Rule to change the color of an element based on certain criteria. In this the battery example, style classes determine the formatting of the battery image. The battery image class rules cause it to change color based on the percentage of battery power that remains.

The "fully-charged" class rule formats the battery image using the "fully-charged" style when the percentage is at least 50 percent:

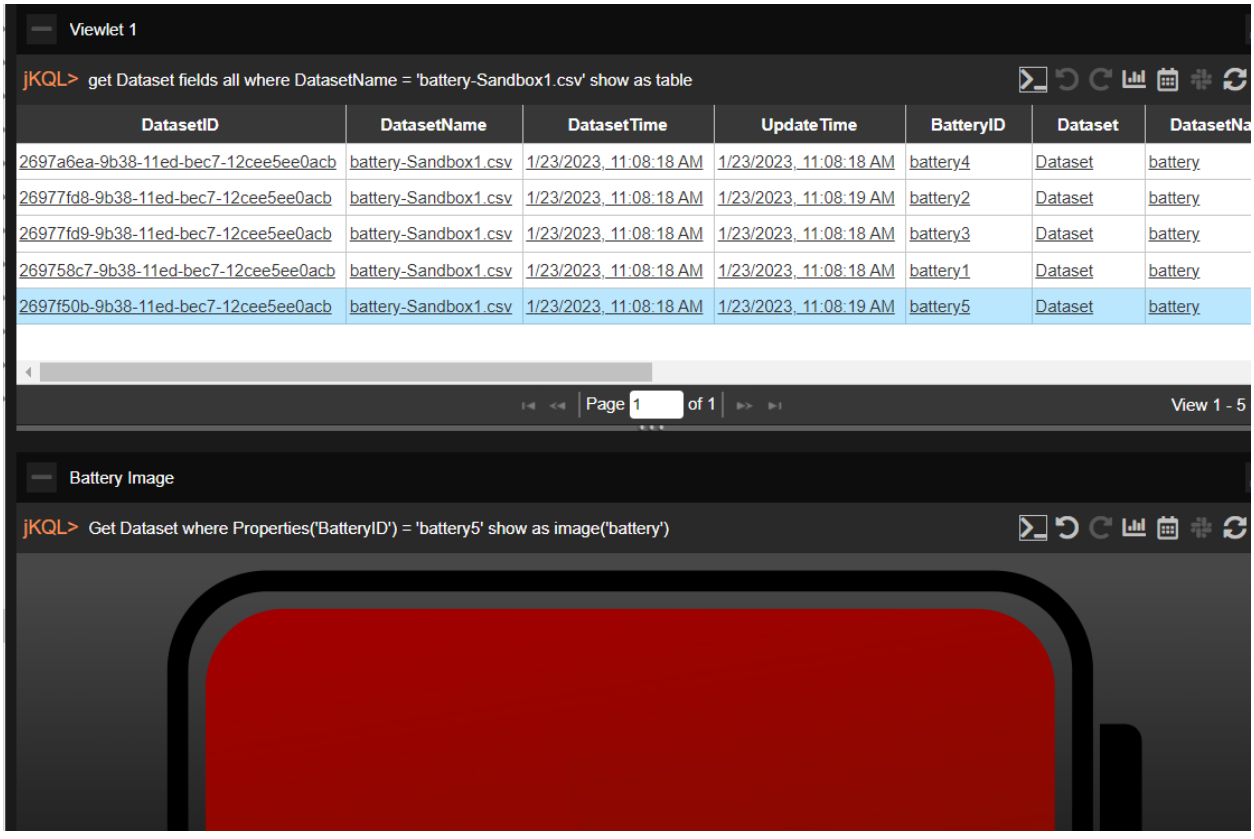


The "partly-charged" class rule formats the battery image using the "partly-charged" style when the percentage is between 25 and 49, inclusive:

The "depleted" class rule formats the battery image using the "depleted" style when the percentage is at 24 percent or less:

For example, the following query shows a depleted battery image:

```
Get Dataset where Properties('BatteryID') = 'battery5' show as
image('battery')
```



Text Rules

When setting up a text element, such as a label, in the battery example, you have three options: Field value, Custom text, or a combination of the two.

- If you select **Field value**, the value from the selected field is displayed. In the first example below, the Percentage field value is used as a label to indicate the battery charge level.

- If you select **Custom text**, you can define any text to be shown. This can be static text, or dynamic, using rules. In the second example below, the descriptive label varies depending on the battery status ("GOOD," "OK," or "LOW").
- You can also combine static and dynamic text in one text element.

In this section we will look at using these different options to add different labels to the battery to reflect its current charge level.

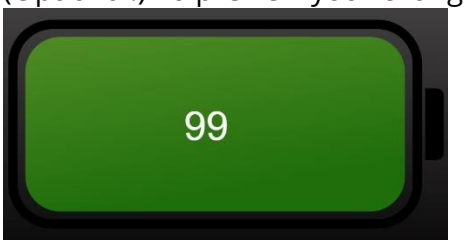
Example 1: Field Value

In this example, we will apply a text rule so that the text label will reflect the actual percentage of the battery. Since this will vary by battery, this rule uses a Field Value, pulling the value for the label from the actual data.

To add a rule for Field Value as Text:

1. In the SVG File Editor (lower-left panel), turn on the Select with Cursor toggle icon (see Select with Cursor Toggle).
2. Select the text element in the Main tab of the SVG editor to open the Selected Element Rules Editor (upper-right panel). You can either click and drag your cursor across the code or, for elements of a single line of code, double-click the line of code to select the element.
3. Select the Element Rules button at the top of the panel. You can now modify the Text rules of the text element by expanding the Text rules (as long as no other text elements use the same rule).
4. A list of the fields in the dataset is provided. Select the field from the list that you want to use in your text. In this case, the Percentage field is selected:

5. (Optional.) To preview your changes, click **Preview**.



6. Continue to make changes until you have completed edits to the image template.
7. To save changes and close the editor, click **Save** in the lower-right corner of the window.

Example 2: Custom Text

In this example, we will apply a text rule so that the text label will change according to the same conditions that affect the color of the battery. When the battery percentage is in the range indicated by the green background, the label will read "GOOD." When it is in the

range indicated by the yellow background, the label will read "OK." When it is in the range indicated by the red background, the label will read "LOW."

What is a Group?

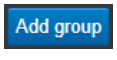
A Group is one logical condition. It can contain one or more value statements. Groups can also contain other groups.

What is a Value?

A value is a data point that is being evaluated. By default, each data point that is being evaluated is a simple value based on the data type of the selected field (that is, for integer fields, the data point is a number; for string fields, it is text). But you can click the Show Hide Advanced Options icon  to make the value a Variable or a Field instead.

Unless the value you are entering is a custom field, you can select from a list of the values that exist for that field within the dataset.

To add a rule for Custom Text:

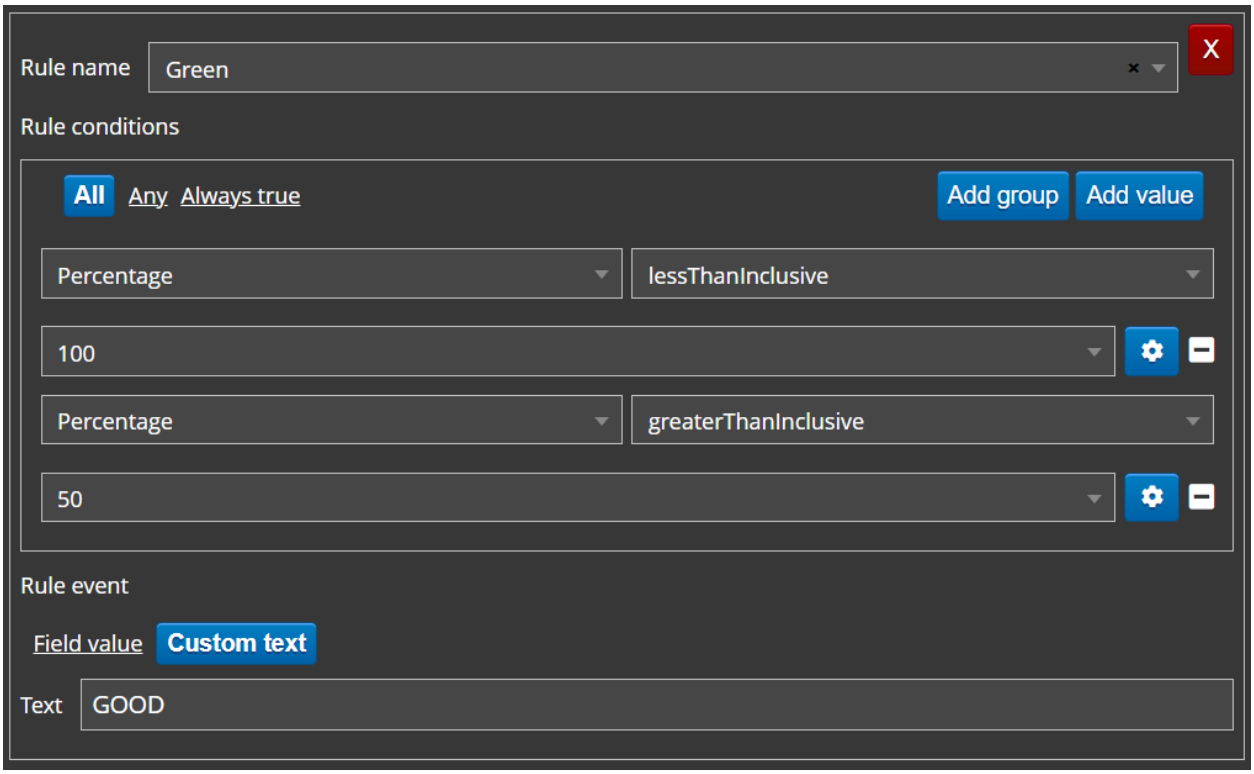
1. In the SVG File Editor (lower-left panel), turn on the Select with Cursor toggle icon (see Select with Cursor Toggle).
2. Select the text element in the Main tab of the SVG editor to open the Selected Element Rules Editor (upper-right panel). You can either click and drag your cursor across the code or, for elements of a single line of code, double-click the line of code to select the element.
3. Select the Element Rules button at the top of the panel. You can now modify the Text rules of the text element by expanding the Text rules. If a Text rule is used by multiple text elements, you must edit it in the Rules tab of the SVG File Editor.
4. Click  .
5. Add a name for the rule in the **Rule name** field.
6. Set Rule conditions. By default, you start with a single Group. You can add value statements to a group by selecting  . The group of one or more value statements as a whole is evaluated according to the "operator" you choose for the group (*All*, *Any*, or *Always true* [*Always apply*]), as follows:
    - If you choose *All*, then the data must "pass" all the value statements in that group for the condition to be met.
    - If you choose *Any*, then the data must "pass" at least one of the value statements in that group for the condition to be met.
    - If you choose *Always true* (or *"Always apply*"), then the rule event is always applied. This option is useful when, for example, you want to use the same custom text for multiple elements. You can apply an *Always true* (or "*Always apply*") rule to each element so that you can set up the custom text in one place for all of them.
7. (Optional.) Click  to add another condition (in the form of another Group, or set, of value statements).

8. (Optional.) To preview your changes, click **Preview**.

9. Continue to make changes until you have completed edits to the image template.
10. To save changes and close the editor, click **Save** in the lower-right corner of the window.

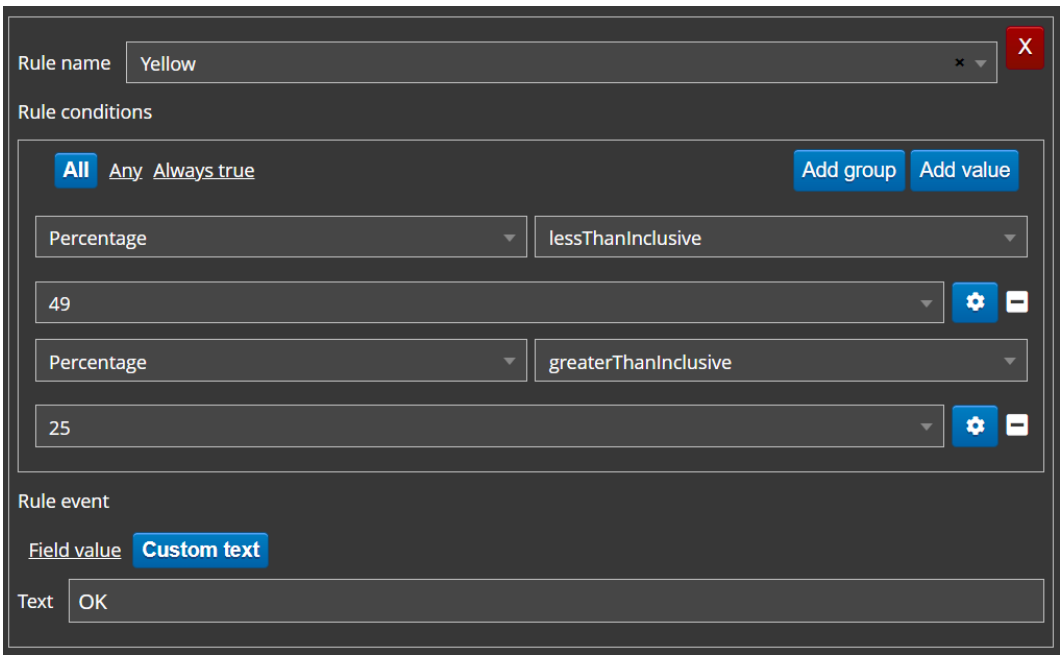
Text Rule Scenarios

In the scenario below, the condition group that represents a battery percentage between 50 and 100, inclusive, is made up of two separate value statements, both ("All") of which must be met: a percentage ≤100 and a percentage ≥50.

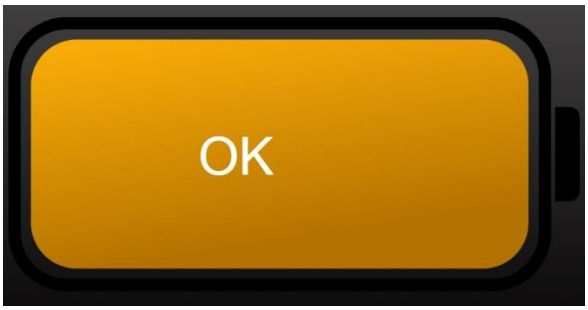


When the condition represented by this group is met, the Green rule will be applied and the text label "GOOD" will be displayed.

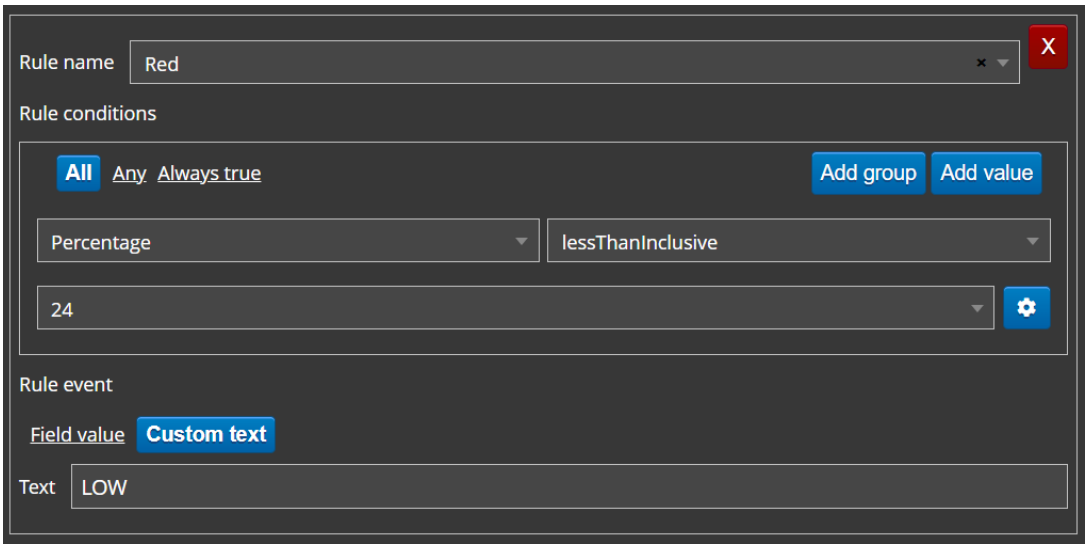
In the scenario below, the condition group of a battery percentage between 25 and 49, inclusive, is made up of two separate value statements: a percentage ≤49 and a percentage ≥25.



When the condition represented by this group is met, the Yellow rule will be applied and the text label "OK" will be displayed.



In the scenario below, the condition of a battery percentage less than 25 only requires one value statement: a percentage ≤24.



When the condition is met, the Red rule will be applied and the text label "LOW" will be displayed.

Example 3: Combining Values and Text

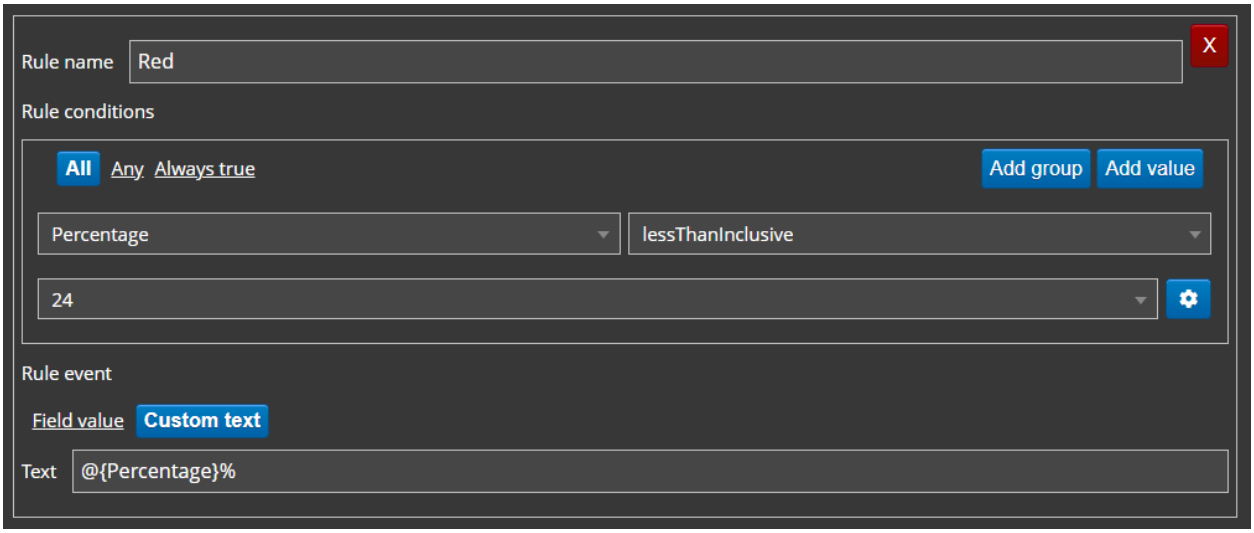
You can also combine custom text and variables. In this example, we will use text to add a percent sign to the percentage field from Example 1.

To include a field value within custom text, introduce the field name with an at (@) sign and wrap it in curly brackets:

```
@{FieldName}
```

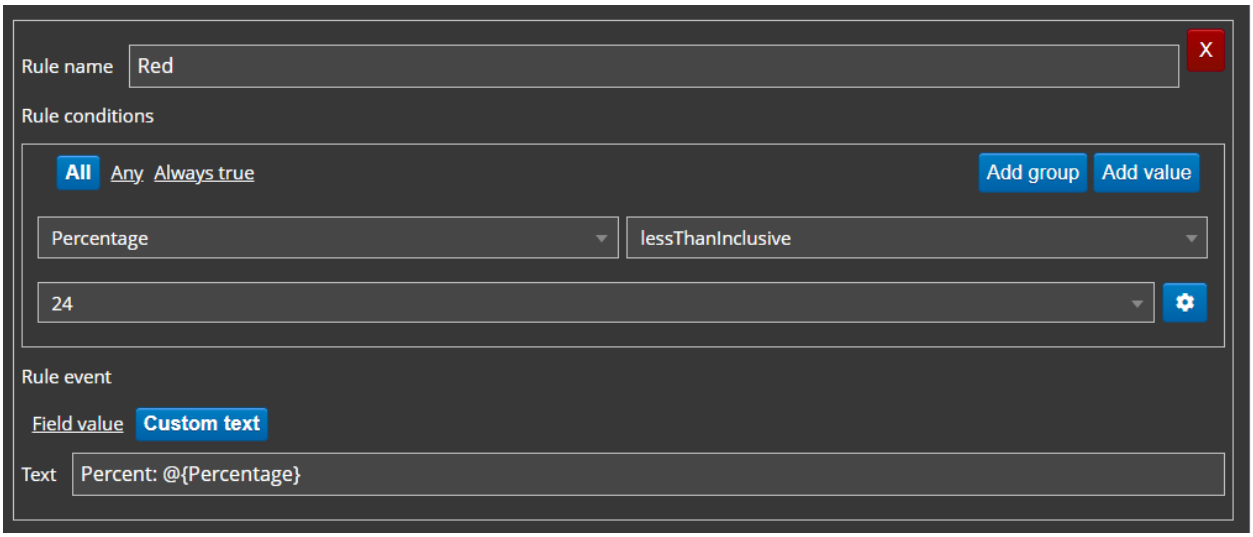
In our example, the field name looks like this:

```
@{Percentage}%
```



If you wanted to put a "Percent" label in front of the percentage value, you would enter:

"Percent: @{Percentage}"



TIP

If there is a possibility that the value you are displaying does not exist, you can set a default value within the field name. For example, if when the battery record did not include a percentage, you wanted the percentage to show "-%" then you would add a colon after the field name within the curly brackets, followed by the default value to be used for the missing value (in this case, "-"). For example: `@{Percentage:-}%`

Fragment Rules

More information about Fragment rules is forthcoming.

2.5.1.3.5 Image Preview

The lower-right panel of the image viewlet editor shows a preview of the image.

Image Preview Panel Buttons:

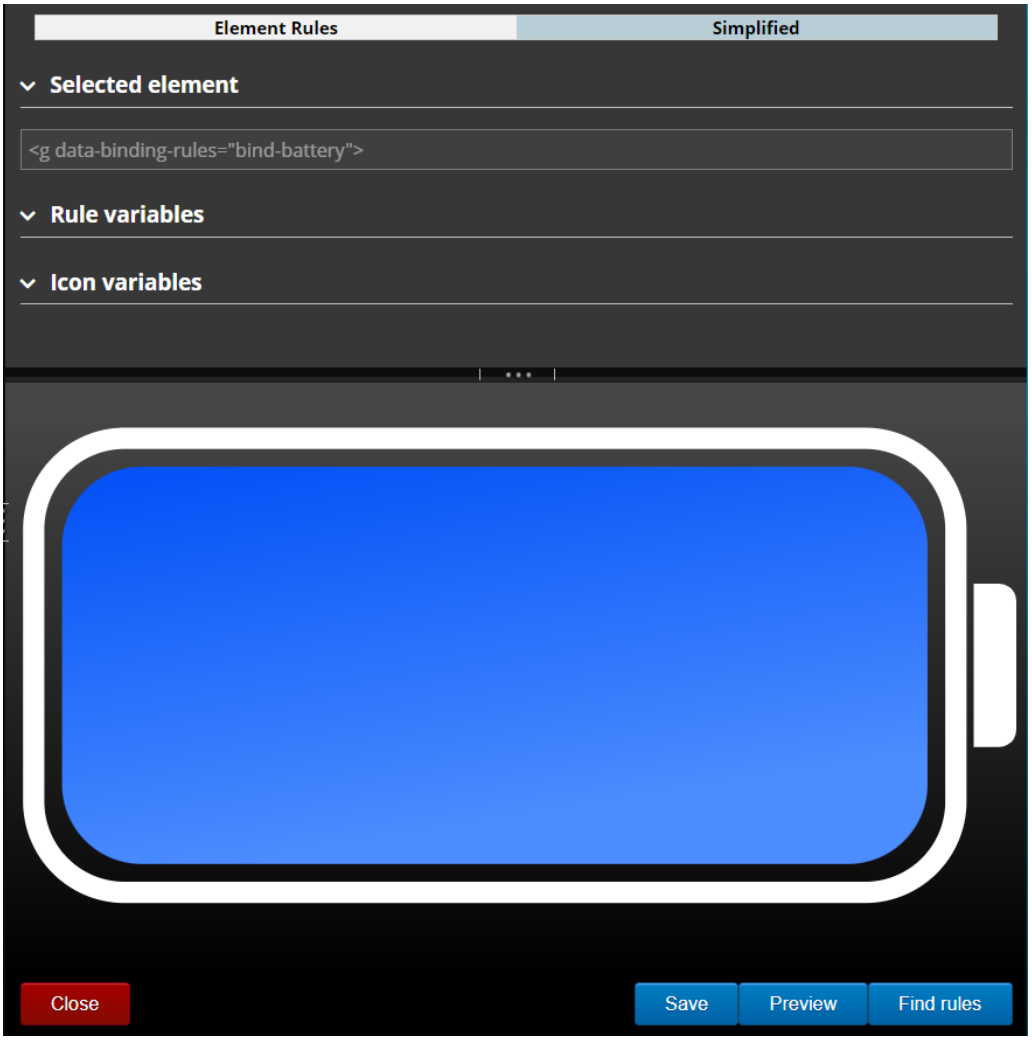
- Click **Close** to close the image template editor and return to the XRay user interface.
- Click **Preview** to preview changes you've made in the SVG File Editor to see their effects before saving.
- Saving your changes to the image template editor will close the editor. When you have completed all desired changes in the SVG File Editor, click **Save** to save changes to the image template editor.

2.5.1.3.5.1 Editing Rules for a Selected Element

You can view the rules that have been applied to an element.

1. Select the specific element using one of the methods below:
   - Select the code for the element in the SVG File Editor.
   - Click the element in the Image Preview panel. (However, see the Important note in the Main Tab section for detailed information.)
2. The Selected Element Rules Editor shows the rules that have been applied to that element.

By default, the Simplified tab shows all existing variables for the element and allows you to specify values for them.



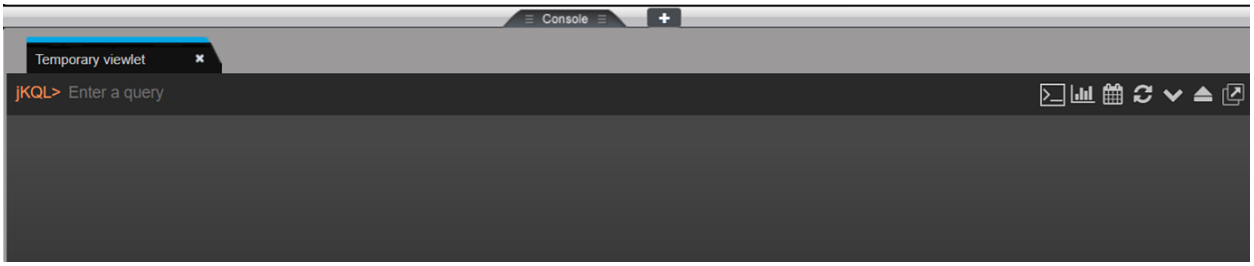
3. To modify rules, select the Element Rules tab. On this tab, you can create new rules. You can also edit existing rules, as long as the rule is not also used by other elements. In that case, you must edit the rule on the Rules tab (see Rules Tab).

4. (Optional.) To preview your changes, click **Preview**.
5. To save changes, click **Save** in the lower-right corner of the window.

2.5.1.4 Create Temporary Viewlet

Create temporary viewlets in the **Console** panel by clicking the **+** button immediately to the right of the **Console** tab. Enter a query in the jKQL query line to generate your desired viewlet. For more information on the Console panel, please see *Section 2.4.4, Console Panel*.

Figure 2.5.1.3-A. Add Console Viewlet



2.5.1.5 Open Existing Viewlet

Selecting **Open Existing Viewlet** on *Create/Open Viewlet* dialog box will open the *Open Existing Viewlet* dialog box. The view can be changed by selecting **Details** from the **View By** drop-down for a more descriptive view (*Figure 2.5.1.4-B*). Use the **Sort By** drop-down to arrange the viewlets in alphabetical order or by chart type. Quickly search for viewlets by viewlet name using the search box.

Select a viewlet and click **Open**. The dashboard's focus will now be the selected viewlet.

The *Open Existing Viewlet* dialog box can also be opened from the **Main Menu** (*Section 2.3.6*).

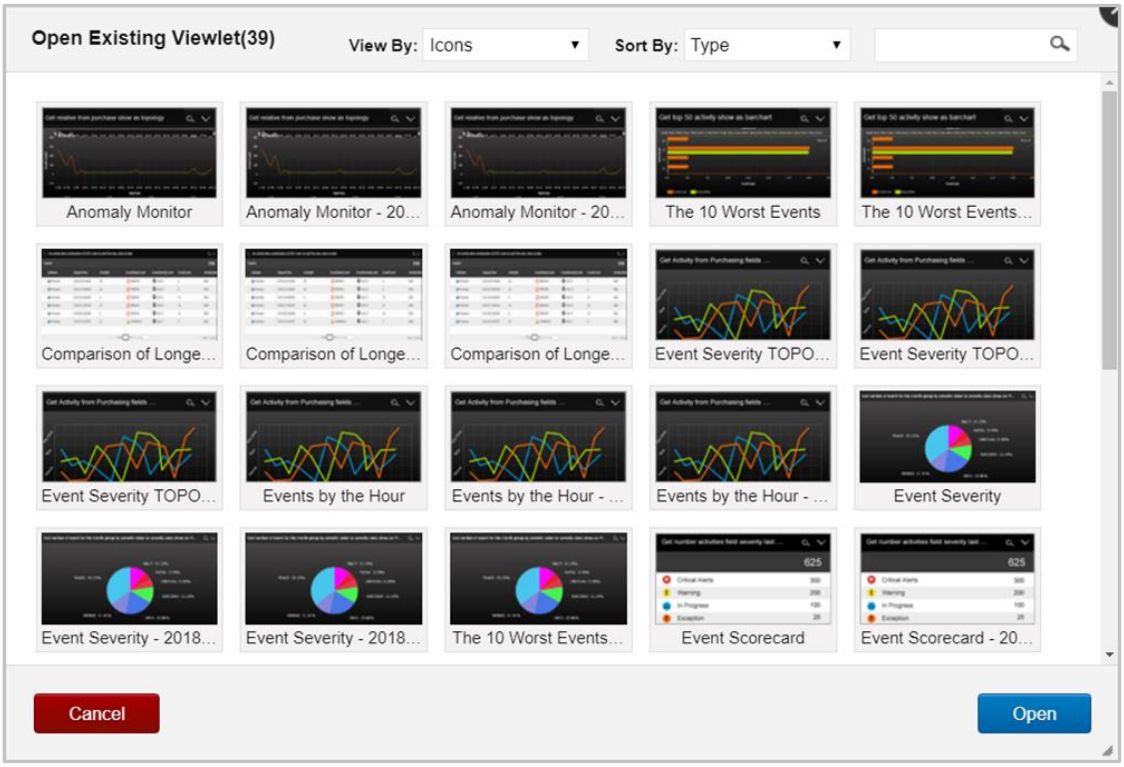


Figure 2.5.1.4-A. Open Existing Viewlet – View By Icons

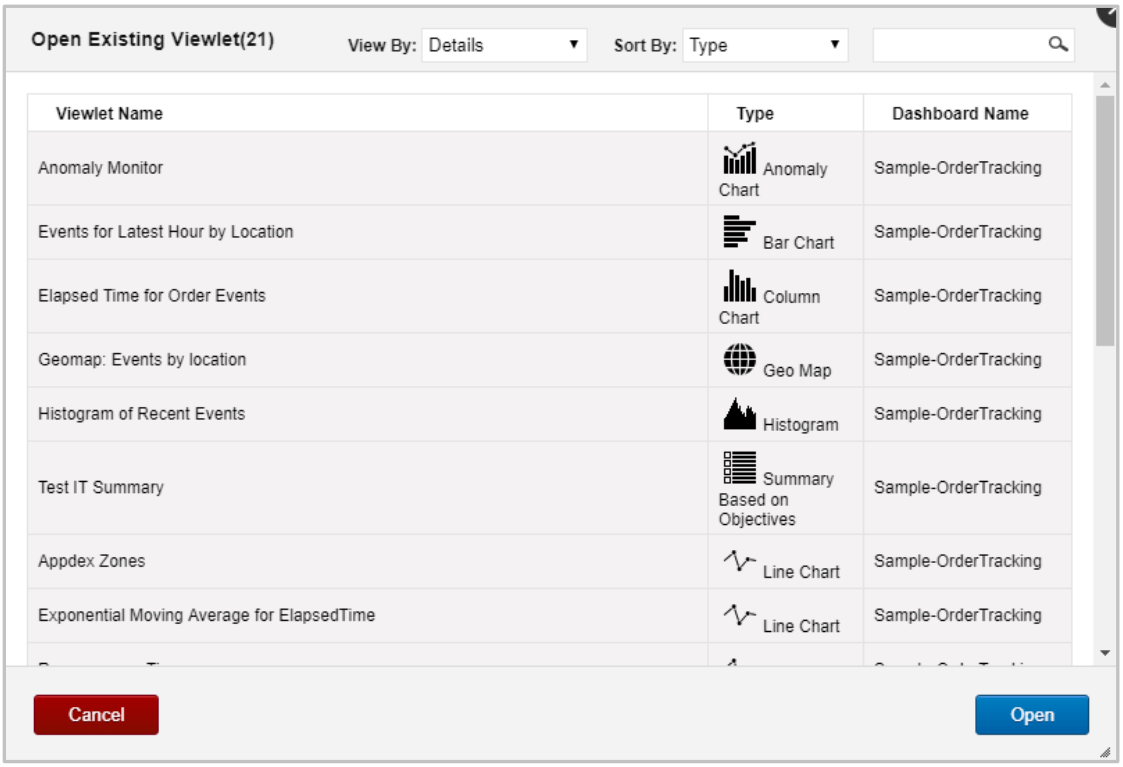


Figure 2.5.1.4-B. Open Existing Viewlet – View By Details

2.5.2 Edit Query

The query line becomes an editable field after you click the edit query icon or you can simply click the query line. Make your changes. As you edit, you will be prompted with suggestions as in *Figure 2.5.1.1-B*.

Figure 2.5.2-A. Edit Query

2.5.3 Undo / Redo

The undo and redo buttons are used to revert or reapply changes from the current user session history. Changes tracked which can be undone or reapplied include updates made to viewlet names, settings, and queries.

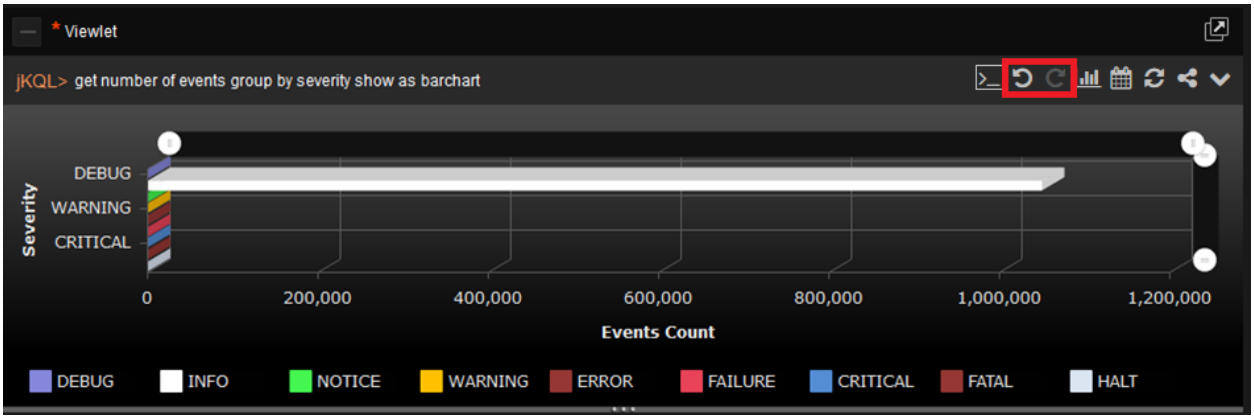


Figure 2.5.10-E. Undo / Redo Buttons

2.5.4 Viewlet Chart Types and Samples

The data in viewlets can be formatted in various chart types.

Easily update a viewlet's chart type by clicking the **Chart** icon . The selected chart type of a viewlet will be highlighted blue. See Sections 2.5.4.1 – 2.5.4.13 below for an explanation of each chart type. Within each section there are samples of the chart types and an explanation of a scenario in which the chart type is useful. Some of these samples can be found in the **Sample-OrderTracking** repository.

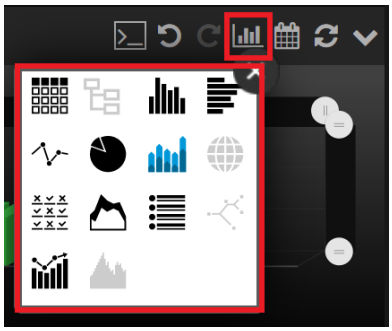


Figure 2.5.4-A. Chart Types

2.5.4.1 Table

Table is the default chart type for viewlets. All data imported will display unless a schema is used. In table viewlets, schemas control what columns are displayed and in what order. See *Sections 3.2.5, Configure Schemas*, and *2.4.2.8 Change Layout* for more information. Use the scroll bar at the bottom of the viewlet to view additional columns. Columns can also be expanded by hovering over the line separators within the header sections.

Click columns headers to sort the data in ascending ( ) or descending ( ) order.

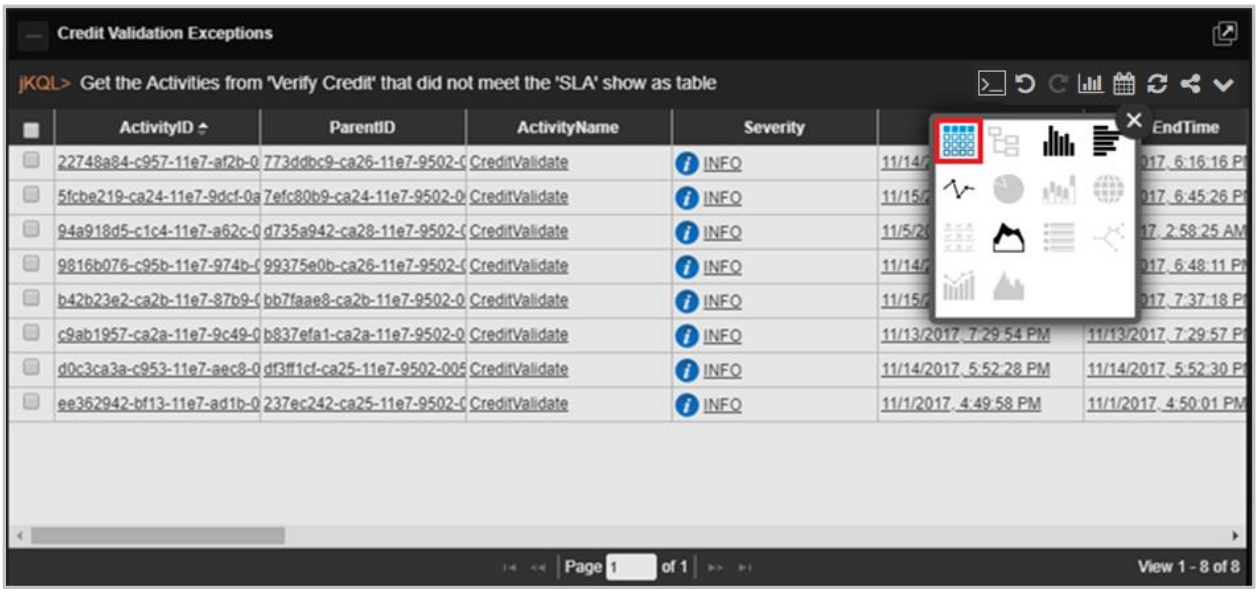


Figure 2.5.4.1-A. Table

Viewlets with a lot of data to display will have multiple pages. Use the left and right arrows to easily navigate through the pages. The **First** and **Last Page** buttons allow users to quickly jump to the first and last pages. Use the **Previous** and **Next Page** buttons to navigate through each page. Enter a page number to load a specific page.

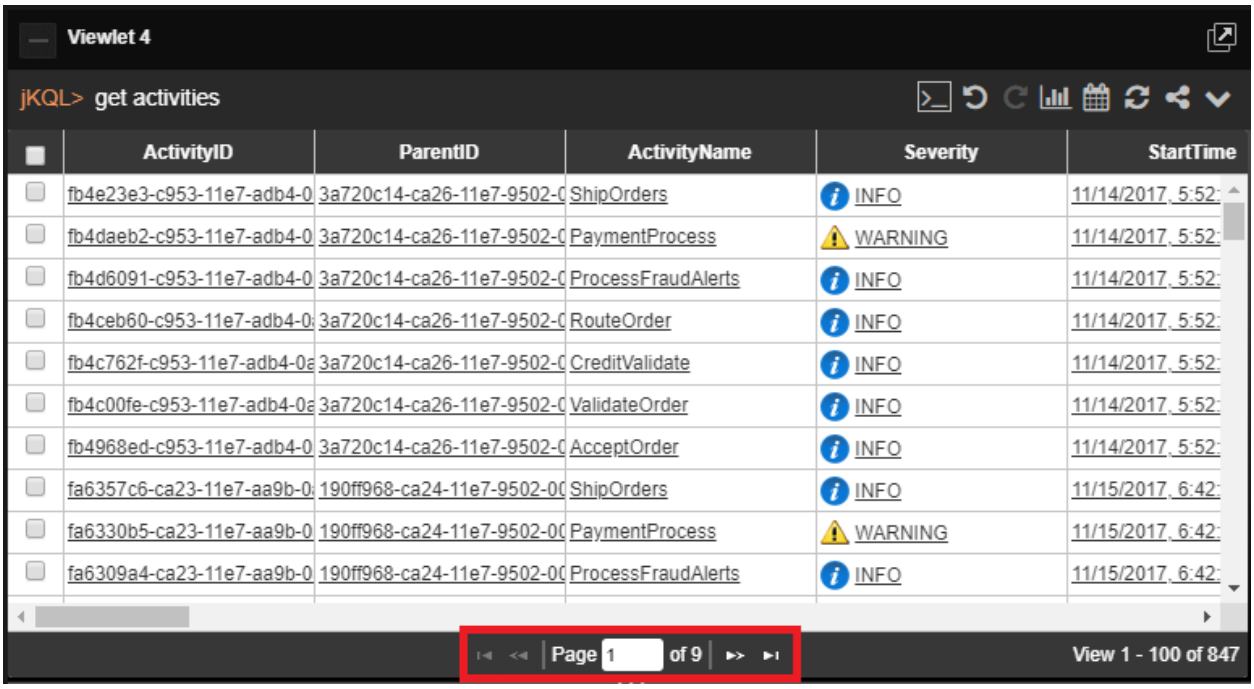


Figure 2.5.4.1-B. Page Navigation

Click on any of the data records to view additional details in a new viewlet within the **Console** section. See *Section 2.4.4, Console Panel*, for more information.

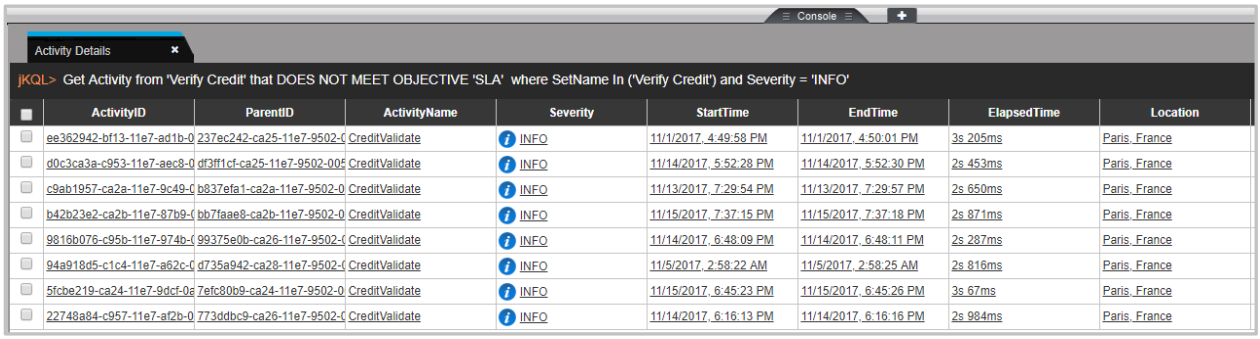


Figure 2.5.4.1-C. Additional Details in Console

2.5.4.1.1 Table Menu Options

Users have different options to dig deeper into the data of table charts. Select all desired records or use the top box to select all records. A pop-up menu appears. Select an option from the pop-up menu to view additional data details within the **Console** section.

The pop-up menu options depend on the data type. A viewlet containing events (*Get events*) will have the following menu items:

- Related
- Parent
- Analyze
- Topology
- Compare (available only when more than one line is selected)

A viewlet containing activities (*Get activities*) will have the following menu options:

- Events
- Related
- Parent
- Topology
- Root Cause (available only for severity status of Error, Halt, Failure, Fatal or Critical)
- Children (available when one or more activities with children are selected)
- Compare (available only when more than one line is selected, see *Figure 2.5.4.1.1-A* and *Figure 2.5.4.1.1-B*).

Tables produced by jKQL queries with the following expressions will not have a pop-up menu: snapshots, logs, actions, active users, count of/number of, token, organization, teams, repository, set, license, fields, items, relatives, provider types, keywords, parameter, dictionary, features, access token, IP location, resource, group by.

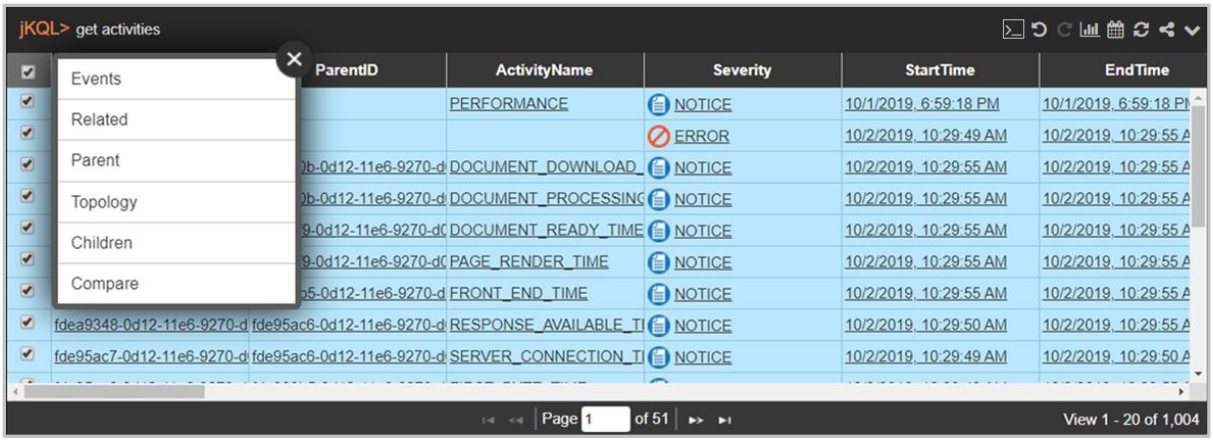


Figure 2.5.4.1.1-A. Table - Select All

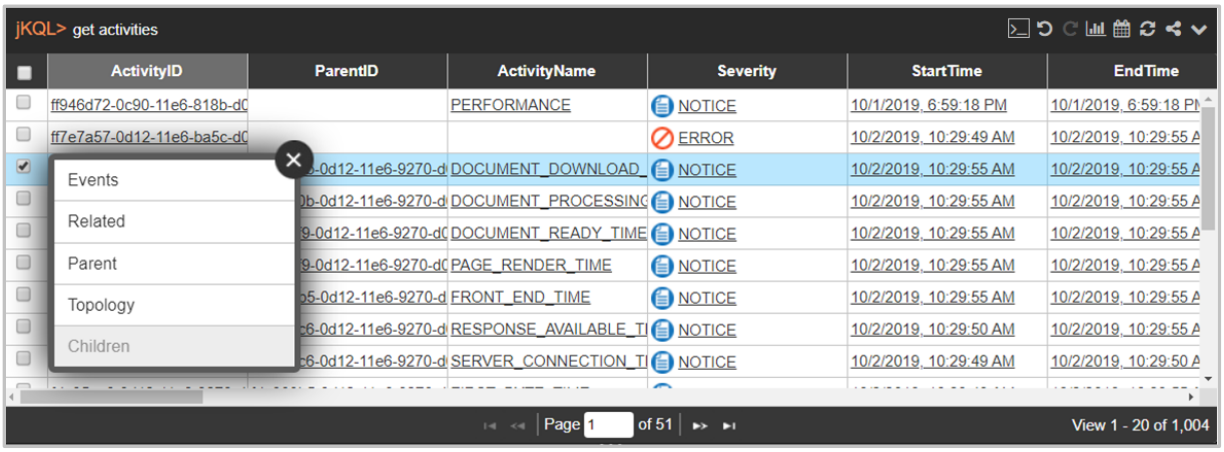


Figure 2.5.4.1.1-B. Table – Select One

After selecting an option on the pop-up menu, a new viewlet related to the option selected will open in the **Console** section.

If **Events**, **Related**, **Parent**, **Children** or **Analyze** were selected, the data will display in a table by default. You can modify the jKQL query to customize the viewlet. These tables have their own pop-up menus which allow users to dig deeper into data. Every selection from the pop-up menu will open a new viewlet within the **Console** section.

2.5.4.1.1.1 Events

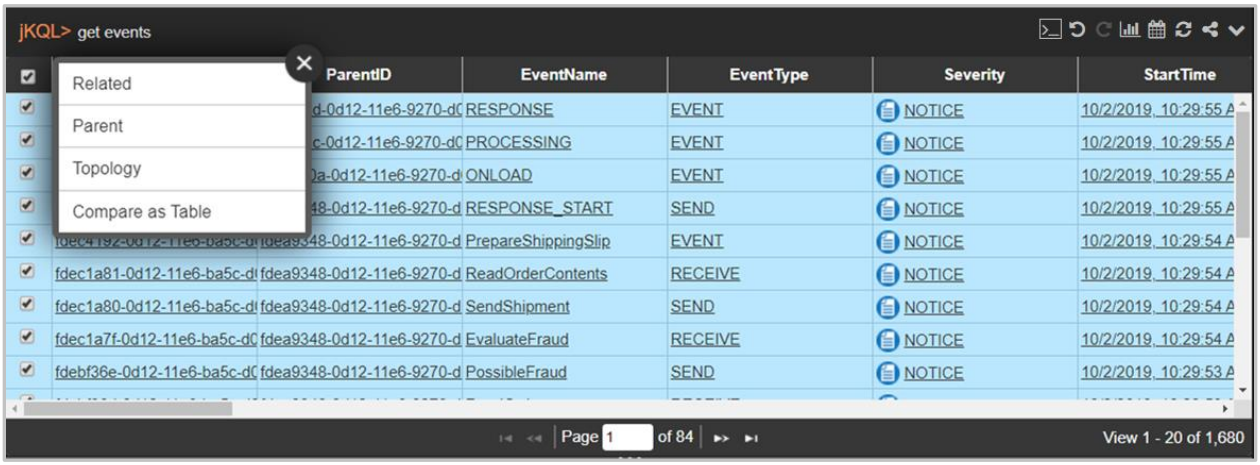


Figure 2.5.4.1.1.1-A Events

The above example was generated by selecting **Events** from the activity's table pop-up menu. The tab name is the selected menu option and the name of the main viewlet. In the example above, the tab name is **Events_Event Severity**.

2.5.4.1.1.2 Related

The below figure is the viewlet that appears when **Related** is selected from the pop-up menu of the **Events_Event Severity** tab**.** The table displays events which have the same selected Activity ID(s).

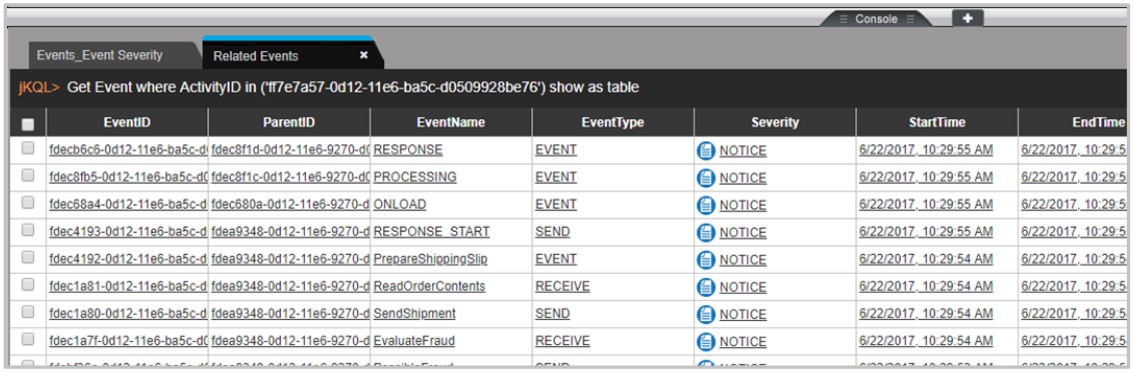


Figure 2.5.4.1.1.2-A. Related

2.5.4.1.1.3 Parent

Select **Parent** from the pop-up menu to open a viewlet which displays parent activities (activities with a greater hierarchical status) of the selected activities or events. Only items with values within the **ParentID** column will have a **Parent** menu option on the table's pop-up menu.

The parent activities of other activities or events will be displayed. An activity without a **ParentID** means that it is the prime activity with the highest hierarchical status.

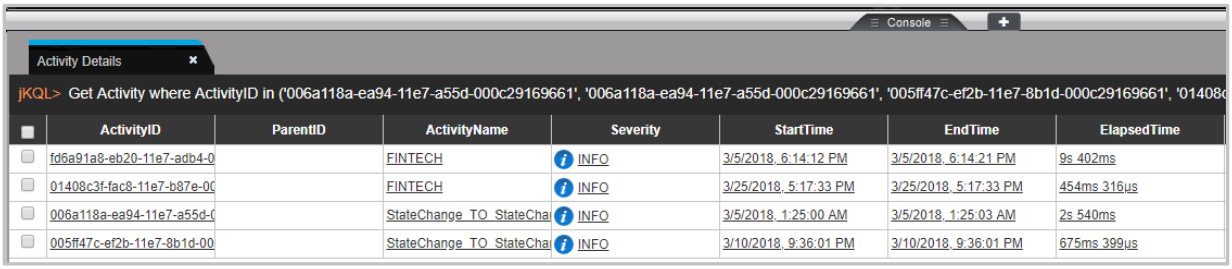


Figure 2.5.4.1.1.3-A. ParentID

2.5.4.1.1.4 Analyze

The analyze function creates an analysis of what factors impacted events. To make the analysis, this function uses event and related event data, which is a collection of snapshots.

After selecting **Analyze** from the pop-up menu of the Event table, the **Create an Analysis Viewlet** window appears. Specify desired options. Please note that **Show Trend Line** becomes active when **Show Elapsed Time** is enabled. Enable the **Remember My Choice** check box if you would like to save your selections.

Click the **Next** button for additional customization or **Create** to create the viewlet.

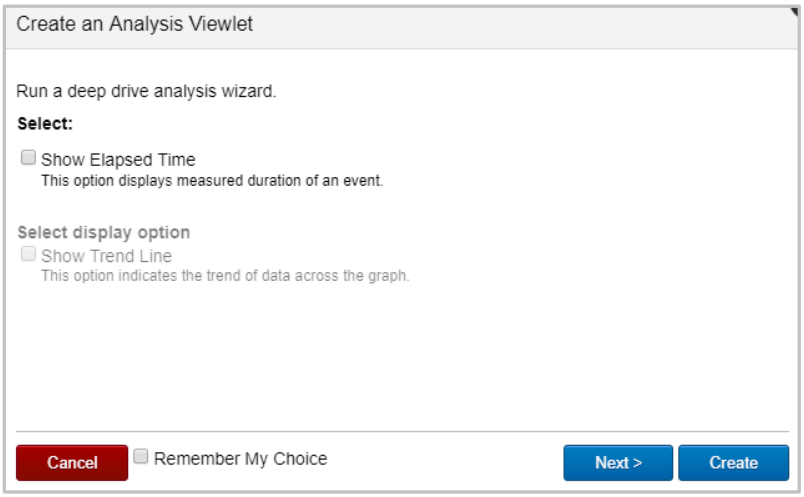


Figure 2.5.4.1.1.4-A. Create an Analysis Viewlet

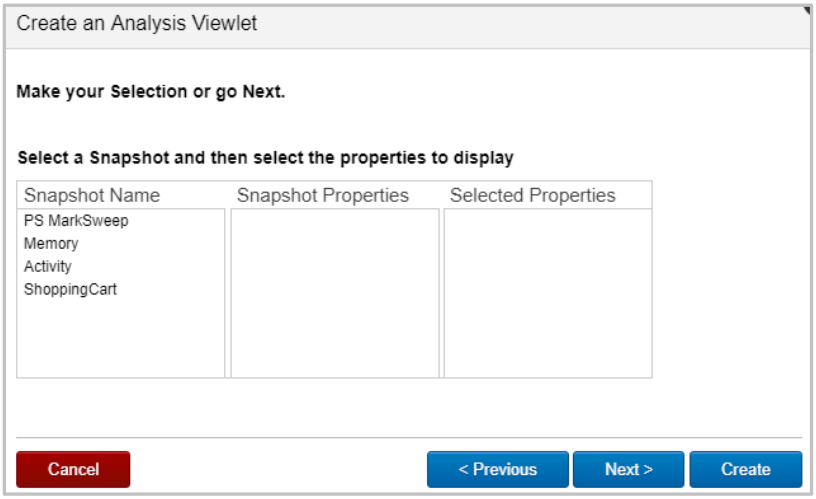


Figure 2.5.4.1.1.4-B. Create an Analysis Viewlet – Additional Options

Select a snapshot from the **Snapshot Name** column and select all desired options from the **Snapshot Properties** column. The selected properties will be displayed in the last column. Click **Create** to finish or **Next** for additional customization.

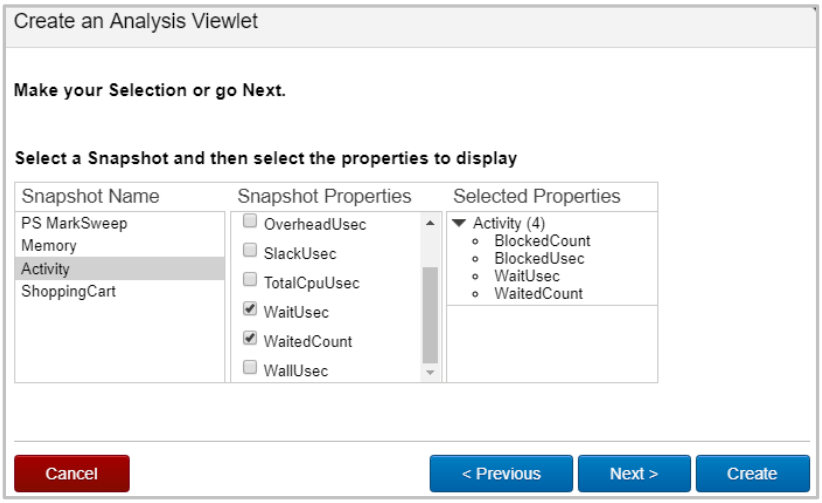


Figure 2.5.4.1.1.4-C. Create an Analysis Viewlet – Select Options

Select which items you would like the viewlet to display and click **Create**. The *Analyze_Event Details* viewlet will appear within the Console panel.

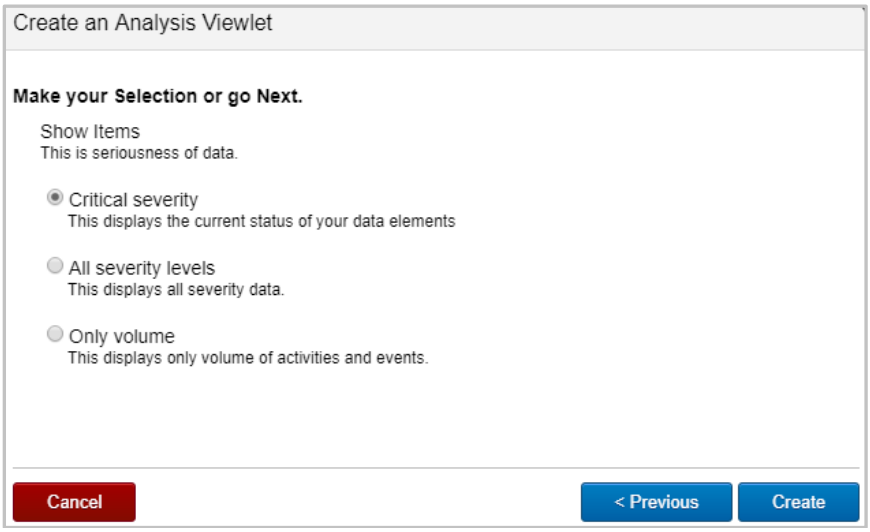


Figure 2.5.4.1.1.4-D. Create an Analysis Viewlet – Select Items

2.5.4.1.1.5 Topology

Select Topology in the pop-up menu to generate a topology viewlet using the selected items within the **Console** section. For more information on topologies, see *Section 2.5.4.11, Topology*.

2.5.4.1.1.6 Root Cause

When selecting activity table records with a severity status of Error, Halt, Failure, Fatal or Critical, **Root Cause** will be an option on the pop-up menu. Clicking this will open a viewlet in the **Console** section displaying a topology. The topology will allow you to dig deeper into the data and find the root cause of the issue.

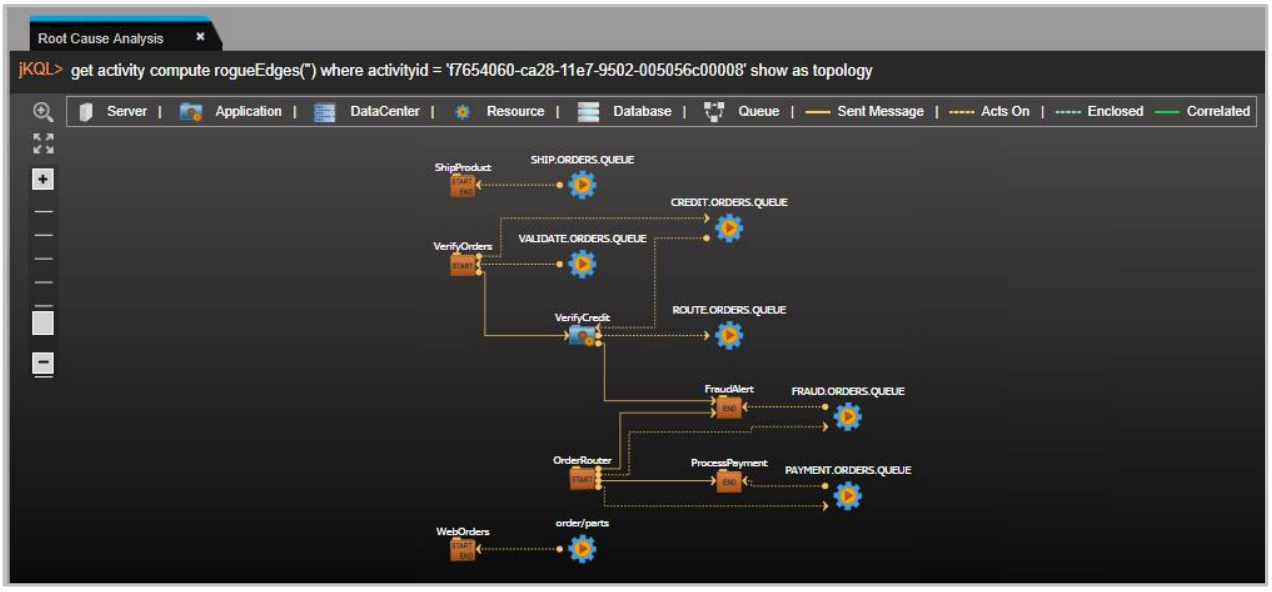


Figure 2.5.4.1.1.6-A. Root Cause

2.5.4.1.1.7 Children

Select **Children** from the pop-up menu to open a viewlet which displays child activities. Please note that not every data record will have child activities and will therefore not have the **Children** option in the pop-up menu. Activities or events of an activity will display.

2.5.4.1.1.8 Compare

Select more than one record to enable the **Compare** option within the table's pop-up menu. This will open a compare table within the **Console** panel (See *Section 2.4.4, Console Panel*).

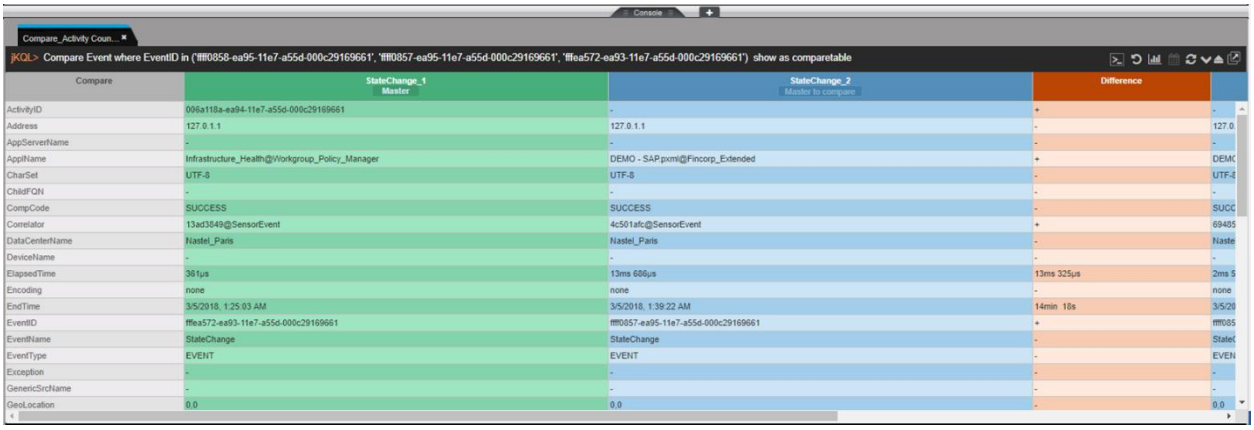


Figure 2.5.4.1.1.8-A. Compare Table in Console

You can also view a compare table in the main workspace by using the 'Compare' command in a jKQL query line, for example:

Query: **jKQL>** `Compare Activity where ActivityID in ('activity ID of first selected activity', 'activity ID of second selected activity') show as comparetable`

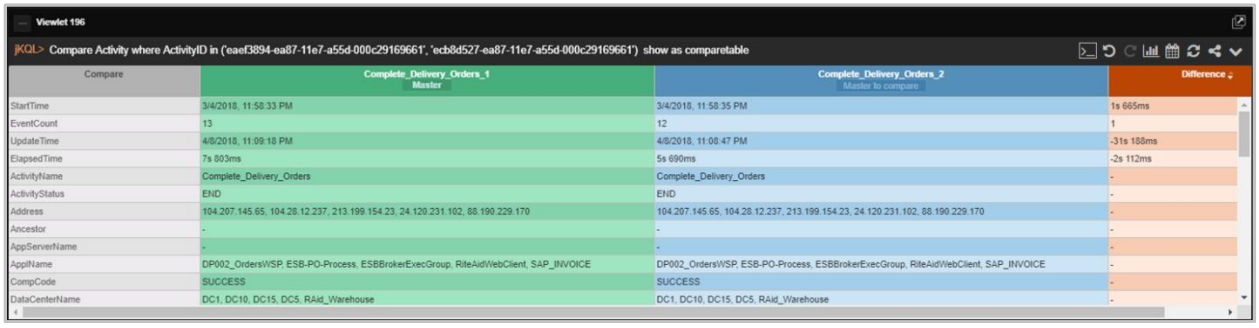


Figure 2.5.4.1.1.8-B. Compare Table in Workspace

In the **Compare** column the items to compare are displayed in alphabetical order. Change the order by clicking the sort buttons, ⬘ (ascending) or ⬙ (descending). This same function is available in all other columns.

The green column with **Master** Master located in the column header signifies the master record. All other selected records will display in blue and will be compared to the master record. To change the master record, simply click the **Master to Compare** button Master to compare within any of the other blue columns. The column will move to the first position and will change to green. All other fields will now be compared against this new master.

In the **Difference** column, the **+** and **-** symbols signify whether there is a difference. If the difference can be measured mathematically, the numeric value will be displayed (for example, the microsecond difference of elapsed time, *Figure 2.5.4.1.1.8-C*).

To change the width of the columns, hover over the lines between column headers until you see the size icon ←‖→. Move it left or right to adjust column width.

Event b1e5ef5c-f31c-11e8-93a7-0242ac12000a Master	Event b1e3a568-f31c-11e8-93a7-0242ac12000a Master to compare	Difference
b1e52c12-f31c-11e8-a2df-0242ac12000d	b1e3092e-f31c-11e8-a2df-0242ac12000d	+
5ms 755µs	4ms 868µs	-887µs
11/28/2018, 4:48:40 PM	11/28/2018, 4:48:40 PM	-
b1e5ef5c-f31c-11e8-93a7-0242ac12000a	b1e3a568-f31c-11e8-93a7-0242ac12000a	+
b1e52c12-f31c-11e8-a2df-0242ac12000d	b1e3092e-f31c-11e8-a2df-0242ac12000d	+
11/28/2018, 4:48:40 PM	11/28/2018, 4:48:40 PM	-
11/28/2018, 4:48:40 PM	11/28/2018, 4:48:40 PM	-
11/28/2018, 4:49:17 PM	11/28/2018, 4:49:17 PM	-

Figure 2.5.4.1.1.8-C. Difference Column

The viewlet can be updated to display only rows with differences. Click the **Viewlet Menu** button and select **Edit Viewlet**. Enable the **Only Show Differences** option on the form. Only rows in which the data is different will now display.

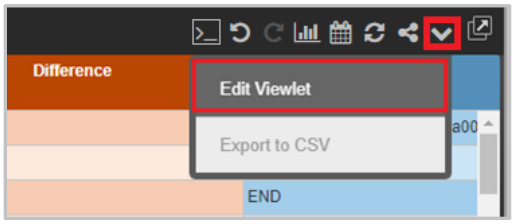


Figure 2.5.4.1.1.8-D. Compare Table - Edit Viewlet

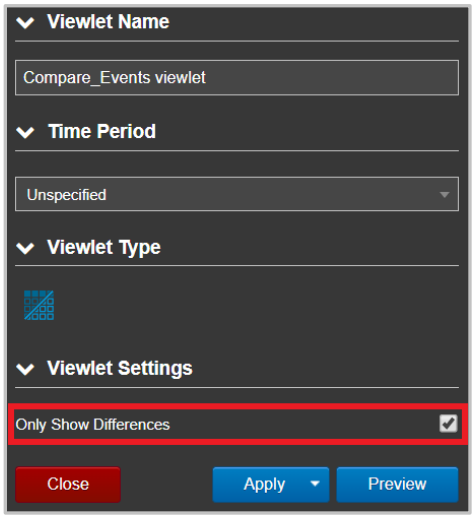


Figure 2.5.4.1.1.8-E. Only Show Differences

2.5.4.1.2 Table Arrangement

Users can create a customized table with specified columns and column order. Use 'fields' and 'order by <*field name*> asc' or 'order by <*field name*> desc' expressions (*asc* is ascending

order and *desc* is descending, see *Section 5.4, Additional Query Options*, for more information on jKQL sorting options).

The following is an example:

Query: jKQL> `Get activities fields ParentID, ActivityID, EventID order by ParentID desc`

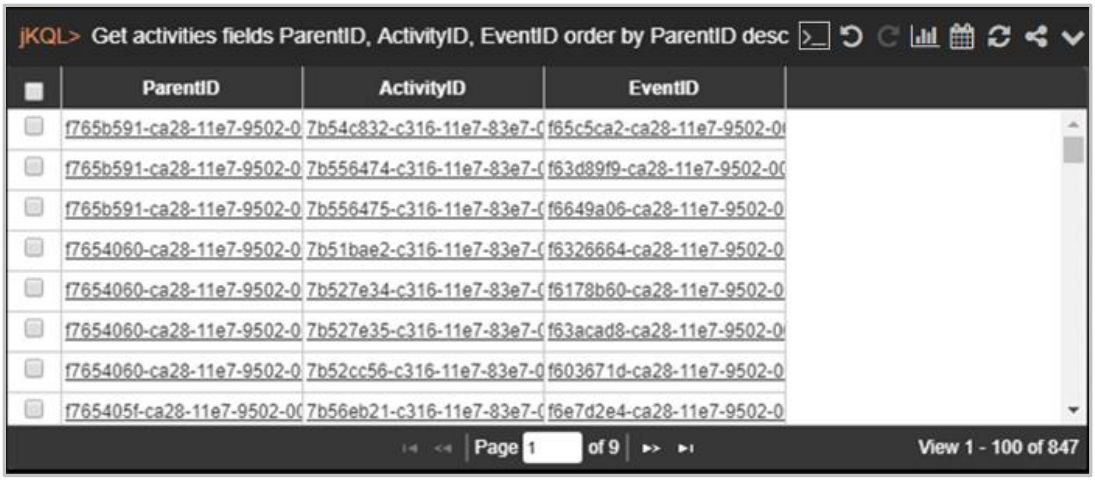


Figure 2.5.4.1.2-A. Custom Table Arrangement

This query will produce a table composed of three columns in the order specified in the query. The data will be sorted by the **ParentID** column in descending order (*Figure 2.5.4.1.2-A*).

Table columns can also be rearranged manually. Simply click and drag a column header to the new desired position.

2.5.4.1.3 Sample: Credit Validation Exceptions

Query: jKQL> `Get the Activities from 'Verify Credit' that did not meet the 'SLA' show as table`

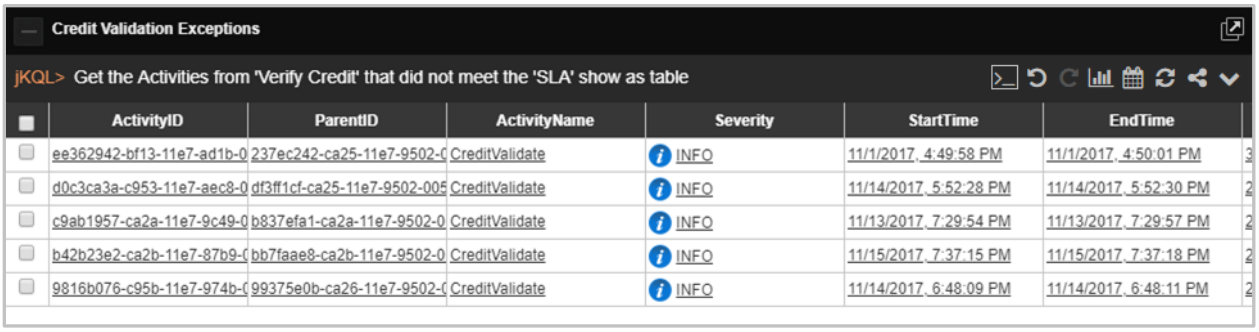


Figure 2.5.4.1.3-A. Sample Viewlet – Credit Validation Exceptions

The viewlet above is in the **Sample-OrderTracking** repository. It shows an example of exceptions or errors for specific activities. Here we are checking for ones that missed their service level agreement (SLA) requirements. A user would utilize this to find the errors and then drill down into the specifics in the **Console** to try and learn why. This is part of the forensics process.

2.5.4.1.4 Sample: Snapshots

A table of snapshots will not have check boxes. To get additional data details, click on the underlined elements. Additional details will display in the **Console** section (*Figure 2.5.4.1.4-B*). See *Section 2.4.4, Console Panel*, for more information.

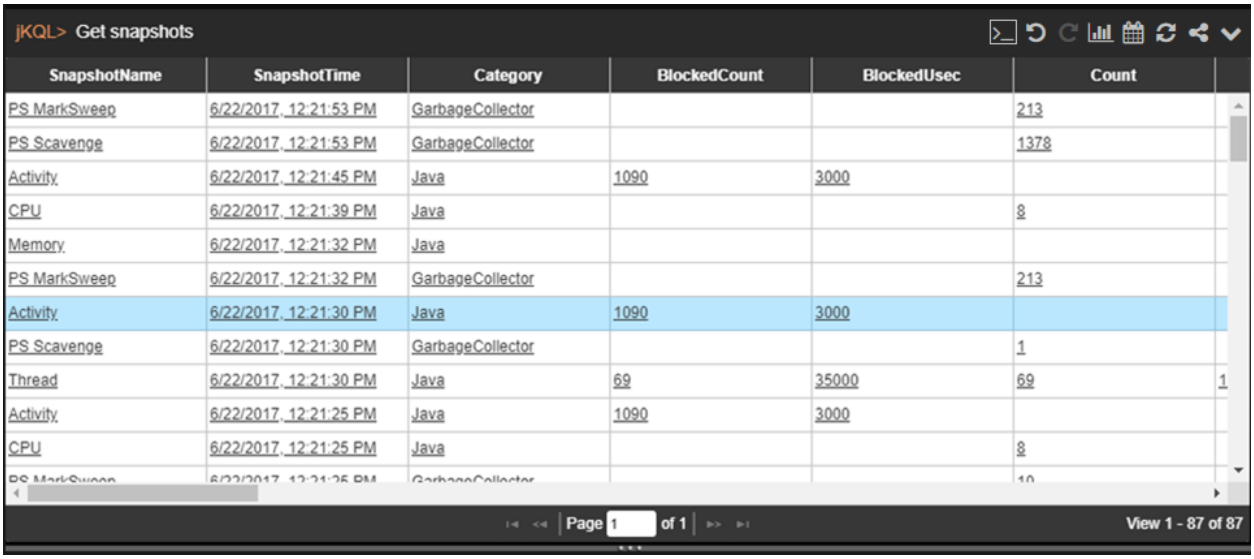


Figure 2.5.4.1.4-A. Snapshot Viewlet

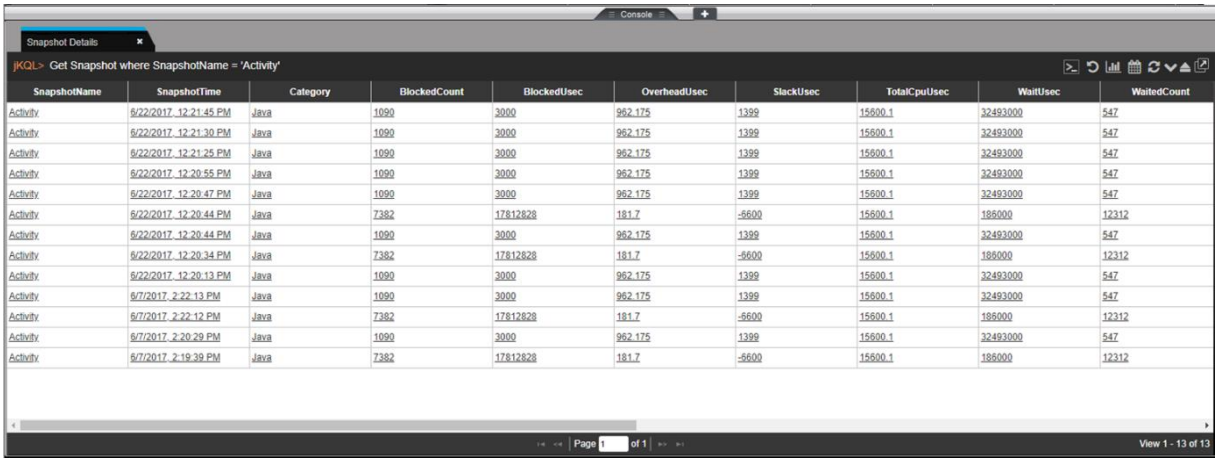


Figure 2.5.4.1.4-B. Detailed Console Viewlet

If a user has permissions to run jKQL queries, the jKQL query can be modified by changing the *Show as* expression.

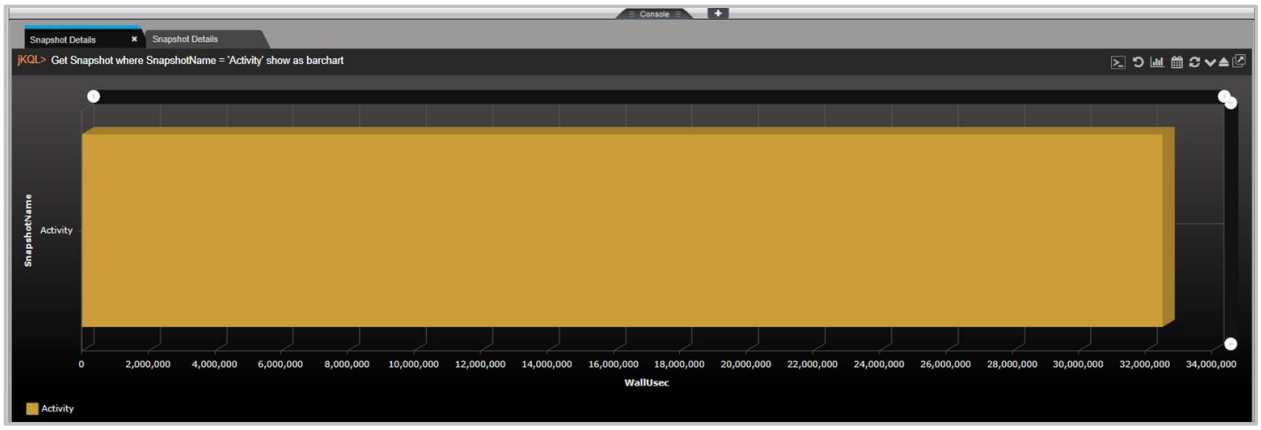


Figure 2.5.4.1.4-C. Changing the Show As Expression

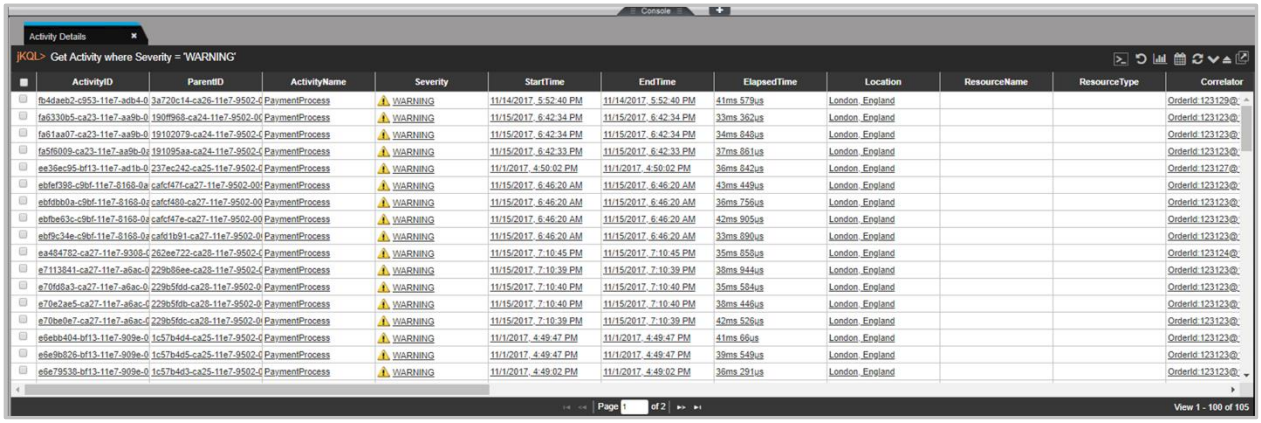


Figure 2.5.4.1.4-D. Activity Details

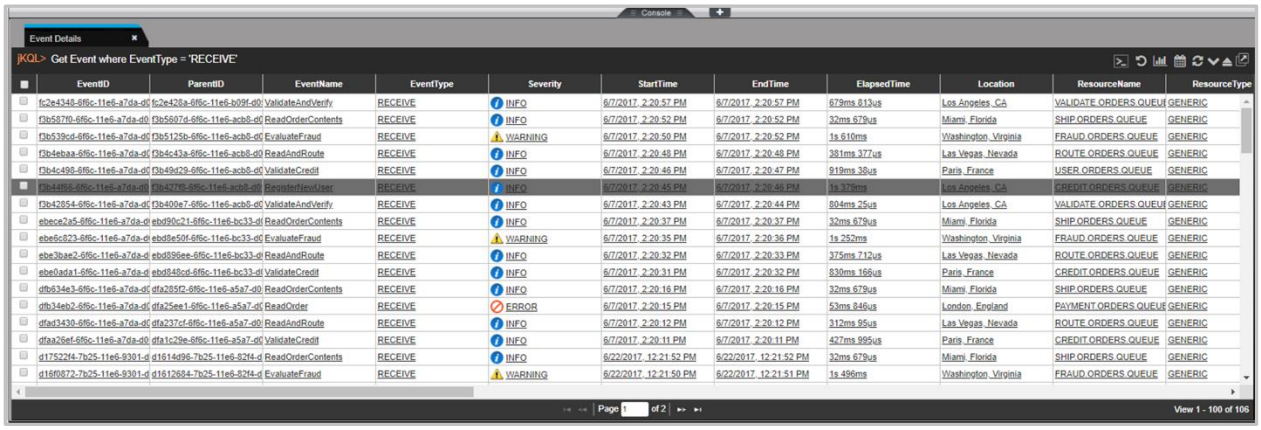


Figure 2.5.4.1.4-E. Event Details

2.5.4.2 Column

Column charts allow users to view a large dataset in an easy-to-read column view. See *Section 2.5.8, Filtering and Display Options*, for information on filtering options available.

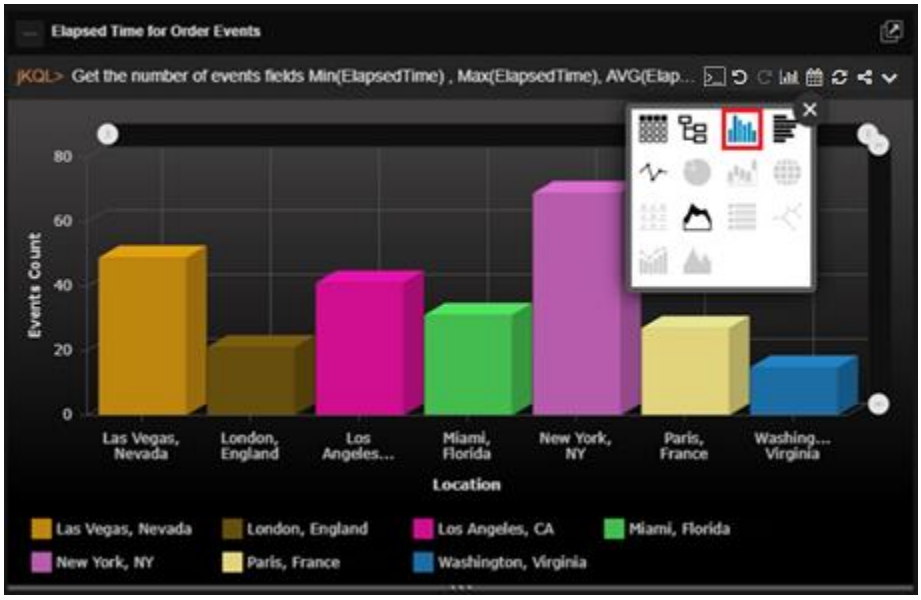


Figure 2.5.4.2-A. Column

Hover over any of the bars in the chart to view a status pop-up.

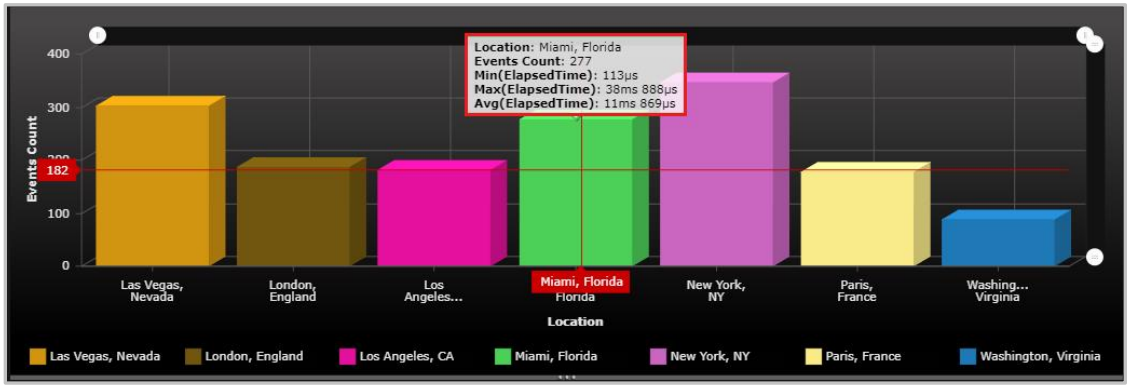


Figure 2.5.4.2-B. Status Pop-up

Click on any of the bars in the chart to view additional details in a **Console** viewlet. See *Section 2.4.4, Console Panel*, for more information.

2.5.4.2.1 Sample: Elapsed Time for Order Events

Query: jKQL> `Get the number of events fields Min(ElapsedTime), Max(ElapsedTime), AVG(ElapsedTime) group by location show as colchart`

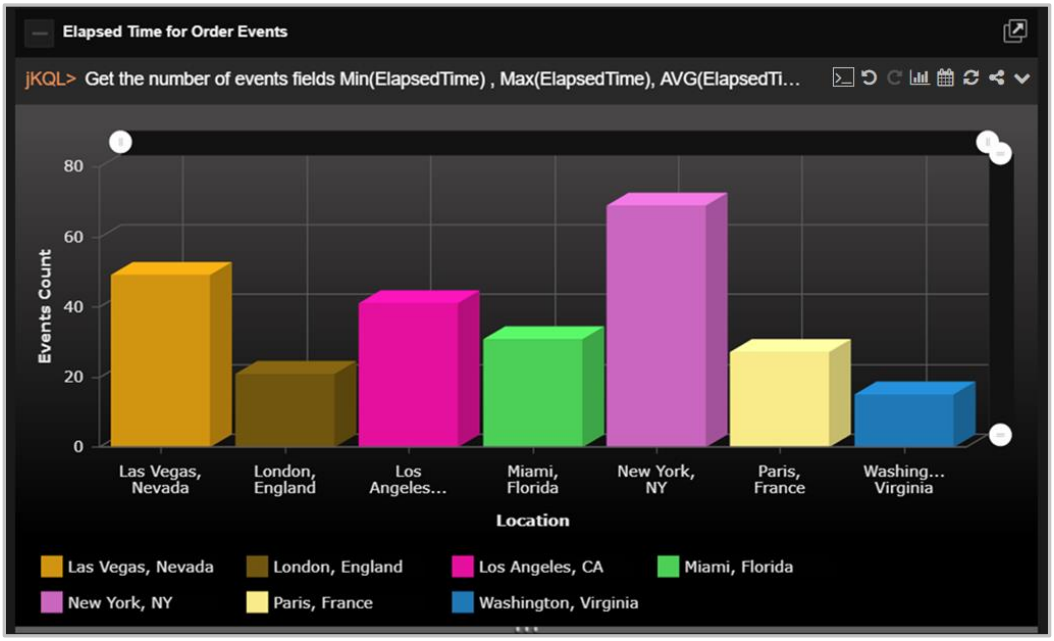


Figure 2.5.4.2.1-A. Sample Viewlet – Elapsed Time for Order Events

The viewlet above is in the **Sample-OrderTracking** repository. It uses the functions min, max, and average as applied to elapsed time for events.

2.5.4.3 Bar

Bar charts generate data in a viewlet with horizontal bars. See *Section 2.5.8, Filtering and Display Options*, for information on filtering options available.

Similar to the Column chart explained above in *Section 2.5.4.2, Column*, hovering over the bars will display a status pop-up and clicking on the bars will open a **Console** viewlet to view additional details. See *Section 2.4.4, Console Panel*, for more information. *Number*, *count* or other numeric expressions must be included in the query or form.

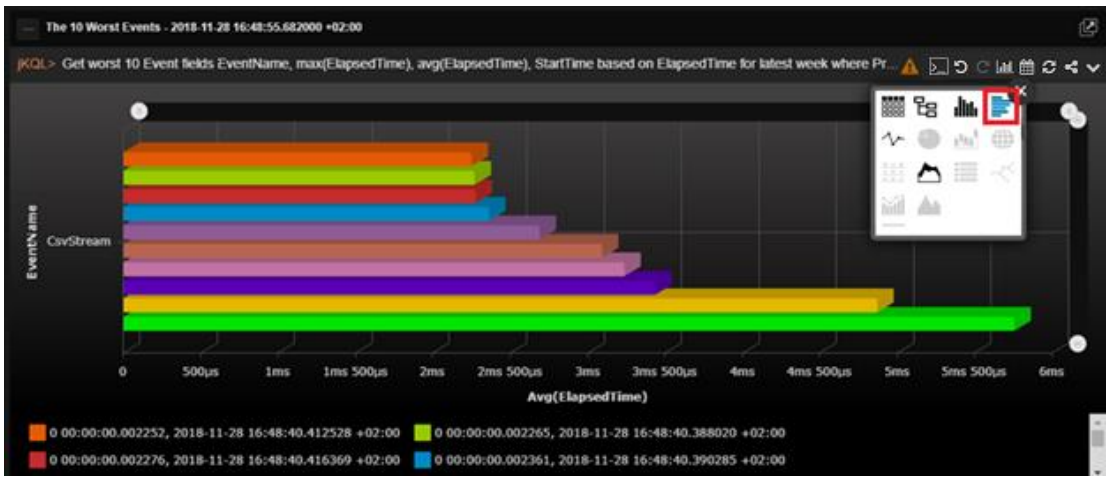


Figure 2.5.4.3-A. Bar

2.5.4.3.1 Sample: Events for Latest Hour by Location

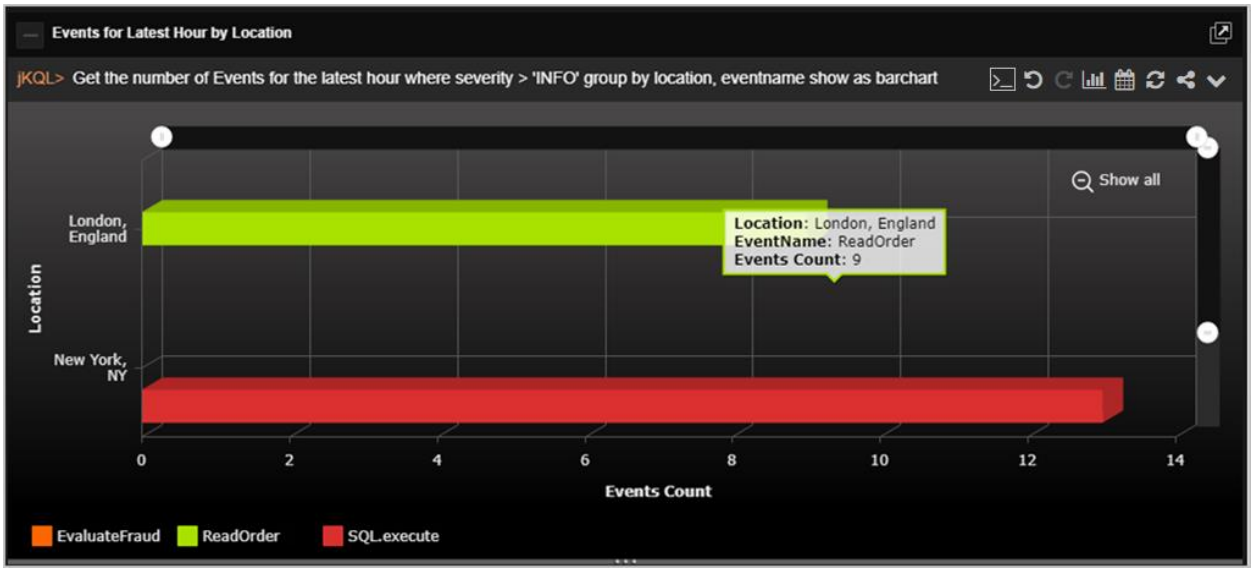


Figure 2.5.4.3.1-A. Bar Chart – Events for the Latest Hour by Location

The bar chart viewlet is useful as it allows you to easily see the differences of various item counts, grouped by location, severity, or another keyword.

2.5.4.3.2 Sample: Events by Severity

Figure 2.5.4.3.2-A. Bar Chart – Events by Severity

This bar chart shows the number of events, grouped by severity. Modify the chart colors in **Main Menu** > **User Settings** > **Manage Settings** (Colors tab) to make the viewlet more informative and easier to analyze (*Section 3.2.5*).

2.5.4.4 Line

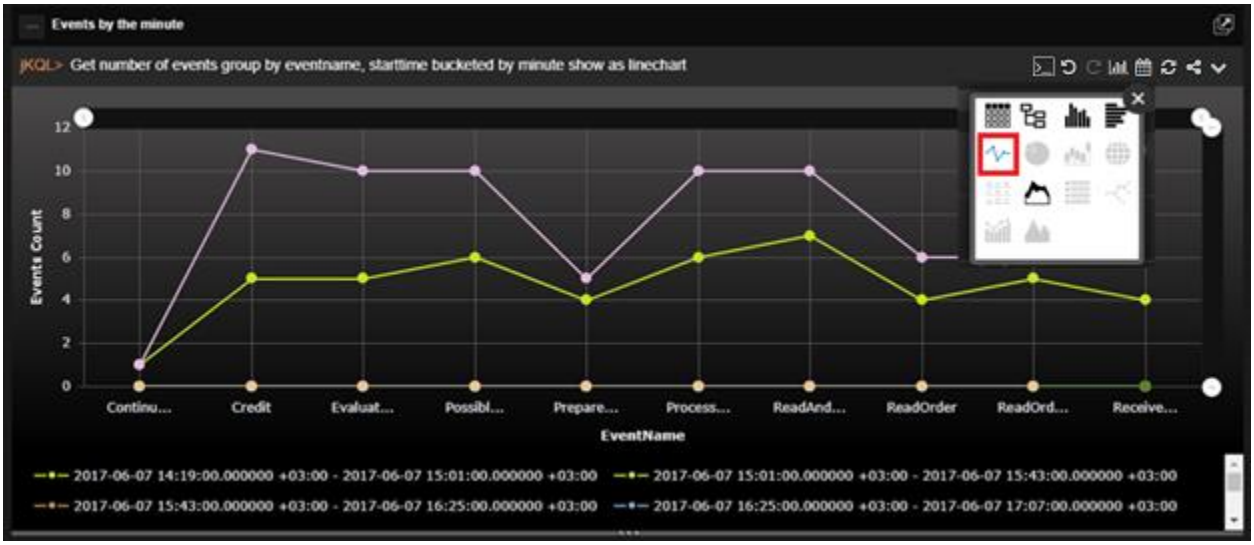


Figure 2.5.4.4-A. Line

Hover over the dots to view a status pop-up and exact axes values.

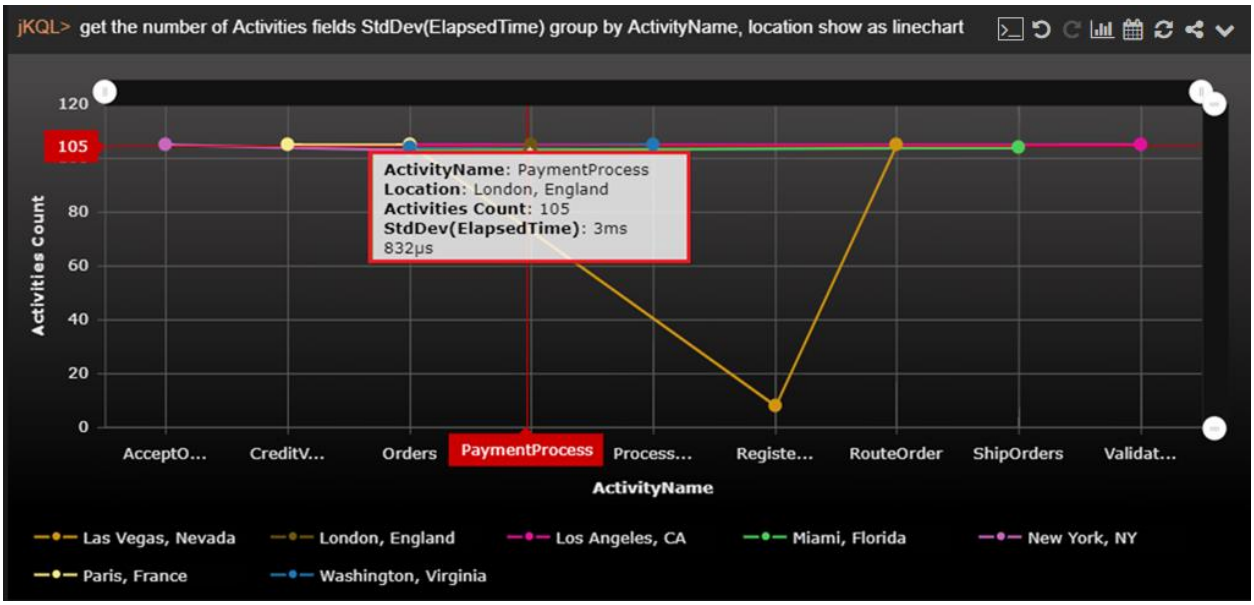


Figure 2.5.4.4-B. Status Pop-up

Click on any of the dots to view additional details in a **Console** viewlet. See *Section 2.4.4, Console Panel*, for more information.

2.5.4.4.1 Sample: Exponential Moving Average for ElapsedTime

Query: jKQL> `get events compute EMA(ElapsedTime, 20) show as linechart`

Figure 2.5.4.4.1-A. Sample Viewlet – Exponential Moving Average for ElapsedTime

The viewlet above is in the **Sample-OrderTracking** repository. An exponential moving average (EMA) is being computed to chart elapsed time over a window of time. EMAs are used with trends and enable one to see the rate of change between one data point and the next.

2.5.4.4.2 Sample: Events for Latest Time Range by Location

Query: jKQL> `Get the number of Events for the latest 4 years group by location show as linechart`

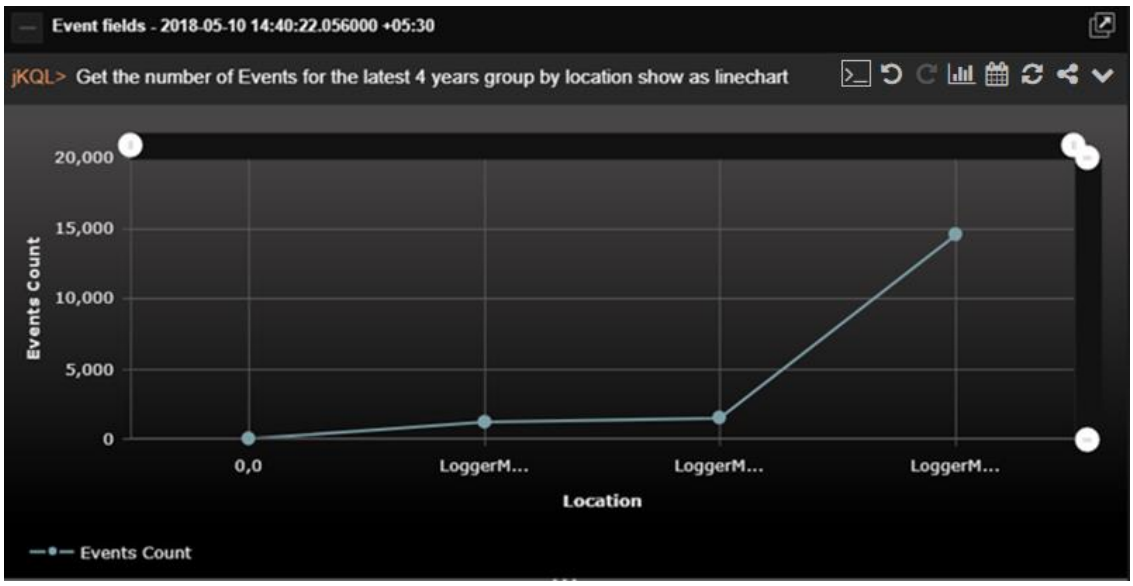


Figure 2.5.4.4.2-A. Sample Viewlet – Events for Latest 4 Years by Location

The viewlet above is in the **Sample-OrderTracking** repository. It is a line chart showing the trend in important event occurrences. Clicking on any of the "dots" or points will take the user to the **Console** where they can see additional details about each event. From there they can compare events or display the topology of an individual transaction.

2.5.4.5 Pie

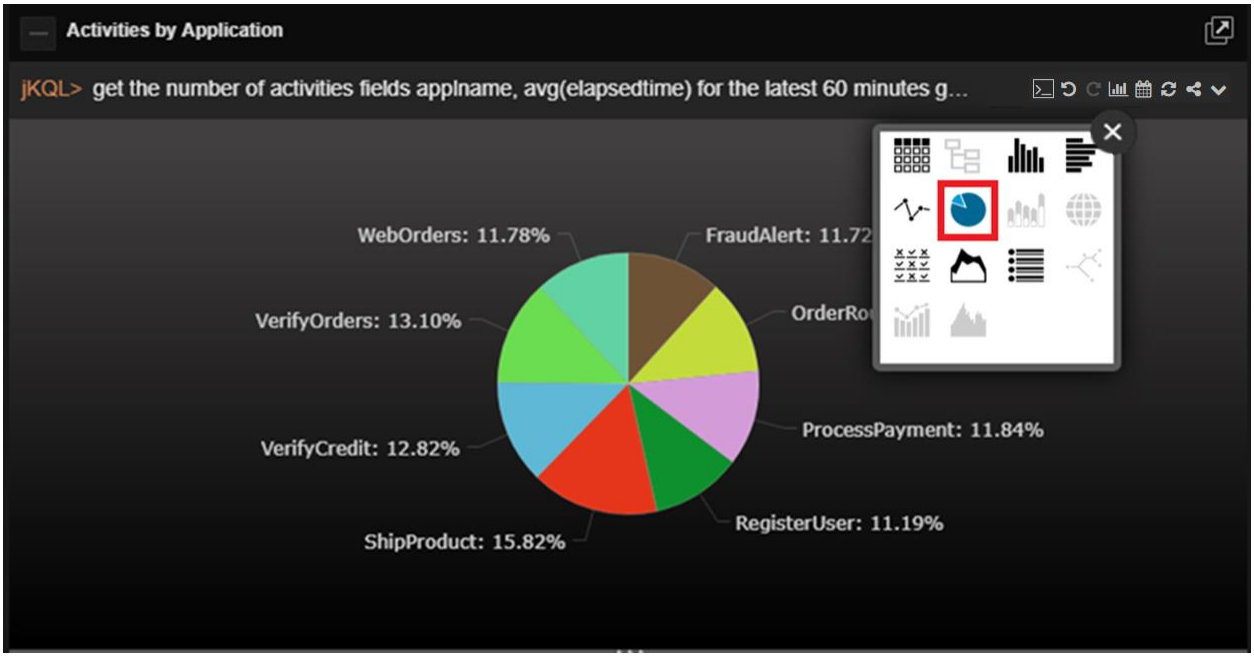


Figure 2.5.4.5-A. Pie

2.5.4.5.1 Sample: Serious Event Distribution

Query: **jKQL>** `Get the number of events for the latest hour where severity > 'WARNING' group by location, severity order by severity show as piechart`

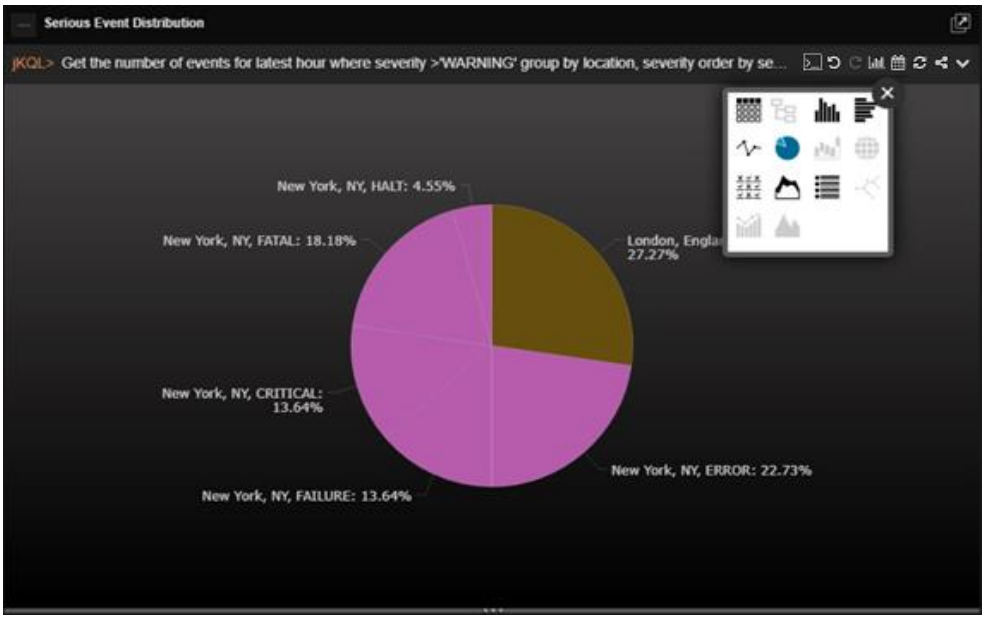


Figure 2.5.4.5.1-A. Sample Viewlet – Serious Event Distribution

The viewlet above is in the **Sample-OrderTracking** repository. It is a pie chart which is often used when counting something and you want to show the distribution of results for each member of a group or specifically severity in this case. This approach makes it easy to see where the biggest groups are that may need attention and further forensic analysis.

2.5.4.6 Stack

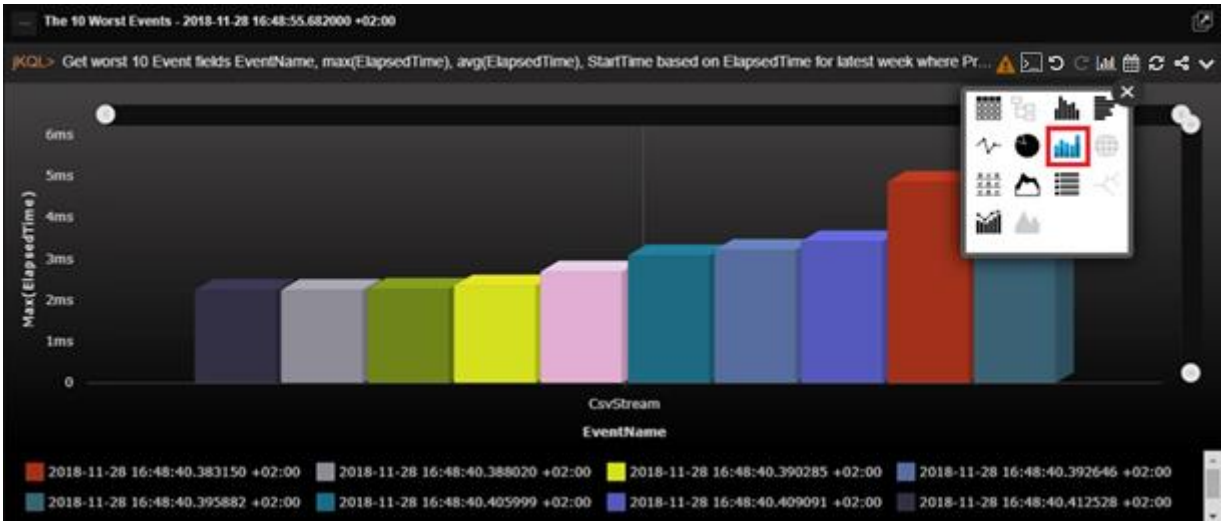


Figure 2.5.4.6-A. Stack

Stack charts must contain **Group By** expressions.

2.5.4.6.1 Sample: Orders for the Latest 3 Days that Missed their SLA

Query: jKQL> `Get the number of activities for the latest 3 days that did not meet 'SLA' group by location, activityname, severity, starttime bucketed by minute show as stackchart`

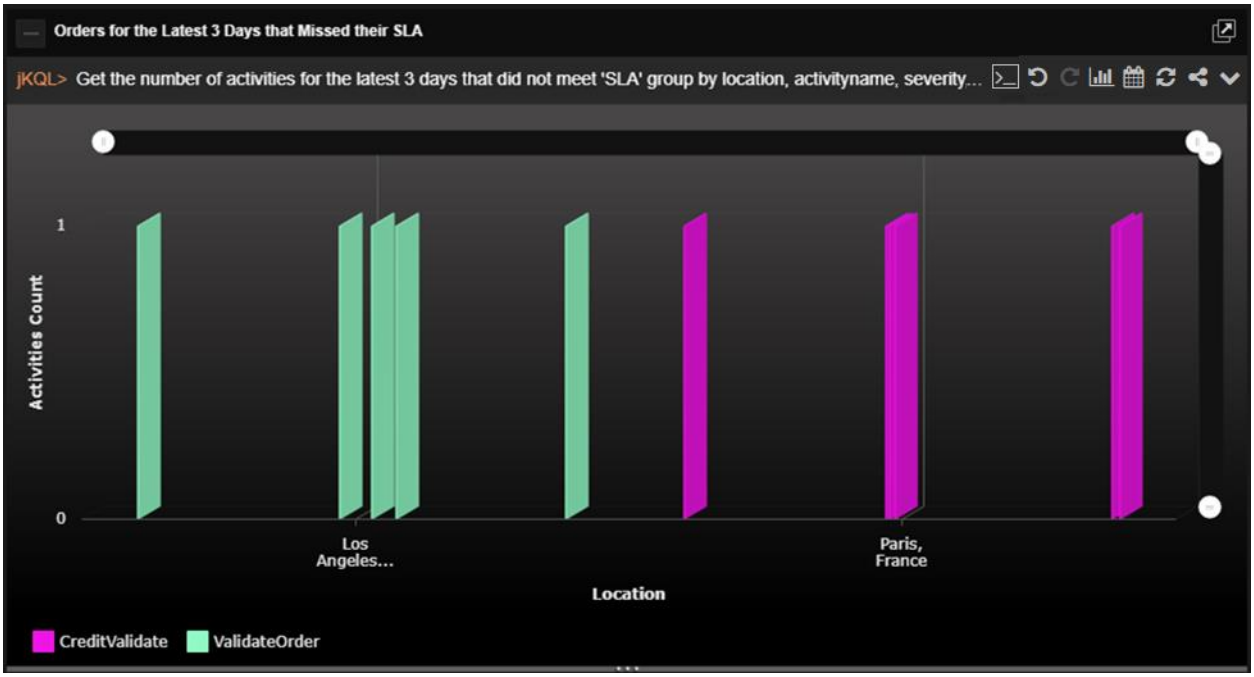


Figure 2.5.4.6.1-A. Sample Viewlet – Orders for the Latest 3 Days that Missed their SLA

The viewlet above is in the **Sample-OrderTracking** repository. It is searching for missed SLAs (service level agreements) and is presenting them in a stacked bar chart grouped by name, location, severity, and time.

Stacked bar charts are a powerful way to display a lot of data about the status of something in a very concise way.

2.5.4.7 Geo Map

Geo map viewlets are useful when transactions and operations between different countries or even continents need to be monitored and analyzed. Geo maps are used when location is important, and you want to first start with that, then drill down to specific applications when troubleshooting a problem.

Supported types for geo map viewlets are relatives and activities – select them while creating a viewlet with a form or specify them in a jKQL query line. If using activity data type, the viewlet must have the 'Group by GeoLocation' expression.

Below is an example of a geo map viewlet, which can be found in the **Sample-OrderTracking** repository (*Figure 2.5.4.7.1-A*).

2.5.4.7.1 Sample: Geo Map Events by Location

Query: jKQL> `Get relatives show as geomap`

Figure 2.5.4.7.1-A. Sample Viewlet – Geo Map Events by Location

The above viewlet displays the set of items within a geographic location. Each icon (push pin) represents a location (for example, United States) and the collection of all the entities such as applications, activities, events, and servers in that location. Each of the arrows shows a relationship between entities in one location with another. The dotted line shows a parent-child relationship (called enclosed) between the locations, while a solid line would represent an observation of an event in one location sending a message to an event in another (called send-to).

You can modify the jKQL query and get a geo map of activities where the data will focus on the perspective of agents.

2.5.4.7.2 Sample: Geo Map Activities

Query: jKQL> `Get activity group by geolocation show as geo map`

Figure 2.5.4.7.2-A. Sample Viewlet – Activities Geo Map

On the left side of the geo map viewlet, there are zoom in (+), zoom out (−) and fit to screen (⌂) buttons for better scope of the data.

On the right side of the viewlet there is a grey copy of the map. It is used as a navigation field to quickly change the map's focus.

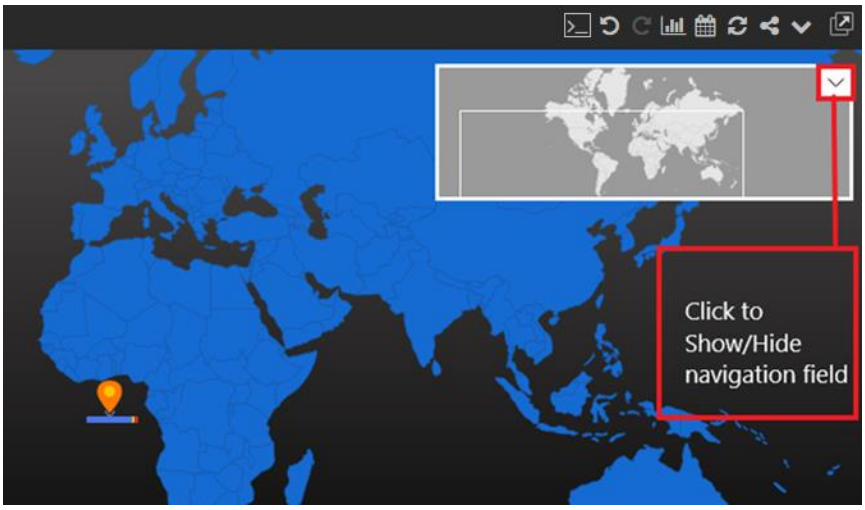


Figure 2.5.4.7.2-B. Geo Map – Navigation Field

After clicking on a specific location, the entire country will appear in light blue.

Figure 2.5.4.7.2-C. Selecting a Country

Click on the health bars above country names to drill into the data.

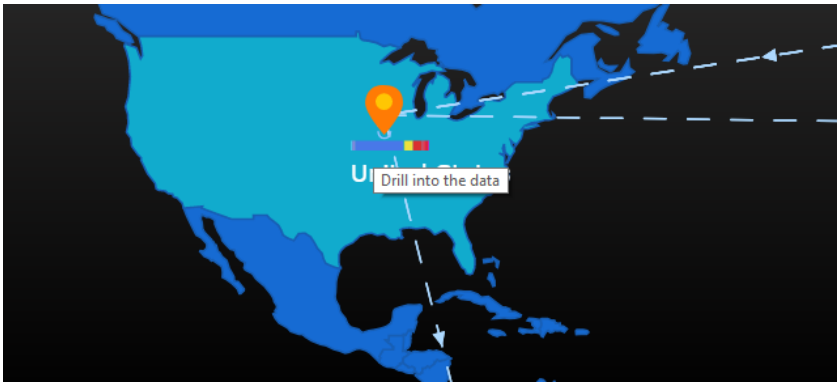
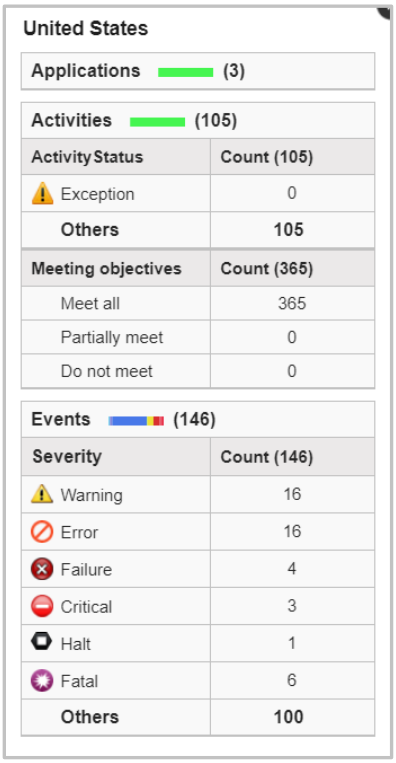


Figure 2.5.4.7.2-D. Country Health Bars

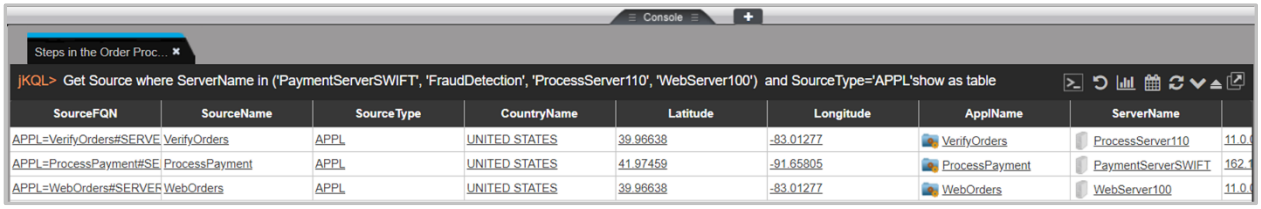
A status window opens. Click on any of the items to view additional details in a viewlet. The viewlet opens in the **Console** section.



United States

Applications (3)	
Activities (105)	
Activity Status	**Count (105)**
Exception	0
Others	**105**
Meeting objectives	**Count (365)**
Meet all	365
Partially meet	0
Do not meet	0
Events (146)	
Severity	**Count (146)**
Warning	16
Error	16
Failure	4
Critical	3
Halt	1
Fatal	6
Others	**100**

Figure 2.5.4.7.2-E. Status Window



Console

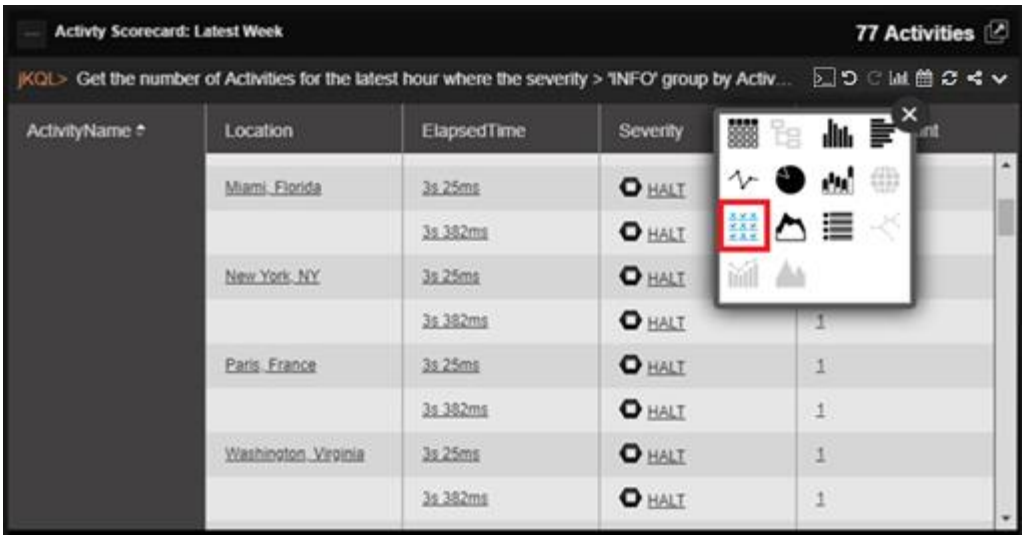
Steps in the Order Proc...

jKQL> Get Source where ServerName in ('PaymentServerSWIFT', 'FraudDetection', 'ProcessServer110', 'WebServer100') and SourceType='APPL'show as table

SourceFQN	SourceName	SourceType	CountryName	Latitude	Longitude	ApplName	ServerName	
APPL=VerifyOrders#SERVE	VerifyOrders	APPL	UNITED STATES	39.96638	-83.01277	VerifyOrders	ProcessServer110	11.0
APPL=ProcessPayment#SE	ProcessPayment	APPL	UNITED STATES	41.97459	-91.65805	ProcessPayment	PaymentServerSWIFT	162.1
APPL=WebOrders#SERVER	WebOrders	APPL	UNITED STATES	39.96638	-83.01277	WebOrders	WebServer100	11.0

Figure 2.5.4.7.2-F. Additional Details

2.5.4.8 Scorecard



Activty Scorecard: Latest Week — 77 Activities

jKQL> Get the number of Activities for the latest hour where the severity > 'INFO' group by Activ...

ActivityName	Location	ElapsedTime	Severity	
	Miami, Florida	3s 25ms	HALT	
		3s 382ms	HALT	
	New York, NY	3s 25ms	HALT	
		3s 382ms	HALT	1
	Paris, France	3s 25ms	HALT	1
		3s 382ms	HALT	1
	Washington, Virginia	3s 25ms	HALT	1
		3s 382ms	HALT	1

Figure 2.5.4.8-A. Scorecard

To create a scorecard viewlet, **Group by** must be used. Use within the jKQL query (see *Section 2.5.1.1, Create Viewlet with a jKQL Query*) or select within the Create / Edit viewlet form (see *Section 2.5.1.2, Create a Viewlet with a Form*).

2.5.4.8.1 Sample: Activity Scorecard Latest Week

Query: jKQL> `Get the number of Activities for the latest week where the severity > 'INFO' group by ActivityName, location, elapsedtime, severity order by ActivityName, severity desc show as scorecard`

Figure 2.5.4.8.1-A. Sample Viewlet – Activity Scorecard Latest Week

The viewlet above is in the **Sample-OrderTracking** repository. It is a Scorecard being used in this example to display details about activities that have an important severity (ones that need attention). The scorecard layout groups activity names in the first column and their details in the subsequent columns. Each row shows an additional instance of activities with the same name. Activity names are not unique. You can differentiate between one activity and another by referring to the activity ID for each one.

The line, **severity desc show as scorecard**, within the jKQL query sorts the results in descending order.

Scorecards are most often used as a grouping mechanism to see the status of a specific application or activity at a glance.

2.5.4.8.2 Sample: SLA Violation Scorecard

Query: jKQL> `Get the number of Activities for the latest week that did not meet the 'SLA' group ActivityName, location, elapsedtime order by ActivityName show as scorecard`

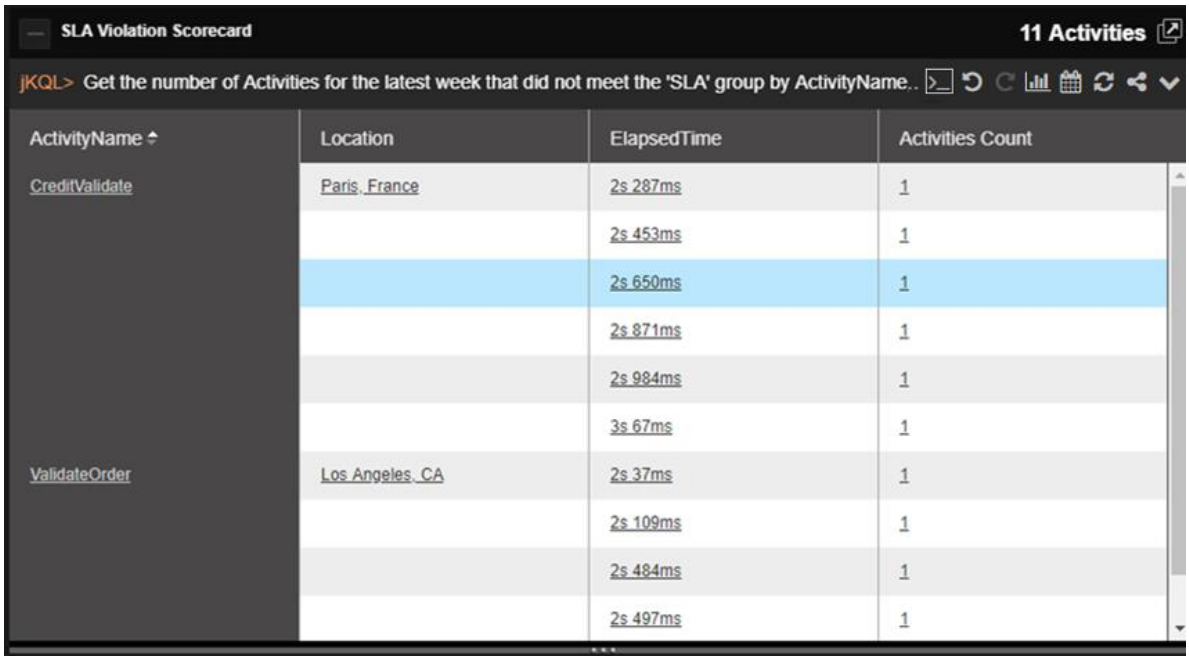


Figure 2.5.4.8.2-A. Sample Viewlet – SLA Violation Scorecard

The viewlet above is in the **Sample-OrderTracking** repository. It is a scorecard displaying SLA violations for each activity grouped by location.

2.5.4.8.3 Sample: Application Performance Index Analytics

Query: jKQL> Get activities fields Apdex(ElapsedTime, 3sec,4.5sec) group by ActivityName, location order by ActivityName show as scorecard

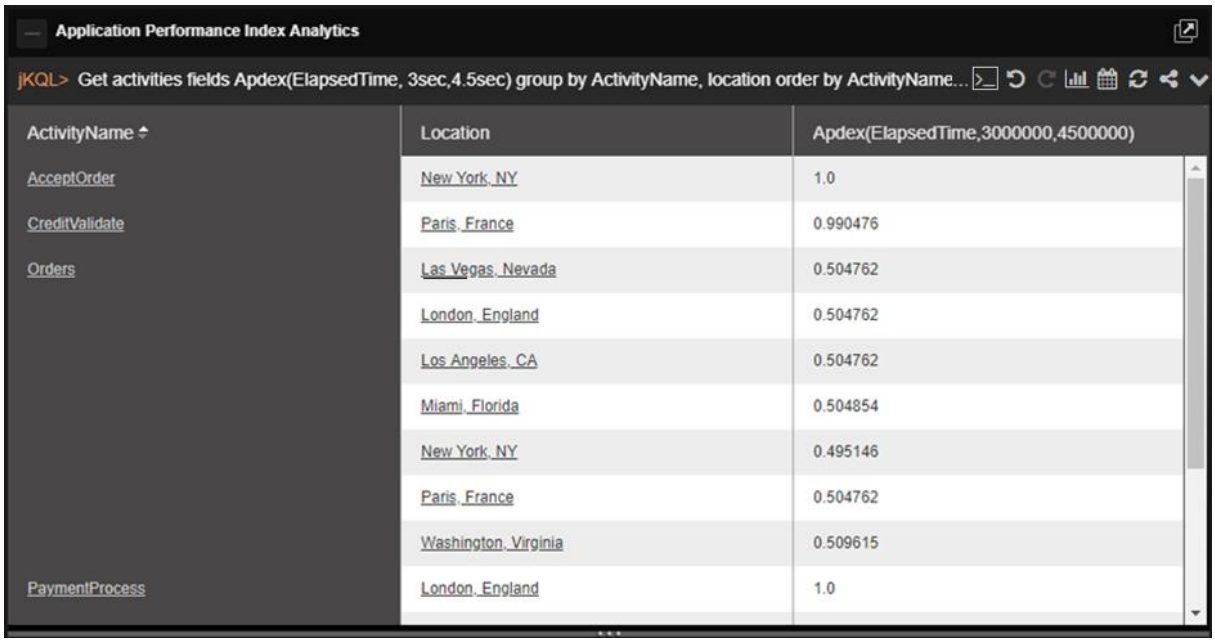


Figure 2.5.4.8.3-A. Sample Viewlet – Application Performance Index Analytics

The viewlet above is in the **Sample-OrderTracking** repository. It is using the statistical function Apdex. XRay comes with a large library of functions built into it including Bollinger bands, EMA, SMA, Floor, Median, Round, Standard Deviation, and many more. Apdex stands for application performance index. It defines a method for reporting and comparing the performance of software applications to measure user satisfaction.

Here it is used to determine the experience of users in each geographic area for each activity and its related applications. A "0" means no users are satisfied, while a "1" means all users are satisfied. A number in-between shows a mix of satisfaction levels. This is measured in relationship to the target elapsed time, in this case, between 3 to 4.5 seconds.

2.5.4.8.4 Sample: Function Analysis

Query: jKQL>
```
Get Activities fields StdDevPop(properties('OrderAmount')),
StdDevSample(properties('OrderAmount')),
VariancePop(properties('OrderAmount')),
VarianceSample(properties('OrderAmount')) for this year group by
props('COUNTRY_NAME') show as scorecard
```

Figure 2.5.4.8.4-A. Sample Viewlet – Function Analysis

TIP

The same query can be written without noting "properties," as in the example below (a simpler way of writing the query). The query will produce the same viewlet. See *Chapter 5: Using jKQL* for more information on jKQL queries.

Query: jKQL>
```
Get Activities fields StdDevPop(OrderAmount),
StdDevSample(OrderAmount), VariancePop(OrderAmount),
VarianceSample(OrderAmount) for this year group by COUNTRY_NAME show as
scorecard
```

Figure 2.5.4.8.4-B. Sample Viewlet – Function Analysis

The viewlet above is in the **Sample-OrderTracking** repository. It is an example of using standard deviation on the order amount field. Standard deviations are used to determine how far a value is from the expected value or mean and can illustrate the volatility of this value over time.

2.5.4.9 Area

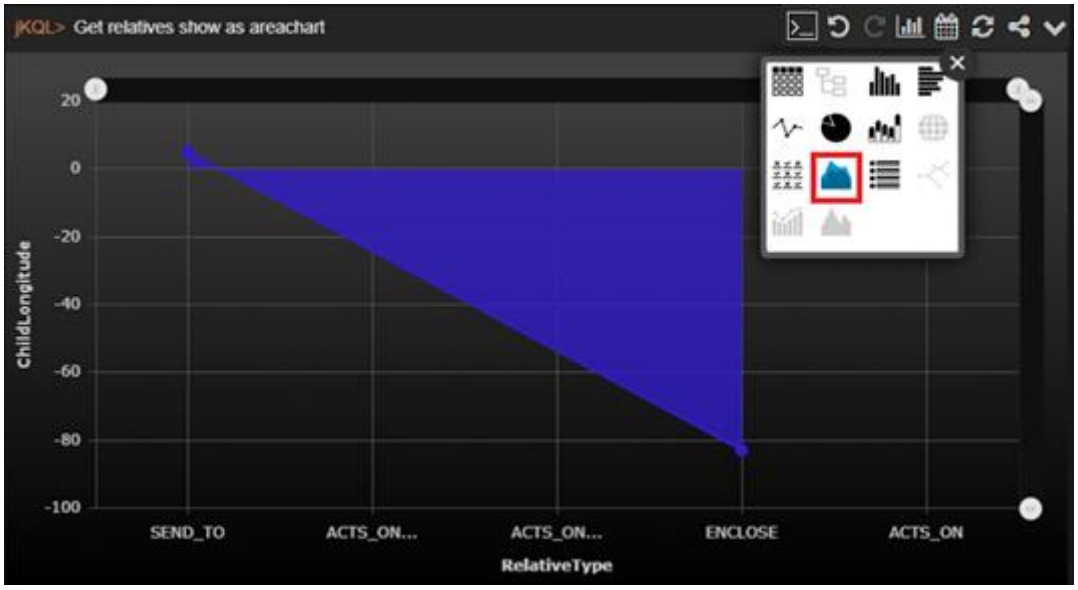


Figure 2.5.4.9-A. Area

Area charts are used to represent values over a specified period of time. The general tendencies of data changes or other items are visually represented. In the example below, the frequency of dpStatusCPUUsage snapshots (with defined word in snapshot name) from the previous 10 months is displayed.

2.5.4.9.1 Sample: CPU Usage

The viewlet below can be found on the **DataPower Metrics** dashboard of the **Sample-Middleware** repository.

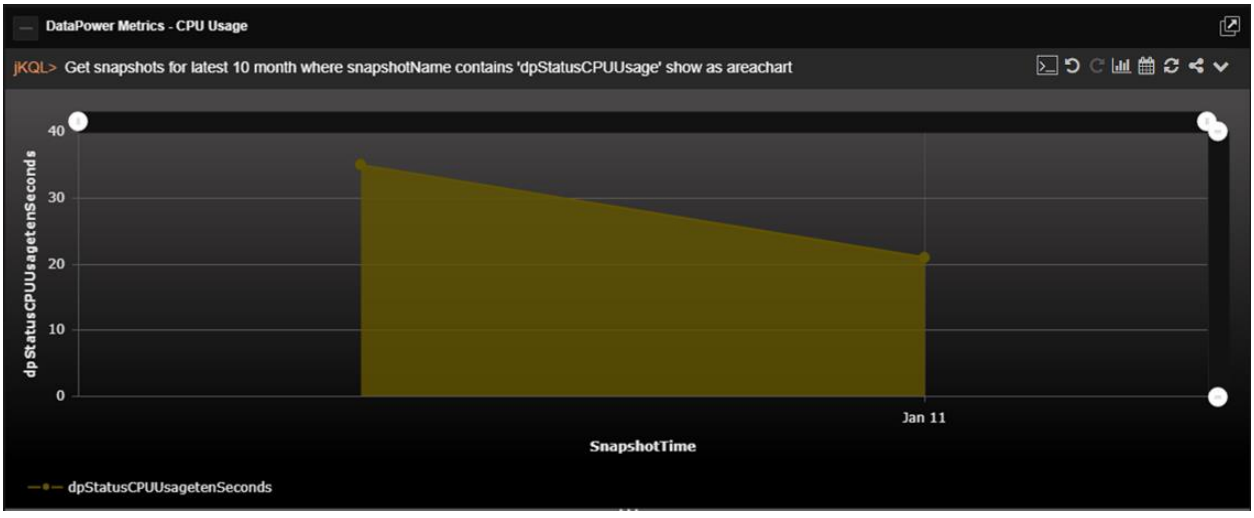


Figure 2.5.4.9-A. Area Chart – CPU Usage

Query: jKQL> `Get snapshots for latest 10 month where snapshotName contains 'dpStatusCPUUsage' show as areachart`

Hover over chart points to view details in a pop-up display, or click a point to view the details in a Console panel viewlet.

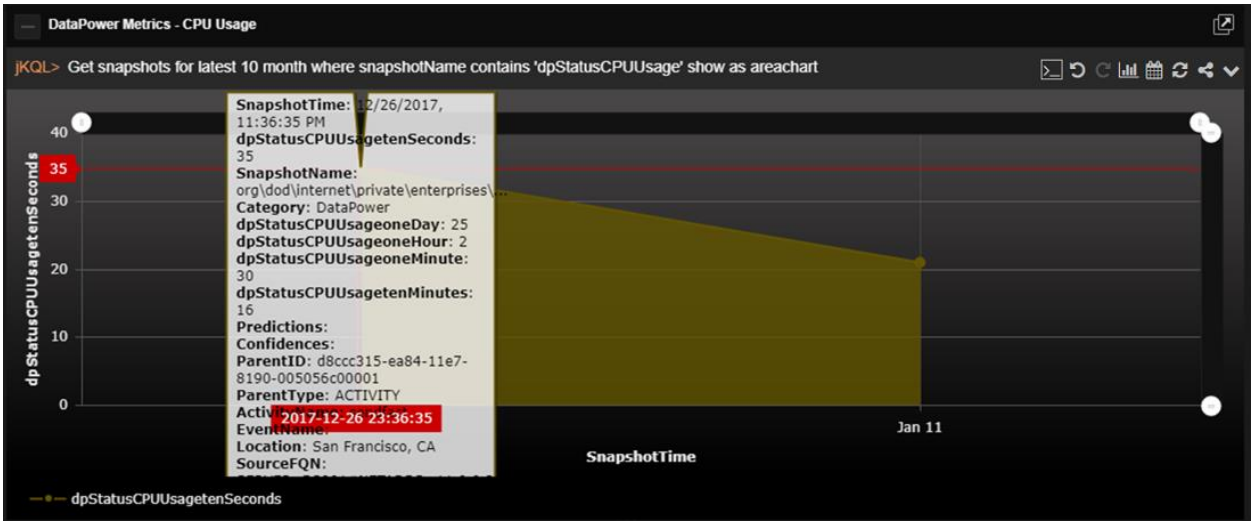


Figure 2.5.4.9-B. Area Chart – Details

2.5.4.10 Summary

Summary viewlets are used to quickly view various data totals of the repository. They are displayed in the Summary panel (*see* *Section 2.4.3, Summary Panel*).

2.5.4.10.1 Adding Summary Viewlets

2.5.4.10.1.1 Add Summary Viewlet from a Dashboard Viewlet

Add a new summary viewlet from a dashboard viewlet by clicking the **Change chart type** and selecting the **Summary** chart type.

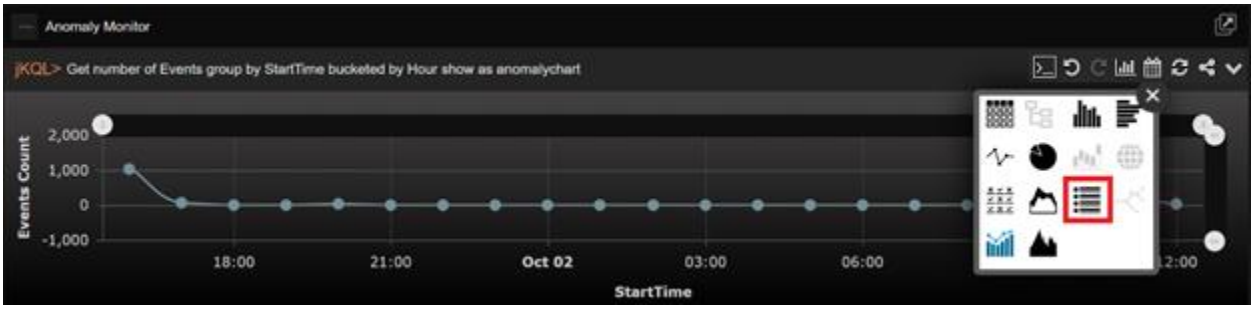


Figure 2.5.4.10.1.1-A. Add Summary from Viewlet

2.5.4.10.1.2 Add Summary Viewlet when Creating a Viewlet with a Form

When creating a viewlet using a form, users can specify to add a summary viewlet. See Create a Viewlet with a Form (*Section 2.5.1.2*) for more information.

Within the **Fields** section, enable the **Count** option. Depending on the data type, this option can be displayed as **Events Count**, **Activities Count**, or **Snapshots Count**.

When **Create** is clicked and the viewlet is generated, a summary viewlet will also be created.

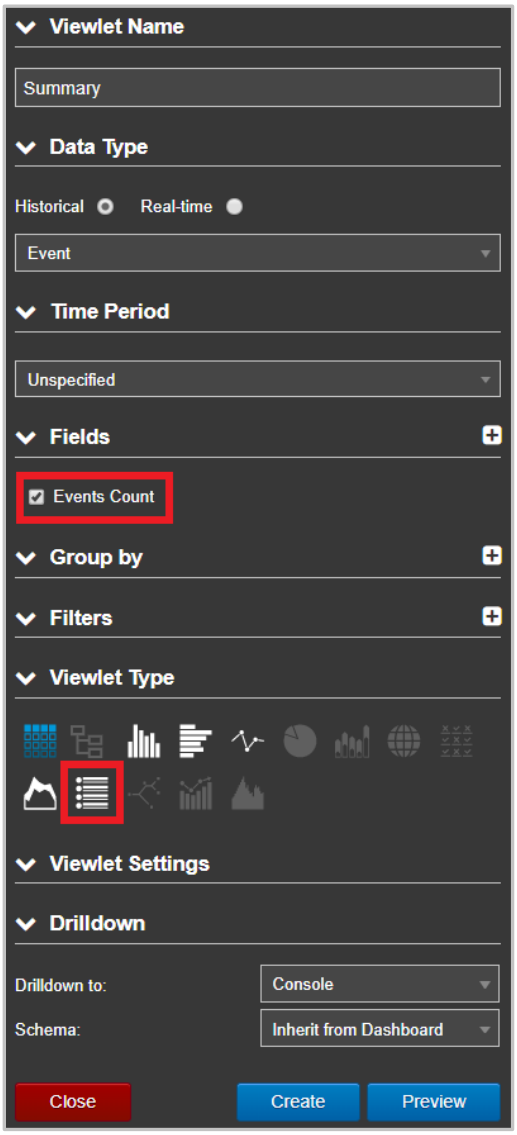


Figure 2.5.4.10.1.2-A. Count Option

2.5.4.10.1.3Add Summary Viewlets When Creating New Dashboards

When adding a new dashboard, enable **Generate Initial Viewlets** on the *Create new Dashboard* dialog box. Please see Create a Dashboard (*Section 2.4.2.1*) for more information on adding a new dashboard. Three default viewlets will be created: Activities Count, Events Count and Snapshots Count.

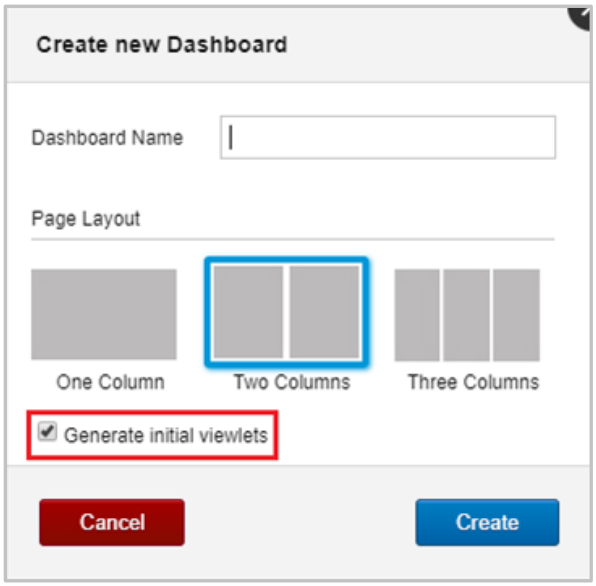


Figure 2.5.4.10.1.3-A. Create New Dashboard – Generate Initial Viewlets

2.5.4.10.1.4Add Summary Viewlet from the Create/Open Summary *Dialog Box*

1. Click the **Create Summary** icon 

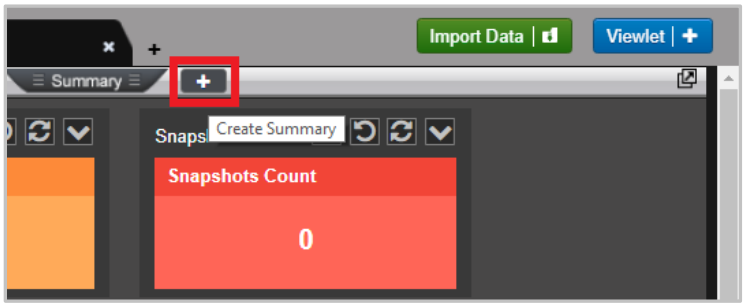


Figure 2.5.4.10.1.4-A. Create Summary Icon

2. The *Create/Open Summary* dialog box opens. An explanation of each option is explained below.

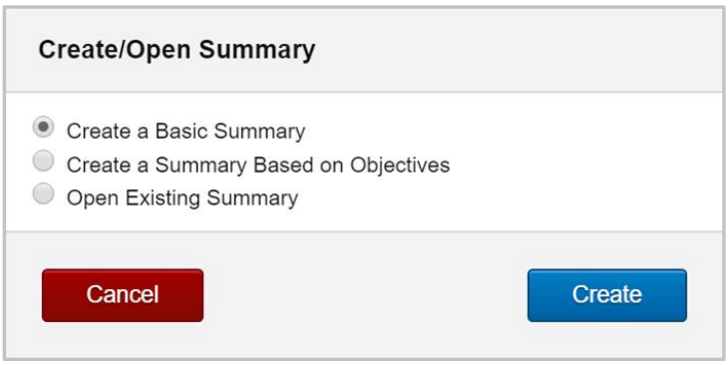


Figure 2.5.4.10.1.4-B. Create/Open Summary Dialog Box

Create a Basic Summary

Selecting the **Create a Basic Summary** option will open the *Create Summary* dialog box. Specify **Define Query** and **Viewlet Name**.

Figure 2.5.4.10.1.4.1-A. Create Summary Dialog Box

The summary viewlet is now created in the Summary panel.

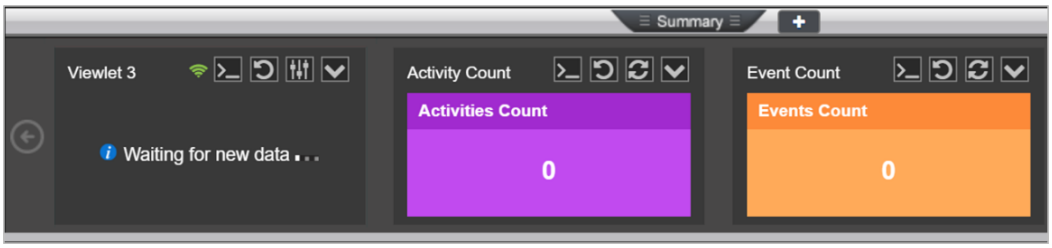


Figure 2.5.4.10.1.4.1-B. Summary Viewlet

Create a Summary Based on Objectives

Selecting **Create a Summary Based on Objectives** on the *Create/Open Summary* dialog box will open the *Wizard: Summary Based on Objectives* dialog box. Perform the following:

1. Select all desired objectives. Multiple objectives can be selected from the same set.

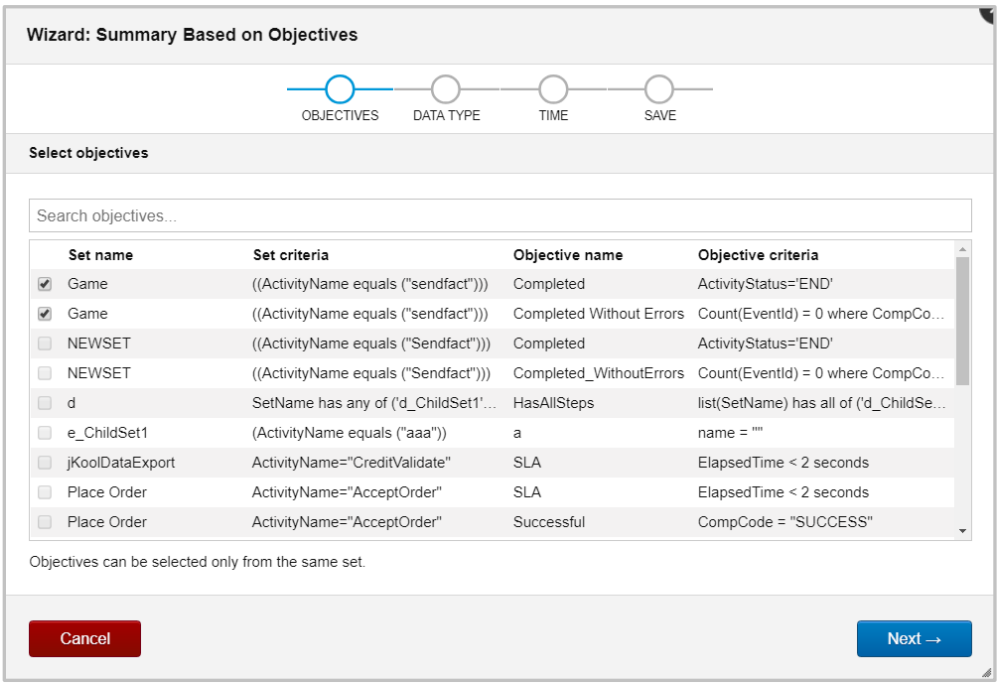


Figure 2.5.4.10.1.4.2-A. Wizard: Summary Based on Objectives – Objectives

2. Select either **Activities** or **Events** for the Data Type.

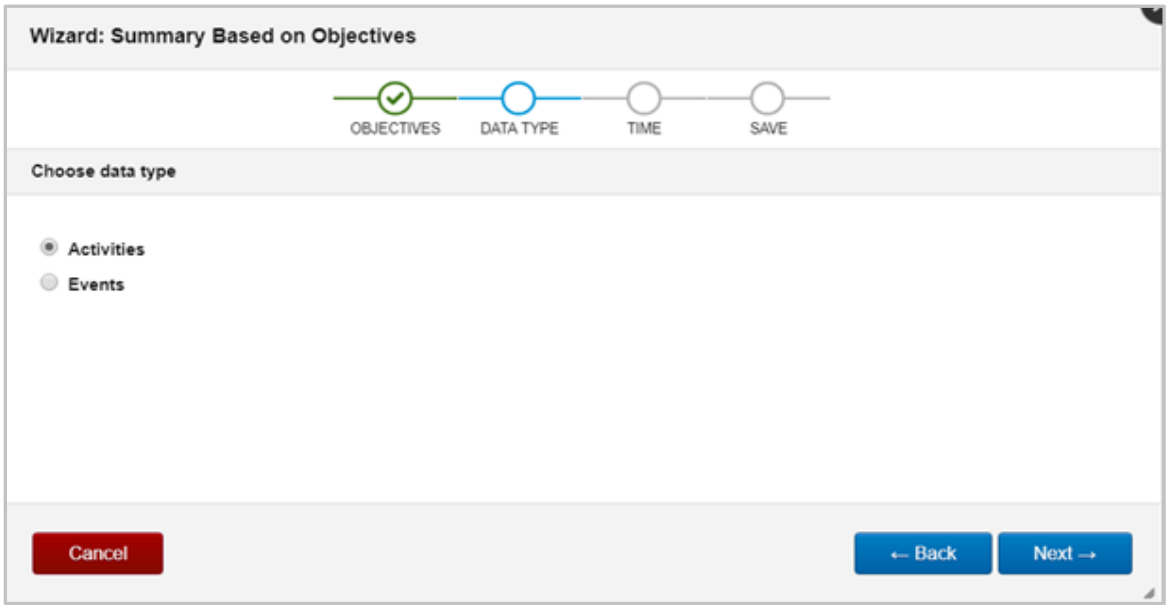


Figure 2.5.4.10.1.4.2-B. Wizard: Summary Based on Objectives – Data Type

3. Select a time limit from the drop-down menu.

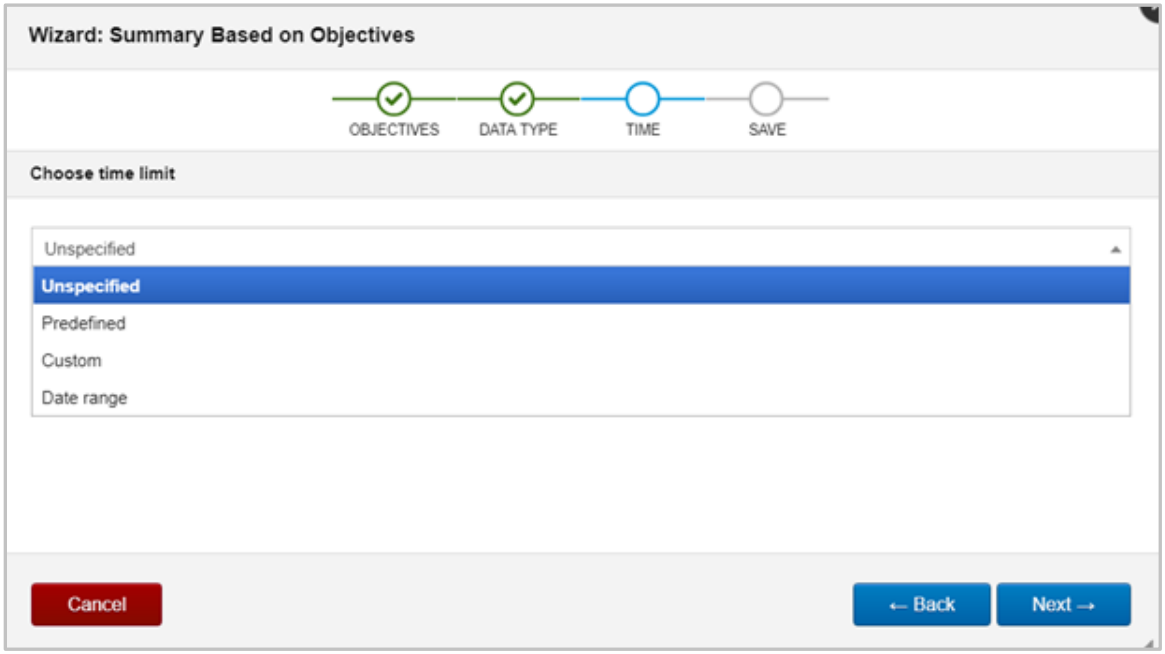


Figure 2.5.4.10.1.4.2-C. Wizard: Summary Based on Objectives – Time

4. Enter a name for the viewlet within the **Viewlet name** field. From the **Dashboard** drop-down, select which dashboard the new viewlet should be added to. Click **Save**.

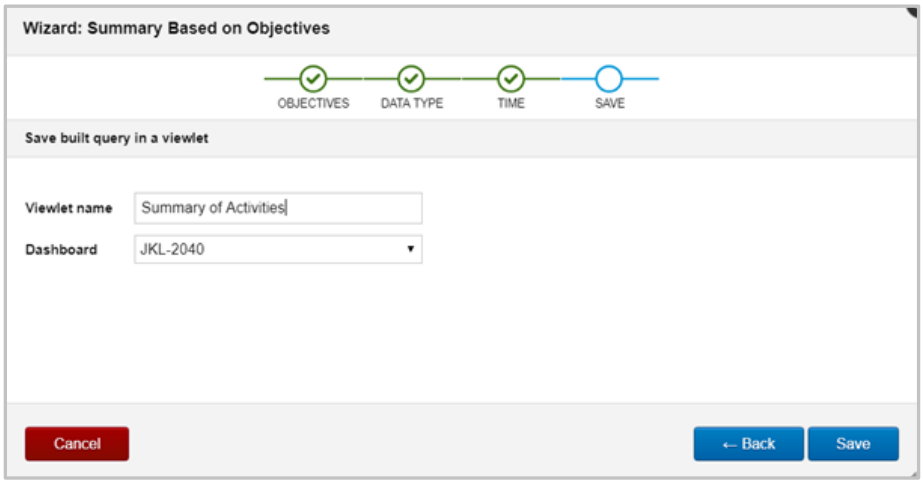


Figure 2.5.4.10.1.4.2-D. Wizard: Summary Based on Objectives – Save

The viewlet is now added to the Summary panel with three fields:

- **All**: Represents the count of activities that met the criteria of all selected objectives.
- **Partial**: Displays the count of activities that met the criteria of at least one of the selected objectives.
- **None**: The count of activities which did not meet any of the selected objectives' criteria.

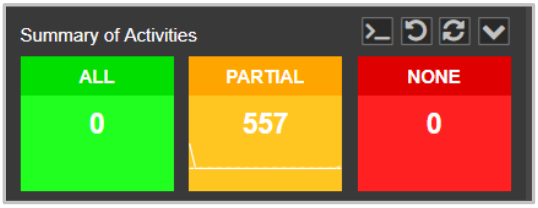


Figure 2.5.4.10.1.4.2-E. Summary Viewlet Based on Objectives

Open Existing Summary

Selecting **Open Existing Summary** on *Create/Open Summary* dialog box will open the *Open Existing Viewlet* dialog box (*Figure 2.5.1.4-A*). See *Section 2.5.1.4, Open Existing Viewlet* for more information on *Open Existing Viewlet*. Select a viewlet and click **Open**. The dashboard's focus will now be the selected viewlet.

The *Open Existing Viewlet* dialog box can also be opened from the **Main Menu** (*Section 2.3.6*).

2.5.4.10.2Tear Off Viewlets

The Summary Dock can be opened in its own browser window. Click the **Tear Off** button to view the summary viewlets in their own window. See *Tear Off* for more information.

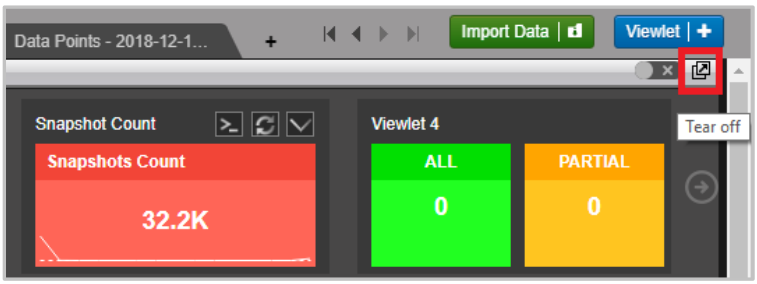


Figure 2.5.4.10.2-A. Summary – Tear Off

2.5.4.10.3Exact Total

Hover over a count to view the exact total.

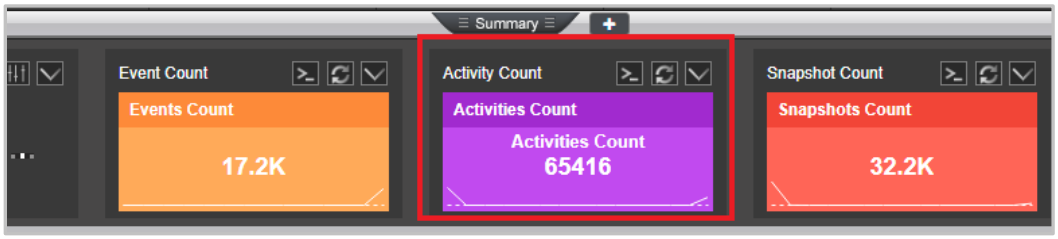


Figure 2.5.4.10.3-A. Summary – Exact Total

2.5.4.10.4Edit Viewlet Query

Click the **Edit Query** button to update the viewlet's query.

Figure 2.5.4.10.4-A. Summary Viewlet - Edit Query Button

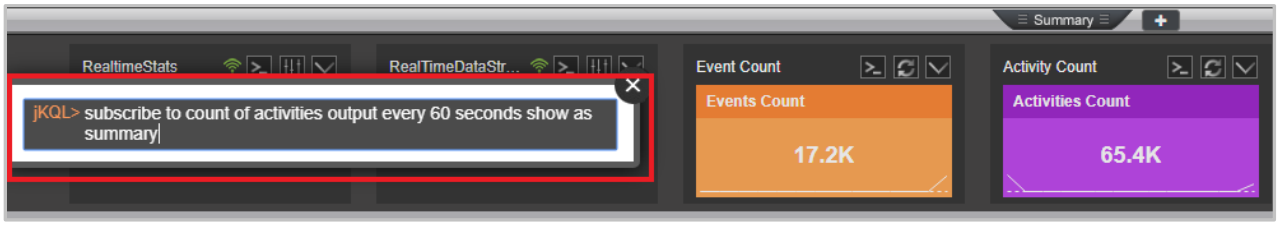


Figure 2.5.4.10.4-B. Summary Viewlet – Edit Query

2.5.4.10.5Reset Query

Click the **Reset Query** button to reset a viewlet's jKQL query to the last saved query.

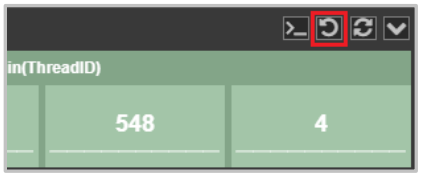


Figure 2.5.4.10.5-A. Reset Query

2.5.4.10.6Refresh Viewlets

To refresh summary viewlets, click the **Refresh Viewlet** button. This process will check for new data.

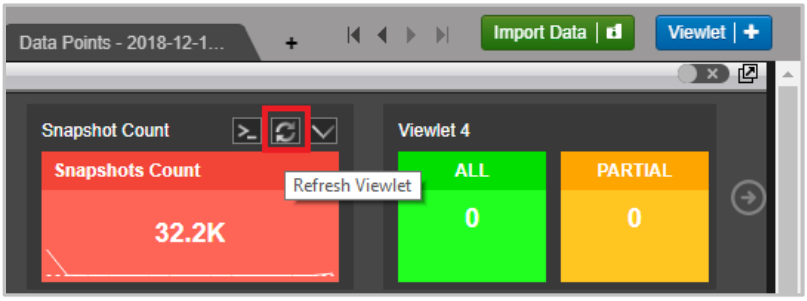


Figure 2.5.4.10.6-A. Refresh Viewlet Button

2.5.4.10.7Viewlet Menu

The **Summary Viewlet Menu** button allows users to edit, save, save as, remove, or delete summary viewlets.

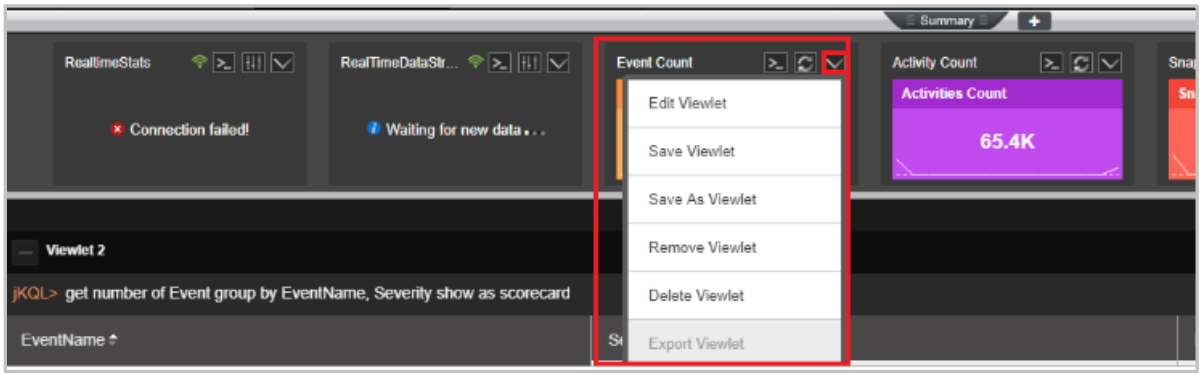


Figure 2.5.4.10.7-A. Summary Viewlet Menu

2.5.4.10.7.1 Edit Viewlet

Selecting **Edit Viewlet** allows users to update the summary viewlet's details using a form. After making updates, click **Preview** to view changes before saving. To cancel and discard changes, click **Close**. To save changes made, click **Apply**.

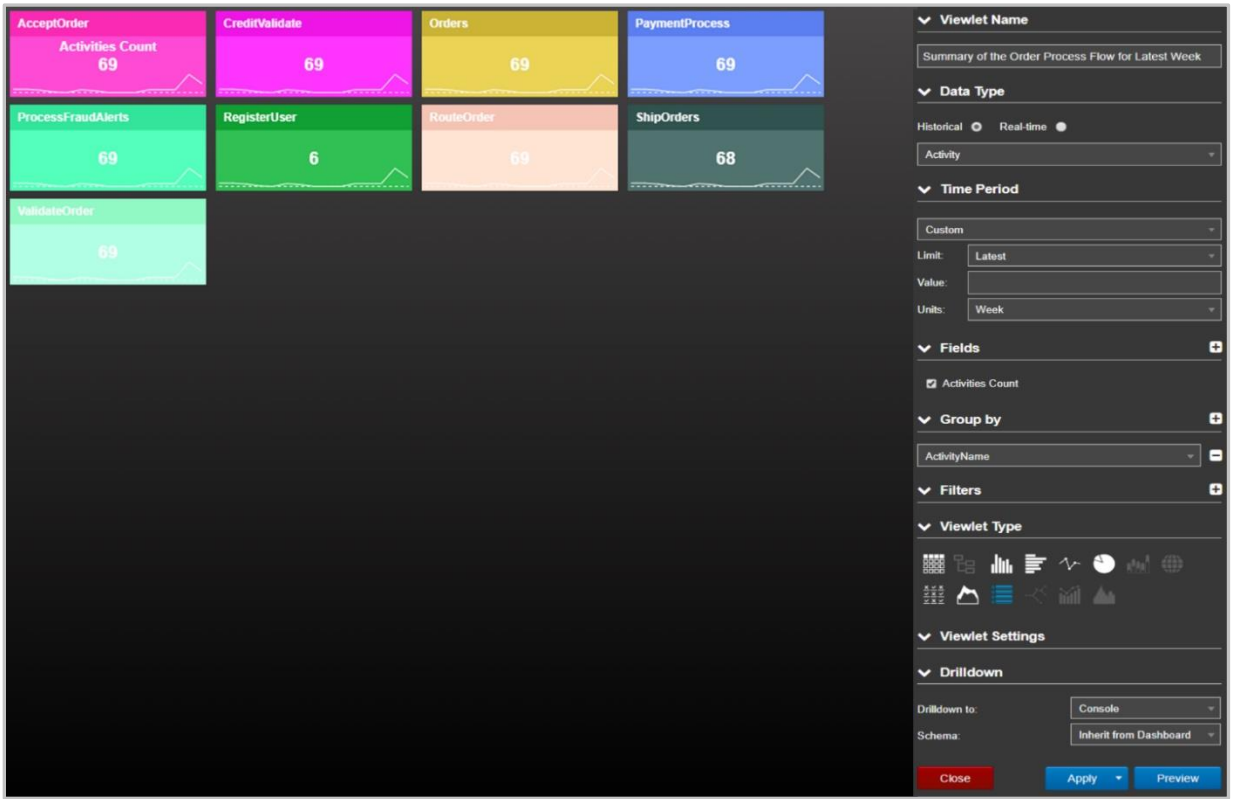


Figure 2.5.4.10.7.1-A. Edit Summary Viewlet

2.5.4.10.8 Real-time Subscription Viewlet Options

For real-time subscription summary viewlets, use the **Real Time Menu** button to set an interval and pause/resume the viewlet data.

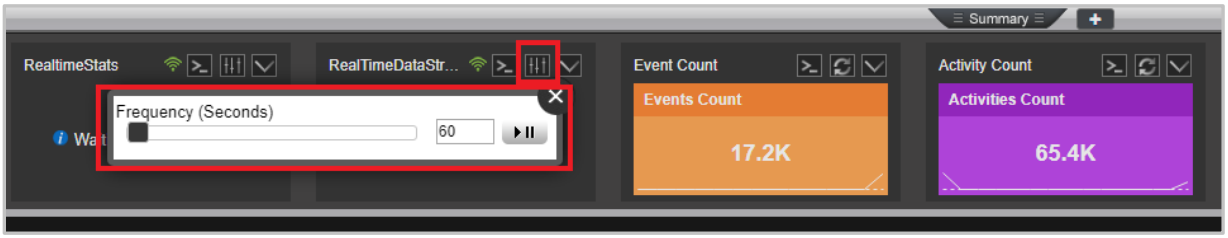


Figure 2.5.4.10.8-A. Real Time Menu Button

2.5.4.10.9Visual History of Changes

The Summary viewlets include line charts displaying increase/decrease count history as seen in the below figure.

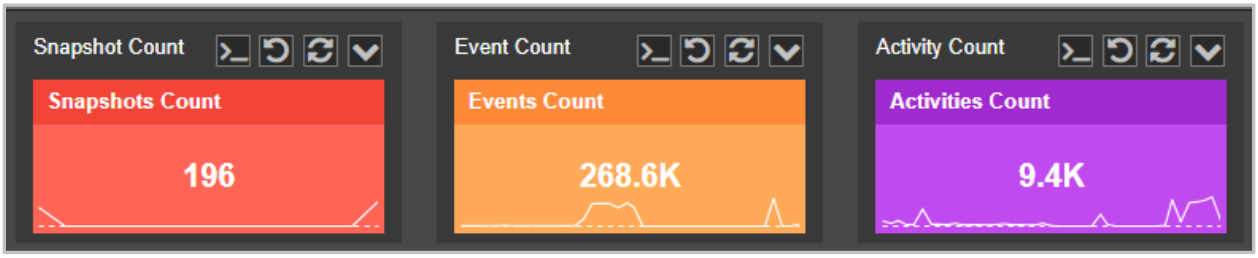


Figure 2.5.4.10.9-A. Count History Chart

2.5.4.11 Topology

Figure 2.5.4.11-A. Topology

A topology is often used to see the "flow" of what happened, and when it happened. This is very helpful in understanding the status of your applications and objectives.

> NOTE
>
> Do not include the @ symbol in Resource names. The @ symbol is reserved in jKQL to identify servers. Using it can cause resource names to be truncated in Topology viewlets.

2.5.4.11.1Sample: Steps in the Order Process Business Milestone

Query: jKQL> `get relatives show as topology`

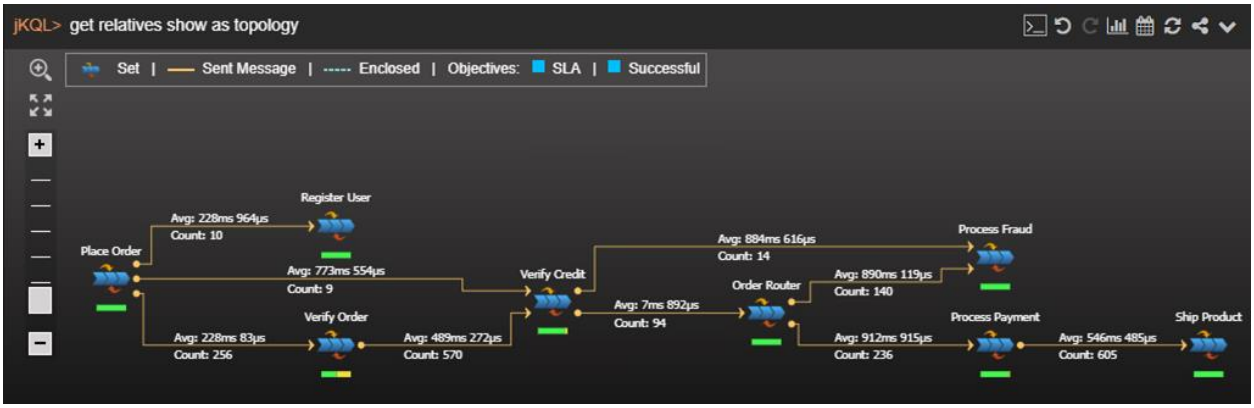


Figure 2.5.4.11.1-A. Sample Viewlet – Steps in the Order Process Business Milestone

The viewlet above is in the **Sample-OrderTracking** repository. It shows the auto-discovered topology of an Order Process, displayed at the business milestone tier. Topologies can be shown at the geographical, datacenter, server, application, or milestone tiers. Each of the blue "chevron"-like icons above represents a specific business milestone. A business milestone is there to represent the completion of a business objective in the "real world." It is defined based on established criteria, while its completion determines its status. Milestones often form a sequence or flow. This happens automatically as the analytics engine determines an observed relationship between them. The colored bars underneath each icon are called a. The health bar under each icon is color coded to reflect status (green = good, yellow = warning, red = critical). It can be clicked to see the status of the milestone. The arrows between icons show data flow between milestones. This is automatically discovered. The numbers surrounding the arrow show statistics for the relationship including elapsed time and count.

2.5.4.11.2Create / Edit Topology Viewlet

To populate *Topology* viewlets, the 'Get relatives <*any criteria*> show as topology' statement must be used. If you're creating a topology with a form, select **Relative** as the data type.

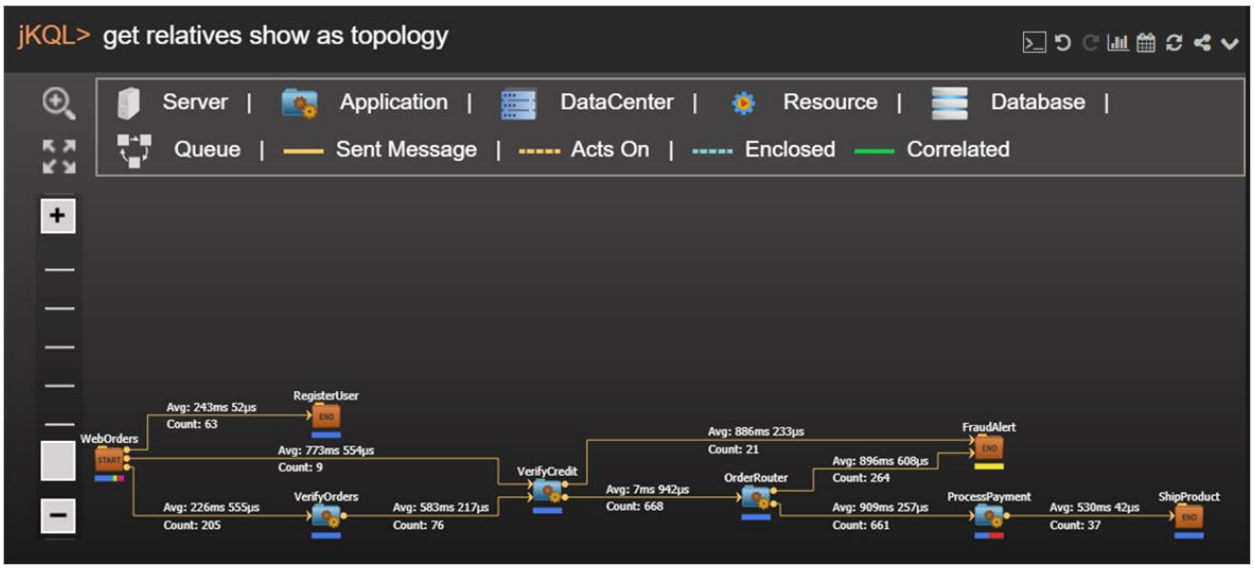


Figure 2.5.4.11.2-A. Topology

To modify the *Topology* viewlet, click the down arrow and select **Edit Viewlet** from the viewlet's drop-down menu. The viewlet's editing form opens.

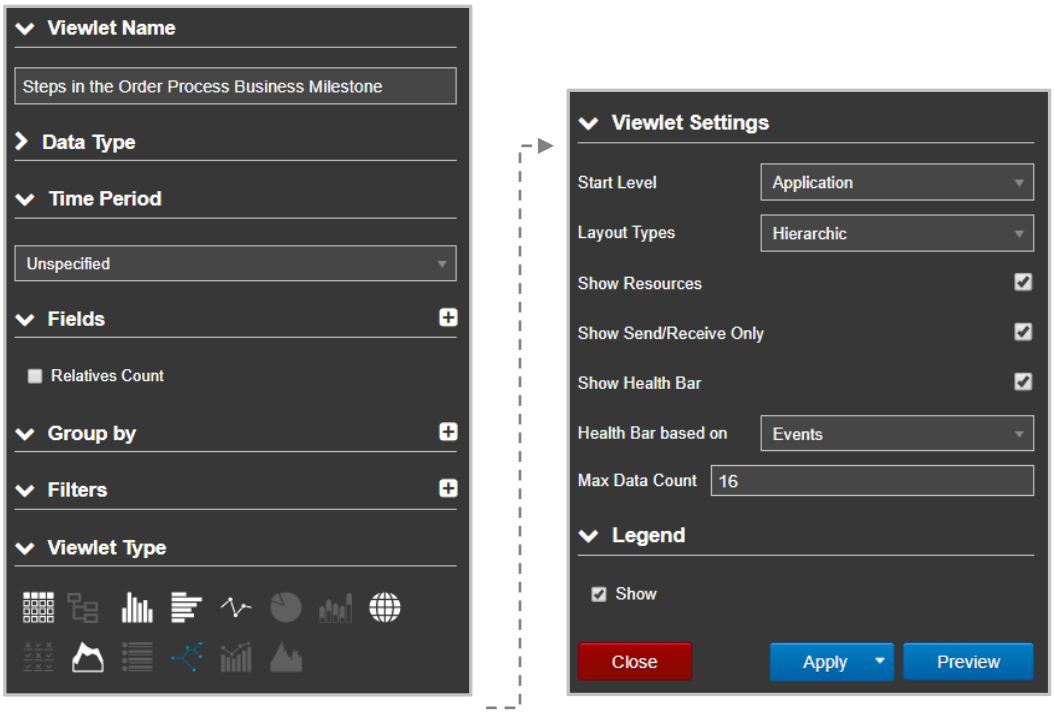


Figure 2.5.4.11.2-B. Edit Topology Viewlet Form

Select desired options from the drop-down menus. Check off the **Legend** check box to enable/disable the displaying of legend icons. The legend is displayed at the top of topology viewlets.

In the **Viewlet Settings** section, you can modify the properties of the topology viewlet. From the **Start Level** drop-down menu, select the item type.

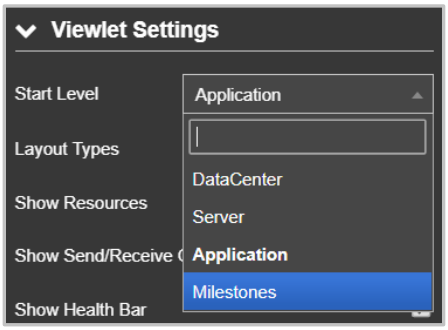


Figure 2.5.4.11.2-C. Start Level Drop-Down Menu

Depending on the start level item type selected, the appearance of the viewlet will differ. In the example below, **Server** was selected to be the start level.

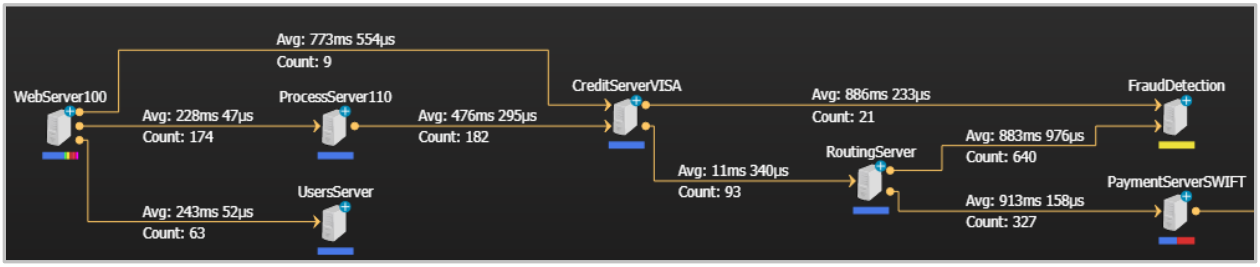


Figure 2.5.4.11.2-D. Topology Viewlet with Server as Start Level

Select a layout type from the **Layout Types** drop-down menu.

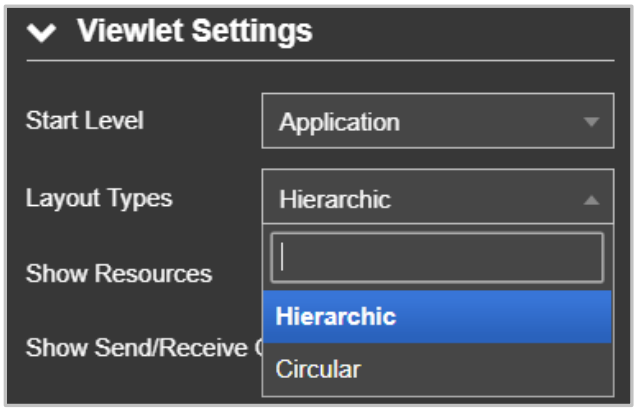


Figure 2.5.4.11.2-E. Layout Types Drop-Down Menu

The topology figures above display the hierarchic layout type. Below is an example of the circular layout type.

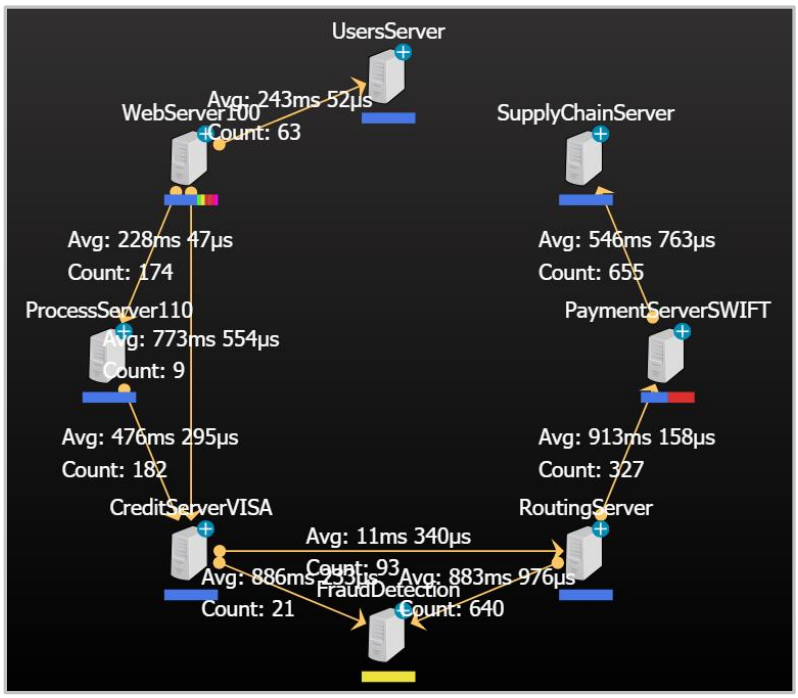


Figure 2.5.4.11.2-F. Circular Layout Type

Select all other desired options within the **Viewlet Settings** section.

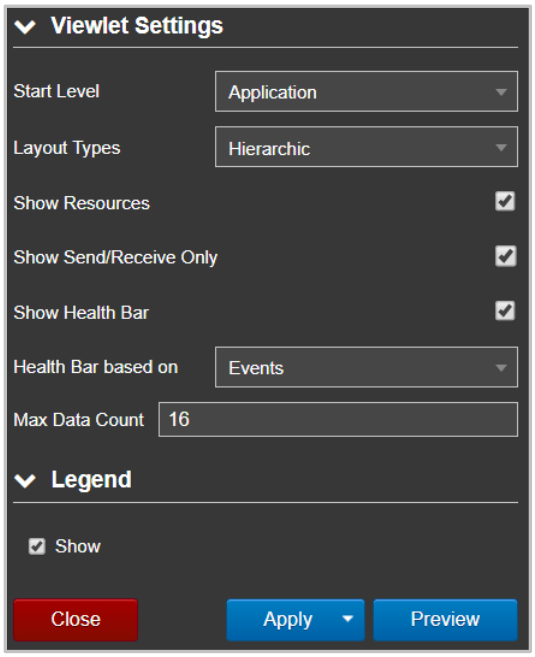


Figure 2.5.4.11.2-G. Circular Layout Type

Click the **Preview** button to view the topology viewlet's updates. Click the **Apply** button to save the changes. The **Close** button will close the form without saving changes.

2.5.4.11.3Topology Viewlet Properties

Topology viewlets can be displayed within the main view or on the Console panel (when generated from a table viewlet, see *Section 2.5.4.1.1.5, Topology*, for information), but the properties of the viewlets are the same in both cases.

The arrows represent the relationships between relatives. Click an arrow to view statistics.

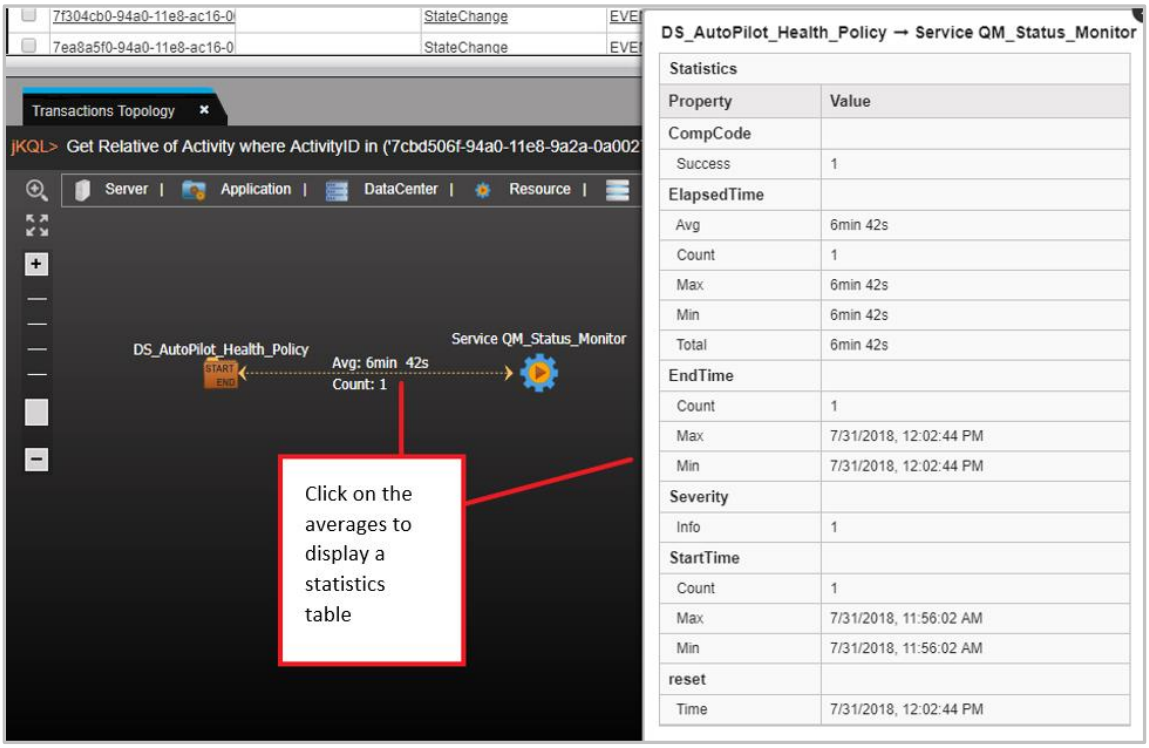


Figure 2.5.4.11-B. Topology - Statistics Chart

Users have the following additional options to customize topology viewlets.

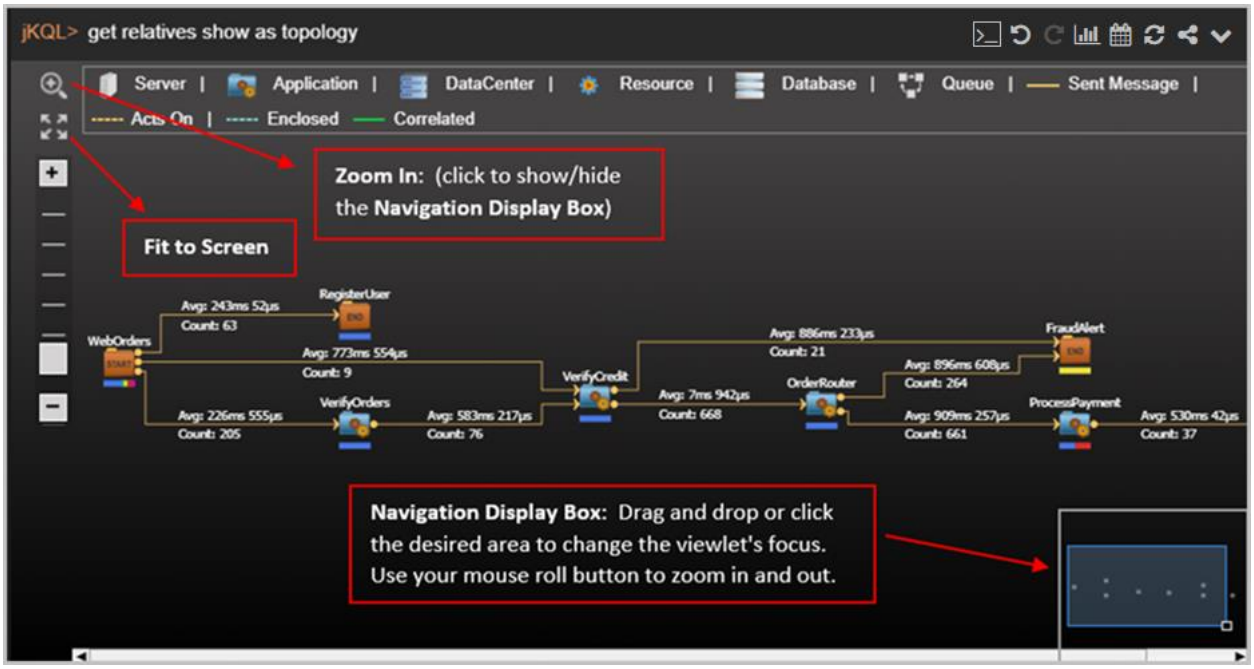


Figure 2.5.4.11.3-B. Topology – Options

2.5.4.12 Anomaly

Anomaly chart viewlets are useful to quickly see data distribution deviations compared to the normal distribution.

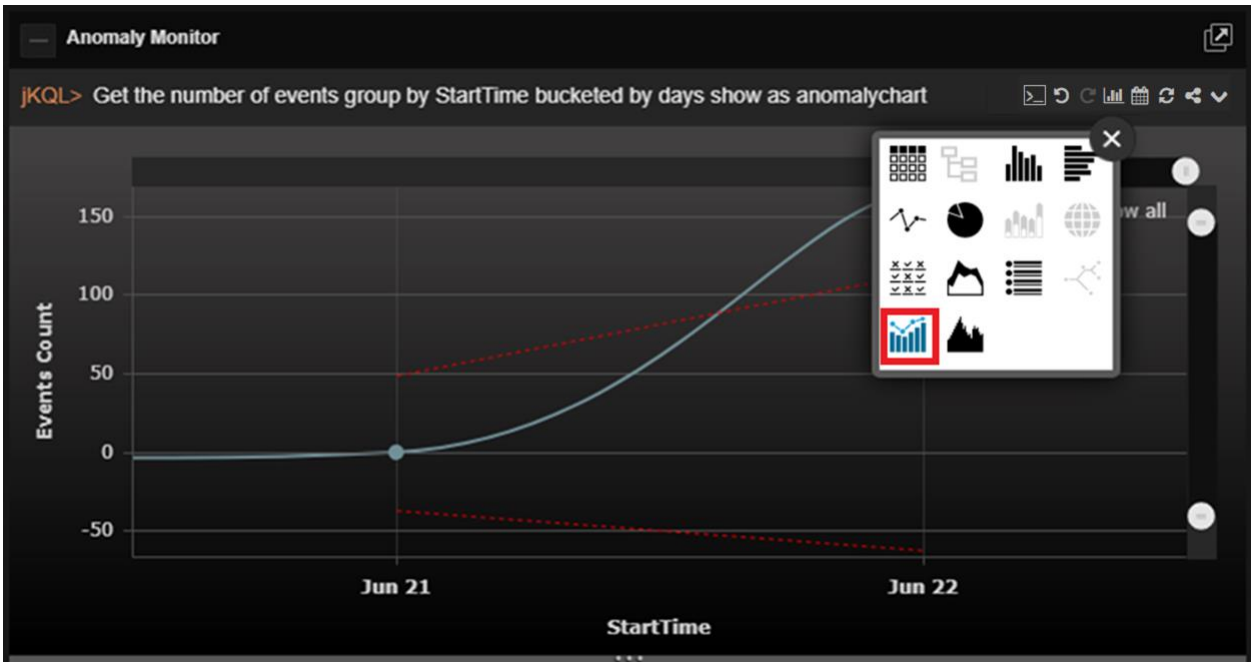


Figure 2.5.4.12-A. Anomaly

2.5.4.12.1Sample: Anomalies via Bollinger Bands

Query: jKQL> `Get number of events group by starttime bucketed by minute show as anomalychart`

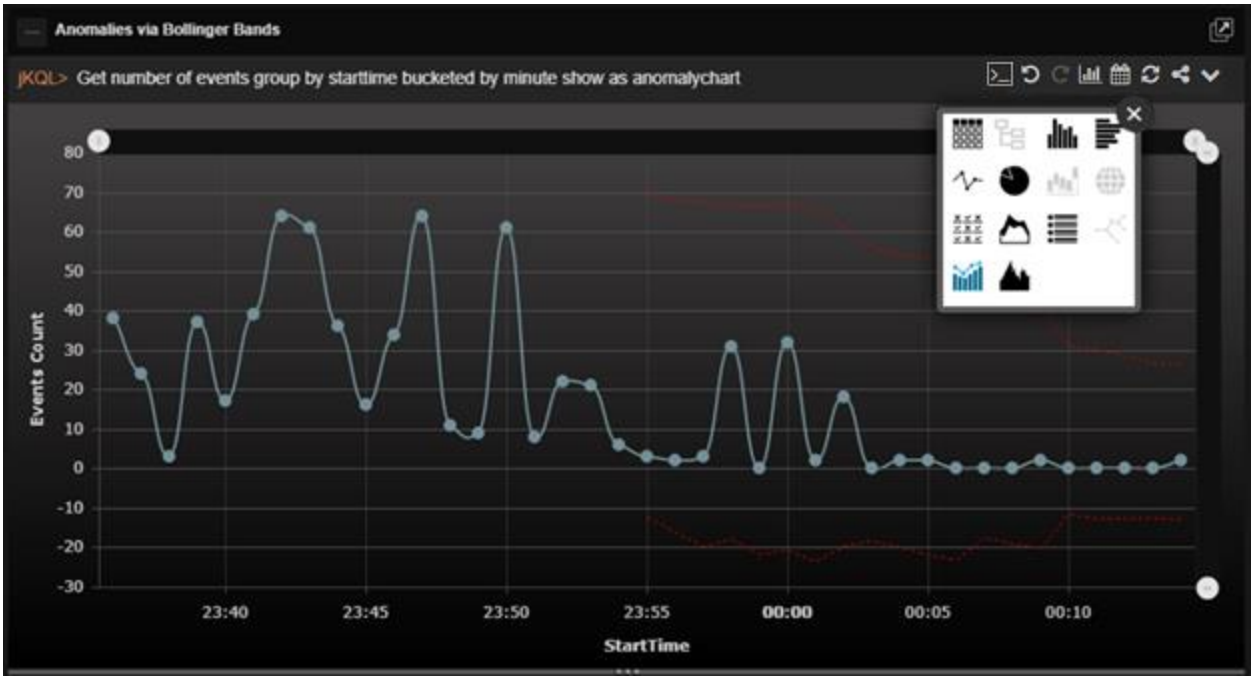


Figure 2.5.4.12.1-A. Sample Viewlet – Anomaly Monitor

The viewlet above is in the **Sample-OrderTracking** repository. This viewlet, which is called an anomaly chart, is using the function Bollinger Bands to automatically detect anomalies

in the number of events per day. The red dashed line displays the average event count, and the blue displays the actual event count. The point at which the blue line surpasses the red dashed line is when the anomaly is suspected.

2.5.4.13 Histogram

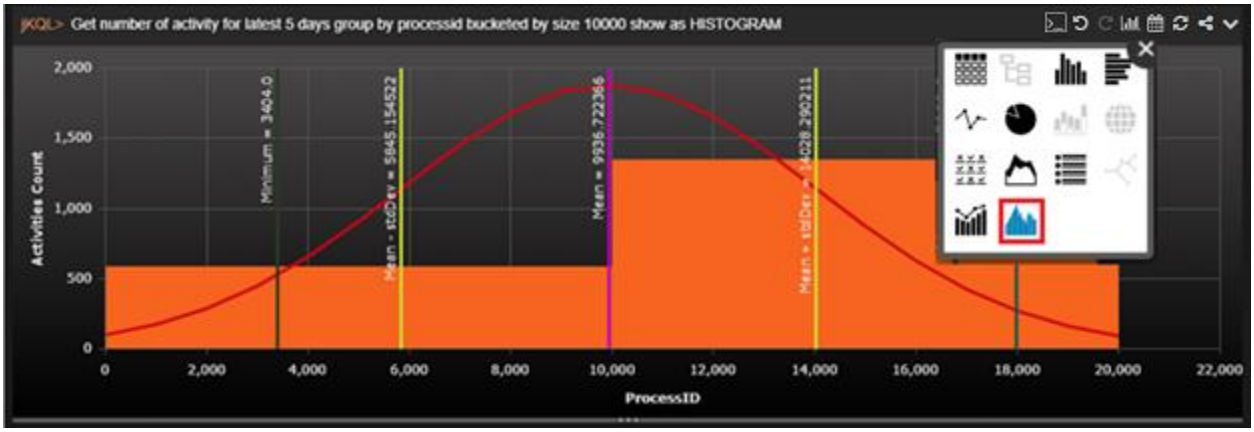


Figure 2.5.4.13-A. Histogram

A **histogram** represents the distribution of numerical data. To create a histogram, the frequency of data within a range of values will need to be bucketed into intervals. Opposed to bar charts, histograms display the frequency of continuous data. The red line displays the normal distribution. The **Mean - stdDev** (stdDev – standard deviation), **Mean** and **Mean + stdDev** lines display statistical means. The **Minimum** line displays the minimum count.

If creating a histogram viewlet using a form (for more information on forms, see *Section 2.5.1.2, Create a Viewlet with a Form*), the following settings are required:

- Within the **Fields** section, enable the **Count** option. Depending on the type of viewlet, this option can be displayed as **Events Count**, **Activities Count**, or **Snapshots Count**.
- From the **Group by** section, select a numerical element that has the **bucket** option. Enable this checkbox and specify all associated options. For more information on bucketing, see *Section 2.5.1.2, Create a Viewlet with a Form*. If **EventCount** is selected, you will have the following options:
    - **Size**: the viewlet generated will divide the data into intervals by the size range defined within the bucket value field.
    - **Count**: distributes the data in the number of intervals specified within the bucket value field.
    - **Auto**: if you do not need to specify a particular bucket range.

- Select the **Histogram** option within the **Viewlet Type** section.

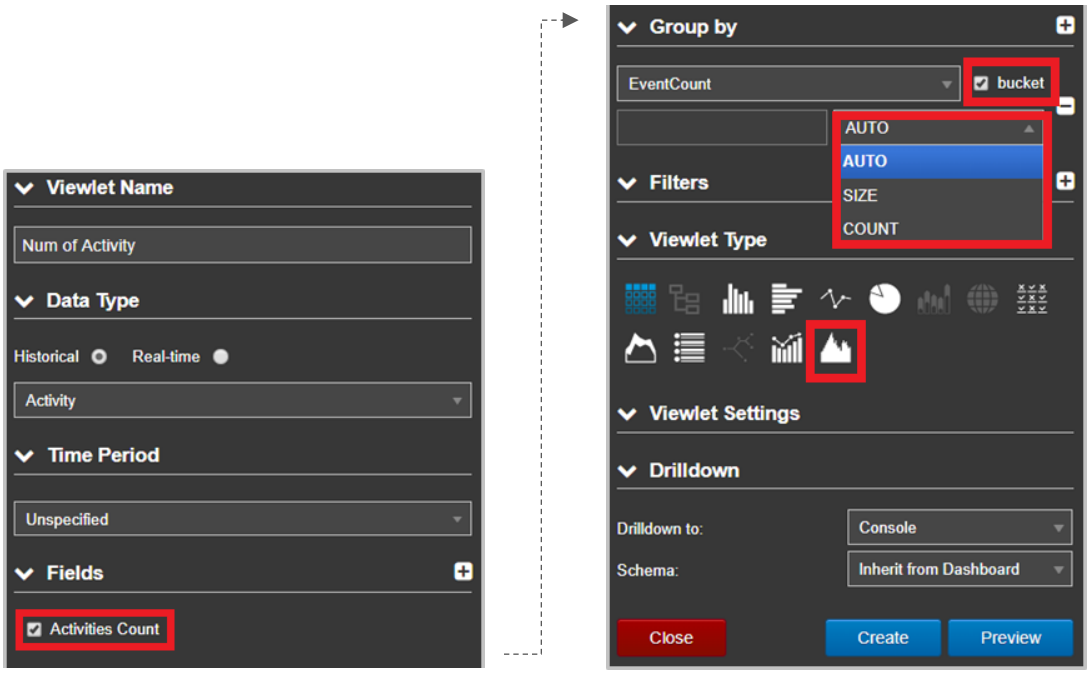


Figure 2.5.4.13-B. Histogram Form Options

The following viewlet gets generated using the options selected in the figure above. The same viewlet can be populated also with jKQL (see *Chapter 5, Using jKQL*, for more information):

Query: **jKQL>** `get number of Activity group by EventCount bucketed by size 3 show as histogram`

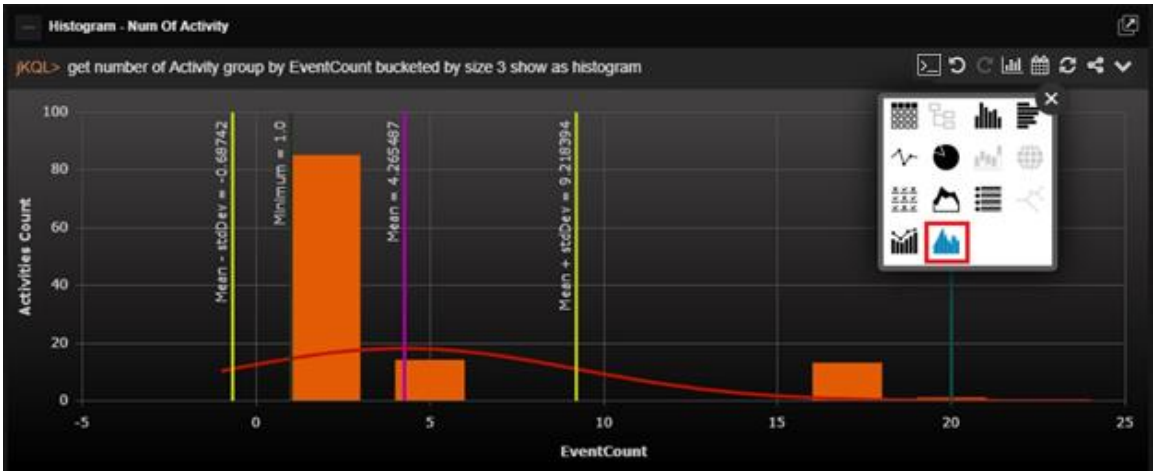


Figure 2.5.4.13-C. Histogram Viewlet

2.5.4.13.1Sample: The frequency of SnapshotCount

In the example below, the frequency of SnapshotCount shows how many activities have snapshot counts within the specific range, defined by bucketing size.

Query: **jKQL>** `get number of Activity group by SnapshotCount bucketed by size 10 show as histogram`

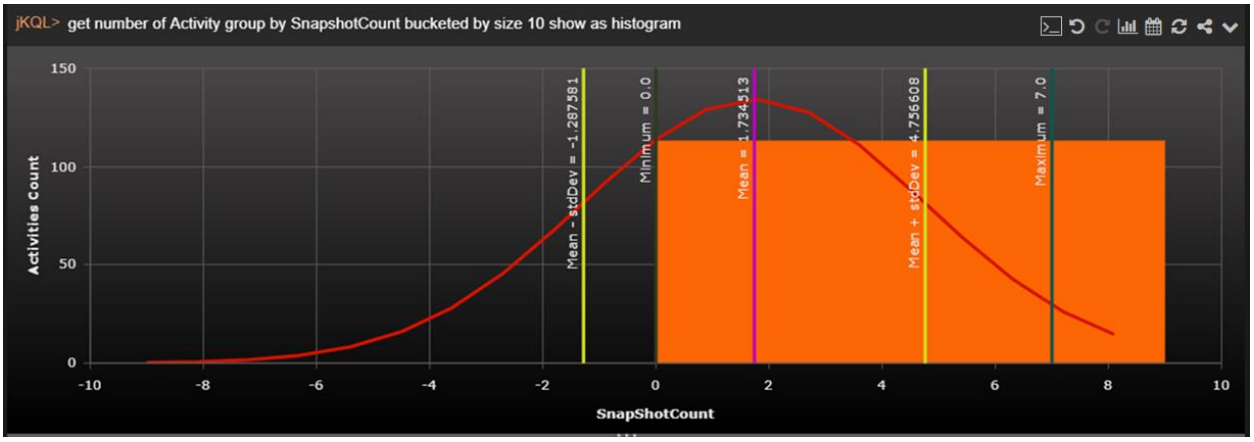


Figure 2.5.4.13-D. Histogram Viewlet – The Frequency of SnapshotCount

You can change the bucketing type from **Size** to **Count** by typing the following expressions in a jKQL query. The expression, *bucketed show as histogram*, corresponds to Auto bucketing type.

Query: **jKQL>** `get number of Activity group by EventCount bucketed show as histogram`

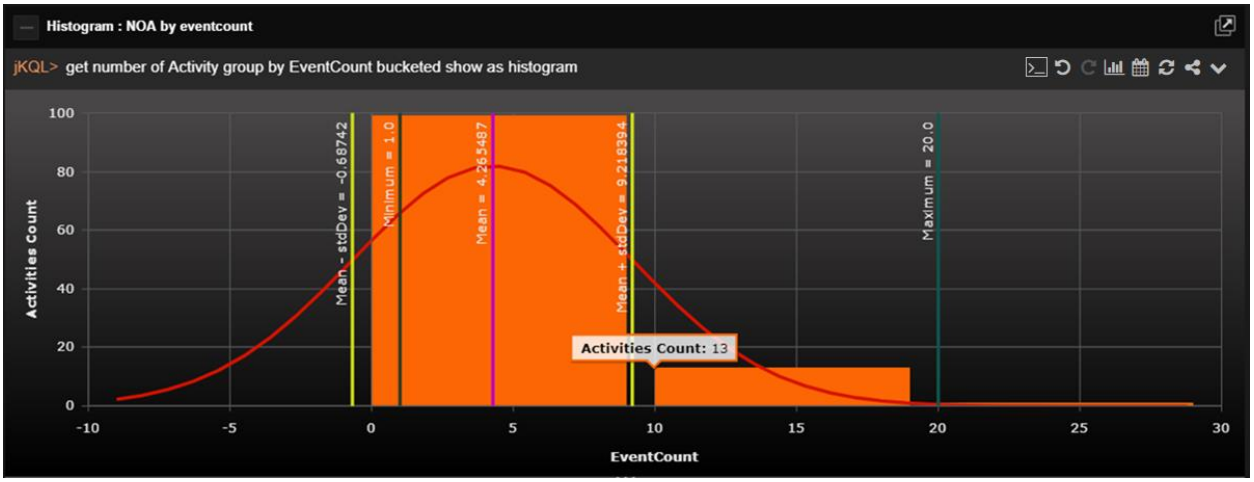


Figure 2.5.4.13-E. Histogram Viewlet – The Frequency of EventCount

You can see the difference between size and count bucketing in the table viewlets below. To generate these examples, click on the **Chart** button from the viewlet's toolbar and select the table chart type, or modify the jKQL query's *show as* expression.

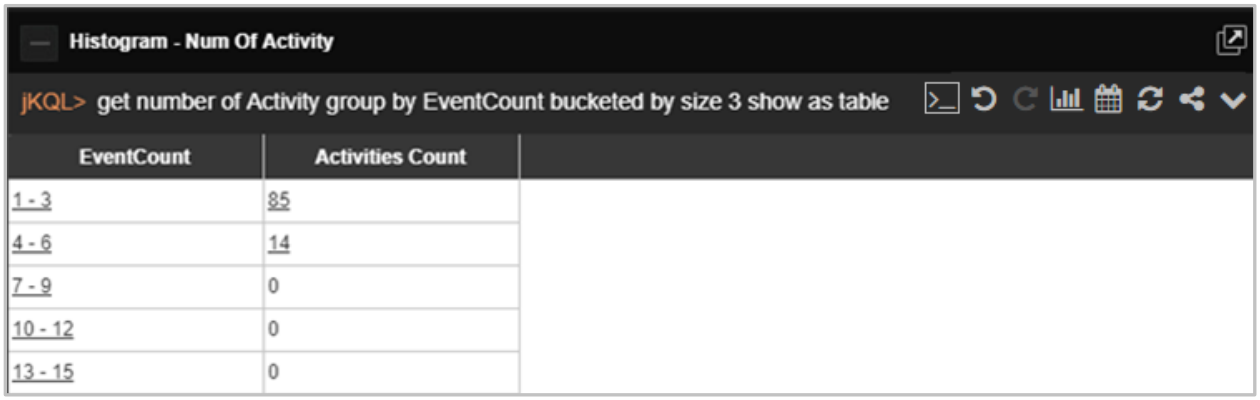


Figure 2.5.4.13-F. Bucketed by Size

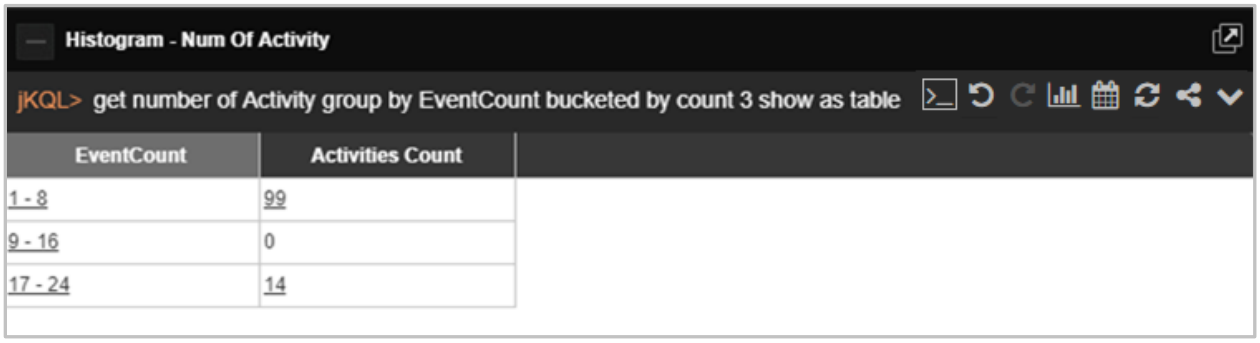


Figure 2.5.4.13-G. Bucketed by Count

In *Figure 2.5.4.13-F*, the EventCount is divided into an unspecified number of groups by 3. For example, the first row of data (EventCount is 1-3 and Activities Count is 85) shows that there are 85 activities taking place within one to three events. The second row of data (EventCount is 4-6 and Activities count is 14) shows that there are 14 activities which occur within four to six events.

Figure 2.5.4.13-G shows the data, divided into a specified number of intervals, by an unspecified range number.

2.5.4.14 Tree

Figure 2.5.4.14-A. Tree

Query: jKQL> `get Activity fields ActivityID, Severity, ActivityName show as tree`

Tree viewlets are only available for *activities*. By default, only the activity's name and severity icon are displayed (basic query example: `get activities show as tree`), but the viewlet can be enriched by adding *field* conditions as in the example above.

Expand an activity's tree by clicking the arrow immediately before the activity name. All the child activities, events or snapshots are displayed. An activity with no child records will have a circle instead of an arrow.

2.5.4.15 Clustering

Clustering charts use machine learning data to group data into clusters so that users can gain insight into the data. This is 'unsupervised' learning; a type of machine learning that looks for previously undetected patterns in a dataset with no pre-existing labels and with a minimum of human supervision

Below are examples of clustering viewlets.

Query: jKQL> `Get dataset compute clusters(3,PETAL_LENGTH,PETAL_WIDTH,SEPAL_LENGTH,SEPAL_WIDTH,'3',false) show as table`

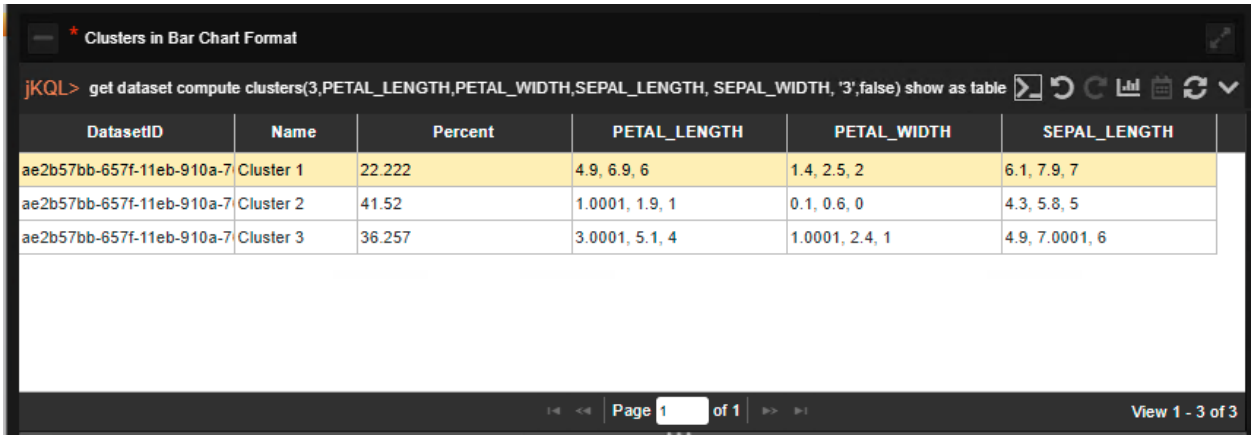


Figure 2.5.4.15-A. Clusters in Bar Chart Format

Query: jKQL> `Get dataset compute clusters3d(PETAL_LENGTH,PETAL_WIDTH,SEPAL_LENGTH,SEPAL_WIDTH,'3',true)`

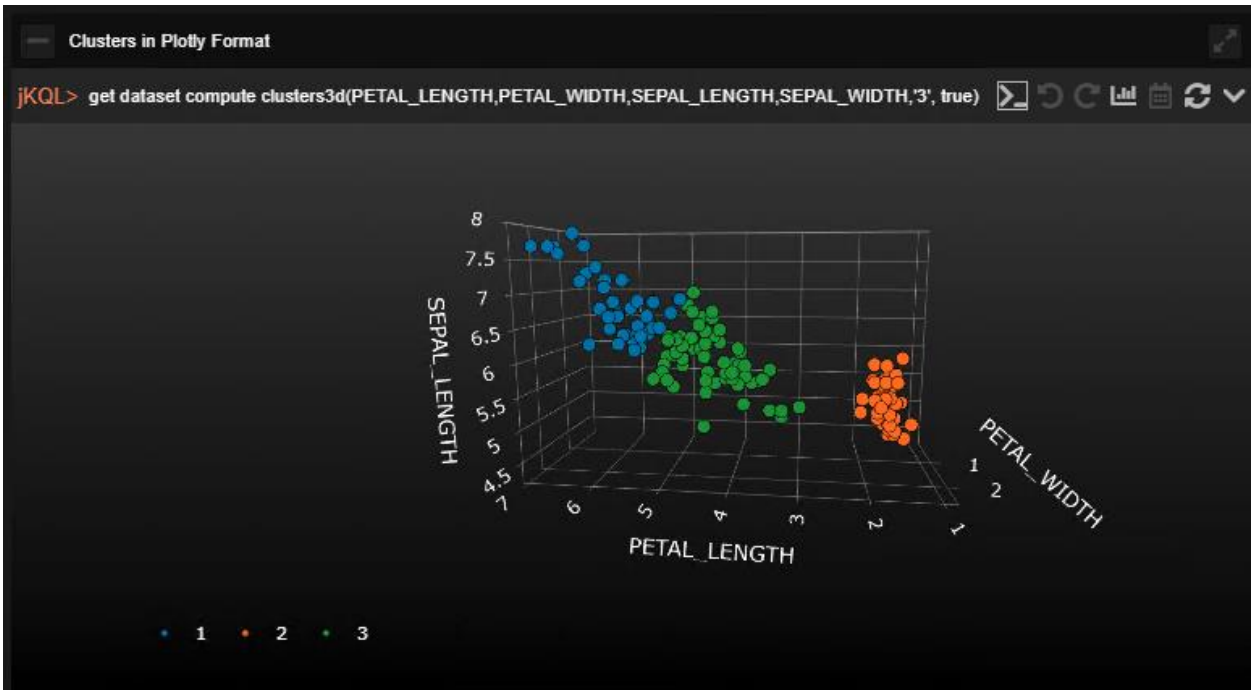


Figure 2.5.4.15-B. Clusters in Plotty Format

2.5.4.16 Correlation

Correlation charts use machine learning data to correlate the data fields. A high positive or negative number indicates a strong correlation. (A negative number indicates a negative correlation: both positive and negative numbers show a similar increase in absolute value.)

The below image is an example of a correlation viewlet.

Query: **jKQL>** `get dataset compute correlate(PETAL_LENGTH,PETAL_WIDTH,SEPAL_LENGTH,SEPAL_WIDTH)`

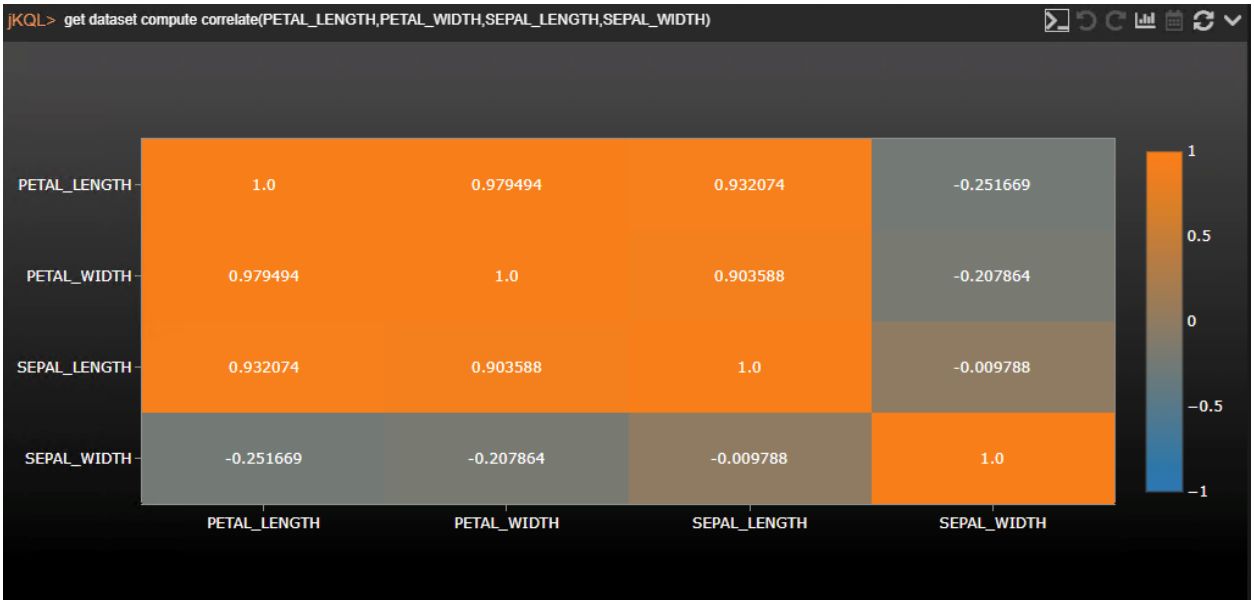


Figure 2.5.4.16-A. Correlation

2.5.4.17 Feature Suggestion

Feature suggestion chart types use machine learning data to display the fields the machine learning model considers the most important when predicting a target variable (the fields that affect the target variable the most).

The below images are examples of feature suggestion viewlets.

Query: jKQL> `get dataset compute featuresuggestion(PETAL_LENGTH,PETAL_WIDTH,SEPAL_LENGTH,SEPAL_WIDTH,SPECIES) show as table`

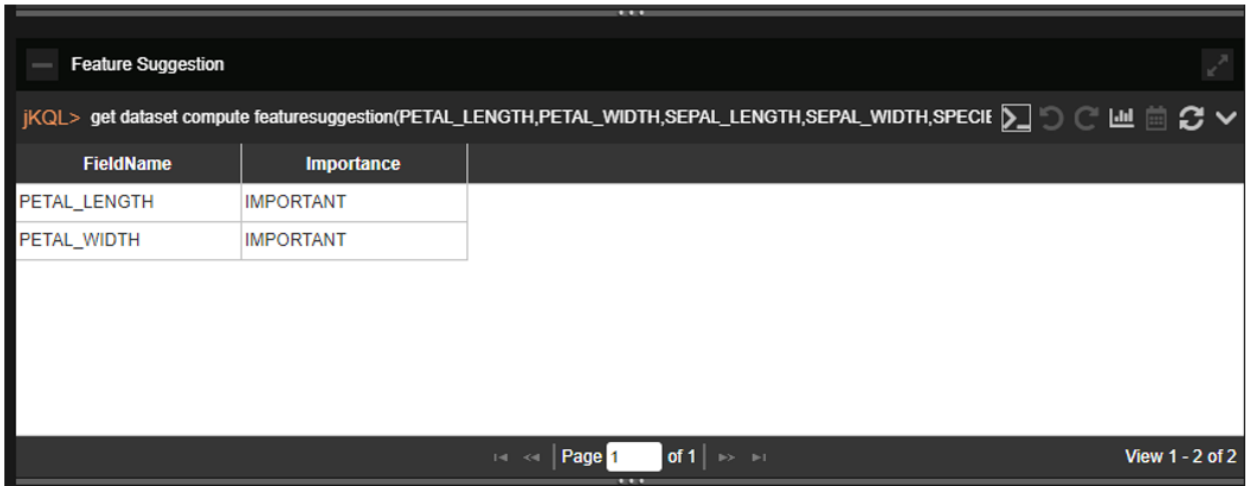


Figure 2.5.4.17-A. Feature Suggestion Table

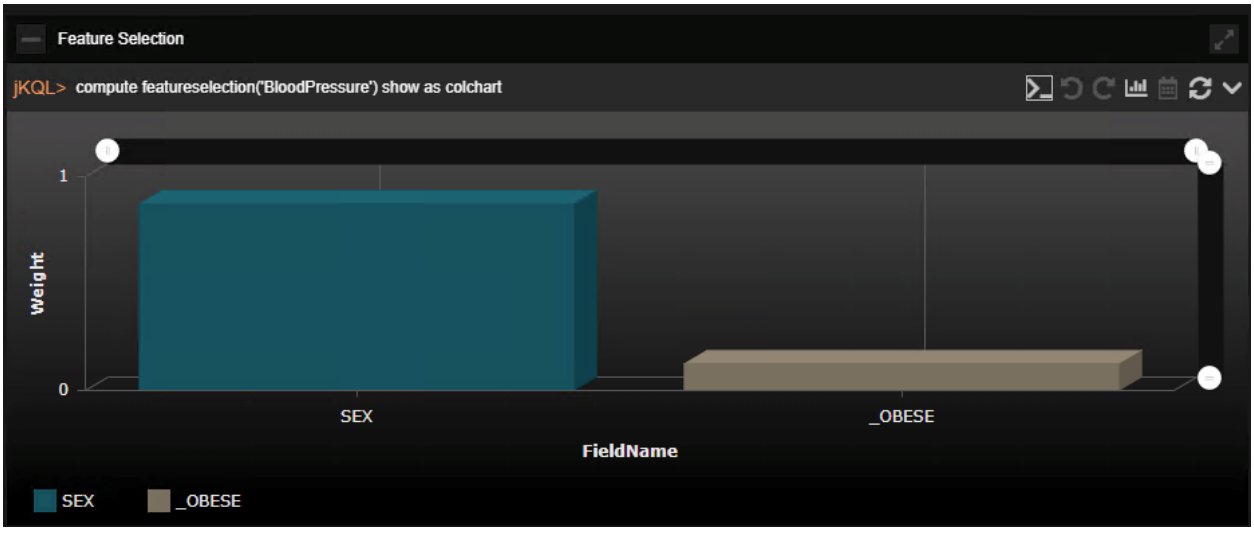


Figure 2.5.4.17-B. Feature Suggestion Diagram

2.5.4.18 Forecast

Forecast charts use machine learning data to display a future projection. The below image is an example of a forecast viewlet.

Query: jKQL> `compute forecast('closingPriceDaily',100)`

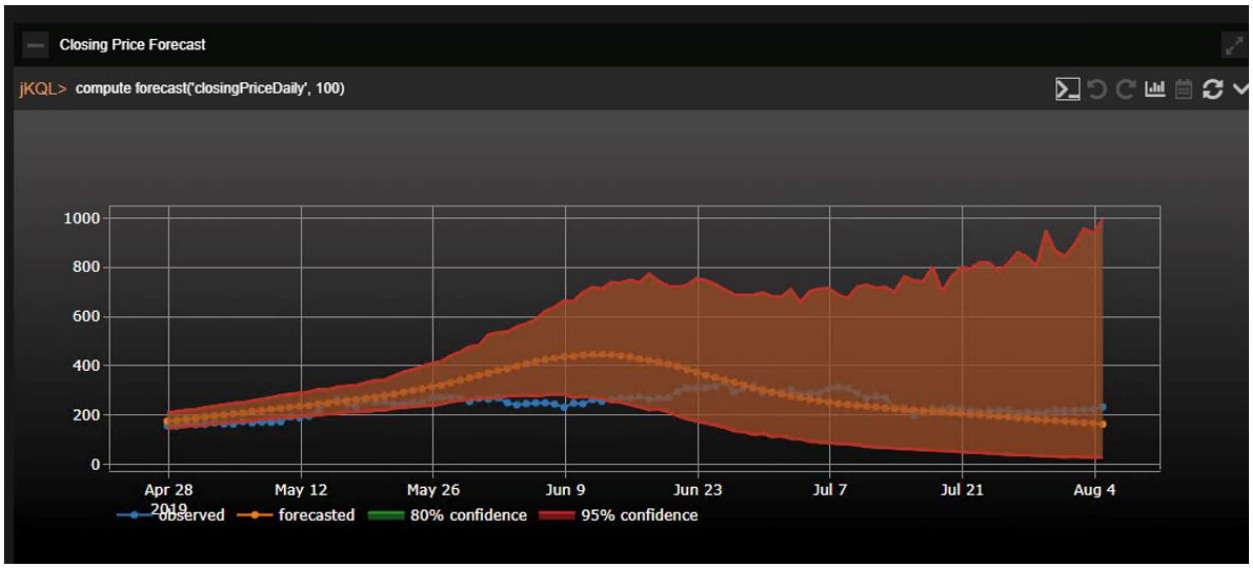


Figure 2.5.4.18-A. Forecast

2.5.4.19 Expected

Expected charts use machine learning data to display predictions. Given certain variables, the expected target variable is displayed. The below images are examples of expected viewlets.

Query: jKQL> `compute expected('SPECIES')show as table`

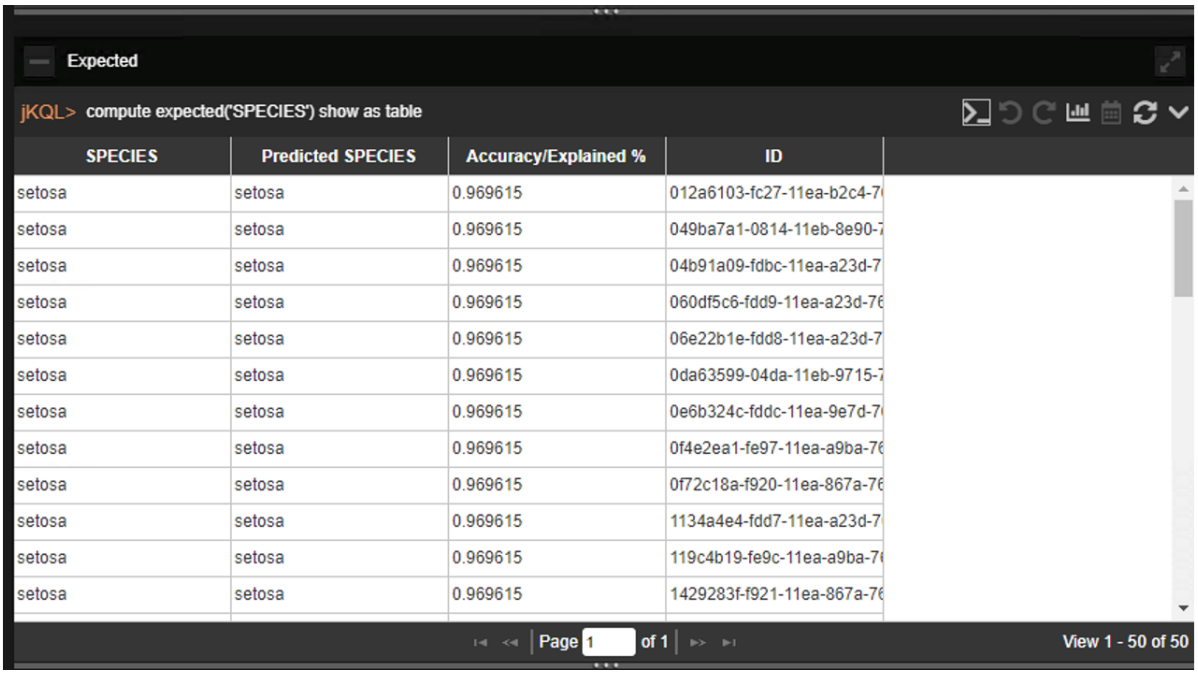


Figure 2.5.4.19-A. Expected Table

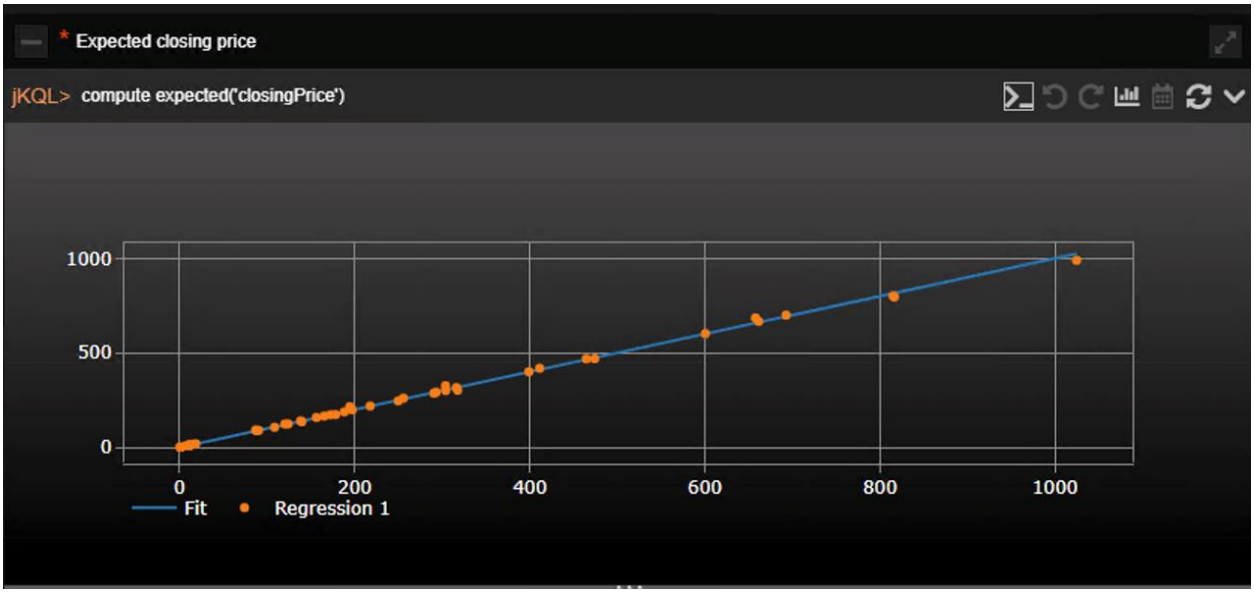


Figure 2.5.4.19-A. Expected Diagram

2.5.4.20 Images

Image viewlets allow you to display data on any predefined SVG image with custom bindings. The powerful, lightweight rules engine, json-rules-engine, is used for binding realization. You can share image viewlets (see the *Share Viewlet* section) for quick access.

SVG images have many elements, but the most important are the following:

- <metadata >
- <style>
- <defs>
- <g>

The following is an example:

Query: jKQL> `Get sensor fields all where PolicyName='DEMO - EAI Workflow Business Process.bsp'Show As Image('demo1')`

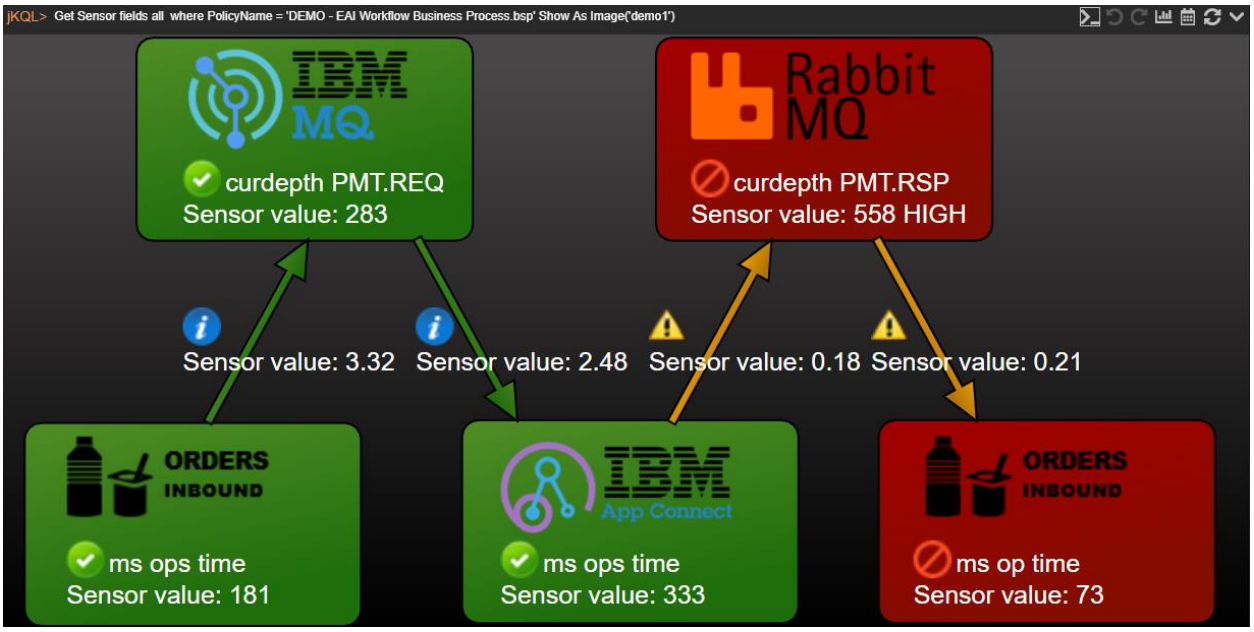


Figure 2.5.4.20-A. SVG Image Viewlet

If interested in utilizing SVG image viewlets, contact Support for assistance.

2.5.5 Date and Time

2.5.5.1 Select Date and Time Range

Viewlet date and time range can be updated. Perform the following:

1. Click the **Date & Time Range** icon.

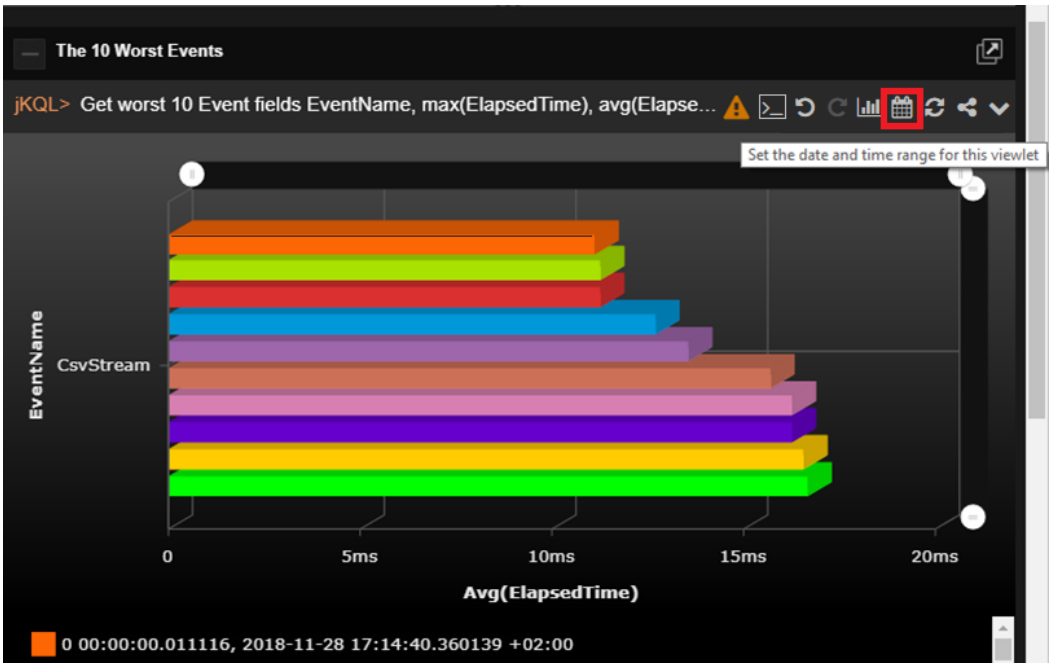


Figure 2.5.5.1-A. Date & Time Range Icon

2. The *Date & Time Range* dialog box appears.

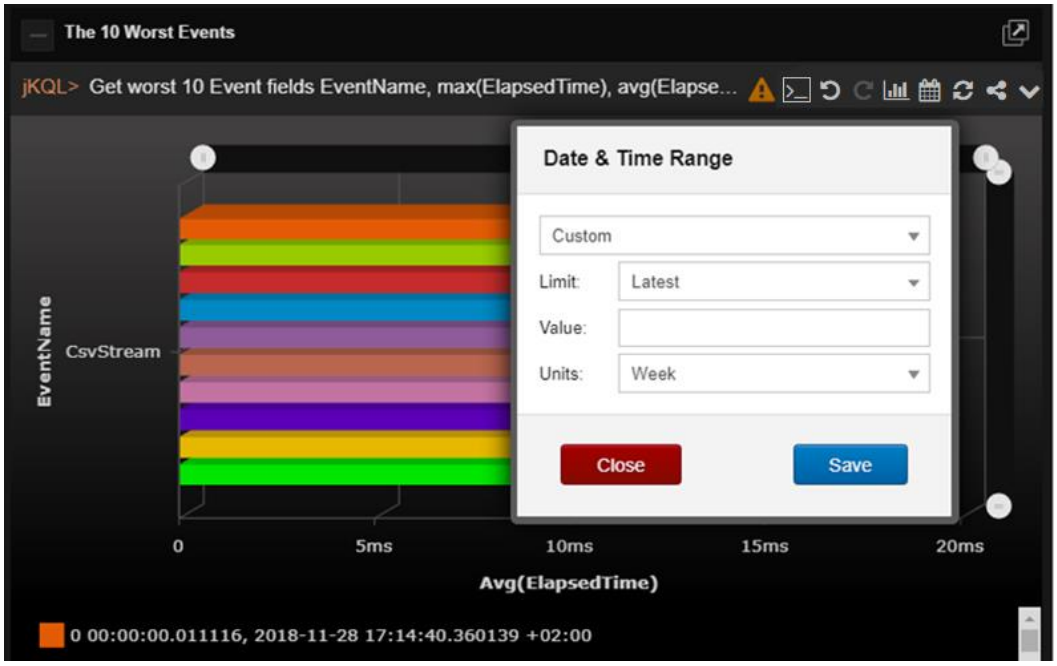


Figure 2.5.5.1-B. Date & Time Range Dialog Box

3. From the drop-down menu select the filter type: **Unspecified**, **Predefined**, **Custom** or **Date Range**. After selecting a type, specify all associated filter options and click **Save**.

2.5.5.2 Date and Time Filtering in Viewlets

Child viewlets inherit date and time conditions (i.e., *for last week, from*, *to*, etc.) from their parent viewlets; however, the time expression will not appear in the child's jKQL query. See the figures immediately below for an example.

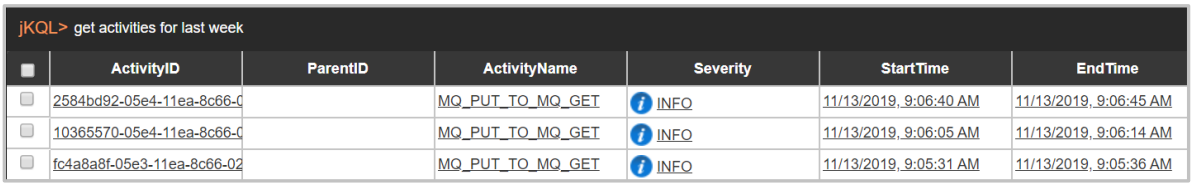


Figure 2.5.11-A. Summary Panel (parent) Viewlet: Includes Time Condition

Figure 2.5.11-B. Console Panel (child) Viewlet: Time Condition Applies (but does not appear in jKQL query line)

To view the date and time condition, simply hover over the Console viewlet's orange "jKQL>" label.

If the originating parent viewlet does not have a "for" condition, or if a Console viewlet does not have a parent viewlet, then the dashboard's default date and time will be applied to the Console viewlet and will display in the label.

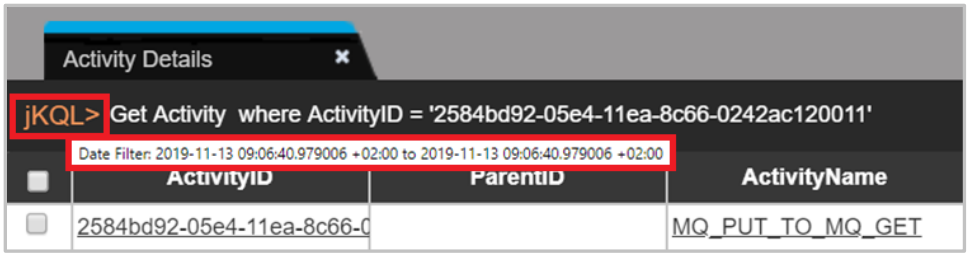


Figure 2.5.11-C. Hover to See Date Condition

TIP

If the dashboard has a default date and time (see *Section 2.3.5, Default Date & Time Range*), it is applied for all viewlets within the dashboard. If the date and time is set for a viewlet, then it will take precedence over the dashboard's default date and time.

2.5.6 Refresh Viewlet

To refresh viewlets, click the **Refresh Viewlet** button. This process will check for new data.

Figure 2.5.6-A. Refresh Viewlet

2.5.7 Viewlet Menu

Click the menu icon to open the viewlet menu. Functions are described below.

- **Edit Viewlet**: See section 2.5.7.1 below for more information.
- **Save Viewlet**: Allows you to save any changes to the viewlet.
- **Save as Viewlet**: Allows you to create and save a copy of the viewlet with a new name. The new viewlet can be found on the **Open Viewlet** dialog box *(Figure 2.5.1.4-A)* and added to any dashboard.
- **Remove Viewlet**: Allows you to remove the viewlet from the dashboard. A dialog box opens asking you to confirm the removal. The viewlet is not deleted and can be restored by opening the **Main Menu** and selecting **Viewlet** > **Open** (*Open Existing Viewlet* dialog box opens), select the viewlet, to be restored, and click **Open**.
- **Delete Viewlet**: Allows you to delete the viewlet. A dialog box opens asking you to confirm the deletion.
- **Export to CSV** or **Export Viewlet**: For table and scorecard viewlets, this option will be **Export to CSV** (to download data to a .csv file). For all other viewlet chart types, this option will be **Export Viewlet** (to download data to a .svg file). Please note that viewlets can also be exported in .json file format, see *Section 2.6.2.2, Export*, for more information.
- **Share Viewlet:** Viewlets can be shared on a web page or internet browser. See *Section 2.5.10, Share Viewlet*, for more information.

2.5.7.1 Editing a Viewlet

Click the **Viewlet Menu** icon and select **Edit Viewlet**.

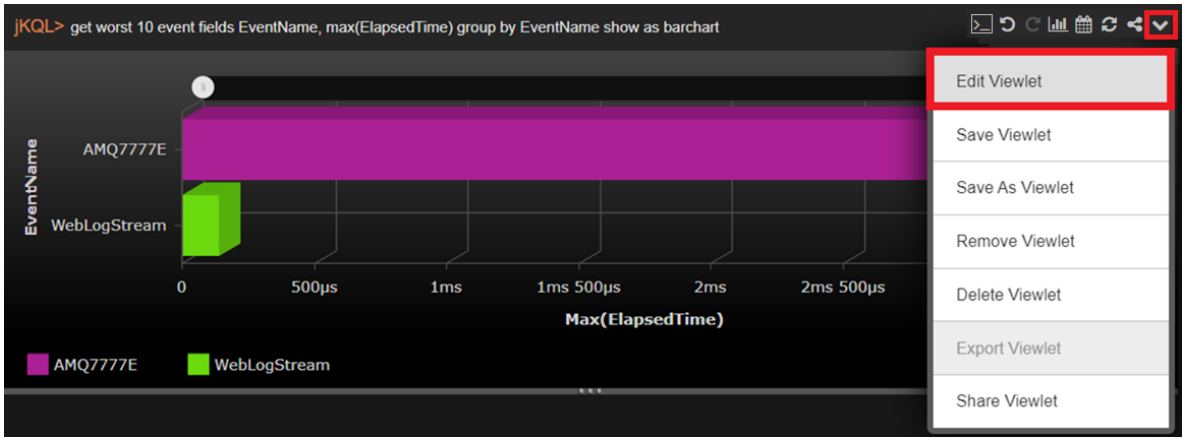


Figure 2.5.7.1-A. Edit Viewlet Menu

Options are different for each display type. All traits of a viewlet are displayed on the right side of the screen.

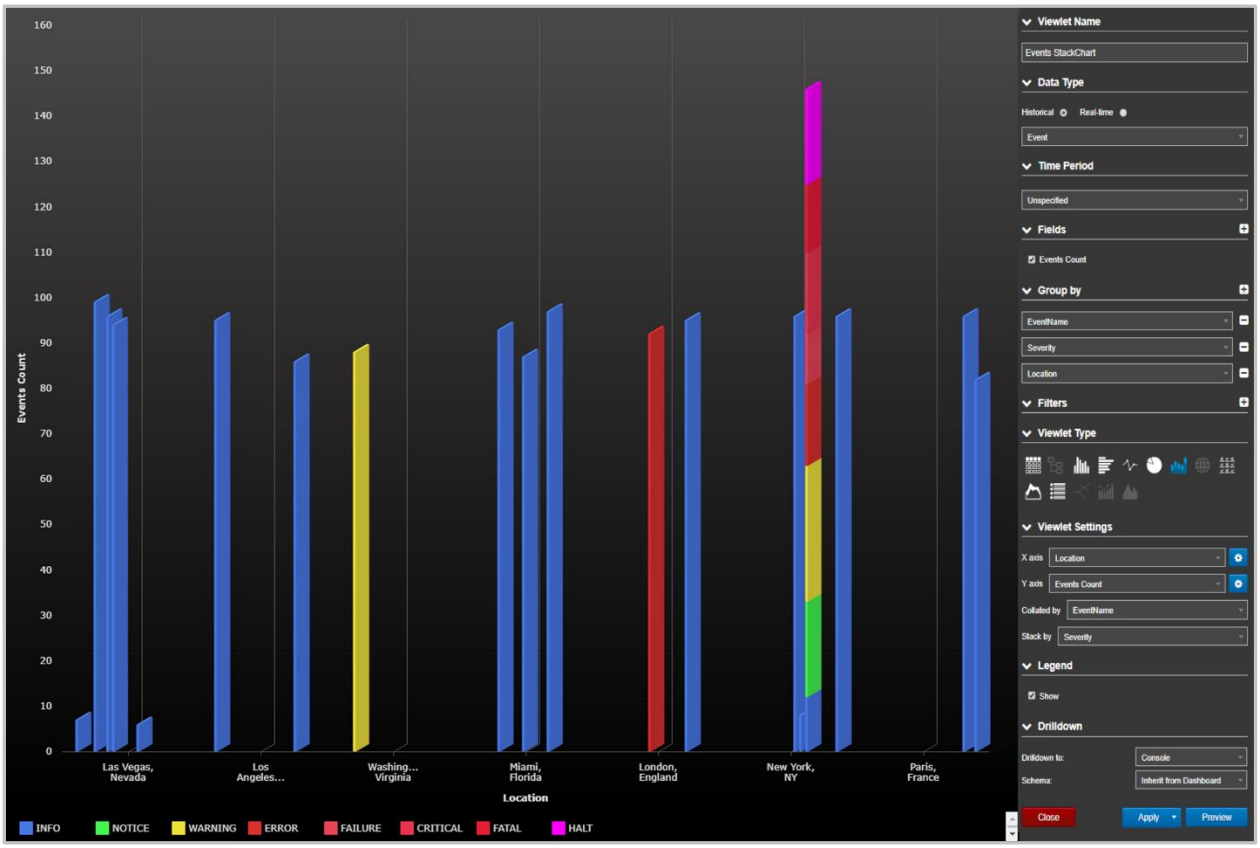


Figure 2.5.7.1-B. Edit Viewlet Form

After making changes, click **Preview** to view updates made before applying. To discard changes and cancel, click **Close**. Click **Apply** to apply the updates (updates will not be saved) or click on the downwards arrow on the right side of the **Apply** button to get the **Save** option to save the changes (this will save the modifications).

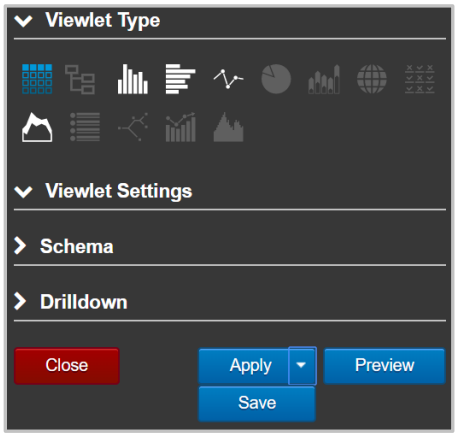


Figure 2.5.7.1-C. Form Options

2.5.8 Filtering and Display Options

Users can view more detailed data displayed within viewlets. Use the following methods to drill into viewlet data and customize how viewlets are displayed.

2.5.8.1 Filtering with Variables

Viewlets can easily be filtered using variables. While creating or editing a viewlet with a Form (see Section 2.5.1.2, Create a Viewlet with a Form, or 2.5.7.1, Editing a Viewlet), click the plus button within the **Filters** section to add a new filter.

Select a filter and the function from the drop-down menu and click the settings button to expand the filters toolbar. Select the **Variable** tab and click the pencil button. Please note, that not every filter will have this tab.

The *Create new variable* window opens. Type in a name for the variable and check off the **Auto suggestion** checkbox if it is not selected to get item suggestions to use as the filter criteria (this option is available only for items which have auto suggestion functionality). When the **Auto suggestion** is not available, the variable will need to be updated manually (no suggestions will be provided).

NOTE

Multiple variables filters can be created by repeating the same steps described above, but their names must be unique.

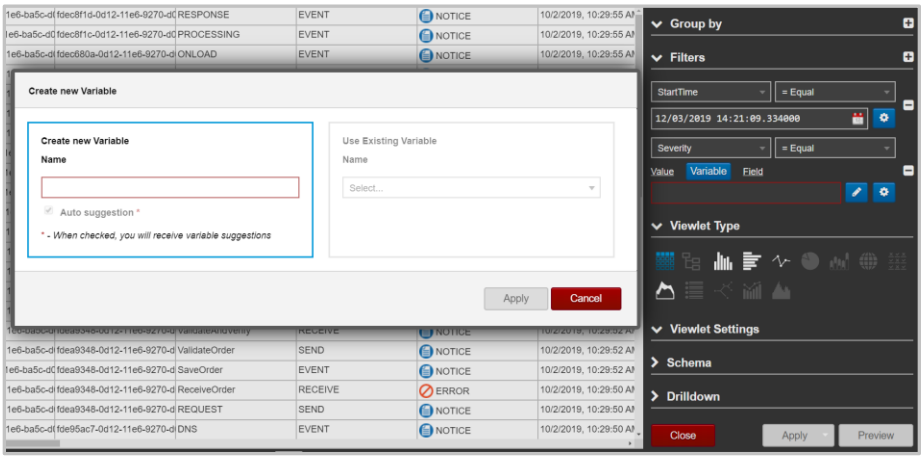


Figure 2.5.8.1-A. Modify Variables

Click the pencil button to edit the name of the variable filter. To remove a filter, click the minus button. The variable value label is placed between curly brackets { } (this is how variable expressions appear in jKQL queries). For example:

```
jKQL> Get Events where Severity = ${E:Severity:Event:Severity} show as table
```

Expression "`${E:Severity:Event:Severity}`" can be manually replaced with a severity type (for example, INFO):

```
jKQL> Get Events where Severity = 'INFO' show as table
```

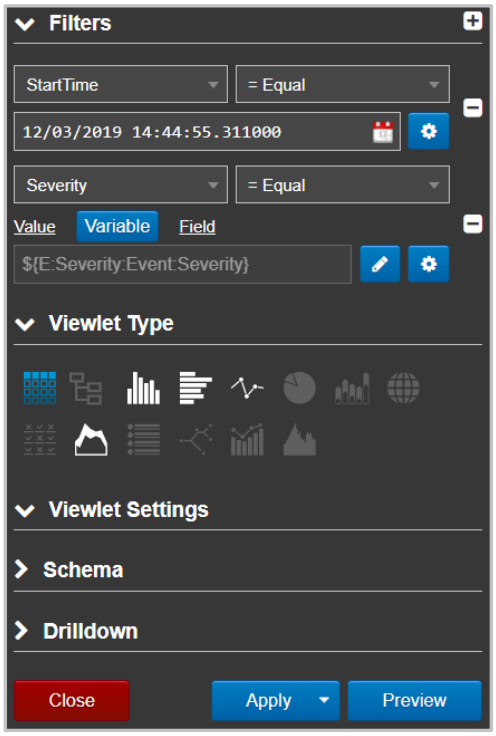


Figure 2.5.8.1-B. Modify Variables

Click **Apply** or **Save** (click the **Apply** button's drop-down menu). The **Modify Variables** window opens. If multiple variable filters were added, they all would appear in this window. If the **Auto suggestion** checkbox was checked off while creating or editing the variable, the *Modify Variables* window will have a drop-down menu with suggestions provided, for example, the viewlet data can be filtered by severity type.

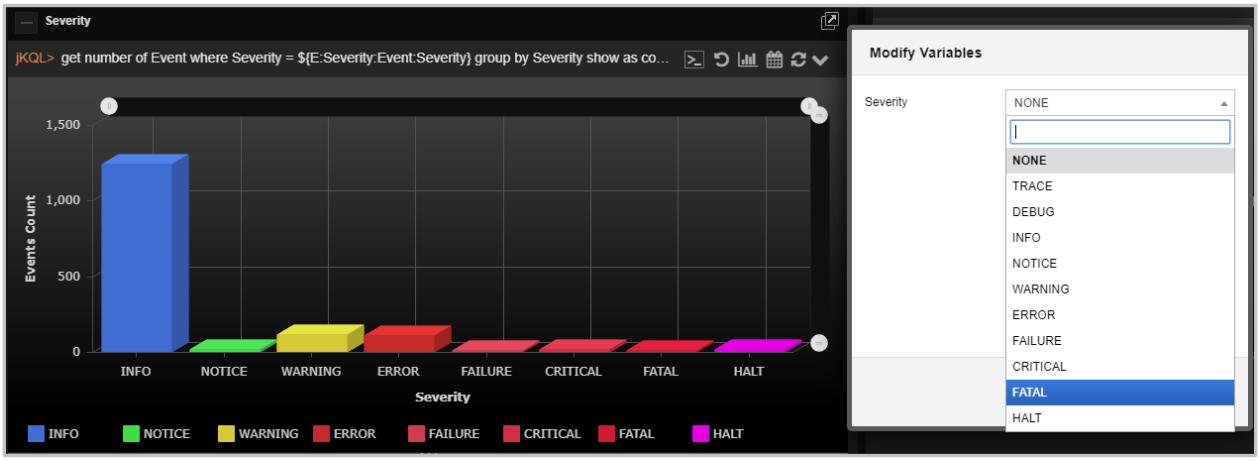


Figure 2.5.8.1-C. Modify Variables

Select the severity type from the suggested drop-down menu and click **Apply**. In the figures above and below, the **Fatal** severity type was selected. The jKQL query and the *Severity* viewlet were modified according to the selected variable.

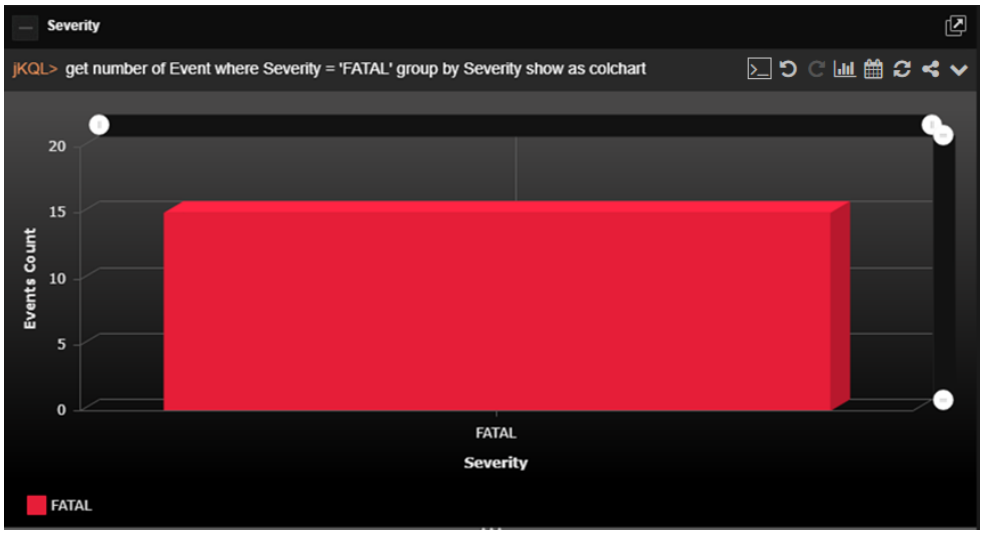


Figure 2.5.8.1-D. Modify Variables

To modify variables, perform one of the following:

- Modify directly in a viewlet's jKQL query by entering a value between the curly brackets { }
- Modify within a viewlet's form
- Click the **Modify** button on the top right corner of the workspace to update all viewlets that use the same variable within the dashboard

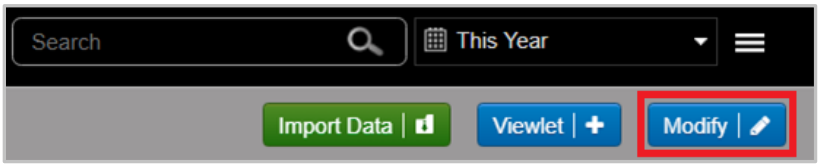


Figure 2.5.8.1-E. Modify Variables

2.5.8.2 Viewlet Scroll Bars

Use the scroll bars within viewlets to view a specific time range and change the amount of data displayed. These scroll bars appear in column, bar, line, stack, area, and anomaly chart types. The scrolls at the top control the X axis of the chart and the scrolls to the right control the Y axis.

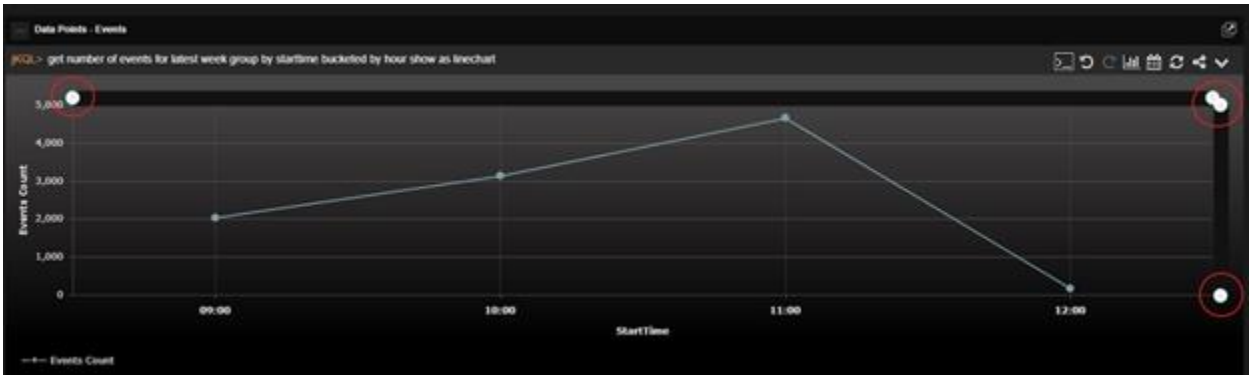


Figure 2.5.8.2-A. Viewlet Scrolling

2.5.8.3 Zoom In / Show All

Select data for a more detailed view. Using your mouse, draw a box around the area you would like to drill into.

Figure 2.5.8.3-A. Zoom In On a Specific Area

Only the area selected will be displayed. This feature functions within the following chart types: column, bar, line, stack, area, topology, and anomaly.

Figure 2.5.8.3-B. Zoomed In Area

To disregard scroll filters and drilling down, click **Show all** to show all data originally displayed.

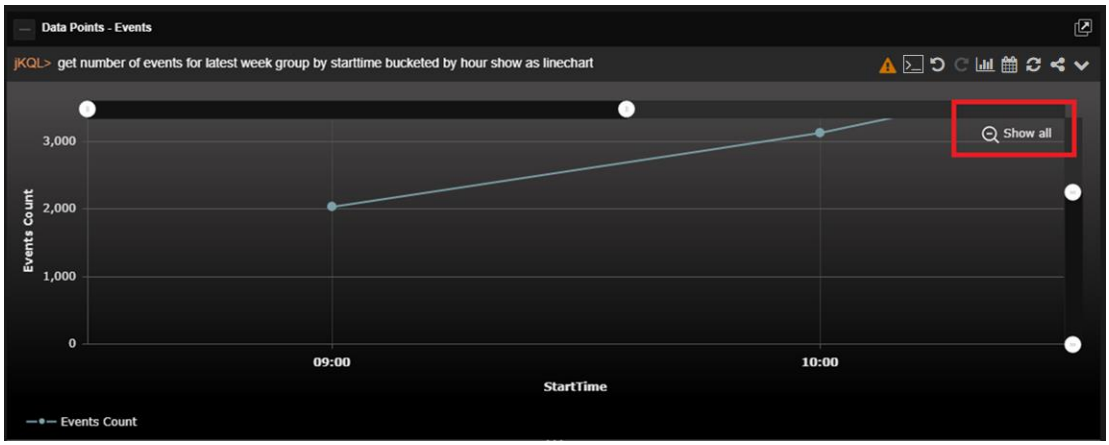


Figure 2.5.8.3-C. Show All

2.5.8.4 Show / Hide Chart Elements

Within line chart viewlets, a unique line is included for each queue manager, queue combination. Click legend keys to turn a specific resource off and on. When off, the resource's line will be removed from the chart and its key in the legend will appear grey.

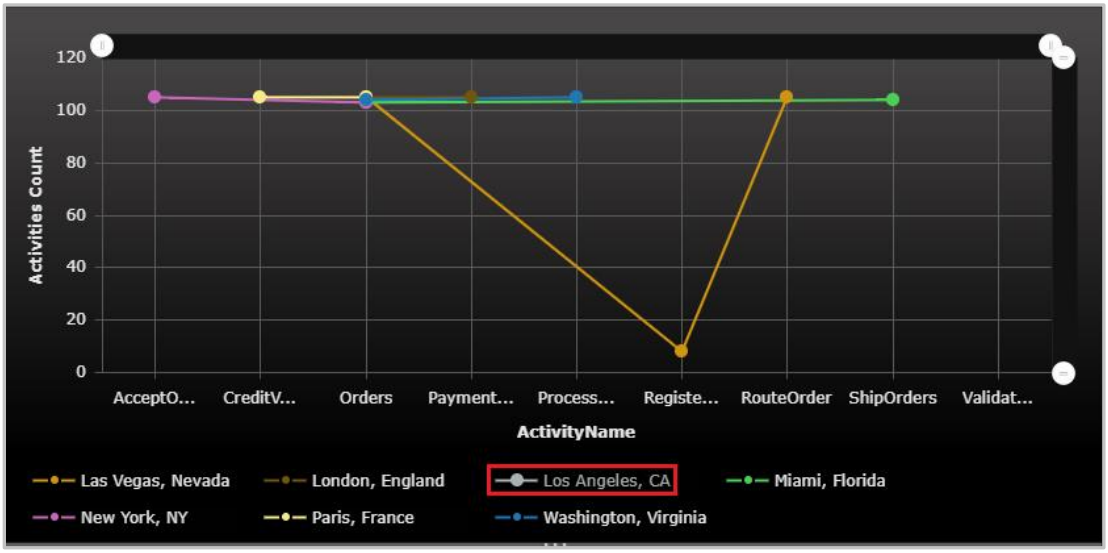


Figure 2.5.8.4-A. Disable Line Chart Lines

2.5.8.5 Viewlet Size

The size of viewlets can be changed using the ellipses symbols appearing on the sides of the viewlets.

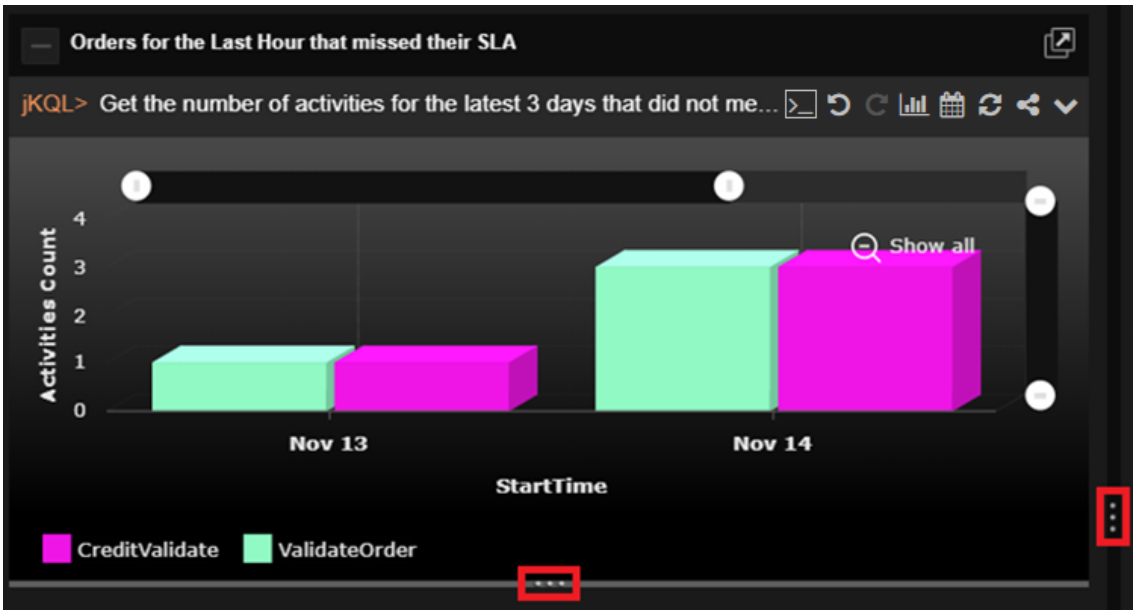


Figure 2.5.8.5-A. Change Viewlet Size

The vertical ellipsis ⋮ appearing on the sides of viewlets allow users to:

- Increase/decrease viewlets by sliding the ellipsis left or right. This will increase/decrease the size of all viewlets appearing within the same column.

- Expand viewlets appearing in the same column to fill the screen by clicking the ellipsis. If there are viewlets appearing in the right column of the dashboard, these viewlets will be hidden until the ellipsis is clicked again.

The horizontal ellipses appearing on the top/bottom of viewlets will increase/decrease the height of viewlets.

2.5.8.6 Show / Hide Viewlets

Click on the box immediately to the left of viewlet titles to collapse or expand viewlets. A viewlet's collapsed or expanded state will remain until changed (even between logins).

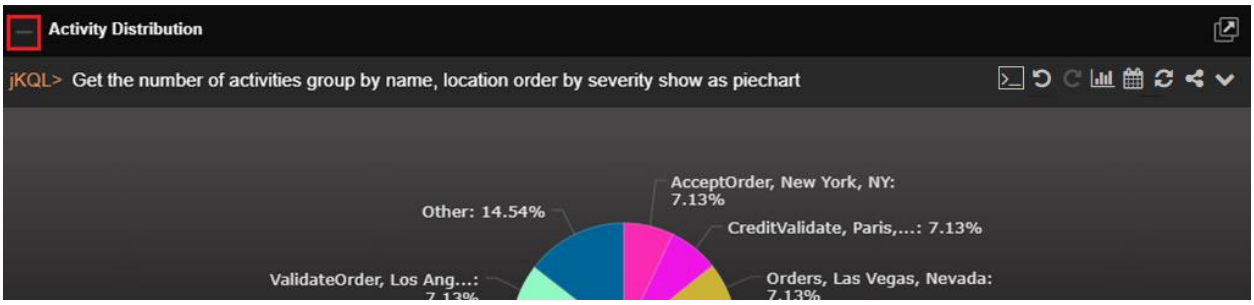


Figure 2.5.8.6-A. Show / Hide Viewlets

2.5.8.7 Tear Off

Notice that all viewlets have a **Tear Off** button located at the top right corner. Clicking the **Tear Off** button will open the viewlet in a new, larger window.

This feature is helpful in a datacenter where you may wish to display a viewlet or dashboard on a large monitor. For example, a large screen of summary viewlets can display a high-level environment status view.

An alternative use case might be for an administrator or developer with multiple screens. They can have the full dashboard on one screen and a specific viewlet they are configuring on the other.

2.5.9 Rename Viewlet

To rename a viewlet, double click the viewlet's name. The field becomes editable and will appear with a blue frame. Specify a new name and hit the **Enter** key on your keyboard to save changes.

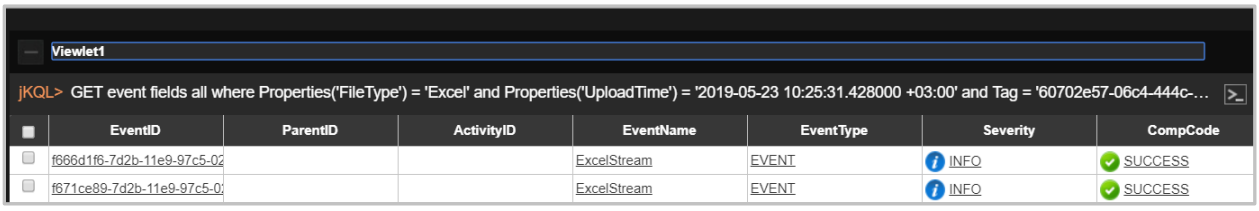


Figure 2.5.9-A. Rename Viewlet

2.5.10 Share Viewlet

Viewlets and their schemas can easily be shared as a URL or embedded on to a web page without requiring the viewer to login. The URLs are public, so there is no need for viewers to log in. This feature is useful for viewlets that need quick, frequent access; simply bookmark the URLs for easy access.

To share a viewlet, select **Share Viewlet** from the viewlet's menu (see Section 2.5.7). For this option to appear on the viewlet's menu, be sure to save the viewlet's dashboard. After selecting this option, the *Share Viewlet* window opens.

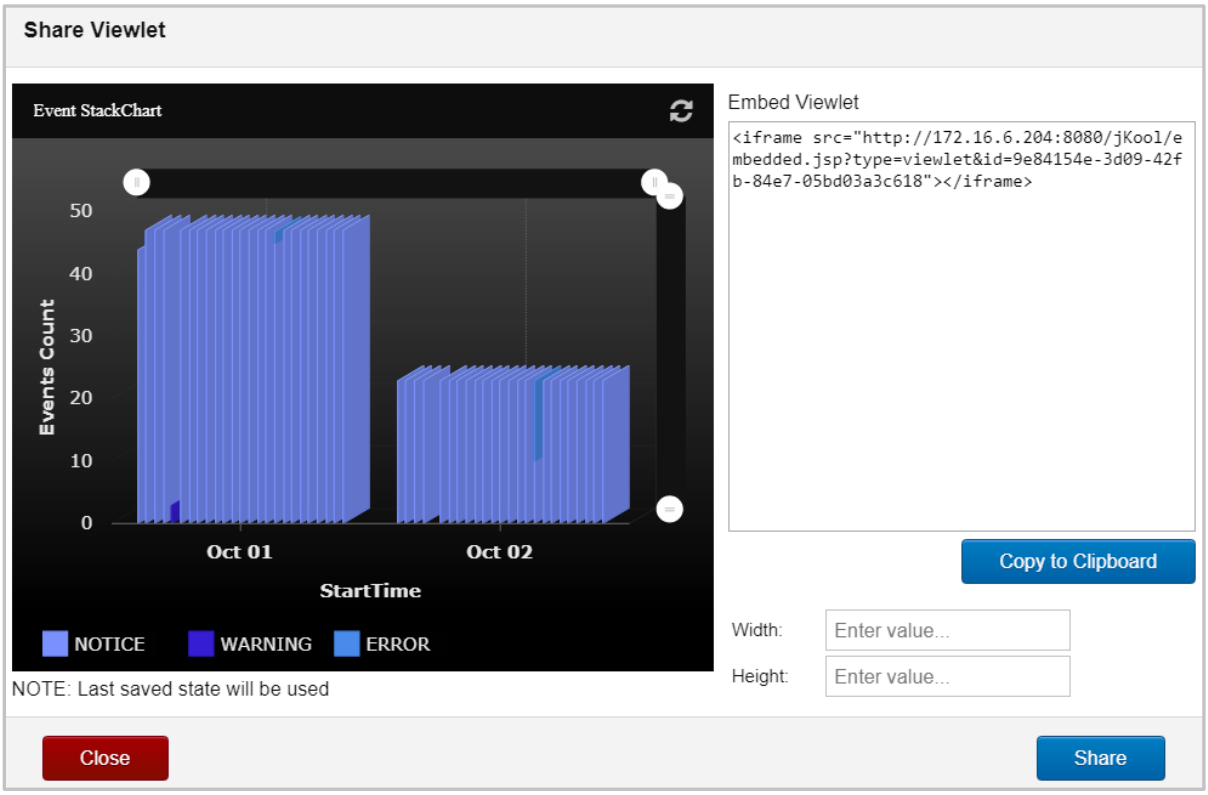


Figure 2.5.10-A. Share Viewlet Window

A preview of the viewlet is displayed on the left side of the menu. Click the **Refresh** icon to refresh the viewlet if needed.

The code to embed the viewlet in a webpage appears in the *Embed Viewlet* section on the right side of the window. Copy this link manually or click **Copy to Clipboard**. Before copying, you can enter dimensions within the **Width** and **Height** fields to specify the size of the viewlet. To simply open the viewlet in an internet browser, copy the link appearing within the double quotation marks and paste into the browser's address bar.

The final step to enable this feature is to click **Share**. After **Share** is clicked, the **Shared Viewlet** icon will appear on the viewlet's toolbar and the viewlet will be viewable.

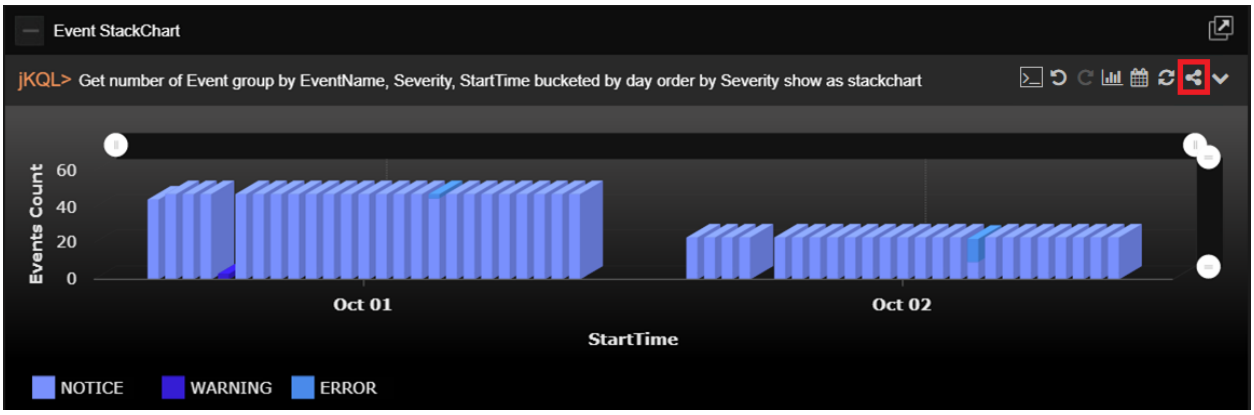


Figure 2.5.10-B. Shared Viewlet Icon

Click the **Shared Viewlet** icon to reopen the *Share Viewlet* window. You can copy the share code, update the viewlet's dimensions or stop sharing the viewlet (click **Stop Sharing**). The **Viewlet Shared** checkbox appears at the top right displaying the shared status.

If **Stop Sharing** is clicked and a user attempts to view the viewlet, the message, **Embedded viewlet is not available**, will appear.

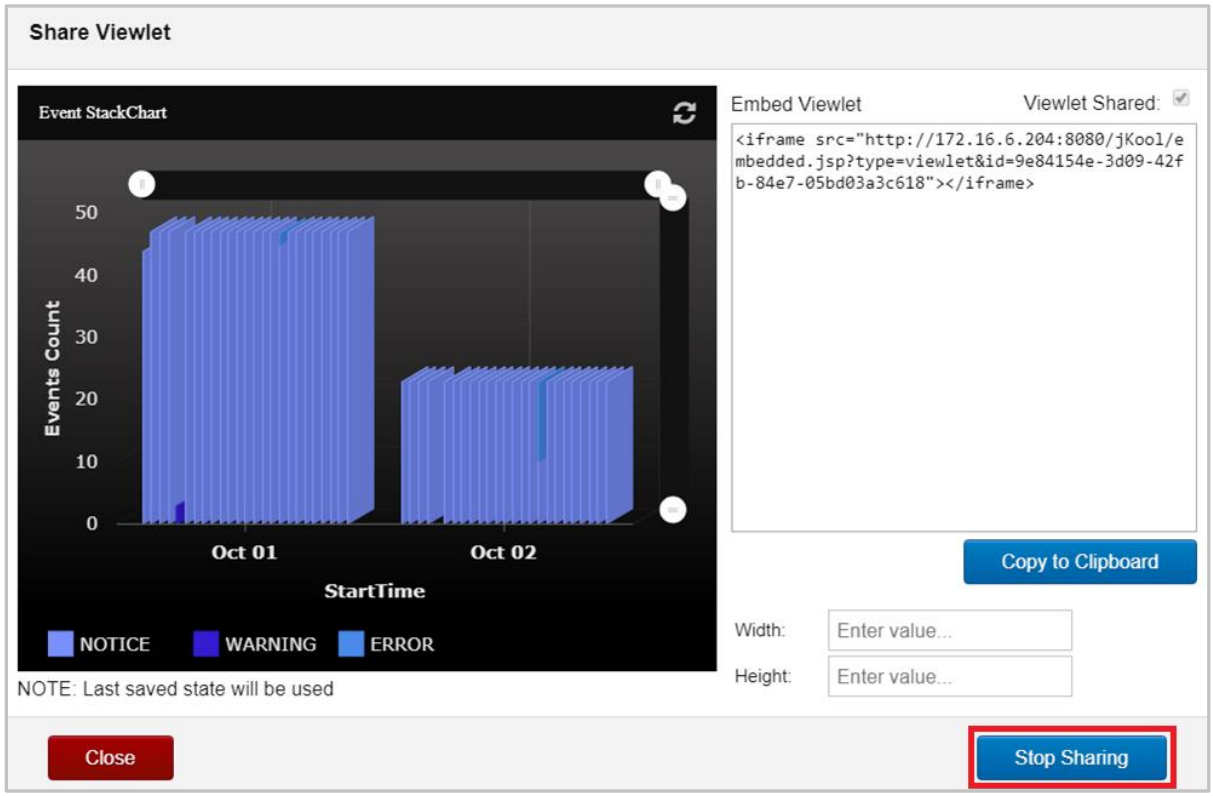


Figure 2.5.10-C. Stop Sharing a Viewlet

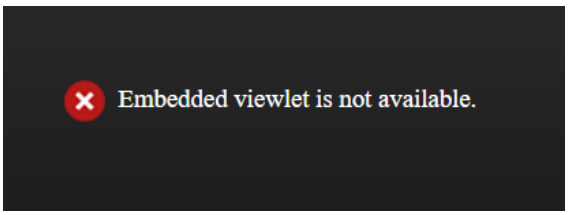


Figure 2.5.10-D. Stop Sharing a Viewlet

2.5.11 AutoPilot Integration

AutoPilot can be used as an External Data Source (EDS), allowing it to be integrated with XRay. With this integration, users can perform AutoPilot functions directly in XRay. Users can query via a jKQL statement to view information about policies that are running on a CEP instance in AutoPilot. In addition, users can also invoke actions such as starting/stopping a policy and acknowledging/unacknowledging a sensor. See below for setup information and for examples of how this integration is useful.

2.5.11.1 Setup: Streaming AutoPilot data to XRay

This section covers instructions for streaming AutoPilot data to XRay. You can choose to stream policies results, facts, or both. Keep in mind that in XRay, AutoPilot policies results become *events*, and AutoPilot facts become *snapshots*.

To set up streaming, you must first have an access token from XRay and use it to configure AutoPilot.

2.5.11.1.1 Generate an access token

1. From the XRay Main Menu, go to **Admin settings**.
2. Go to the **Organization** tab. The Organization Manager dialog opens.
3. Open the **Repositories** tab.
4. Select a repository and click **Edit**.
5. Select the *Manage Repository Tokens* tab.
6. At least one token should have been created for you as part of XRay backend installation process. Click  to view the token. Copy the token value and go to step 8. If no tokens are present, proceed to the next step to generate a new token.
7. Click **Generate Token**.
8. Verify that for the token you are using, the **Stream** setting is set to *Active*. All other settings can be left Inactive.

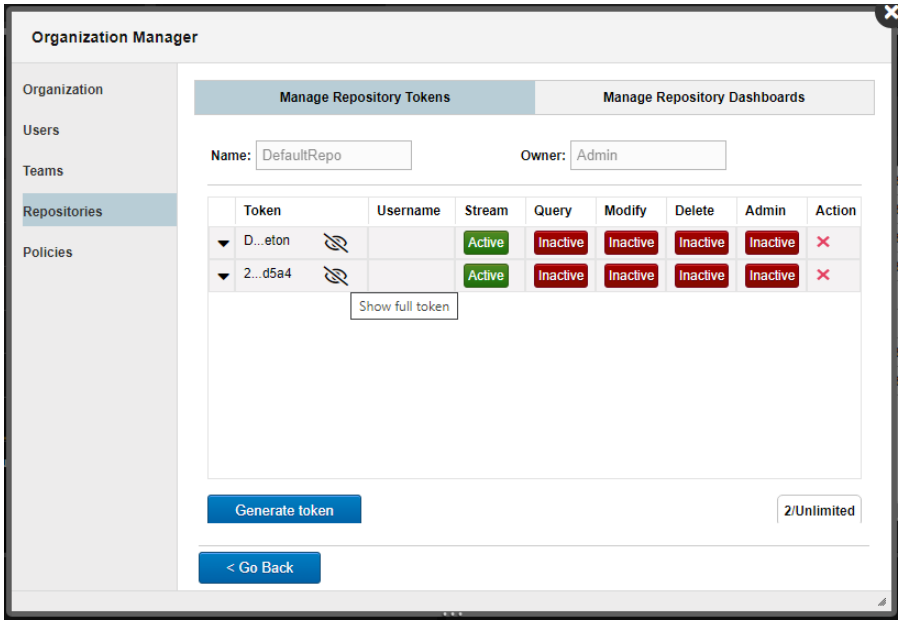


Figure 2.5.11.1.1-A. Repository Tokens

2.5.11.1.2Configure AutoPilot for Streaming

It is recommended that you set up the streaming token in global.properties so that it applies to all servers. It can also be set up in node.properties for a single server.

1. Go to the AutoPilot root directory (for example, `/opt/Nastel/AutoPilotM6/`) and open global.properties.
2. Find the line that reads: `"; uncomment if streaming events or metrics to Nastel XRay."`
3. Uncomment the following two lines by removing the semicolon from the beginning of the line:

```
;property tnt4j.source.hosturl=https://data.jkoolcloud.com
;property tnt4j.source.access.token=your_accesstoken
```

4. On the first line, replace the placeholder host URL with the address of your local streaming server (for on-premises installations) or cloud service (for cloud installations). The host URL must be a full address. The following ports are examples only. Ports from the gateway settings could be used instead.
   http: 6580
   https: 6585
5. On the second line, replace *your_accesstoken* with the access token from XRay.

2.5.11.1.3Stream events (Recommended)

If you want to stream events to XRay, do the following:

1. In the AutoPilot root directory, open log4j2.xml.
2. Find the line that reads:

   `<!-- uncomment the following to stream policies events to Nastel XRay -->`

3. Uncomment the following lines:

```
<!-- Logger name="policies" Level="info" additivity="false">
              <AppenderRef ref="Tnt4j"/>
</Logger-->
```

Event streaming is now enabled. AutoPilot policies results are streamed to XRay as events.

2.5.11.1.4Stream facts

If you want to stream facts to XRay, such as facts from Domain Manager, Operating System Monitor, or from the workgroup server expert, you can do so by following the instructions below. Every expert that can stream data to AutoPilot has a Streaming Options tab where streaming can be configured. The example below covers the Domain Manager.

1. In AutoPilot, right-click the Domain Manager and go to **Properties**.
2. Look for the **Streaming Options** tab. This tab controls which facts get streamed and the location they stream to. Many of the fields are optional and have default values

provided in the **tnt4j.properties** file that is part of the XRay installation package. However, they can be overridden.

3. For streaming the facts as they update, select **Stream Facts**. Again, keep in mind that AutoPilot facts become snapshots in XRay.
4. For streaming periodic derived metrics, select **Derived Metrics** and set a frequency by entering a time interval in milliseconds in the **Interval of derived metrics** field.
5. Since some experts produce a large number of facts, we do not recommend that you stream all of them. Instead, in the **Exclude filter** and **Include filter** provided, it is recommended that you enter regular expressions to control which facts are streamed.
6. The default Streaming configuration is com.nastel.autopilot.
7. Click **Apply** to apply changes.

Fact streaming is now enabled. AutoPilot facts are streamed to XRay as snapshots.

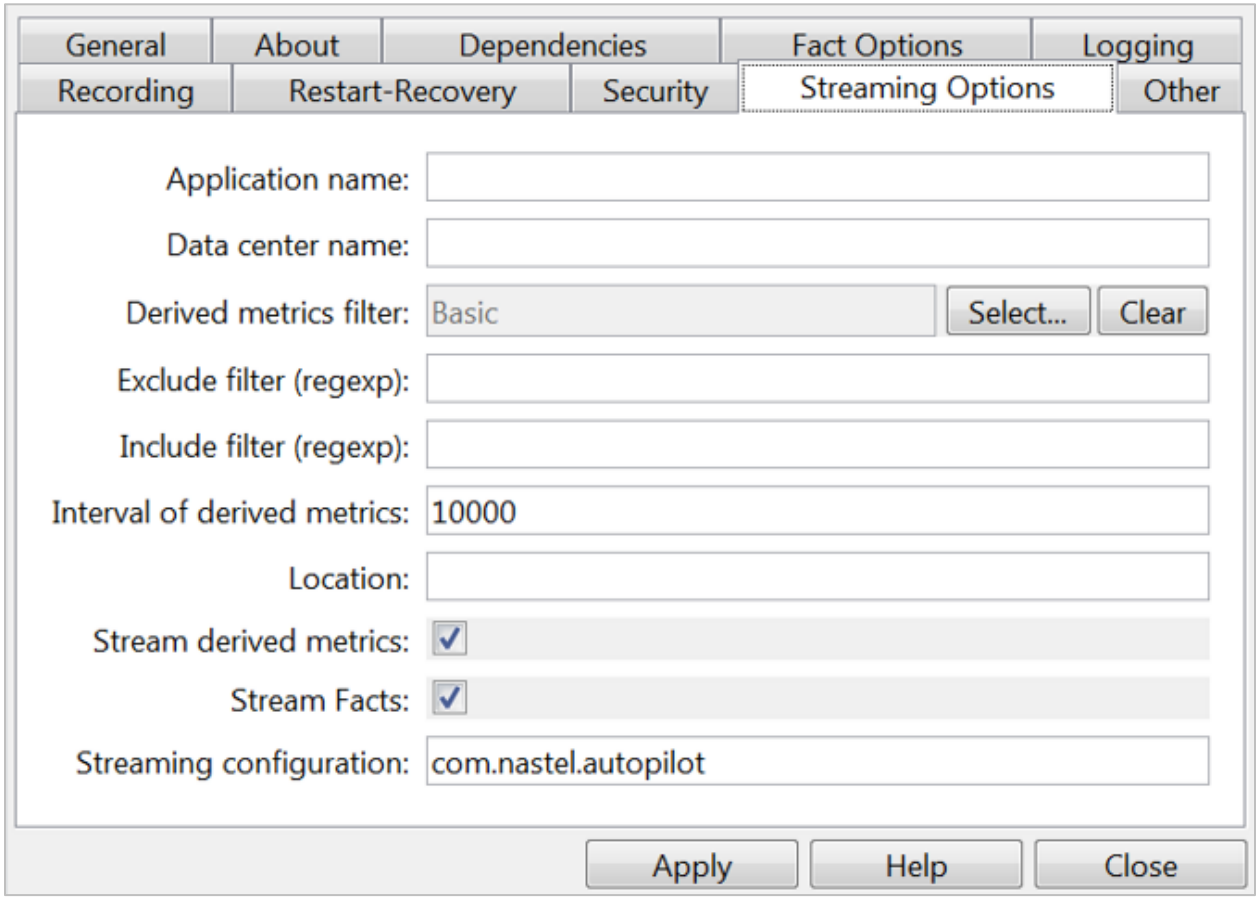


Figure 2.5.11.1.4-B. AutoPilot Streaming Options

2.5.11.2 Examples: Using the AutoPilot Integration

The following examples show a variety of ways in which you can use XRay to query AutoPilot data.

Query: jKQL> `get policies fields all`

This will return a list of all the policies and policy managers that they belong to. It shows all policies over all policy managers. This also will return all fields defined in the external data source, not just the default ones.

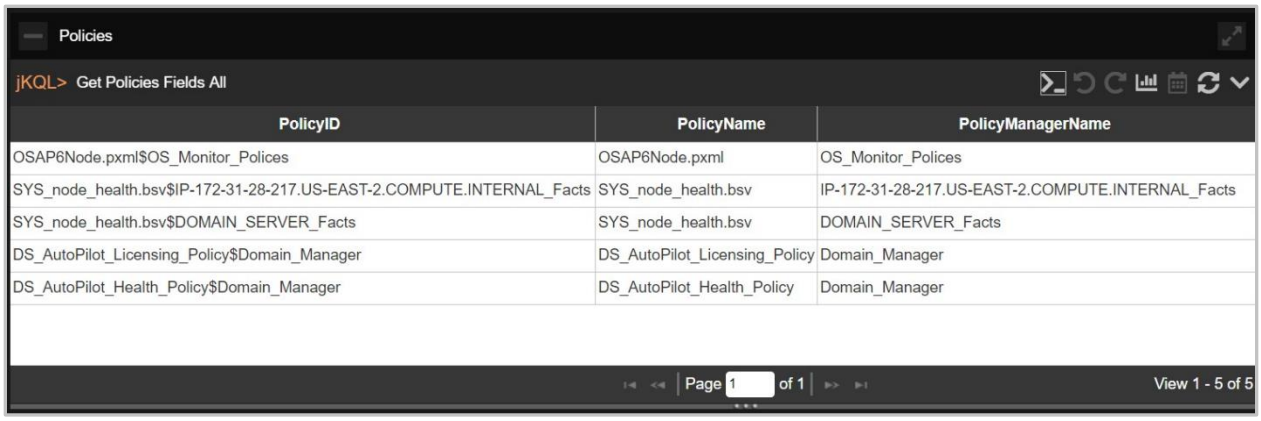


Figure 2.5.11-A. Get Policies Fields All

Query: **jKQL>** `get policies fields all where policy_manager_name='Domain_Manager'`

This will return a list of all the policies under the policy manager Domain_Manager. This would be the policies in AP EM under the policies folder under the Domain_Manager policy manager. Again, this returns all fields, not just the default ones.

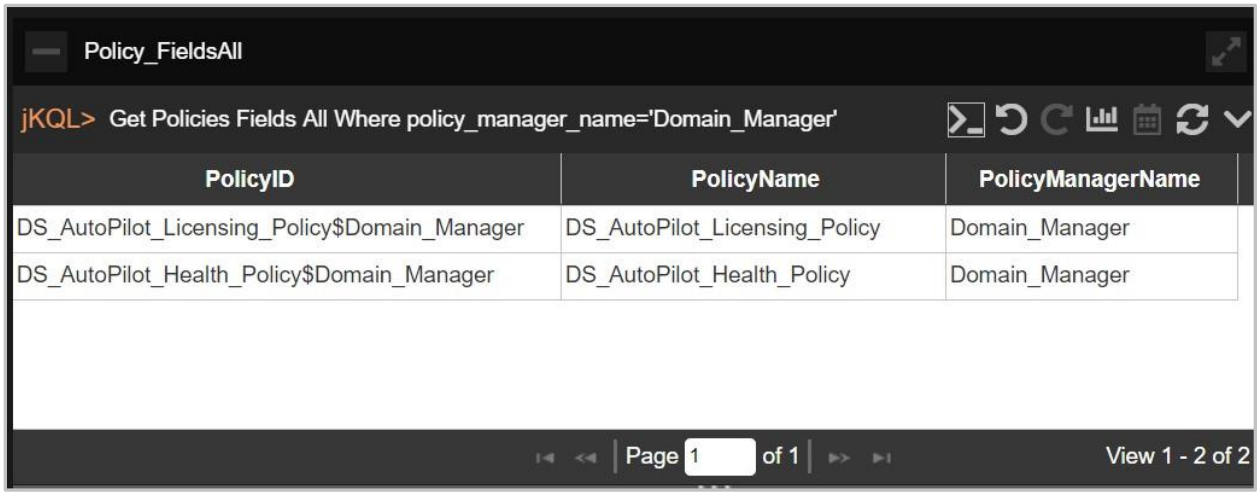


Figure 2.5.11-B. Get Policies Fields All Where…

Query: **jKQL>** `get policies of policymanager 'Domain_Manager' fields all`

This is equivalent to the previous query.

Query: jKQL> `get fields for policies`

This will give the list of fields that are associated with the "policy" item.

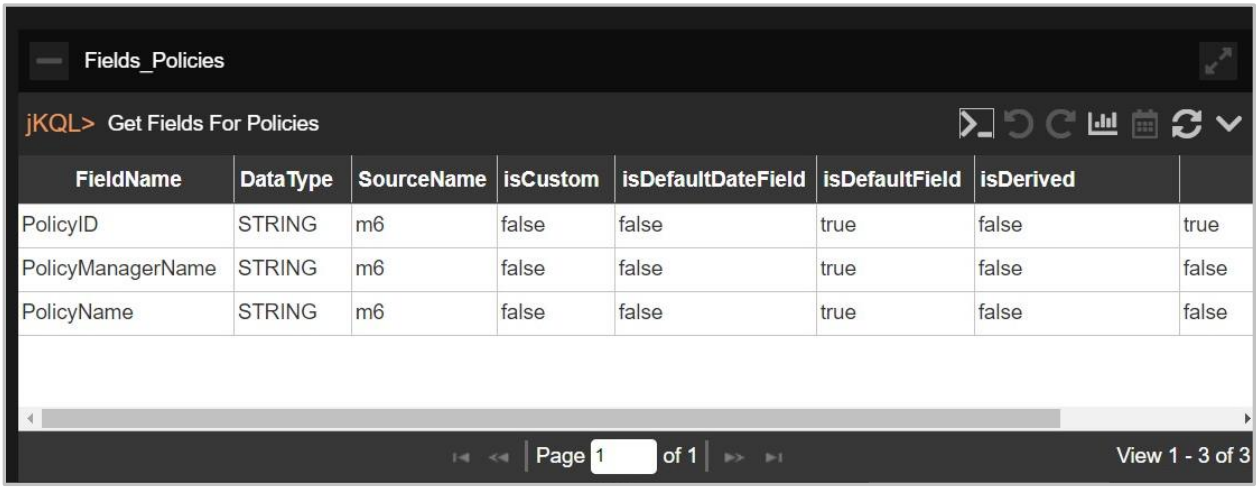


Figure 2.5.11-C. Get Fields for Policies

Query: jKQL> `get sensors fields all where policy_name='SYS_node_health.bsv' and policy_manager_name=' DOMAIN_SERVER_Facts'`

This will give a list of all sensors for policy SYS_node_health.bsv under the DOMAIN_SERVER_Facts policy manager. Omitting the policy manager name will give a list of the sensors under all policies with the name "SYS_node_health.bsv" regardless of the policy manager.

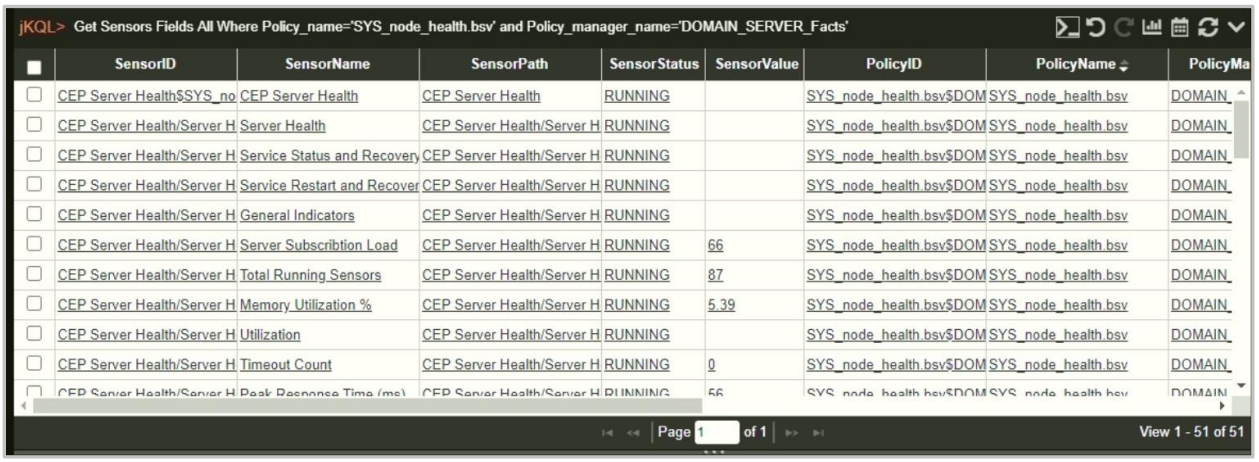


Figure 2.5.11-D. Get Sensors Fields All Where…

Query: jKQL> `get sensors fields all where policy_manager_name='LOCALHOST.LOCALDOMAIN_Facts'`

This will give all sensors for all policies under a given policy manager. Note that each sensor has a unique path (for example, PolicyManager\Policy\Sensor1\Sensor2\Sensor3) and so this would be the unique identifier for a sensor.

Figure 2.5.11-E. get sensors fields all where policy_manager_name='LOCALHOST.LOCALDOMAIN_Facts'

Another way to get sensors would be to issue a statement like:

Query: jKQL> `Get sensors of policy 'SYS_node_health.bsv'`

This is equivalent to a statement like "get sensors where policy_name='SYS_node_health.bsv'" but provides a sort of "shorthand." These are consistent with the jKQL query language.

Query: jKQL> `invoke AcknowledgeSensor using (PolicyManagerName='MyPolicies',PolicyName='Tester1', SensorPath='Untitled/Sense1')`

This will acknowledge a sensor called Sense1 in the Tester1 policy. Similarly using the verb UnacknowledgeSensor will cause the Sense1 sensor to clear the ack flag.

Query: jKQL> `invoke StopPolicy using (PolicyManagerName='MyPolicies',PolicyName='Tester1')`

This will stop the policy named Tester1. Similarly using the verb StartPolicy will start the Tester1 policy under the MyPolicies policy manager.

You can also get sensor facts by issuing statements like:

Query: jKQL> `get sensor_fact where sensorid='CEP Server Health/Server Health/Utilization/Timeout Count$SYS_node_health.bsv$PC_152_Facts'`

This will return the facts of the Timeout Count sensor that is part of the SYS_node_health.bsv policy. It will also return a predetermined set of properties that are associated with this fact (things like 'last-changed','max','min').

2.6 Import / Export

Go to **Main Menu** > **Import / Export** for the following import and export options:

- **Data** (*Section 2.6.1*).
- **Viewlets** (*Section 2.6.2*)
- **Dashboard** (*Section 2.6.3*)
- **Sets** (*Section 2.6.4*)

2.6.1 Import Data into a Dataset

Import data into your XRay system with an easy-to-use wizard. The instructions below describe the import process for XRay versions 1.5 and later. For instructions for the version 1.4 import process, refer to the *XRay 1.4 User's Guide* in the *XRay Documentation Library*.

Supported formats

The following file formats are supported to import data:

- .XLS, .XLSX
  As of version 1.5, Microsoft® Excel files are imported as datasets. In versions 1.4 and earlier, they are imported as events.
- Apache log
  Imported as events
- .CSV
  As of version 1.5, Microsoft® Excel files are imported as datasets. In versions 1.4 and earlier, they are imported as events.
- Custom
  Imported as configured by the user

To import data into XRay:

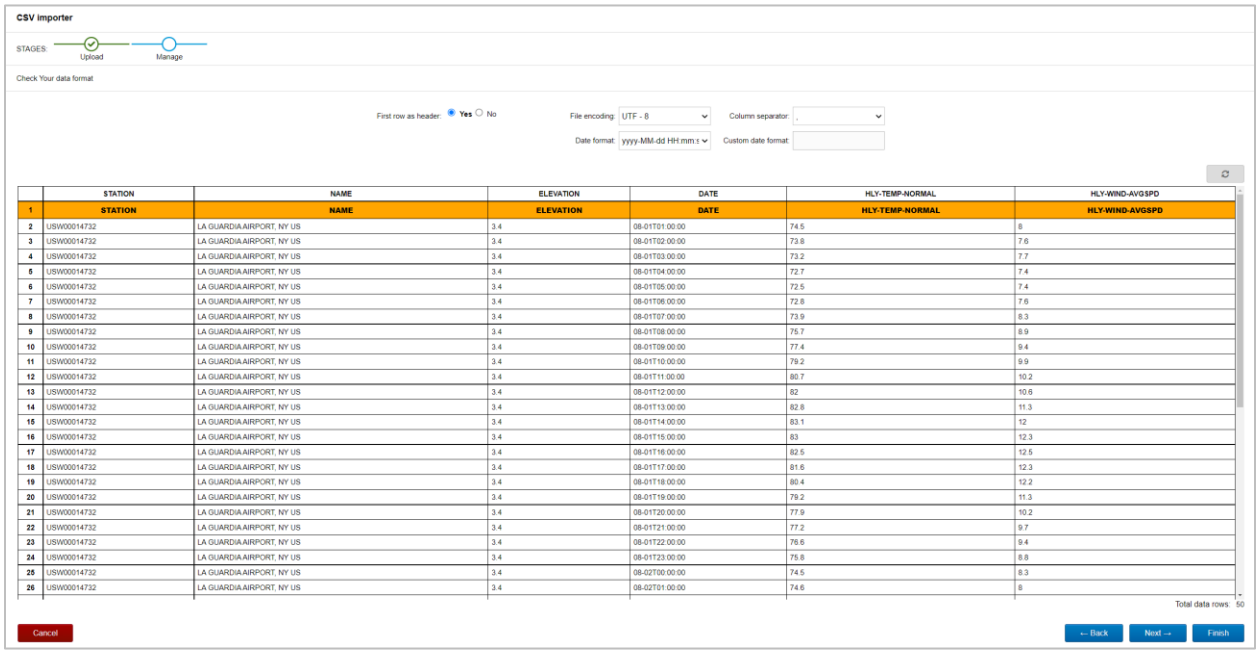
1. Go to **Main Menu** > **Import / Export** > **Data** to start the import wizard. (You can also begin by clicking the **Import Data** button 

at the top right of your dashboard or by selecting the **Analyze Your Data** option on the landing page.)
2. Select your file format on the left side of the screen.


3. Click **Choose File** to navigate to and select the import file.
4. Click **Next**.

5. Analyze the preview. Make any desired updates. The following processing options are available:

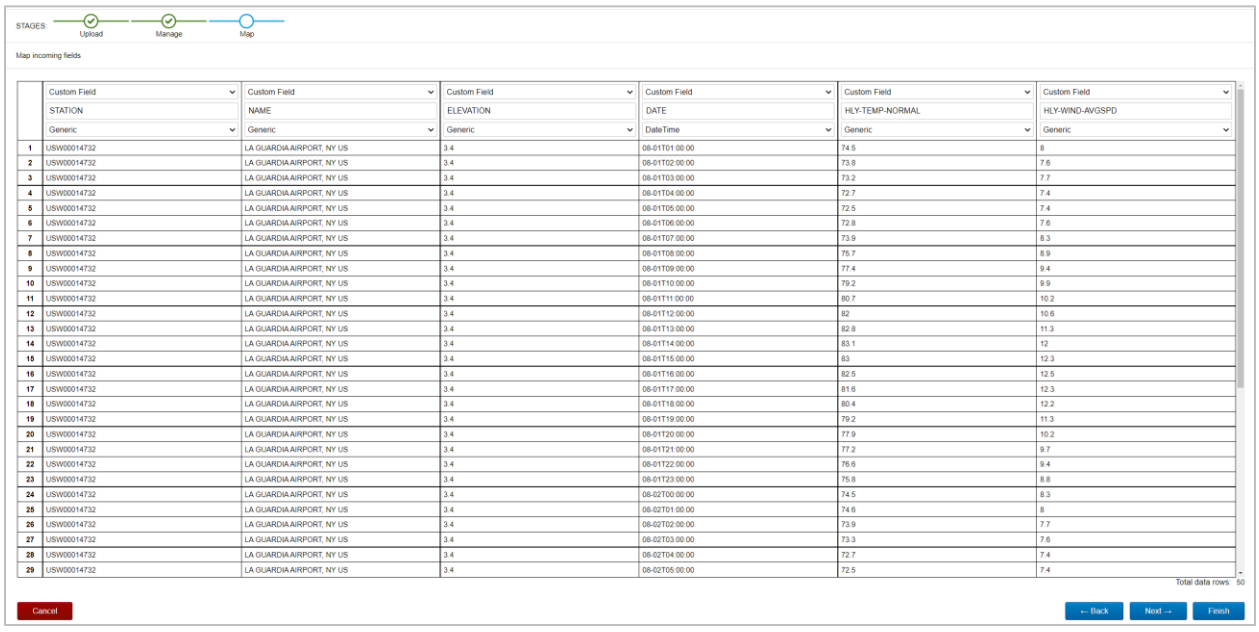
    - **First row as header:** Specify whether the first row is a header row.
    - **File encoding:** Select encoding type.
    - **Column separator:** From the drop-down menu select the column delimiter: comma, semicolon or tab.
    - **Decimal character:** Specify the decimal number delimiter: period or comma.
    - **Date formats:** Select a predefined **Date format**, or enter a **Custom date format** in the field provided.



	STATION	NAME	ELEVATION	DATE	HLY-TEMP-NORMAL	HLY-WIND-AVGSPD
1	STATION	NAME	ELEVATION	DATE	HLY-TEMP-NORMAL	HLY-WIND-AVGSPD
2	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T01:00:00	74.5	8
3	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T02:00:00	73.8	7.6
4	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T03:00:00	73.2	7.7
5	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T04:00:00	72.7	7.4
6	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T05:00:00	72.5	7.4
7	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T06:00:00	72.8	7.6
8	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T07:00:00	73.9	8.3
9	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T08:00:00	75.7	8.9
10	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T09:00:00	77.4	9.4
11	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T10:00:00	79.2	9.9
12	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T11:00:00	80.7	10.2
13	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T12:00:00	82	10.6
14	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T13:00:00	82.8	11.3
15	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T14:00:00	83.1	12
16	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T15:00:00	83	12.3
17	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T16:00:00	82.5	12.5
18	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T17:00:00	81.6	12.3
19	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T18:00:00	80.4	12.2
20	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T19:00:00	79.2	11.3
21	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T20:00:00	77.9	10.2
22	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T21:00:00	77.2	9.7
23	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T22:00:00	76.6	9.4
24	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T23:00:00	75.8	8.8
25	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-02T00:00:00	74.5	8.3
26	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-02T01:00:00	74.6	8

Figure 2.6.1-A. Import Upload

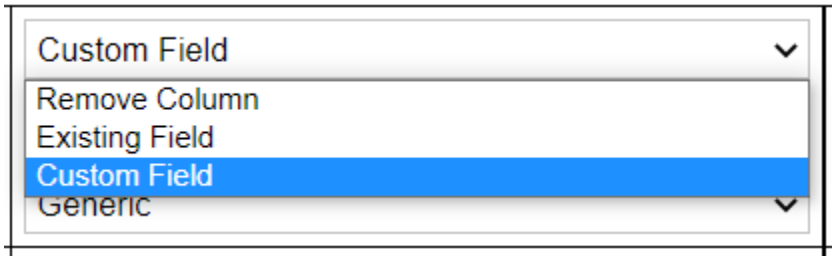
6. Click **Next**.



	Custom Field	Custom Field	Custom Field	Custom Field	Custom Field	Custom Field
	STATION	NAME	ELEVATION	DATE	HLY-TEMP-NORMAL	HLY-WIND-AVGSPD
	Generic	Generic	Generic	DateTime	Generic	Generic
1	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T01:00:00	74.5	8
2	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T02:00:00	73.8	7.6
3	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T03:00:00	73.2	7.7
4	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T04:00:00	72.7	7.4
5	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T05:00:00	72.5	7.4
6	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T06:00:00	72.8	7.6
7	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T07:00:00	73.9	8.3
8	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T08:00:00	75.7	8.9
9	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T09:00:00	77.4	9.4
10	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T10:00:00	79.2	9.9
11	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T11:00:00	80.7	10.2
12	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T12:00:00	82	10.6
13	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T13:00:00	82.8	11.3
14	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T14:00:00	83.1	12
15	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T15:00:00	83	12.3
16	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T16:00:00	82.5	12.5
17	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T17:00:00	81.6	12.3
18	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T18:00:00	80.4	12.2
19	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T19:00:00	79.2	11.3
20	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T20:00:00	77.9	10.2
21	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T21:00:00	77.2	9.7
22	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T22:00:00	76.6	9.4
23	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-01T23:00:00	75.8	8.8
24	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-02T00:00:00	74.5	8.3
25	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-02T01:00:00	74.6	8
26	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-02T02:00:00	73.9	7.7
27	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-02T03:00:00	73.3	7.6
28	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-02T04:00:00	72.7	7.4
29	USW00014732	LA GUARDIA AIRPORT, NY US	3.4	08-02T05:00:00	72.5	7.4

Figure 2.6.1-B. Import Manage

7. Additional processing options are available on this screen. You can:
    - Change the name of column headers. Select from the drop-down list or type over the existing header name.
    - Map imported data to an existing field within the XRay data model. Alternatively, select **new property** to import custom data.
    - Remove a column



8. Click **Next**.
9. By default, .CSV and .XLS (or XLSX) files are imported as datasets. Choose whether you want to import the data into a new dataset or, if one or more datasets exist, import it into an existing one. By default, a new dataset has the same name as the imported file, but you can change it.

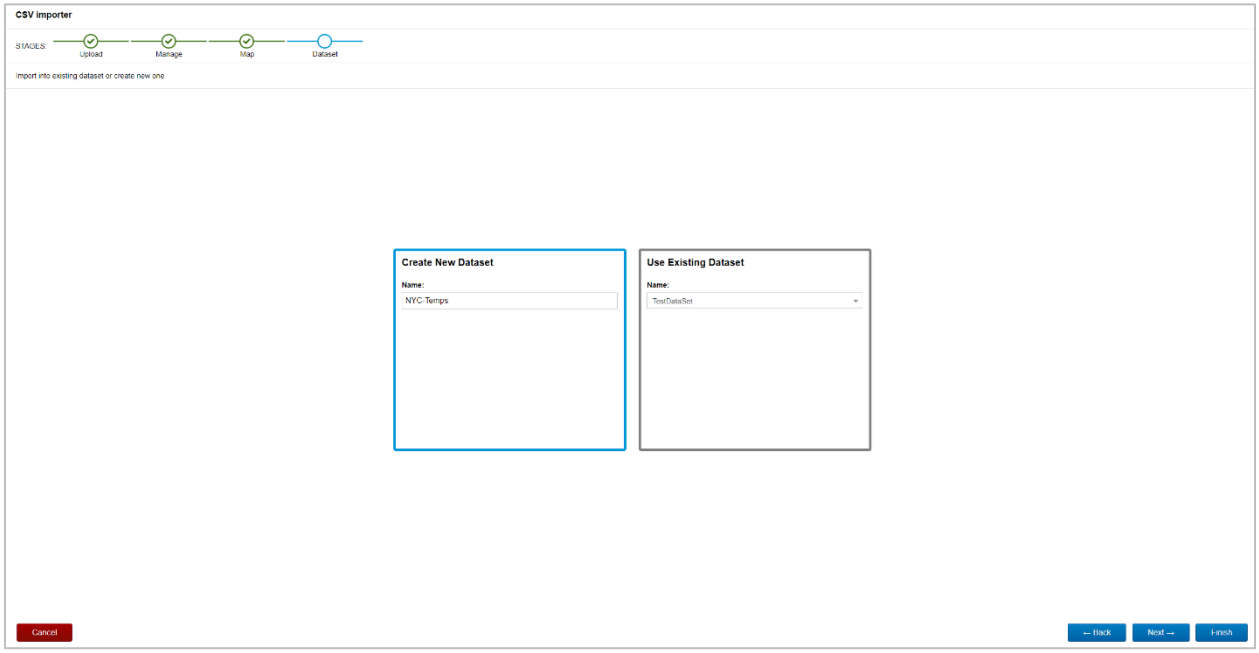
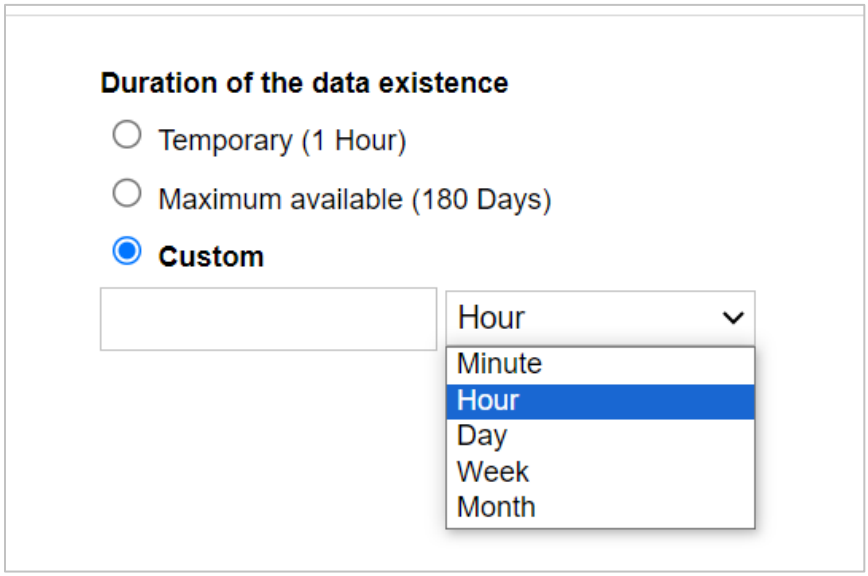


Figure 2.6.1-C. Import Map

10. (For version 1.6 and later only. If you're running version 1.5, skip to the next step.) Click **Next**. You can specify the length of time you want the dataset to be retained. Selections are *Temporary (1 Hour)*, *Maximum available (180 Days)*, or a *Custom* length of

time in minutes, hours, days, weeks, or months.



11. Click **Next**. An overview is displayed so that you can confirm the origin of the imported information.

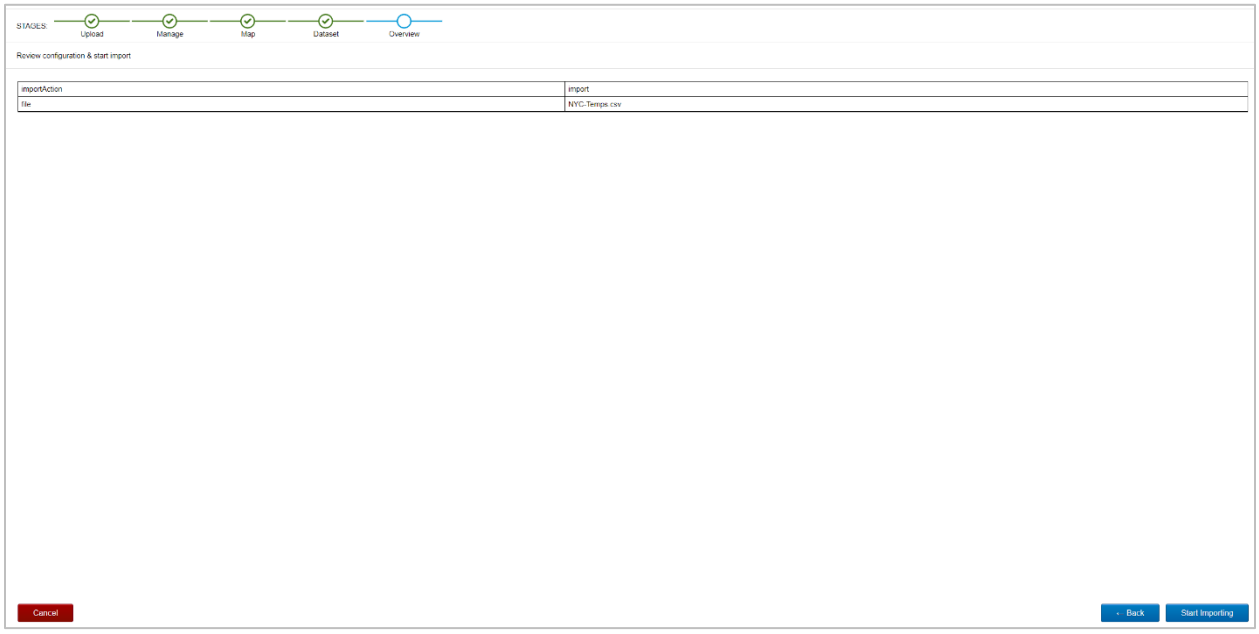


Figure 2.6.1-D. Import Dataset

12. Click **Start Import** to import the prepared file. Depending on the amount of data, the import may take a while. After you've started the process, you may close the import wizard to perform other tasks and come back to the wizard when the import has been completed.

    - If you choose to wait, then when the import process is complete, statistics about the import process are displayed on the Upload information page. Click **Next** and skip to step 3 of the View Imported Data section below.
    - If you've closed the wizard, you can resume it at this step later. See View Imported Data below.

Figure 2.6.1-E. Upload Information

2.6.1.1 View Imported Data

To view all data previously imported, go to Main Menu **> Import / Export > Data**. The *Manage Imported Data* dialog box opens. It lists all data files that have been imported into the system. Perform the following to select a file and generate a dashboard to view the imported data.

1. Click the Edit icon of the imported file for which you want to view data. The Upload information page of the Wizard is displayed with statistics about the import results.
2. Click **Next**.
3. New viewlets are automatically created based on the content of the imported data. At a minimum, a table viewlet is created. Based on the data provided, XRay also may display other suggested viewlets that you may find useful, such as charts or maps. By default, all viewlets are selected. Deselect the viewlets you do not want to add to your dashboard. Click **Next**.

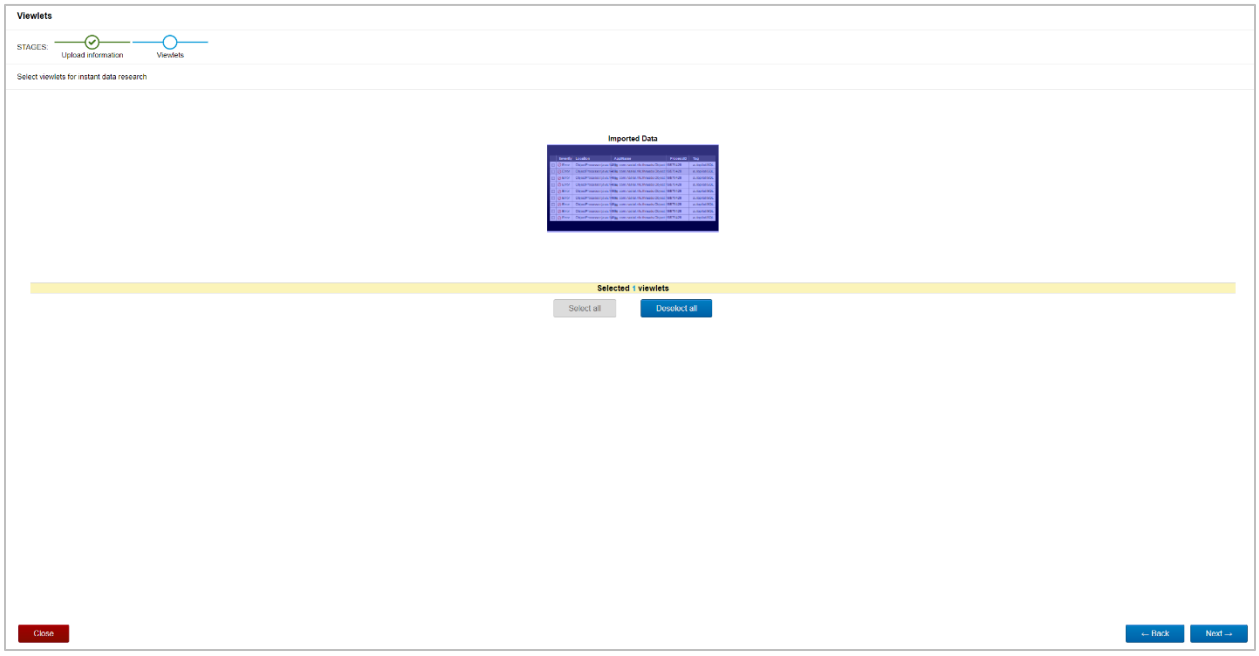


Figure 2.6.1.1-A. Import Viewlets

4. Create a dashboard for the new viewlets, or select an existing one.

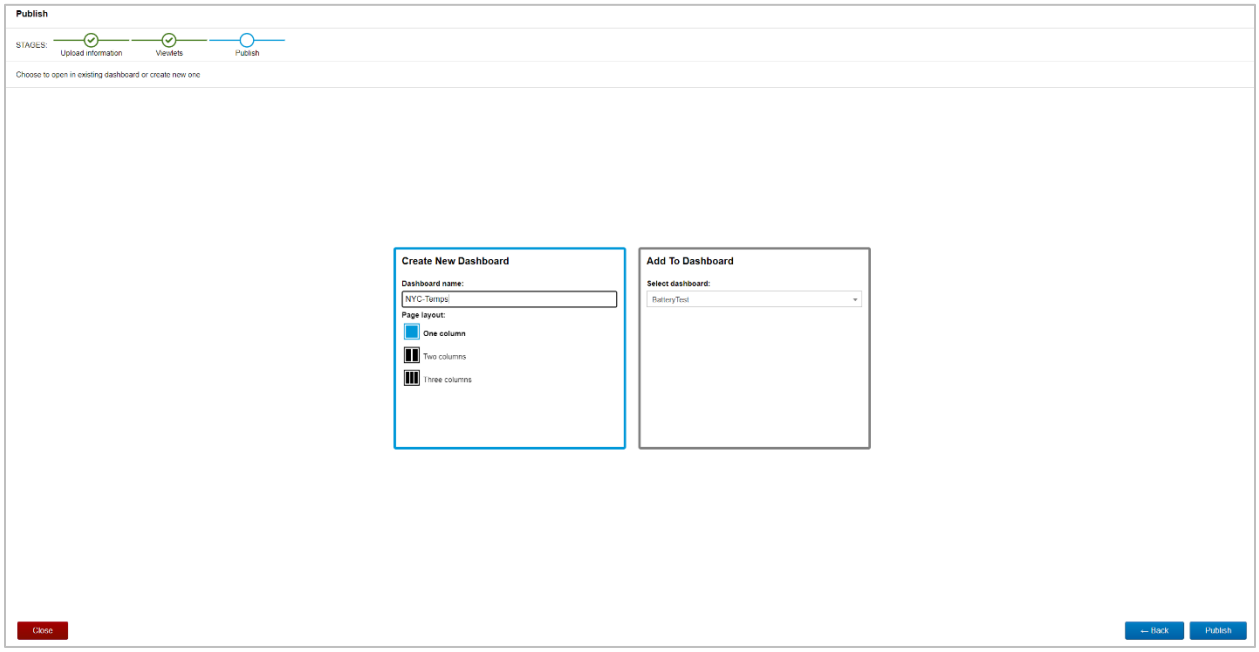


Figure 2.6.1.1-B. Publish Viewlets to a Dashboard

5. Click **Publish** to publish your viewlets and view them in the XRay user interface.

2.6.1.2 Delete Imported Data

To view all data previously imported, go to **Main Menu** > **Import / Export** > **Data**. The *Manage Imported Data* dialog box opens which lists all data files that have been imported into the system.

To delete imported data, click the delete icon at the end of the row of the Source file that you want to delete. The file and its data will be deleted.

2.6.2 Viewlets

To import or export viewlets, go to **Main Menu** > **Import / Export** > **Viewlets**. The *Import/Export Viewlets* dialog box opens. The file format used is .json or .csv.

NOTE

Viewlets can also be exported in .svg or .csv format. See Section 2.5.7, Viewlet Menu, for more information.

2.6.2.1 Import

Perform the following to import a viewlet:

1. On the *Import / Export Viewlets* dialog box, go to the **Import** tab.

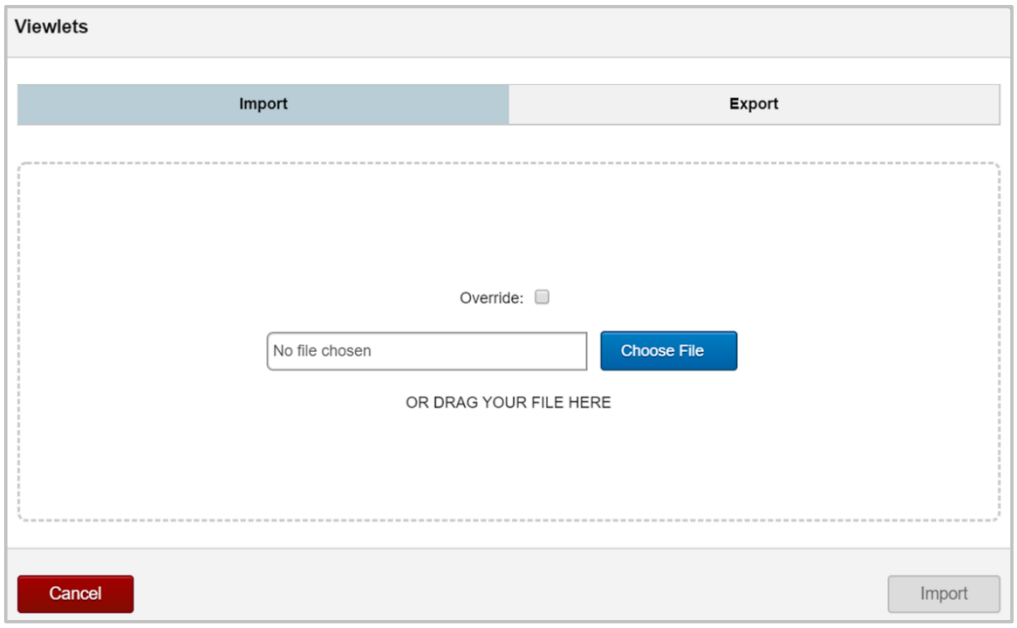


Figure 2.6.2.1-A. Import / Export Viewlets – Import Tab

2. Check off the **Override** checkbox to replace an existing viewlet. A viewlet with the same ID will be overwritten. Please note, IDs are generated automatically. The **Override** option is useful when exporting viewlets and importing them back into the same repository (for example, to restore previous viewlets).
3. Click **Choose File** to select the .json or .csv file.
4. Click **Import**. The viewlet is added to the *Open Viewlet* dialog box *(Figure 2.5.1.4-A)* and can be added to any dashboard.

2.6.2.2 Export

Perform the following to export a viewlet:

1. On the *Import / Export Viewlets* dialog box, go to the **Export** tab.

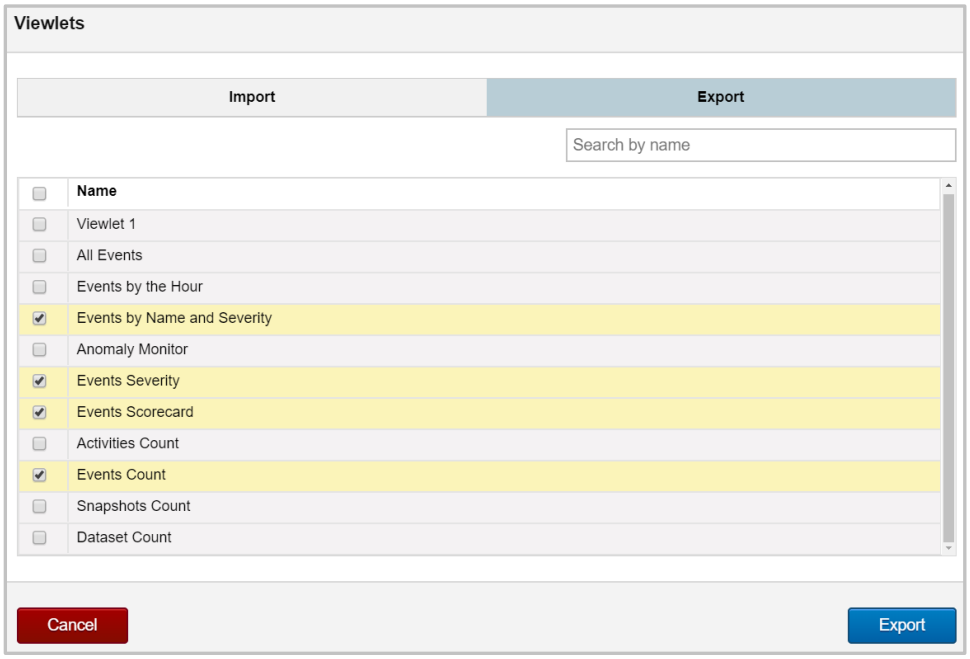


Figure 2.6.2.2-A. Import / Export Viewlets – Export Tab

2. Select the viewlets you would like to export or check the **Select All** option.
3. Click **Export**. The viewlets are downloaded in .json file format.

2.6.3 Dashboard

To import and export dashboards, go to **Main Menu** > **Import / Export** > **Dashboards**. The *Import / Export Dashboards* dialog box opens. The file format used is .json.

2.6.3.1 Import

Users can import a dashboard with a .json file. To import the file, confirm you are on the **Import** tab (the tab will be blue). Click **Choose File** to specify the import file.

Check off the **Override** checkbox to replace an existing dashboard. A dashboard with the same ID will be overwritten. Please note, IDs are generated automatically. The **Override** option is useful when exporting dashboards and importing them back into the same repository (for example, to restore previous dashboards).

Click **Import** to start the import process. Once the import is finished, a confirmation message will appear.

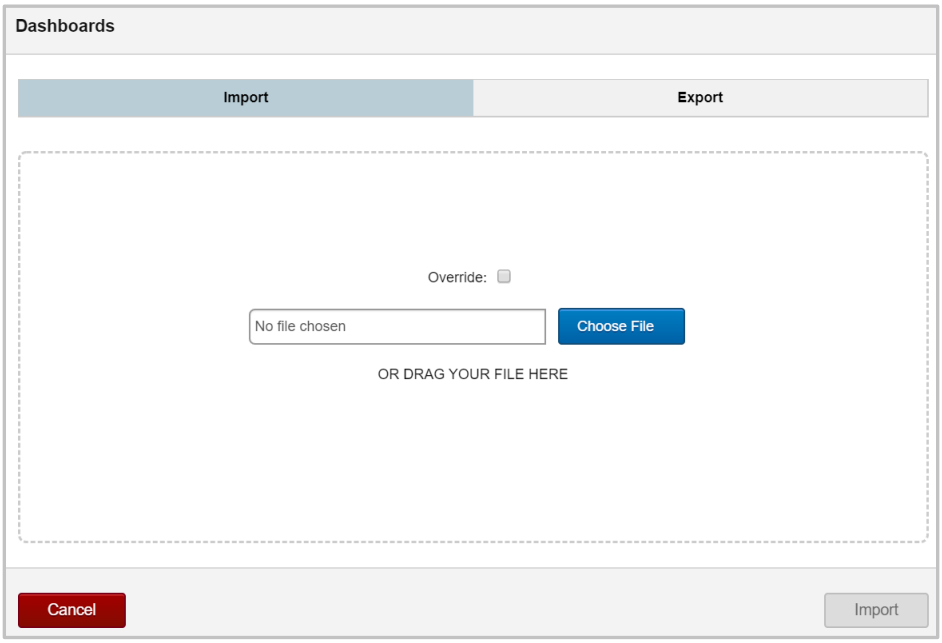


Figure 2.6.3.1-A. Import Dashboards

2.6.3.2 Export

The dashboards can be exported to a .json file. To export a dashboard, go to the **Export** tab of the *Import / Export Dashboards* dialog box. A list of your saved dashboards will display. Select all desired dashboards or check off **Select All**. Click **Export** and the file, **ExportDashBoards.json**, will download.

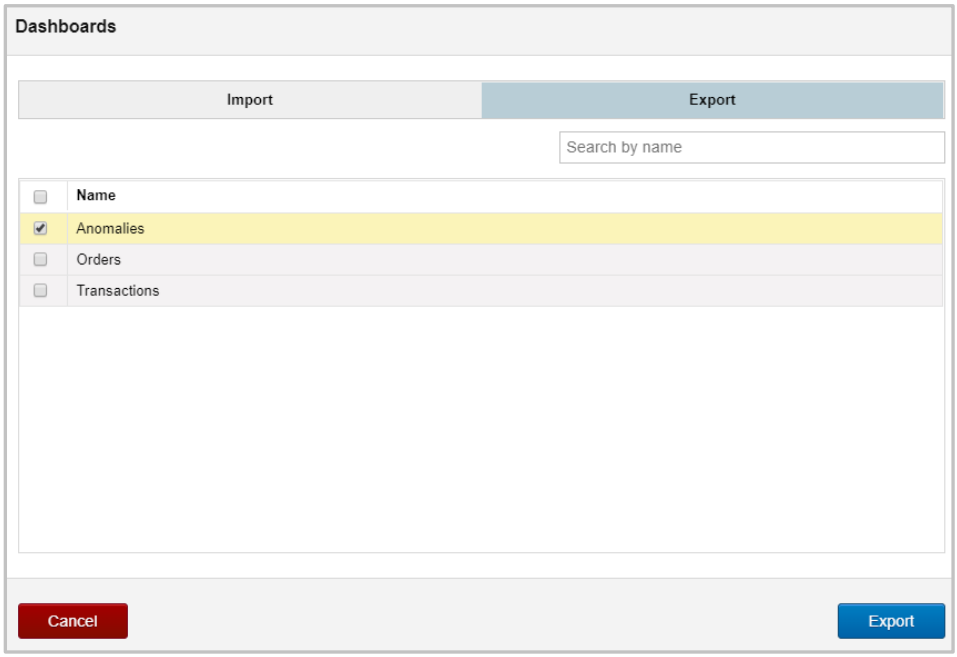


Figure 2.6.3.2-A. Export Dashboards

2.6.4 Sets

To import and/or export sets, go to **Main Menu** > **Import / Export** > **Sets**. The *Sets* dialog box opens. The file format used is .csv. For more information on sets, please see Sets.

2.6.4.1 Import

To import sets, go to the **Import** tab of the *Sets* dialog box. Click **Choose File** to specify the import file or drag and drop your file.

Check off the **Override** checkbox to replace an existing set. A set with the same ID will be overwritten. Please note, IDs are generated automatically. The **Override** option is useful when exporting sets and importing them back into the same repository (for example, to restore previous sets).

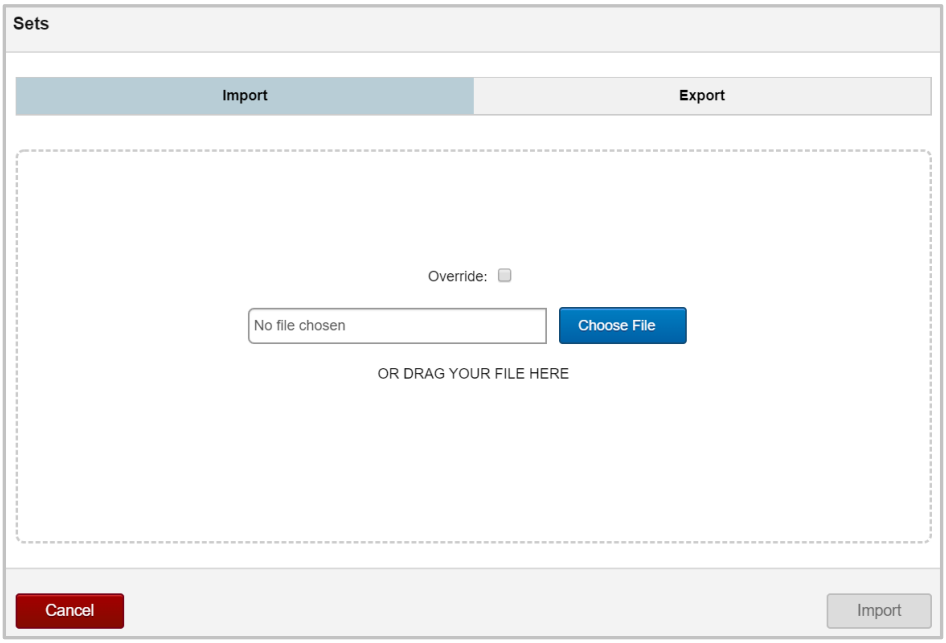


Figure 2.6.4.1-A. Import Sets

To change the file, click the **Change** button or drag a new file. Click **Import** to start the import process. Once the import is finished, a confirmation message will appear.

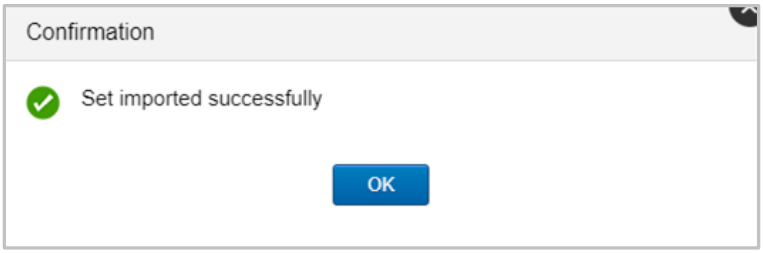


Figure 2.6.4.1-B. Import Sets – Confirmation

2.6.4.2 Export

Go to the **Export** tab of the *Sets* dialog box to export a set. A list of existing sets (imported or created) will appear. Select all desired sets or check the **Select All** check box immediately to the left of the **Name** header. Click **Export** to download the files.

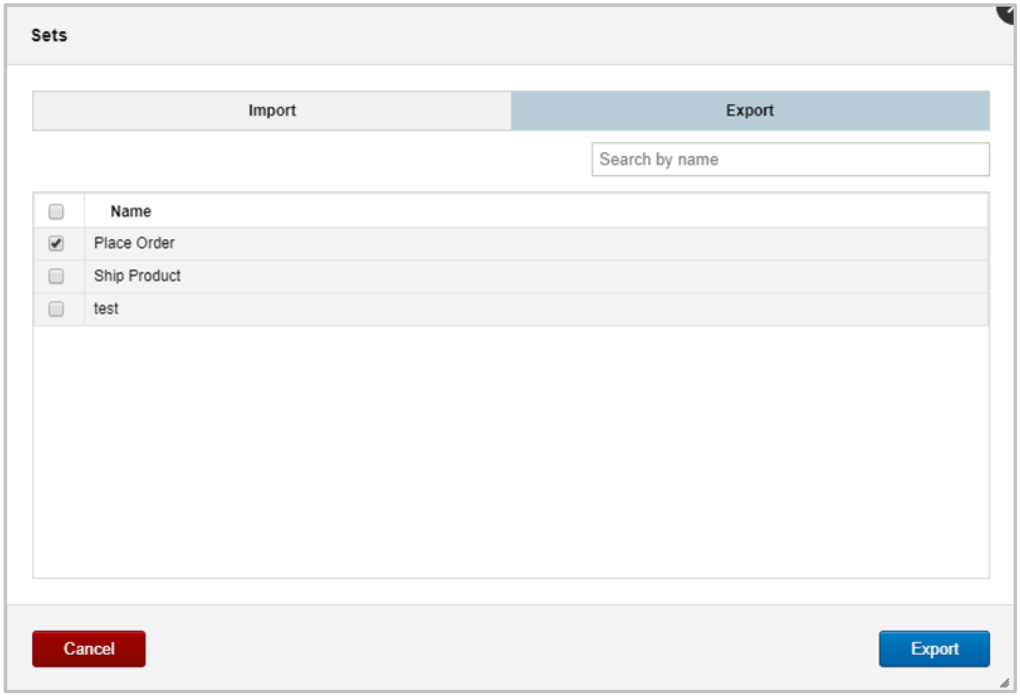


Figure 2.6.4.2-A. Export Sets

2.7 Social Integration: Slack

XRay Slack integration offers the following benefits:

- Create alerts that send Slack messages to individual users or channels based on specific criteria.
- Send queries directly from the XRay user interface.
- Send a link to any viewlet (even shared viewlets) directly from the XRay user interface.

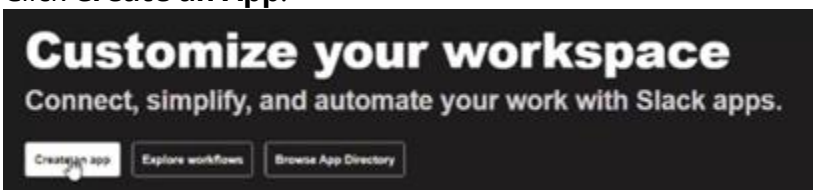
2.7.1 Setup and Use

Step 1. Create a Slack application

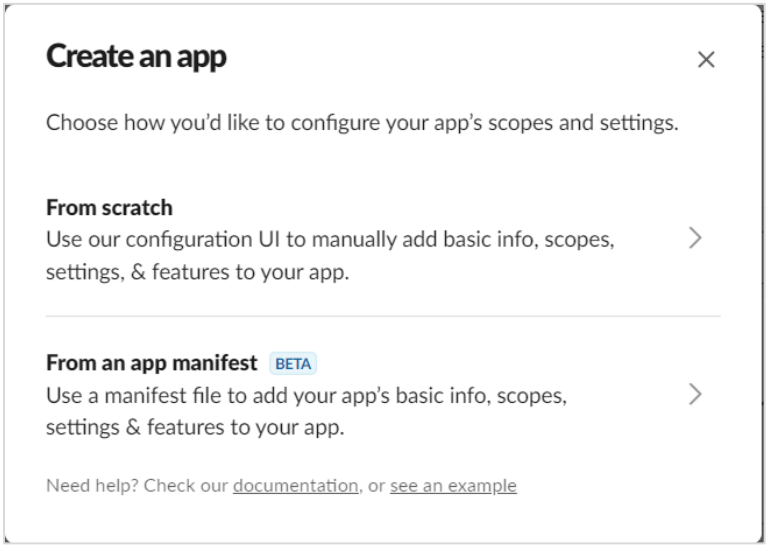
The first step to setting up Slack integration in XRay is to create an application in Slack. Creating and configuring an application in Slack allows XRay to interact with Slack. You can do this from a browser or through a desktop client.

1. On the Slack website, begin the process of creating an app.
    a. Log into Slack.
    b. Go to *https://slack.com/apps*.
    c. Click (in the upper-right corner).

d. Click **Create an App**.

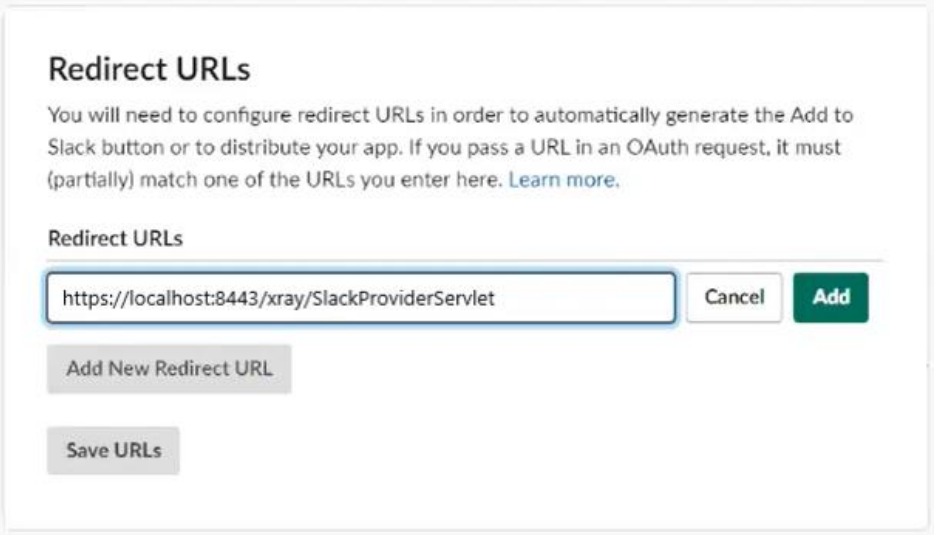


2. On the Create an app dialog, select **From Scratch >**.

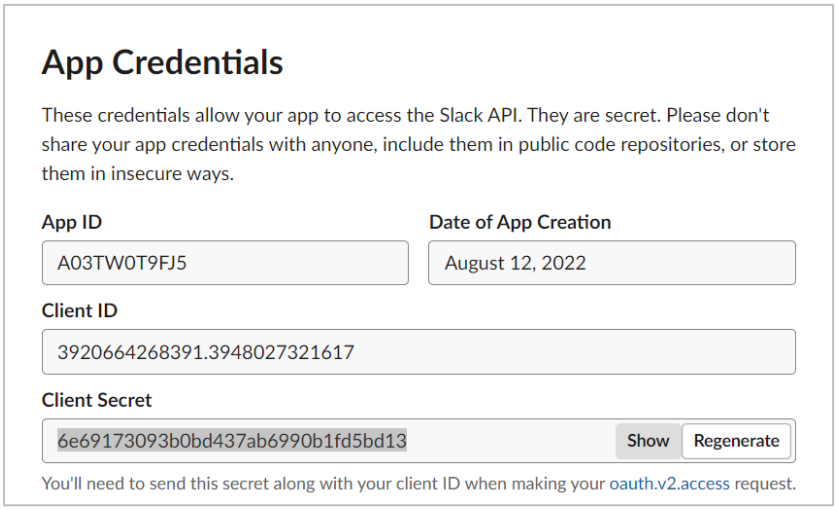


3. Enter an **App Name**. For future reference, make sure the name indicates that the app is for XRay.
4. If applicable, specify a workspace.
5. Click **Create App**.
6. Set up permissions for the app.
    a. In the left panel, under Features, select **OAuth & Permissions**.
    b. In the Scopes section, under **Bot Token Scopes**, click **Add an OAuth Scope**.
    c. Add the following permissions. As you type, suggested permissions will be listed for you to choose from.
        1. team:read
        2. channels:read
        3. users:read
        4. profile:read
        5. users:read.email
        6. chat:write
7. Add Redirect URLs.
    a. In the Redirect URLs section, provide the address of the XRay site in the format <host>/xray/SlackProviderServlet. At a minimum, the address you provide must include the entire URL up to and including the **/xray/** subdirectory. For example: *https://xraytest.nastel.com/xray/SlackProviderServlet*.
    b. Click **Add**.

c. Click **Save URLs**.



8. The **Client ID** and **Client Secret** are generated by Slack and are needed to add the Slack application to XRay. Go to the Basic Information section in the left pane and make note of these two values. Click **Show** to view the value in text form so you can copy it for use in XRay setup.



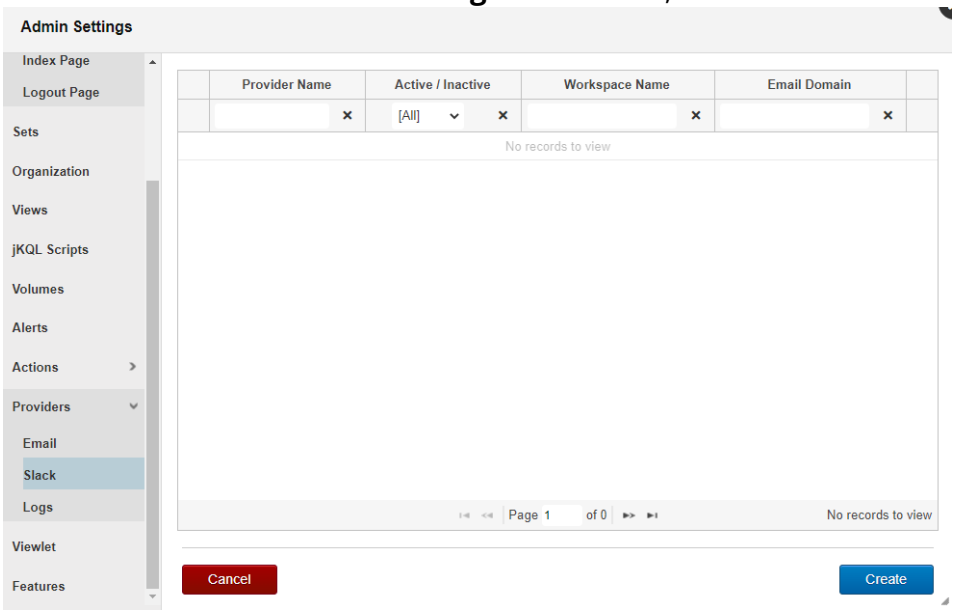
Step 2. Set up Slack as a Provider in XRay

IMPORTANT!

Before performing step 2, complete the following prerequisite steps to configure Slack as an XRay external data source:
1. Load config-slack.xml from the jkool-dbapi/jkool-dbapi-solr-1.5.X/config/ext-data-src folder.
2. Restart the CEP server and Tomcat.

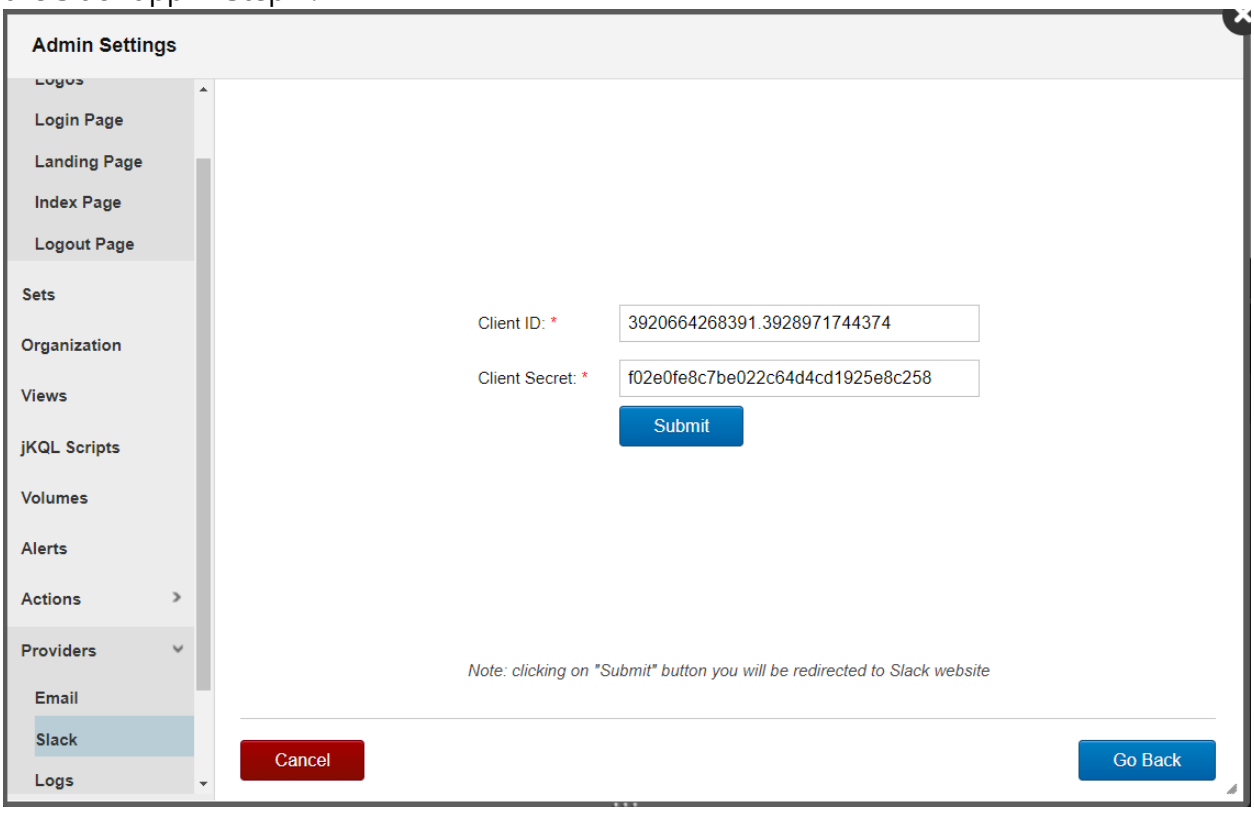
Now you'll use the Client ID and Client Secret values from Step 1 to add Slack as a Provider in XRay.

1. From **Main Menu > Admin Settings** >**Providers**, select **Slack**.

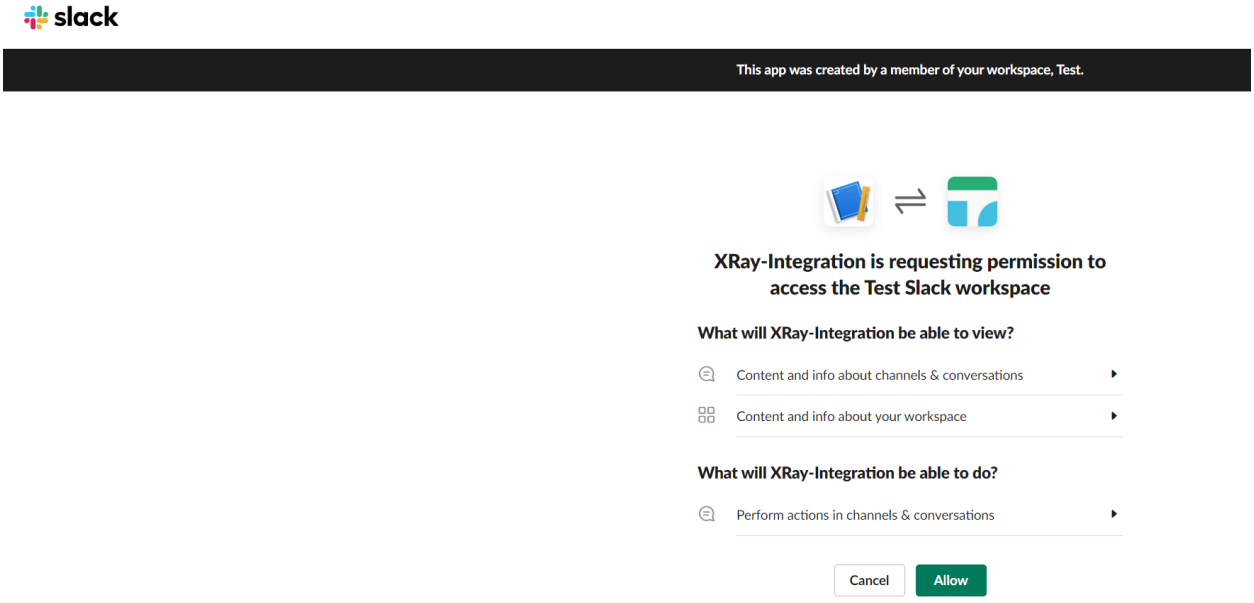


2. Click **Create**.

3. Enter the **Client ID** and the **Client Secret** that were generated when you created the Slack app in Step 1.



4. Click **Submit**. The Slack web page displays a prompt that asks you to allow the app to access the workspace.



5. Click **Allow**. You are returned to the XRay interface.

Step 3. Set up XRay Actions for Slack

By setting up an XRay Slack action, you lay the framework for messages to be sent to users or channels (groups of users). Slack actions send Slack messages; email actions send emails. (To learn how to set up an email Provider, see section *3.1.10*, Providers.) In Step 4, you'll associate these actions with alerts so that messages are sent in certain cases.

A Slack action sends a message to Slack users, Slack channels, or both. Messages can include text, variables, or both.

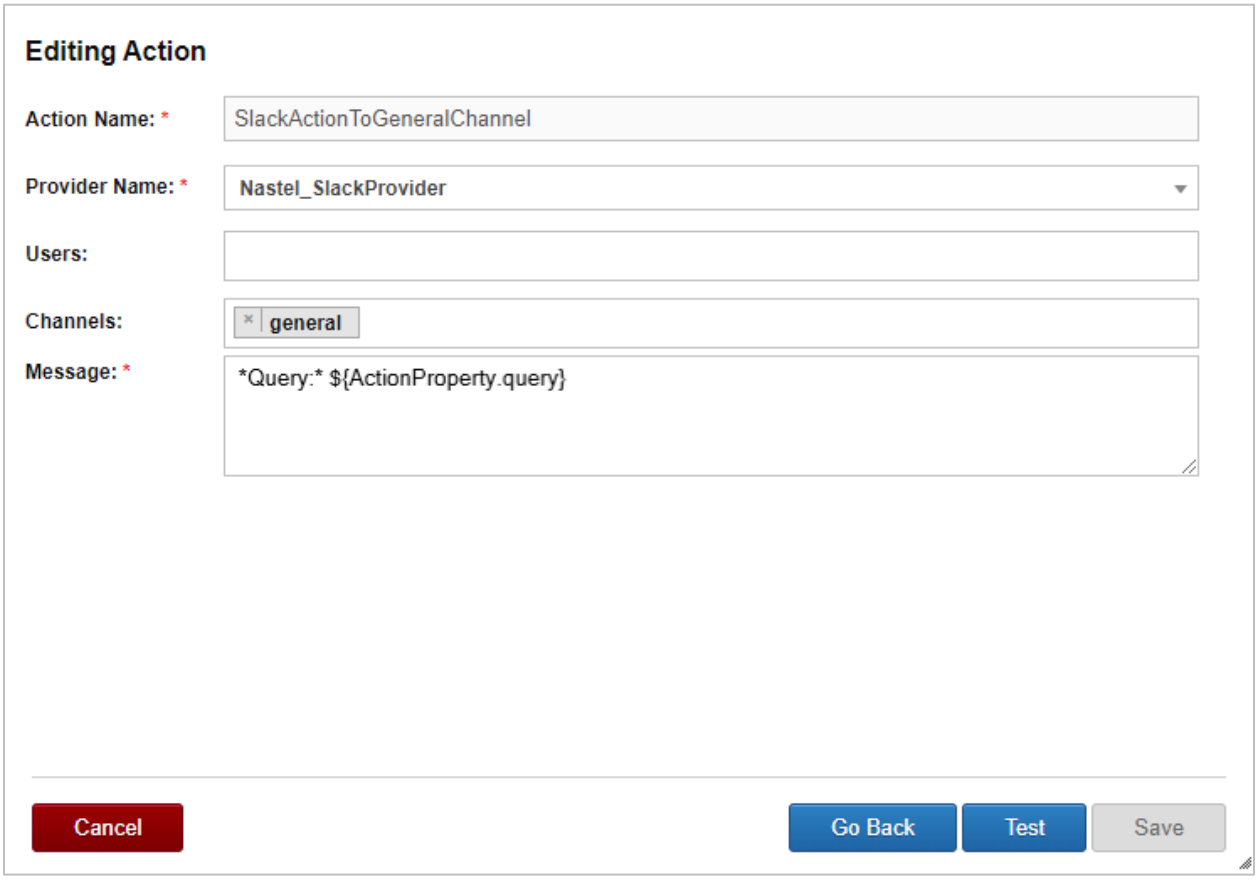
To create an action:

1. From **Main Menu > Admin Settings > Actions**, select **Slack**.
2. Click **Create**.
3. Enter an **Action Name**.
4. Select the **Provider Name**. Users and channels are made available based on the provider you select.
5. Enter users, channels, or both.
    - Enter one or more **Users** in the field provided.
    - Enter one or more **Channels** in the field provided.

NOTE You can remove a user or channel at any time by clicking the **X** on the user or channel you want to remove.

6. Enter a **Message**. For more information about creating and formatting messages, see Templates and Message Formatting.

7. To verify that messages can be sent through the provider, click **Test**. XRay displays a confirmation message to inform you that the Slack message was sent successfully.
8. Check the user account or channel to ensure that a test message has arrived.
9. Click **Save**.



Step 4: Create Alerts to Trigger XRay Actions

Alerts inform individuals or groups about specific scenarios.

You can create an alert that is based on a set of objectives. Alerts are associated with actions, which send an email or a Slack message based on the defined objectives.

Follow the instructions for creating an alert in section *3.1.8*, Alerts. During alert setup, select the Slack action you want to associate with the alert.

2.7.1.1 Send Queries and Links from Viewlets to Slack

You can send queries and links directly to Slack from the XRay user interface.

2.7.1.1.1 Send Queries to Slack

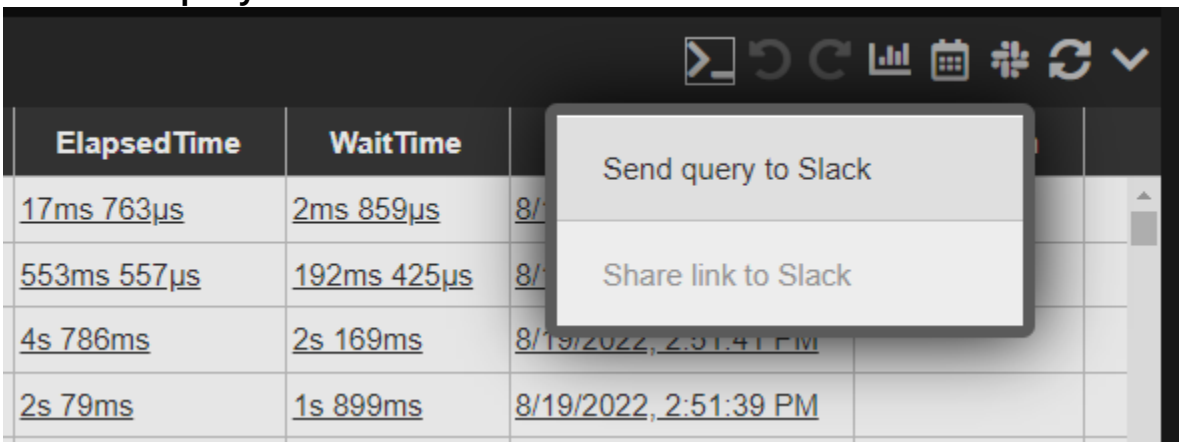
NOTE

To send queries to Slack as described here, you must have a Slack action named "QueryToSlack". For suggested message templates, see *Templates and Message Formatting*.

From a viewlet, you can send a query to Slack.

1. Click the Slack icon in the viewlet tool bar.

2. Select **Send query to Slack**.



XRay displays a confirmation message when the message is sent.

3. Click **OK**.

A Slack message has been sent with the query.

2.7.1.1.2 Send Links to Slack

> NOTE To send links to Slack as described here, you must have a Slack action named "LinkToSlack". For suggested message templates, see *Templates and Message Formatting*.

You can send a link to any viewlet, including shared viewlets, to Slack.

1. Click the Slack icon in the viewlet tool bar.
2. Select **Send link to Slack**.

2.7.2 Templates and Message Formatting

This section covers creating templates and formatting messages.

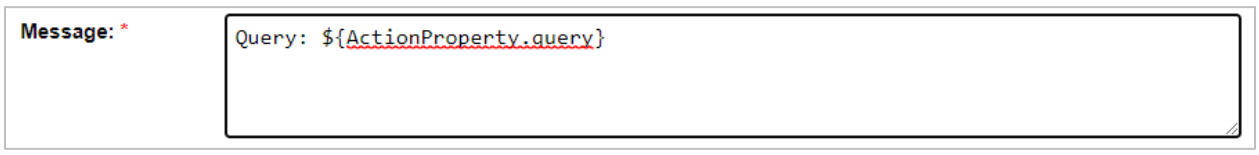
Creating "Templates" Using Variables

Messages can include a combination of text and variables.

Example 1

In the example below, a query is sent within the message. The word "Query" is text. The remainder of the message allows a variable to be inserted.

Query: ${ActionProperty.query}



Example 2

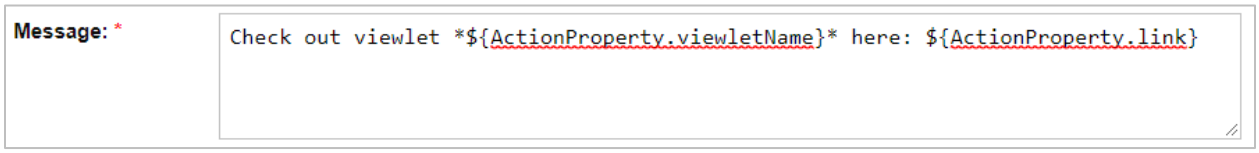
This is an example of a message for the QueryToSlack action.

RepositoryID: ${ActionProperty.repositoryID}
Dashboard: ${ActionProperty.dashboardName}
Viewlet: ${ActionProperty.viewletName}
Query: ${ActionProperty.query}
Error: `${ActionProperty.errorMessage}`

Example 3

This is an example of a message for the LinkToSlack action.

Check out viewlet *${ActionProperty.viewletName}* here: ${ActionProperty.link}



Message Formatting

Slack messages can be formatted using mrkdwn. For example, to make text bold, add asterisks around it, as shown here:

Query: ${ActionProperty.query}

TIP

For more Slack message formatting options, see *https://api.slack.com/reference/surfaces/formatting*.

Chapter 3: Settings

3.1 Admin Settings

When **Admin Settings** is selected on the **Main Menu**, the *Admin Settings* window opens. See sections 3.1.1 – 3.1.13 below for an explanation of the administrative options available.

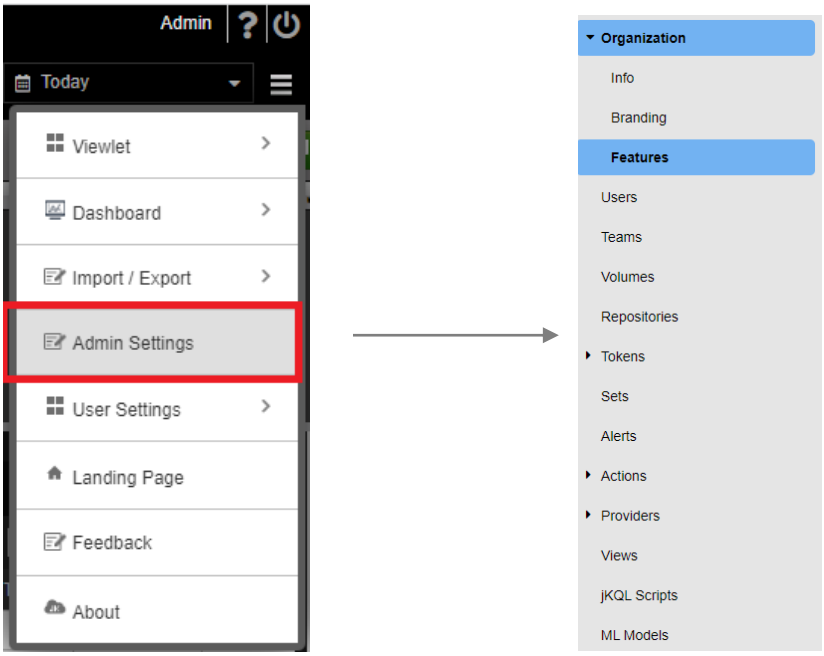


Figure 3.1-A. Admin Settings Menu

NOTE

Only administrative users with repository permissions will have this option.

3.1.1 Organization

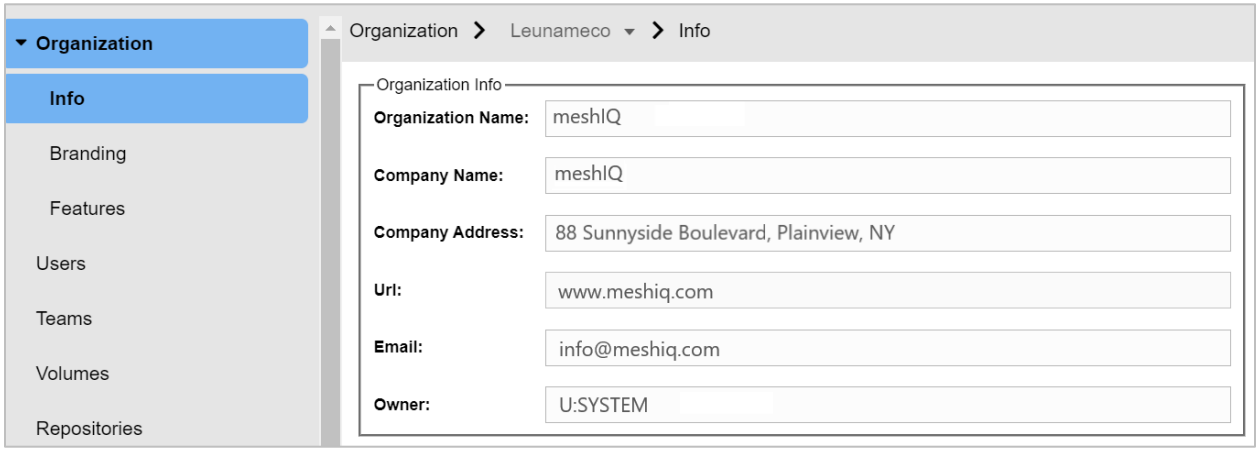
Select **Organization** from the side menu of *Admin Settings* (*Figure 3.1-A*) to modify organization information or branding, or to view the features that are turned on for your account.

NOTE

If your license permits, you can add/update users to organizations on the following screen.
If running on premise, external security is used where the users are defined externally and come from AutoPilot although the teams and other options are still configured there (teams cannot be created).

3.1.1.1 Info

The **Info** tab stores your organization's contact information.



3.1.1.2 Branding

The Branding tab allows users to configure organization-specific branding of application pages.

Logos. Select Logos from the Branding sub-menu to change the logos appearing on the main application window. Color images are subject to a maximum bit depth of 24.

Login Page. Configure the login page settings.

Landing Page. The Landing Page portion of Branding allows you to specify the text of the landing pages. The character limit is displayed immediately to the right of each field.

Subscription. Stores the Subscription URL.

Contacts. Here you can add the contact email, and specify URLs for support, feedback, and collectors. The Collectors URL is for open-source collector download links.

System. The Query prompt text field allows administrators to customize the "jKQL>" field label in viewlets (this is the orange field label that appears immediately to the left of a viewlet's query). The Logout Text allows you to specify the text a user sees when they log out.

3.1.1.3 Features

Select **Features** from the side menu of *Admin Settings* (*Figure 3.1-A*) to view a list of features which are active and available to your organization.

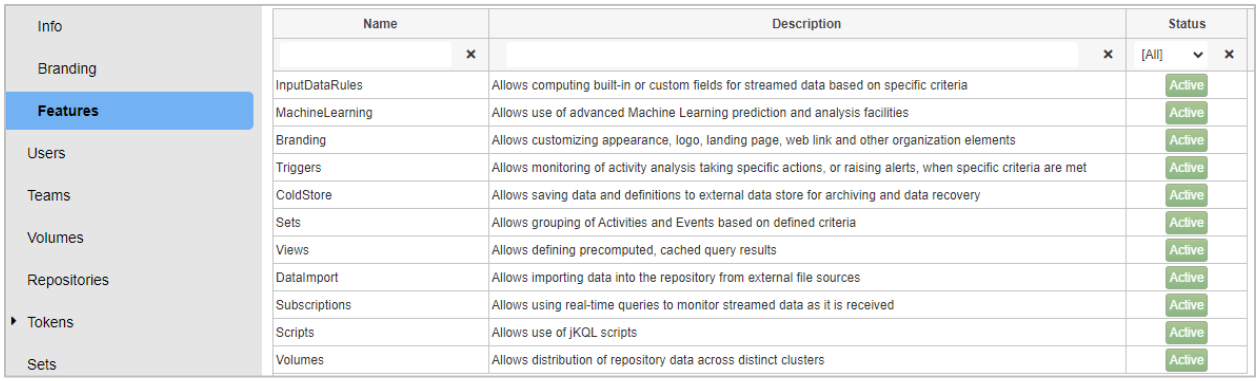


Figure 3.1.1.3-A. Features

3.1.2 Users

For on-premises installations, external security is used. Users are defined externally and come from AutoPilot.

3.1.3 Teams

For on-premises installations, external security is used. Teams are defined externally and come from AutoPilot.

3.1.4 Volumes

Volumes allow XRay to be configured so that different types of data can be held in separate clusters. For example, one small master SolrCloud instance on one node (or two nodes for replication) might serve as a master volume. Other volumes that are separate clusters would connect to the master one.

Volumes can handle additional storage for different repositories. Only the XRay Administrator user can administer (create, edit, and delete) volumes, and only the Administrator user can assign repositories to volumes. Non-Administrator users only have read access: on the Volumes tab, they can see the list of volumes, but they cannot do anything with them.

You can only assign a repository to a volume when the repository is being created, not thereafter. If a repository is not on a dedicated volume, you cannot move it there later. Likewise, if it is on one volume, you cannot then move it to another volume.

3.1.4.1 Prerequisites

To use volumes, the following requirements must be met:

- You must have more than one Solr instance running. You can install another Solr instance either on a remote server or on the same server using a different port. (Two Solr instances on same server require different ports.)
- All volumes must be at the same Solr version as the master volume (for XRay version 1.5, Solr 8; for XRay versions 1.4 and earlier, Solr 6).

3.1.4.2 When Solr Authentication is Enabled

XRay versions 1.4 and prior:

- If the Solr cluster for any volume requires authentication, then its password must be the same as that of the master volume. Therefore, you must configure Xray to use Solr authentication by specifying the user/password in global.properties (jkool.db.server.user and jkool.db.server.pwd), even if the master cluster is not using it.

In XRay 1.5 and later:

- XRay is enhanced to allow each volume cluster to have a different user and password. When setting up the volume, use the new Volume User and Volume Password fields to fill in the Solr user and password needed to connect to the cluster.

3.1.4.3 Create a Volume for a Solr Installation on a Remote Server

Volumes can be created using the Query Browser. To install the Query Browser, look for jkool-qb.sh in either the jkool-dbai-1.x.tar.gz package or the separate Windows package.

1. Download and install Solr on the remote server.
2. Start the remote Solr instance.

3. Check the SolrCloud console.
4. In the solr.in.sh file, provide the Zookeeper host. Example: ZK_HOST="172.16.31.27:2181/xraysolr". If you are running Solr in cloud mode, you must provide the Zookeeper Root (zkroot). If the Zookeeper Root is not provided, an exception is generated while creating Solr cores.
5. Enter the IP address of the master volume as the 'ZKHOST' and the IP address of the remote volume as the 'SOLRHOST'.
6. Create the volume using the query browser.

   For XRay Version 1.4, or for Version 1.5 *without Solr Authentication enabled*:

   ```
   Create Volume '<Volume Name>'
       Description='Local node Solr Cluster',
      Url='http://<solr host>:<solr port>',
      Properties=('SOLRHOST'='<solr host>',
                  'SOLRPORT'=<solr port>,
                  'ZKHOST'='<ZooKeeper host>',
                  'ZKPORT'=< ZooKeeper port>),
                  'ZKROOT'='< ZooKeeper root>')
   ```

   In XRay Version 1.5, when Solr Authentication is enabled and used to connect to the volume, you can add the user and password, as shown bolded below:

   ```
   Create Volume '<Volume Name>'
      Description='Local node Solr Cluster',
      Url='http://<solr host>:<solr port>',
      VolumeUser='<username'>,
      VolumePassword='<pwd>,
      Properties=('SOLRHOST'='<solr host>',
                  'SOLRPORT'=<solr port>,
                  'ZKHOST'='<ZooKeeper host>',
                  'ZKPORT'=< ZooKeeper port>),
                  'ZKROOT'='< ZooKeeper root>')
   ```

7. Create a repository and assign the newly created volume to this repository using the following query:

   ```
   Create Repository '<Repo Name>', OrganizationName='<Organization Name>', VolumeName='<Volume Name>'
   ```

   
   NOTE

   Query Browser is required because repositories created through the user interface use the master volume.

8. Go to misc/schemas and run the create-core.sh command to create the Solr cores in the remote Solr instance.

   ```
   ./create-cores.sh -sh <SOLR HOST> -sp <SOLR PORT> -zh <ZOOKEEPER HOST> -zp <ZOOKEEPER PORT> -zr <ZOOKEEPER ROOT>
   ```

   This command will create all the Solr cores in the new Solr cluster. Validate that the cores have been created by going to the remote Solr instance in the browser console.

9. Log in to XRay, go to the new repository, and create the dashboard. If the dashboard is not created successfully, and instead you receive a message saying that the

volume is not assigned to the repository, then check your Volume configuration in XRay Admin Settings.


TIP

You can use the 'get repository' query to find out which repository a volume is assigned to. You can also use the 'get volumes' query to determine how many volumes you have, and which volume is assigned to each repository.

3.1.4.4 Field Descriptions

Table 3. Manage Settings: General tab

Field	Definition	
	Solr	Zookeeper
Name	Name of Volume	
Description	Description of Volume	
Type	Solr	Zookeeper
URL	Solr address or Zookeeper address.	
Organization	Optional. Fill in which organizations can use this volume. Usually this is a single organization for a dedicated volume.	
Solr Host	IP address of your Solr Host. If the volume is being created for a Solr instance on a remote server, the Solr Host is the remote volume. If the volume Type is Solr, the Solr Host is filled in automatically based on the Solr URL entry.	Not applicable.
Solr Port	The Solr port is filled in automatically based on the Solr URL entry.	Not applicable.
Zookeeper Host	If the volume is being created for a Solr instance on a remote server, the Zookeeper Host is the IP address of the master volume.	Not applicable.

Table 3. Manage Settings: General tab

Field	Definition	
	Since Solr comes with its own Zookeeper, if you do not provide the Zookeeper Host properly, the remote Solr will start the Zookeeper that comes with it by default.	
Zookeeper Port	Zookeeper port. For example: 2181 (default).	Not applicable.
Zookeeper Root	Root directory of Zookeeper. If you are creating a volume for a Solr installation on a remote server, you must provide the Zookeeper Root to be able to create the Solr cores.	Not applicable.

3.1.5 Repositories

Select **Repositories** tab in Admin Settings to display the list of all system repositories. Click on the **Active** and **Inactive** buttons to change the repository status.

A default repository can be specified so that each time you log into XRay, the default repository's data will load. This is a user-based setting. See *Section 3.2.5, Manage Settings*, for more information.

For information on repository data limits, see *Section 2.3.1, Repository*.

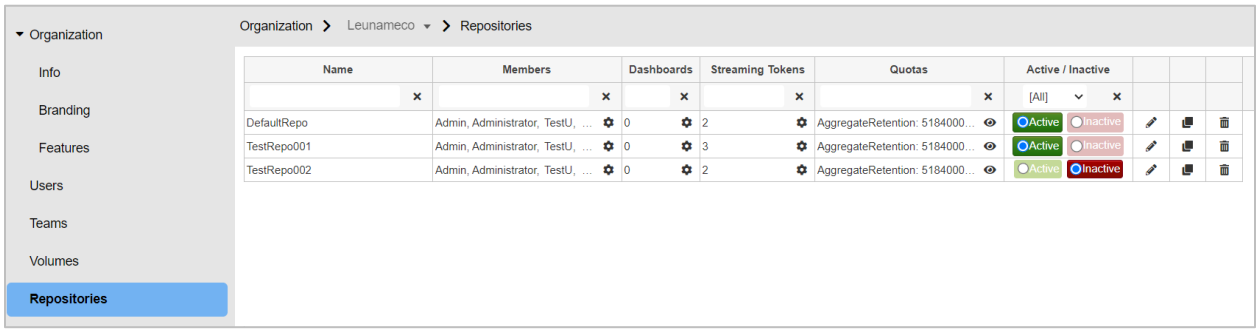


Figure 3.1.5-A. Repositories

To configure a dashboard's teams, click the setup icon for the dashboard. The list of dashboards for the repository is displayed. The assigned teams for each dashboard are included.

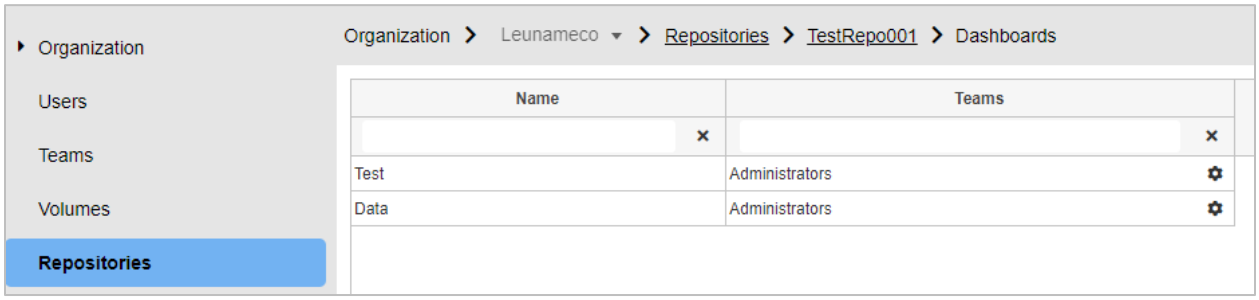


Figure 3.1.5-B. Repository Dashboards List

Search fields are provided at the top of the Name and Source columns. Click an entry to edit it. You can also assign a new team to a dashboard by clicking the setup button for the team. To assign a team or teams, select them from the Teams list at the top of the

page. Using the Role list, choose whether you want the team to be able to view the dashboard only (View) or view and modify it (Modify). Then click **Assign**.

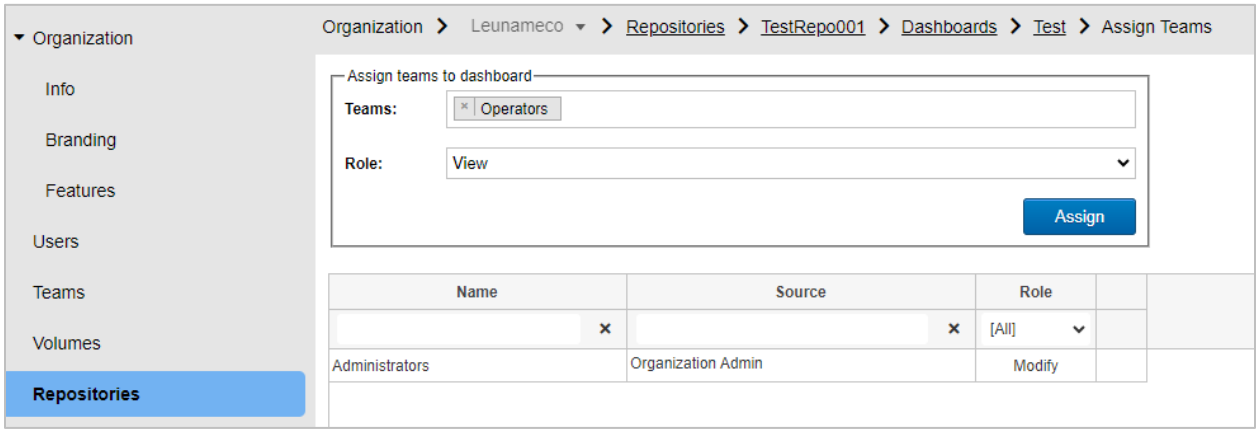


Figure 3.1.5-C. Repository Dashboard Teams

3.1.5.1 Create Repository

On the Repositories tab, click the **Create** button to add a new repository to the system. Review the settings. Click **Create** to create the new repository.

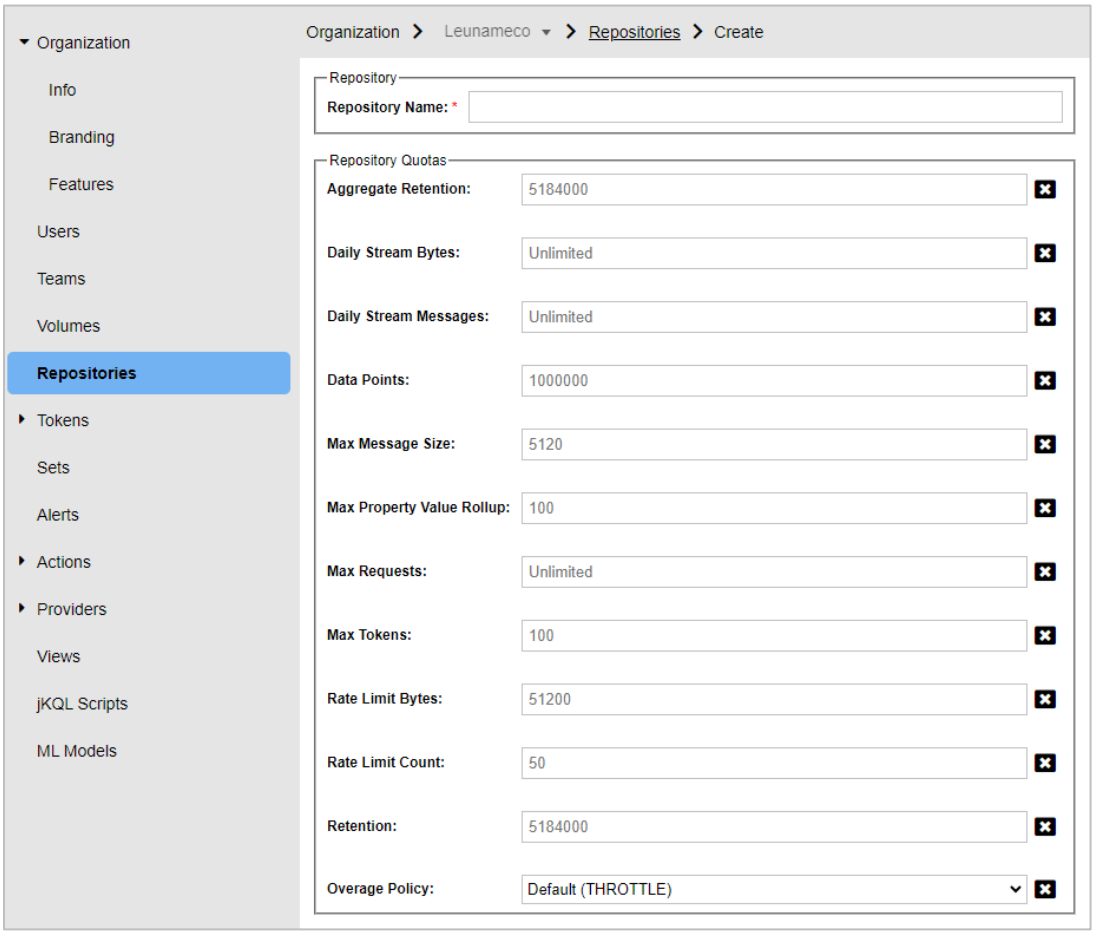


Figure 3.1.5.1-A. Create Repository

Table 4. Repository Field Descriptions	
Field	**Definition**
Aggregate Retention	Defines the length of time, in seconds, that aggregated data stored in Datasets table as the result of View evaluations is kept, after which it is deleted from database.
Daily Stream Bytes	Number of bytes that can be streamed per calendar day.
Daily Stream Messages	Total number of individual messages that can be streamed per calendar day.
Data Points	Defines the total number of data points (total number of Activities, Events, and Snapshots) that can be stored in the data store at any one time (based on Retention).
Max Message Size	Defines the maximum number of bytes that is stored in the Message field of Events (generally represents the payload of the data involved in the Event).
Max Property Value Rollup	Deprecated.
Max Tokens	The maximum number of Access Tokens that can be defined in the entire system (for Master License) or in a specific organization (for Default or organization-specific license)
Rate Limit Bytes	Controls the rate at which your data is processed. This is a limit on the number of bytes per second.
Rate Limit Count	Defines the maximum streaming rate, in messages per second, which data can be sent to the system. If data comes in at a higher rate, the defined Overage Policy will be applied to the connection.
Retention	Defines the length of time, in seconds, that data is kept. When the Retention time expires, the data is deleted from the database. 5,184,000 seconds = 60 days 604,800 seconds = 7 days
Overage Policy	Defines what action is taken when the streaming rate exceeds either Rate Limit Bytes or Rate Limit Count: THROTTLE – the connection is throttled so that the processing rate on the connection is the minimum of RateLimitBytes and RateLimitCount

Table 4. Repository Field Descriptions	
Field	**Definition**
	DROP – messages are dropped until the streaming rate slows down to the limits defined by RateLimitBytes and RateLimitCount ALLOW – no action is taken, and the streaming is allowed to continue at the current rate

3.1.6 Tokens

NOTE Token Setup in XRay 1.6 has moved from its location in previous versions.

You can access token setup in the following ways:

From Admin Settings:

 > Admin Settings > Tokens > Streaming

or

 > Admin Settings > Tokens > Access

or

 > Admin Settings > Tokens > Repositories; click the Setup icon  in the Streaming Tokens column.

From User Settings:

 > User Settings > Personal Tokens

3.1.6.1 Streaming Tokens (Admin Settings)

TIP Users who are not admins can see the streaming tokens for the repository they are in by selecting  > About and looking for the Stream Access Tokens line.

Stream Access Tokens	e2dee5c7-bb20-4d1c-8bd7-e068ee4d2447

An administrator defines and configures streaming tokens for the following scenarios and shares these tokens with all team members as required:

- Streaming data to XRay
- Set up an AWS CloudWatch metrics HTTP endpoint
- Streaming AutoPilot data to XRay

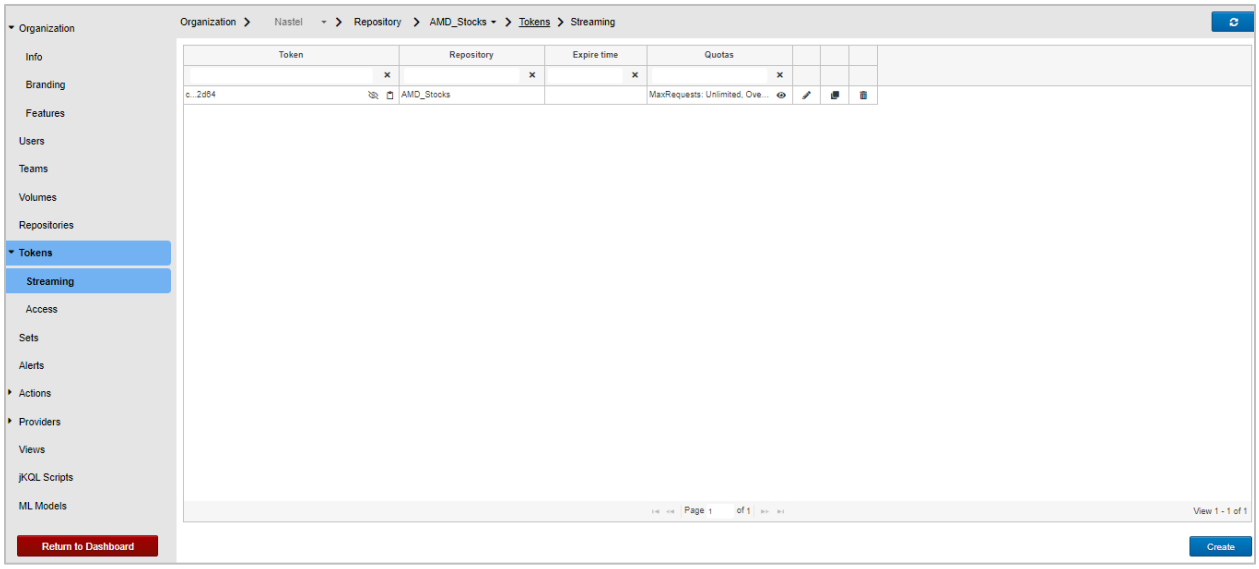


Figure 3.1.6.1-A. Streaming Tokens

To set up a streaming token:

1. Click **Create**.
2. (Optional.) Enter a **Token prefix** to identify the type of token, if needed.
3. Select the **Repository** to which the token provides access.
4. Indicate which types of objects you want to stream. By default, all objects are streamed (the **Select all** checkbox is selected). You can clear the checkbox to choose individual items, one at a time. Options are Activity, Dataset, Event, and Snapshot. (To remove a selection, click the X provided for that object.)
5. (Optional.) To set up the token to expire, use the **Expire Time** field to select a specific date and time.
6. Update the Token Quotas as needed. See Token Quotas fields below for details.
7. Click **Create**.

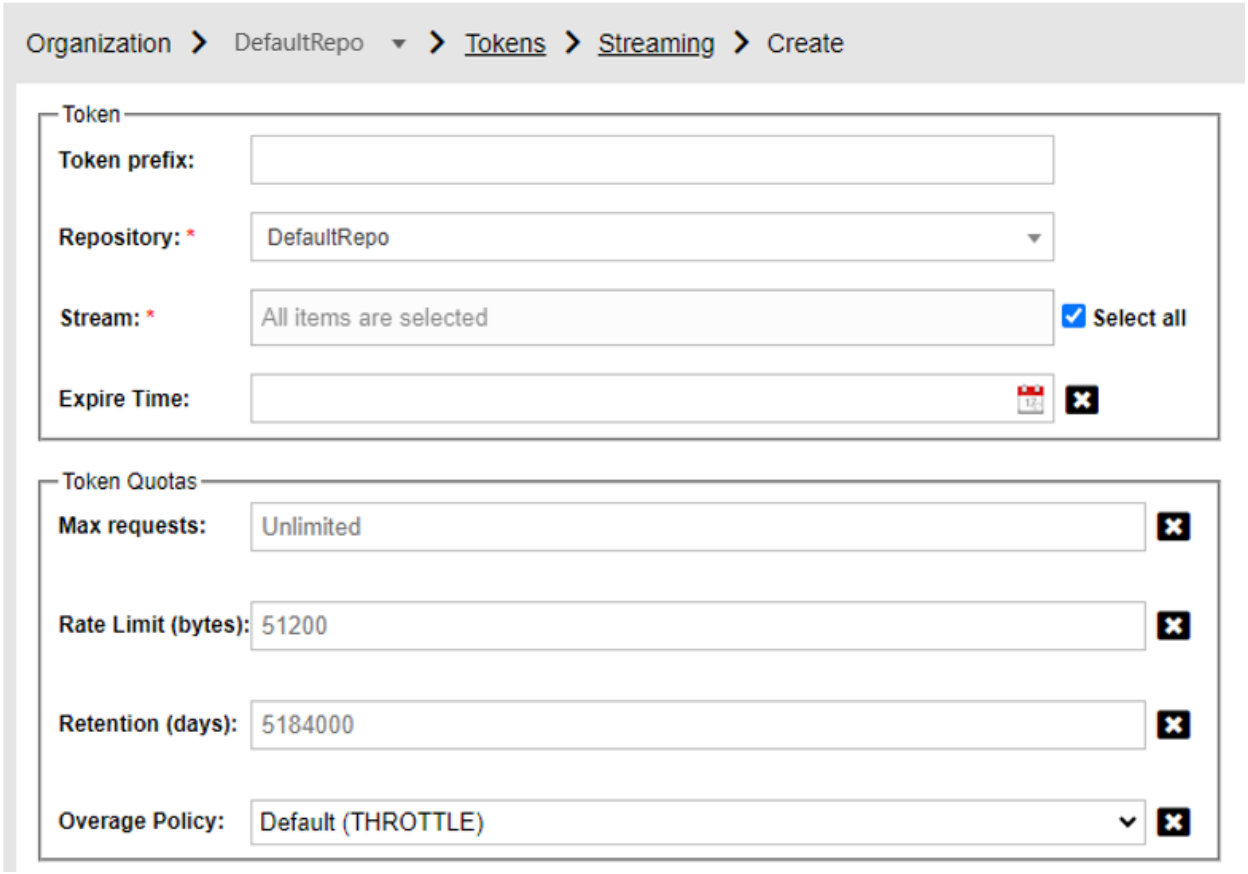


Figure 3.1.6.1-B. Create Streaming Token

Token Quotas fields

When you set organization or repository quotas, the values you set must not exceed the values specified the max limits defined by your license.

Table 5. Token Quotas Field Descriptions

Field	Definition
Max requests	Limit on requests for a single personal access token. After this number, requests made with this token are rejected. However, you can continue by resetting the quota value on the token.
Rate Limit (bytes)	Controls the rate at which your data is processed. This is a limit on the number of bytes per second.
Retention (days)	Length of time (in seconds) that streamed data is retained in a repository. 5,184,000 seconds = 60 days 604,800 seconds = 7 days

Table 5. Token Quotas Field Descriptions	
Field	**Definition**
Overage Policy	Defines what takes place when the Rate Limit quotas have been exceeded. THROTTLE – the connection is throttled so that the processing rate on the connection is the minimum of Rate Limit Bytes and Rate Limit Count. DROP – messages are dropped until the streaming rate slows down to the limits defined by Rate Limit Bytes and Rate Limit Count. ALLOW – no action is taken, and the streaming is allowed to continue at the current rate.

From the list of streaming tokens, you can view the quotas for a particular token in the list by clicking the View quotas icon: in the Quotas column.

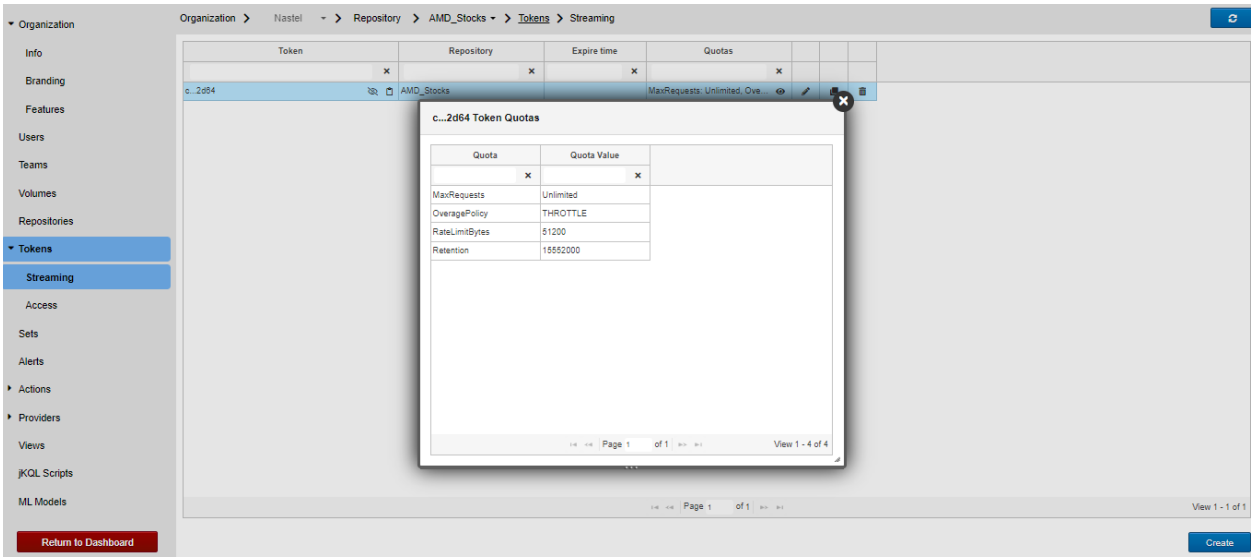


Figure 3.1.6.1-C. Token Quotas

3.1.6.2 Access Tokens (Admin Settings or User Settings)

All tokens except streaming tokens can be used for the REST API.

NOTE All tokens except streaming tokens can be used for the REST API.

Users define and generate their own access tokens, or personal tokens, for the following scenarios:

- For yourself:

- Generate a QR code to connect to a repository with a mobile device
- Generate a token to be able to import data into a repository

- For others:
  - Authorize someone else to access a repo on your behalf (for example, during vacation). You can remove their token/revoke access when no longer needed.
  - Give someone else the right to stream data to your repository. This might be a 3rd party that you want to collect data from. Once they have provided the required data, you can set an **Expire Time** or remove the token to discontinue streaming access.

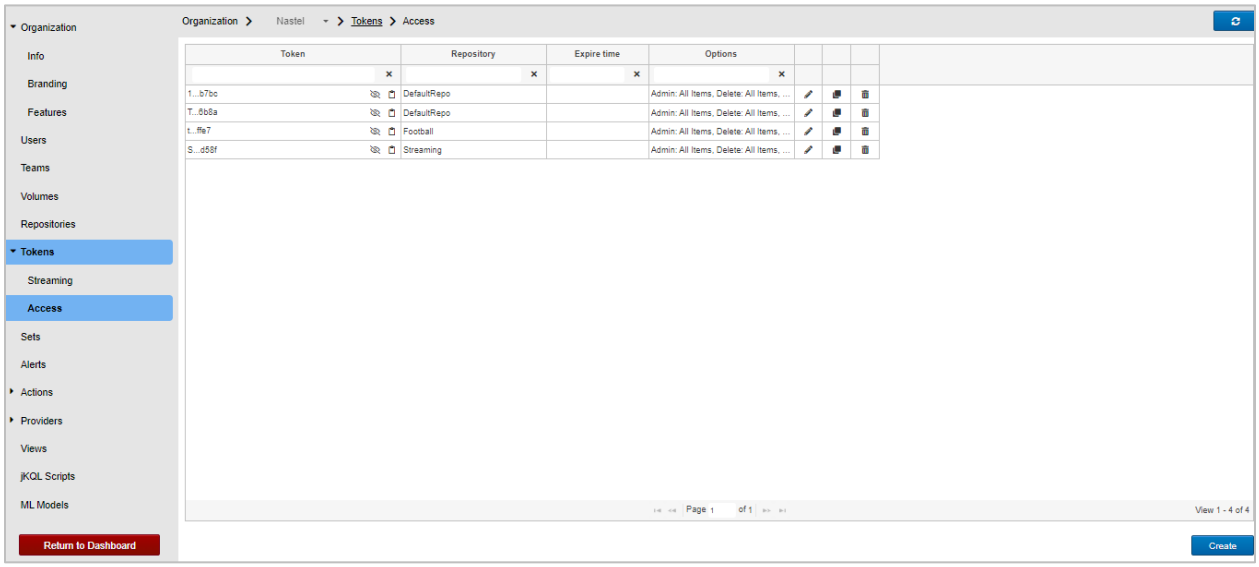


Figure 3.1.6.2-A Access Tokens

To set up an access token:

1. Click **Create**.
2. (Optional.) Enter a **Token prefix** to identify the type of token, if needed.
3. Select the **Repositories** to which the token provides access. You can choose individual repositories, one at a time. (To remove a selection, click the X provided for that repository.)
4. (Optional.) To set up the token to expire, use the **Expire Time** field to select a specific date and time.
5. Update the Token Options as needed. By default, all objects are selected (the **Select all** checkbox is selected). You can clear the checkbox to choose individual items, one at a time. See Token Options below for a list of objects.

   - **Query** (for getting data). Allows a user to access the repository data (for example, from the mobile application)
   - **Modify** Allows users to modify non-administrative data in the repository. This includes the ability to both add and update items.
   - **Delete** Allows users to delete non-administrative data in the repository
   - **Admin** allows users to manipulate administrative data only (that is, the following items): Users, Organizations, Teams, Repositories, and Tokens (listed as "AccessTokens").

- **Execute** allows users to use Invoke commands (to invoke scripts, actions, or external actions).

6. Click **Create**.

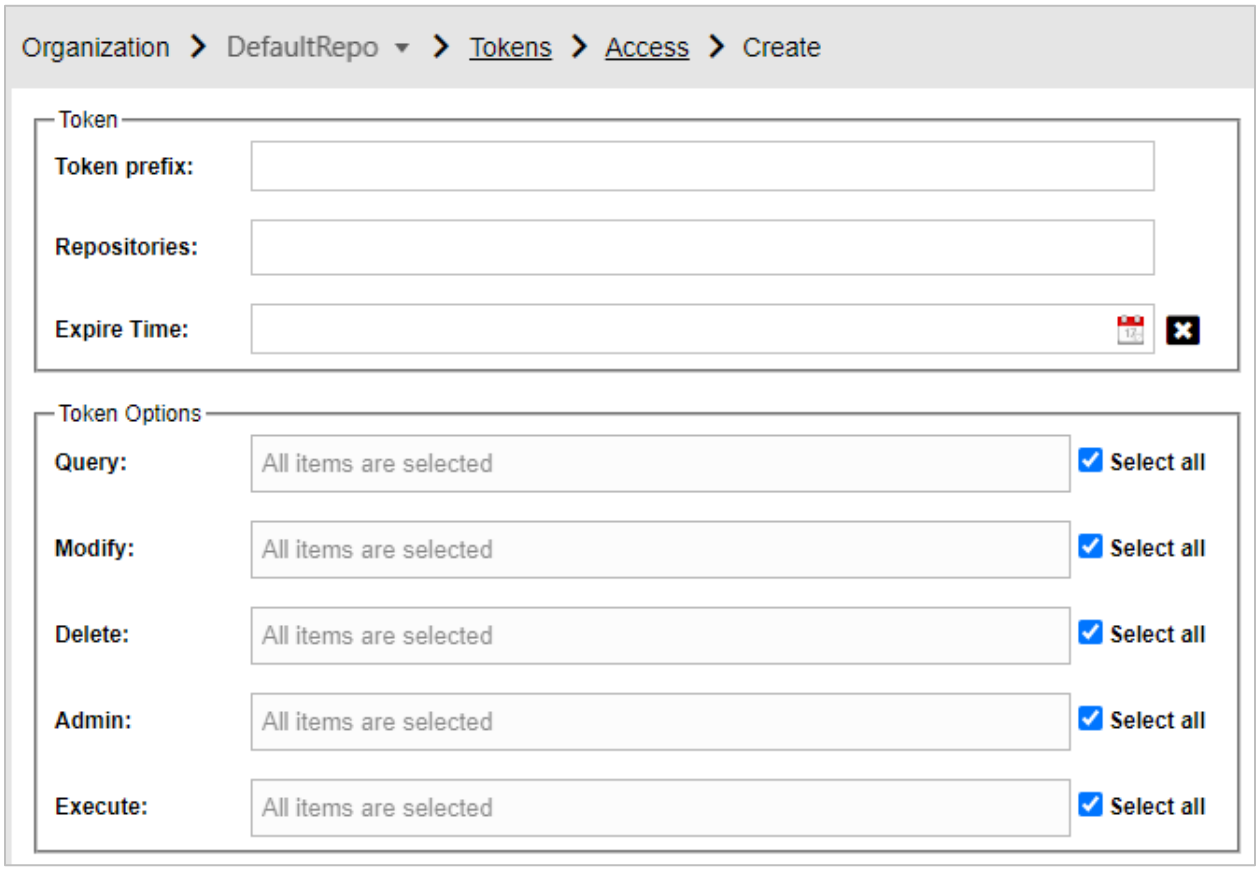


Figure 3.1.6.2-B. Create Access Token

Token Options

Items marked with asterisks below are administrative items. These are controlled by the Admin token option.

Source	Team*
Geolocation	Repository*
Address	AccessToken*
Server	Word
Process	Job
AppServer	Log
Application	DATASTORE
SourceUser	Provider
Runtime	Action
VirtualSource	Trigger

Network	InputDataRules
Device	BayesSourceFields
DataCenter	Parameter
GenericSource	Keyword
Event	Function
Activity	ProviderType
Resource	Query
Set	Dataset
Snapshot	Script
Dictionary	Policy
Relative	PolicyManager
Variable	Sensor
Topic	SensorFact
IpLocation	MLModel
Enumeration	License
Item	QuotaUsage
Value	View
Field	ViewTemplate
User*	Volume
Organization*	Feature

3.1.7 Sets

Select **Sets** from the side menu of *Admin Settings* (*Figure 3.1-A*). On this window, users can create, edit, clone, or delete sets (see sections 3.1.7.1 – 3.1.7.4 below for more information).

Sets are configurations used to group repository data. The system uses sets to determine how data is displayed in event and activity viewlets. Sets can be created manually or imported with a .csv file (see Section 2.6.4).

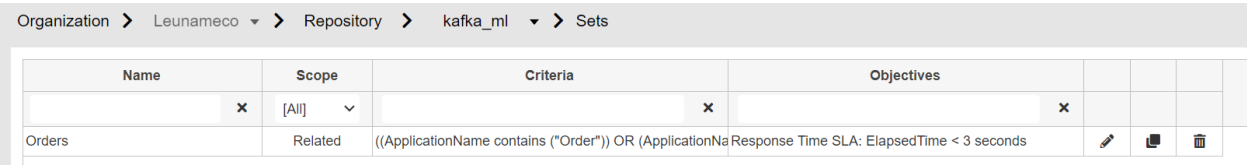


Figure 3.1.7-A. Admin Settings – Set Actions

3.1.7.1 Create Set

Click the **Create** button on the *Sets* window (*Figure 3.1.7-A*) to create a new set.

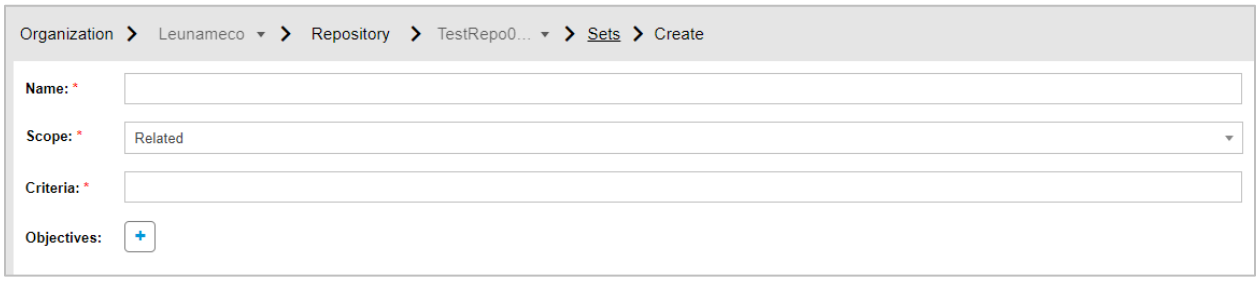


Figure 3.1.7.1-A. Admin Settings – Sets – Create

Specify the following:

- **Name**: Enter a name for the set. It must be at least 3 characters long (required).
- **Scope**: Select either **Singular** or **Related** from the drop-down menu.
- **Criteria**: Enter a criteria value.
- **Objectives**: Add objectives in this section. Click the blue add icon +. Enter the objective name and value, for example, `ElapsedTime < 3 seconds`. To remove an objective, click the blue minus - icon.

Click **Save**. The details of your new set will display.

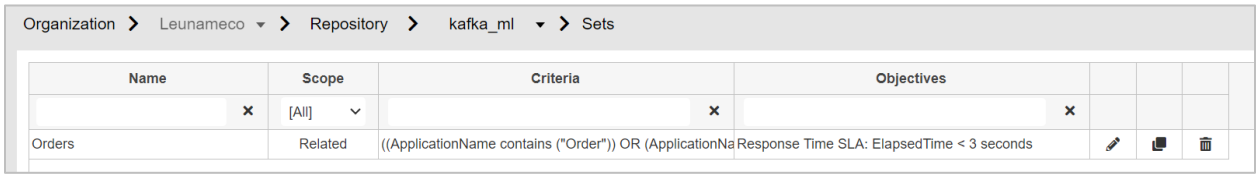


Figure 3.1.7.1-B. New Set

3.1.7.2 Edit Set

On the *Sets* window (*Figure 3.1.7-A*), click the **Edit** button of the existing set you want to edit.

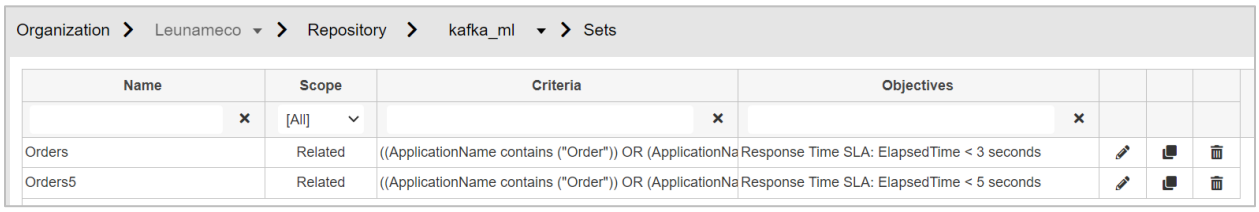


Figure 3.1.7.2-A. Admin Settings – Sets

Update all desired fields. Please note that the name of a set cannot be updated.

Click **Save** when finished.

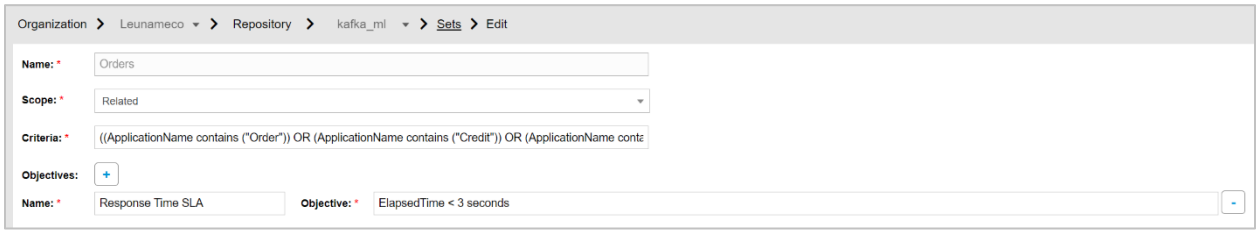


Figure 3.1.7.2-B. Admin Settings – Sets – Edit

3.1.7.3 Clone Set

Choose the set you want to copy and click the Clone icon . The Script editor opens. The **Name** of the new set is the name of the cloned set followed by "Copy." Update the **Name** field to rename the new set as needed.

Make your changes to the new script. For details, see the instructions for creating a set above.

Click **Save** when finished.

3.1.7.4 Delete Set

On the *Sets* window, click the **Delete** button next to the set you want to delete. A confirmation message is displayed. Click **Yes** to delete the set or **No** to cancel.

3.1.8 Alerts

Select **Alerts** from the side menu of *Admin Settings* (*Figure 3.1-A*) to set up email notifications about events.

3.1.8.1 Create an Alert

Click the **Create** button on the lower left to create alerts for specific activities or events. In your repository, if you have imported sets with objectives, you will see a list of possible sets. Only one set or sets with the same name can be selected at the same time to be used for the creation of an alert.

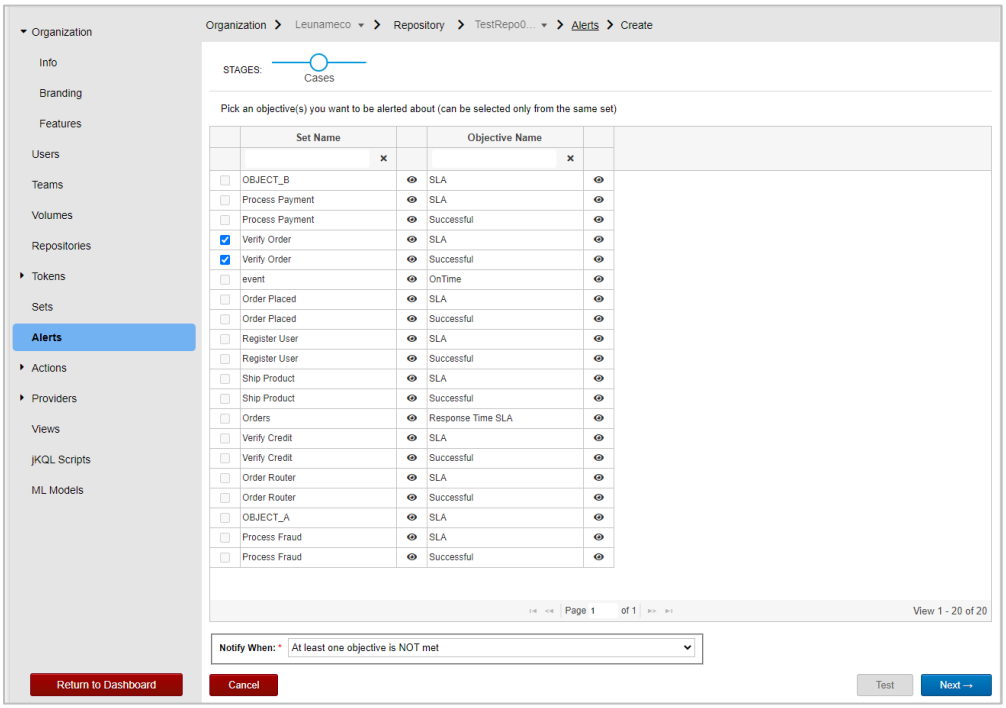


Figure 3.1.8.1-A. Objectives

Select all desired sets. From the **Notify when** drop-down menu, select when you want to send a notification. Click **Next** to continue configuring.

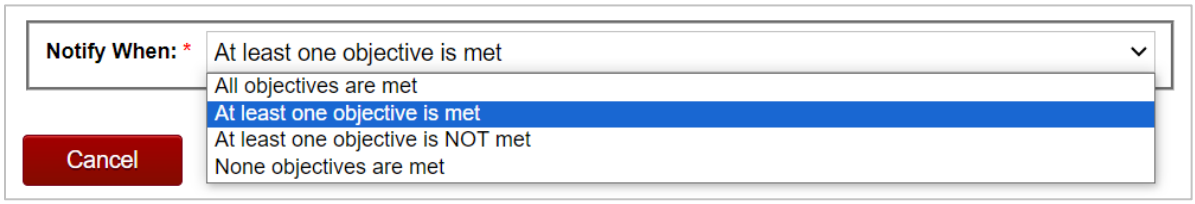


Figure 3.1.8.1-B. Notification

On the next screen, select the severity type from the **Severity** drop-down menu, and choose how often you would like to get the reports. Click **Next** to continue.

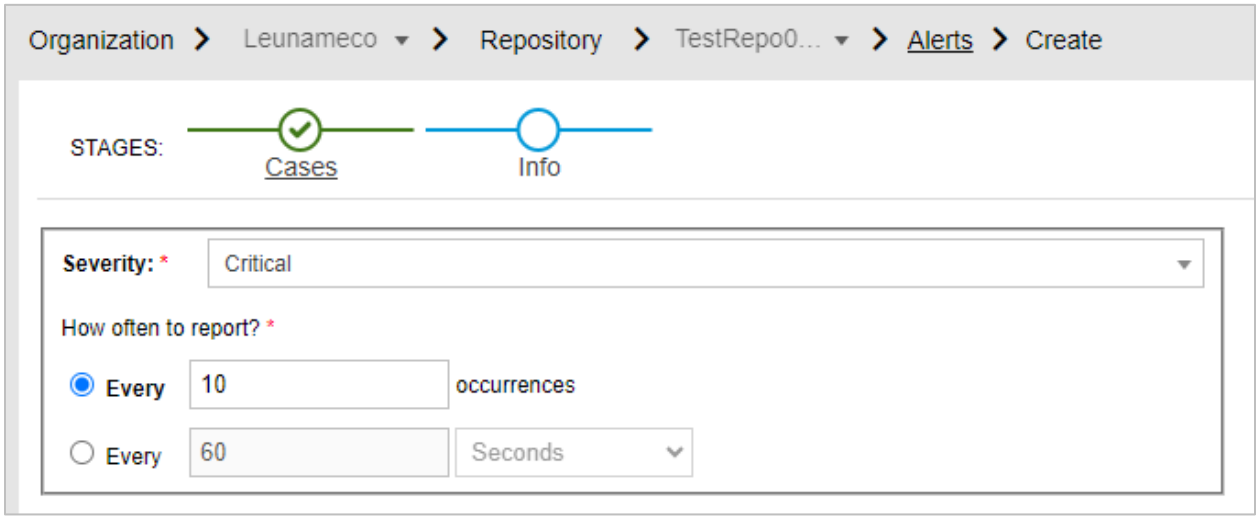


Figure 3.1.8.1-C. Alert Cases

On the next screen, enter the **Name** of the alert, and choose the **Action** you want to use to inform users about the problem. (For instructions on how to create actions, see section *3.1.9,* Actions.)

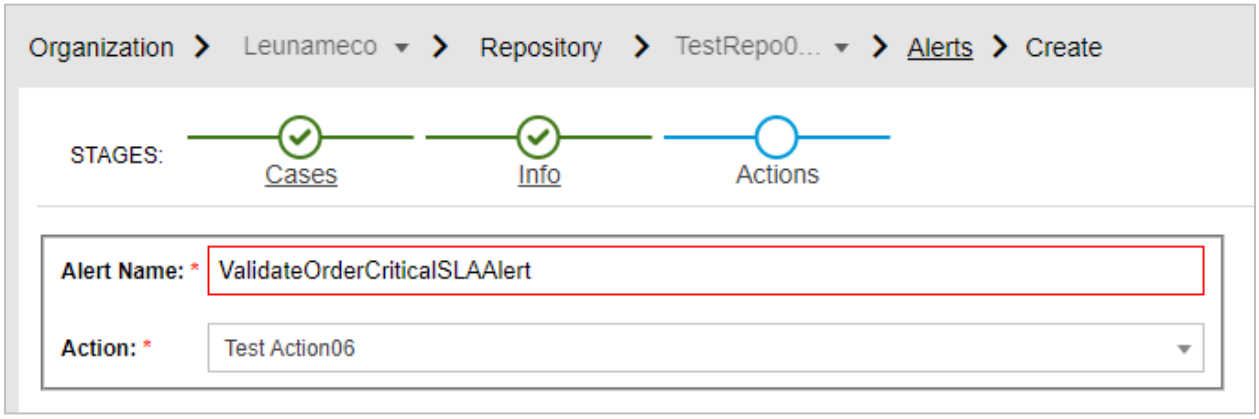


Figure 3.1.8.1-D. Alert Information

Click the **Test** button (enabled when alert name and email address are specified) to send a test message. An Alert test result message is displayed to inform you whether the test was successful.

Click **Finish** to save the newly created alert.

3.1.8.2 Clone an alert

You can create a new alert by starting with a copy of an existing one. Clone an alert by clicking the clone icon. When cloning, the Alert Name of the new alert is the name of the cloned alert followed by "Copy." You can change it as needed.

3.1.8.3 Edit Alert

After clicking the pencil icon to edit a selected alert, the same window seen when creating an alert will display. The only difference is that the **Test** button is now active.

Click the **< Back** button to modify the previous step's configurations. Click **Test** to send a test notification to your email address (specified on the *Create an Action* window). Click **Next >** to continue editing. After all edits have been made, click **Finish**.

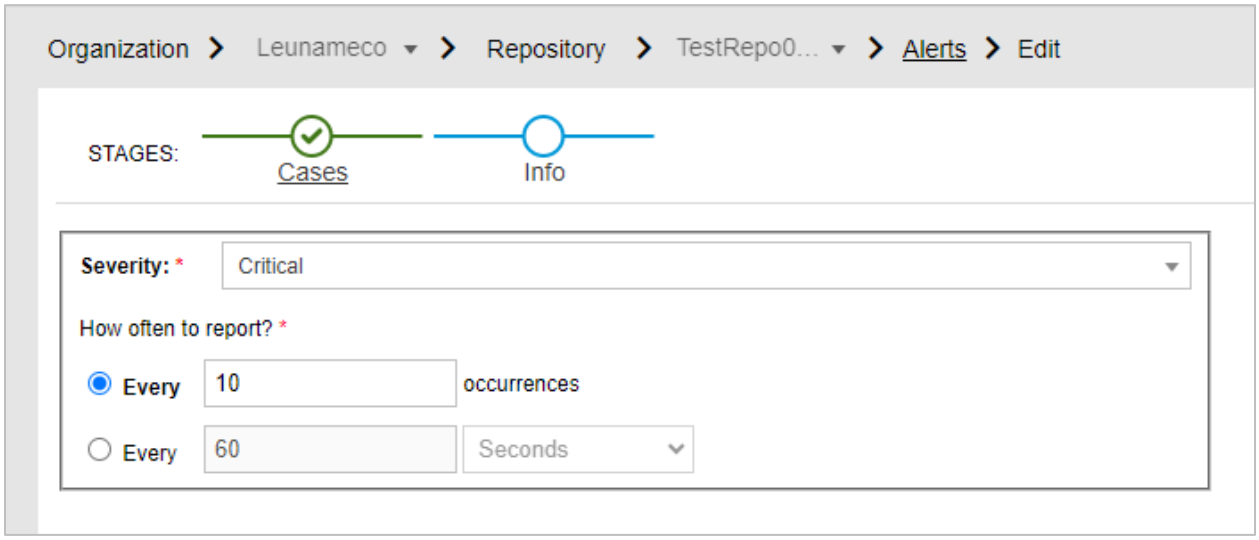


Figure 3.1.8.3-A. Edit Alert

3.1.8.4 Delete Alerts

Delete alerts by clicking the trashcan icon .

3.1.9 Actions

By setting up an XRay action, you lay the framework for informing users or groups of users about a condition that has been met in the data. Slack actions send Slack messages; email actions send emails. An action is associated with an alert, which defines the conditions (see section *3.1.8*, Alerts).

3.1.9.1 Slack Actions

For information about setting up Slack actions, see step 3 of Setup and Use under Social Integration: Slack. In Step 4 of that section, you'll associate Slack actions with alerts so that messages are sent in certain cases.

3.1.9.2 Email Actions

Before creating an email action, you must set up an email provider. See section *3.1.10*, Providers.

You can send an email to any email address (individual or group). Messages can include text, variables, or both.

1. From **Main Menu > Admin Settings** > **Actions**, select **Email**.

2. Click **Create**.
3. Enter an **Action Name**.
4. Select the **Provider Name** that will send emails for this action.
5. Select the MIME type: **Text/plain** or **text/html**.
6. Enter "from" and "to" email addresses in the **Mail From** and **Mail To** An optional **Mail CC** field is available to copy the message to additional addresses.
7. Enter the **Subject** of the message to be sent.
8. Enter the **Message** text and variables, if desired. For more information about creating and formatting messages, see Templates and Message Formatting.
9. To verify that messages can be sent through the email provider, click **Test**. XRay displays a confirmation message to inform you that the test email was sent successfully.
10. Click **OK**.

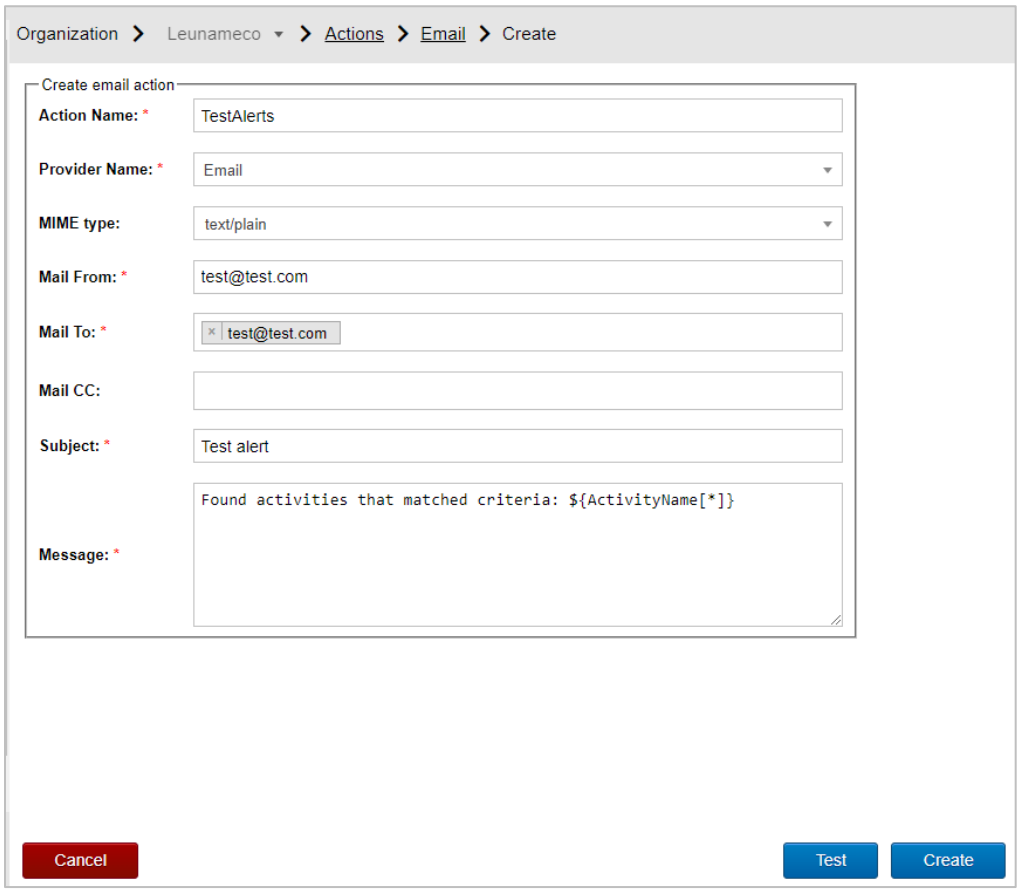


Figure 3.1.9.2-A. Email Action

3.1.9.3 Log Actions

Instead of sending a message through Slack or email, a Log action saves data to a log for future reference.

1. From **Main Menu > Admin Settings** > **Actions**, select **Logs**.
2. Click **Create**. The Editing Provider console opens.
3. Enter the Provider Name; this name must match the **Provider Name** as set up in Providers (see section *3.1.10*).
4. The **Line** is filled in automatically, as set in the Editing Provider console.
5. Click **Save**.

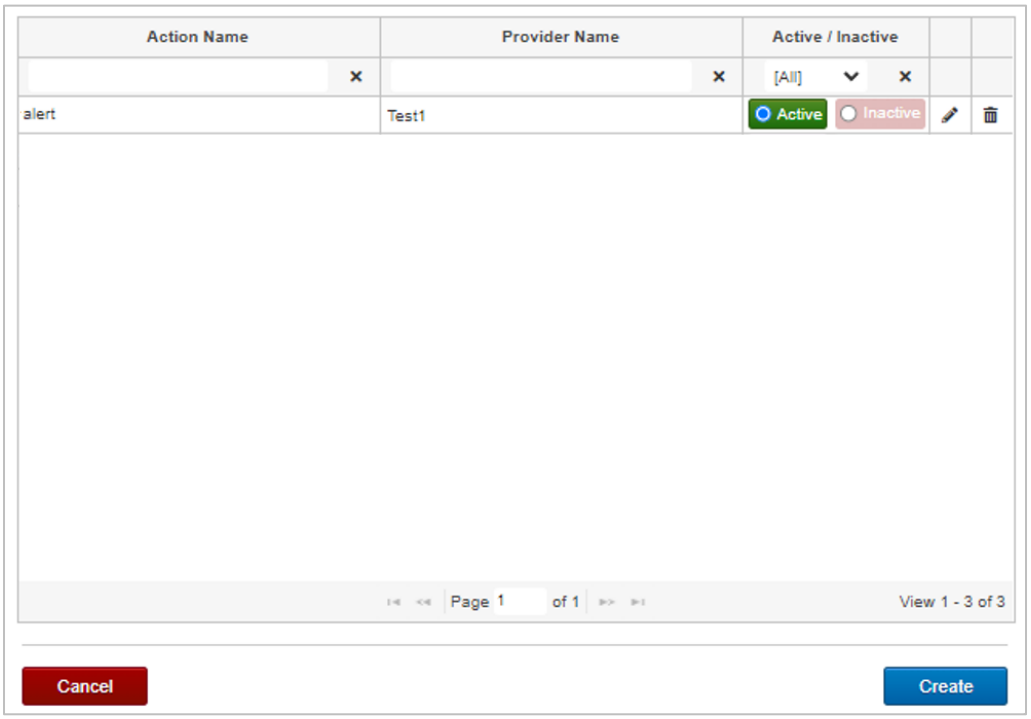


Figure 3.1.9.3-A. Log Actions

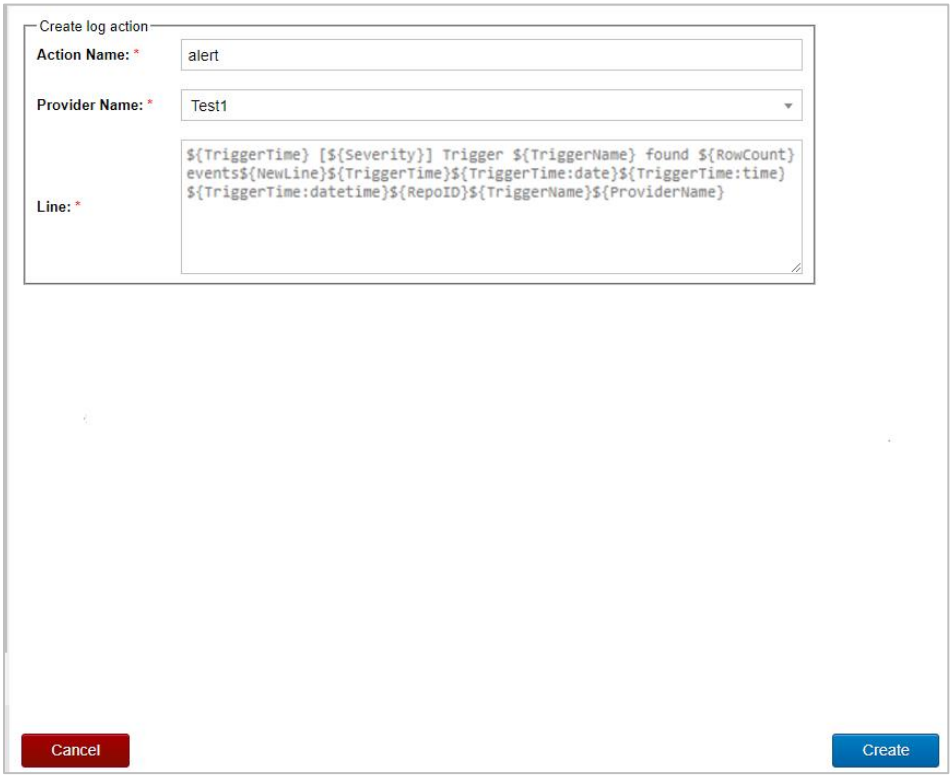


Figure 3.1.9.3-B. Creating Log Actions

To test the log action:

1. In the jKQL Workbench, open any repository.
2. Type "invoke action '<action name'>" and run the query.
3. To check the log file, go to AutopilotM6/localhost. The log file can be found there.

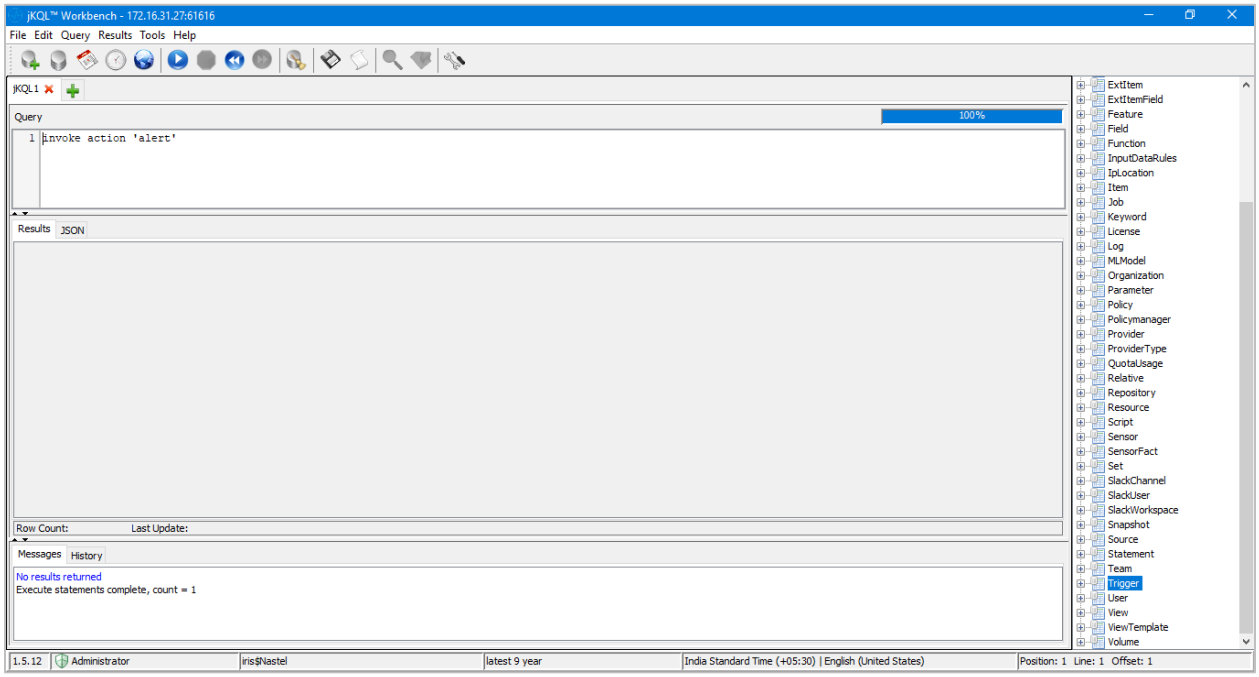


Figure 3.1.9.3-C. jKQL Workbench

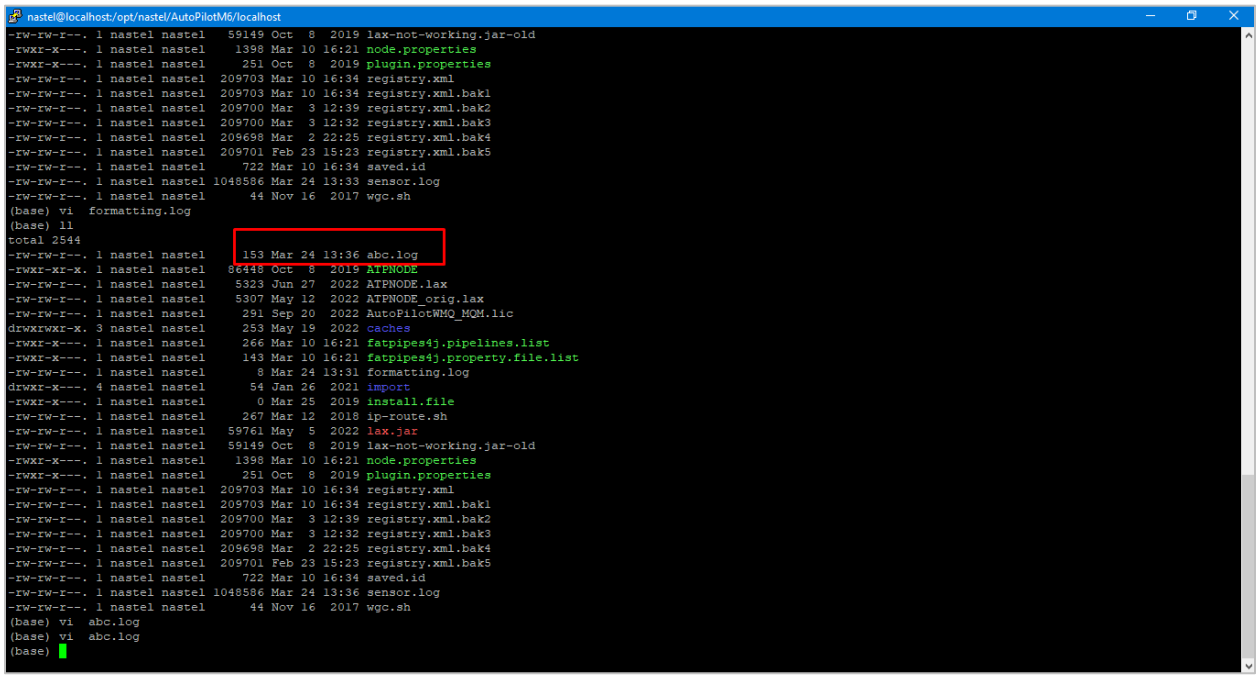


Figure 3.1.9.3-D. Log File in AutopilotM6/localhost

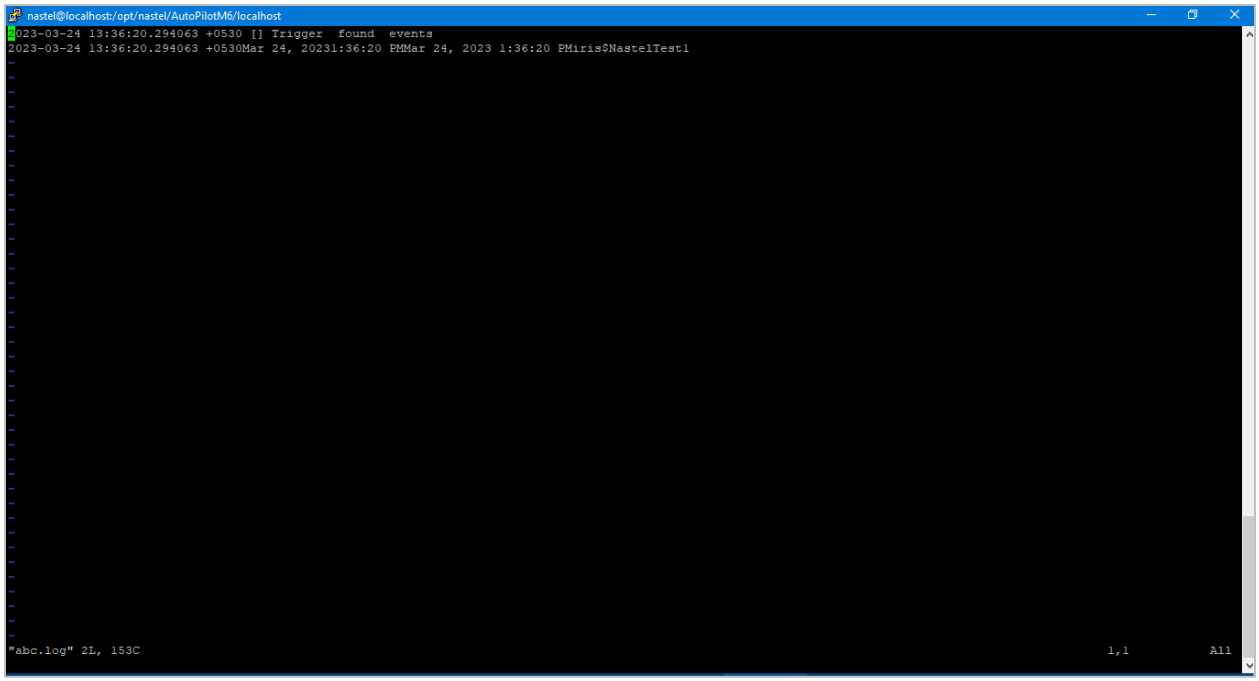


Figure 3.1.9.3-E. Log File Contents

3.1.10 Providers

XRay providers are the services you can select from when setting up messaging actions for alerts. There are currently three provider options available:

- **Email.** Information about setting up Email as a provider is covered in this section.
- **Slack.** For information about setting up Slack integration, including how to set up Slack as a provider, see Step 2. Set up Slack as a Provider in XRay in Social Integration: Slack.
- **Logs.** Information about setting up Logs as providers is covered in this section.

3.1.10.1 Set up an Email Provider

1. From **Main Menu > Admin Settings** > **Providers**, select **Email**.
2. Click **Create**.
3. Enter a **Provider Name**.
4. Enter a **Host** and **Port**.
5. Enter a **Username** and **Password**.
6. If your email provider is required to use STARTTLS, select the **Enable STARTTLS** option.
7. Click **Create**.

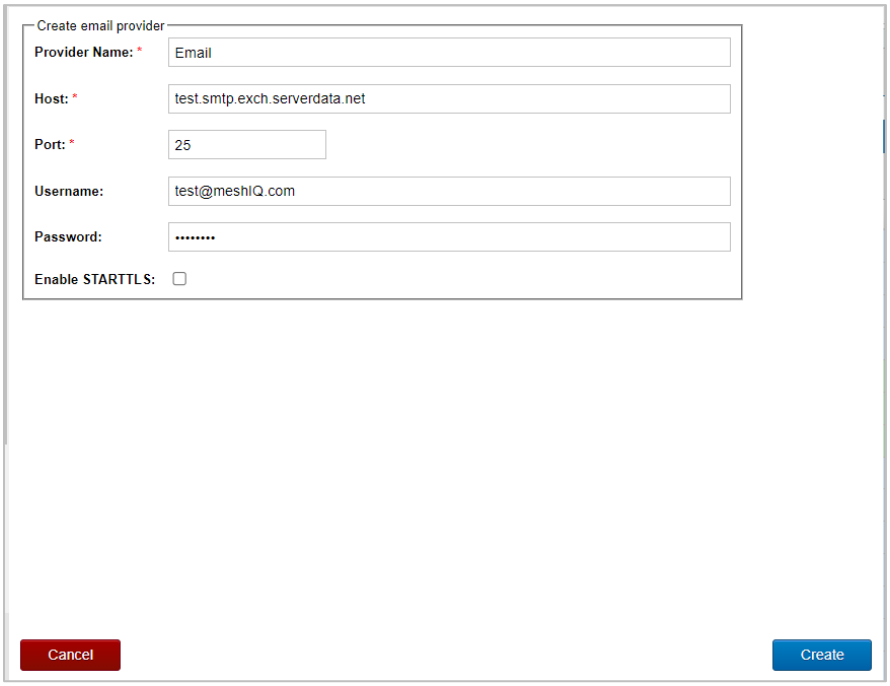


Figure 3.1.10.1-A. Email Provider

3.1.10.2 Set up a Log Provider

Log providers are used to log data into a log file in file system. They are only applicable to on-premises servers, where clients have access to the log files.

1. From **Main Menu > Admin Settings** > **Providers**, select **Logs**.
2. Click **Create**.
3. Enter a **Provider Name**.
4. Enter a **File Name**. This is the name of the file, as it is stored in the CEP server.
5. If you want to append the action to the file, select the **Append** checkbox.
6. In the **Line** field, a trigger is created using the shortcuts provided in the Line Editor. Click the **Edit** button below the Line field to access the Line Editor and make changes. When changes are complete, click **Apply**.
7. Click **Create**.

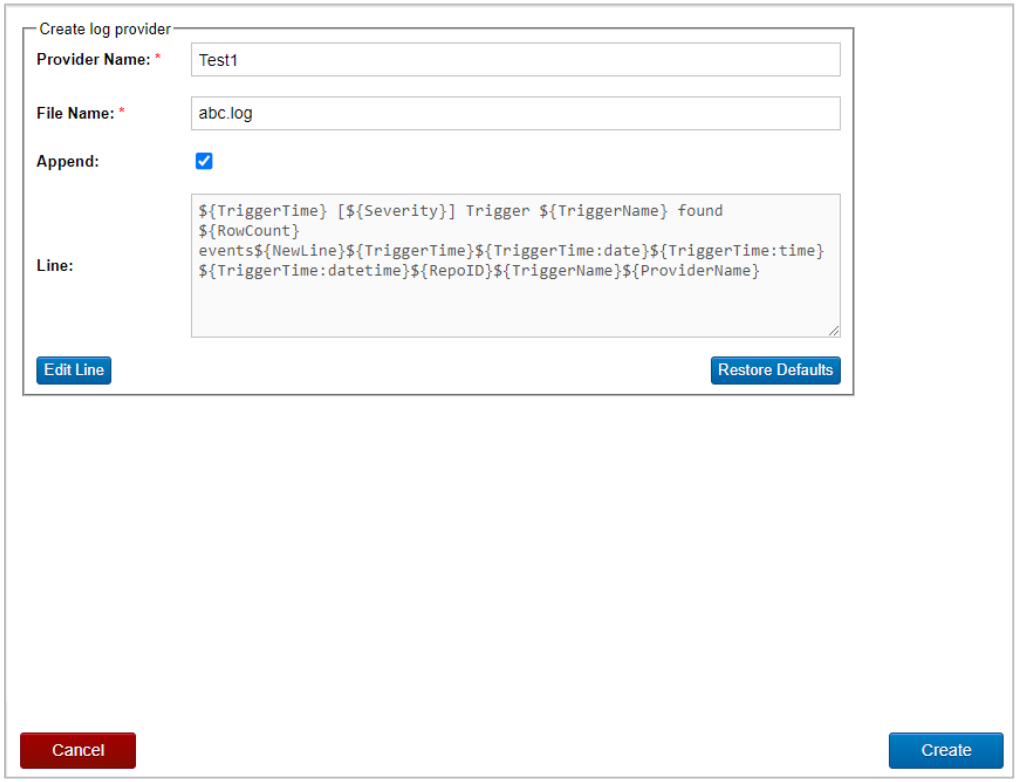


Figure 3.1.10.2-A. Log Providers

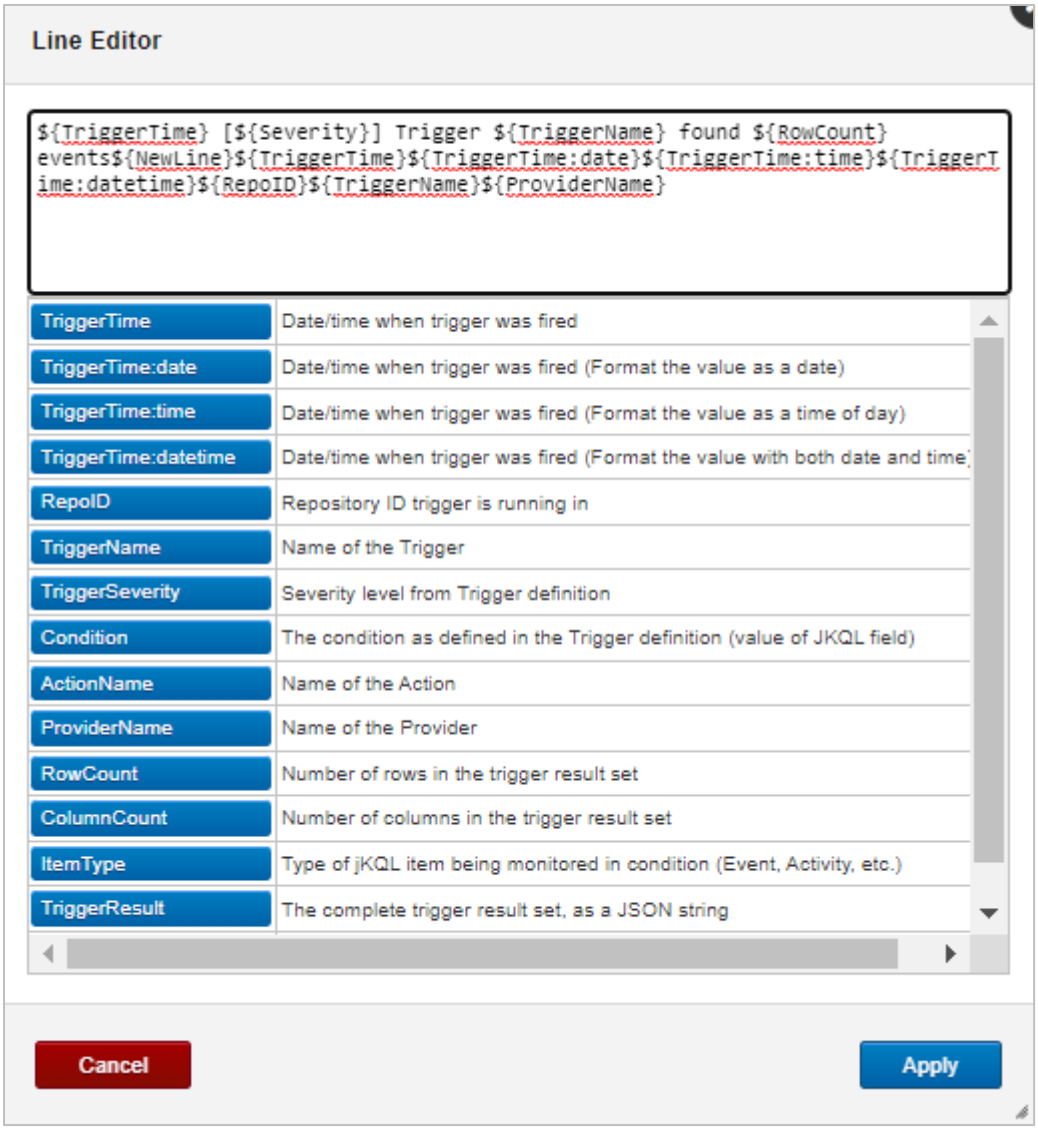


Figure 3.1.10.2-B. Line Editor

3.1.11 Views

A View represents a named query whose result is evaluated on a defined interval. After each periodic evaluation, the results are cached for quick retrieval. The implementation is analogous to an SQL Materialized View.

Views can be implemented directly using jKQL, or through the user interface.

Let's define a simple View using jKQL:

```
Upsert View Name='TestView',
    jkql='Get Number Of Events Group By EventName',
    Schedule='1 day';
```

This view will be evaluated every 1 day, and the result of the query will be cached.

Let's define the same view through the user interface.

3.1.11.1 Create a view

1. On the Views tab of Admin Settings, click the **Create** button.
2. Enter a **Name** for the view.
3. Enter the **Query** that defines the data that the view obtains.
4. Select the format for the Schedule expression: **jKQL** time interval expression or **CRON** expression.
5. In the **Schedule** field, enter the interval at which the query will be run and the frequency (*Day*, *Hour*, *Minute*, or *Second*).
6. Select the **Repository**.
7. Enter the **Dataset** name to be used.
8. Click **Create**.

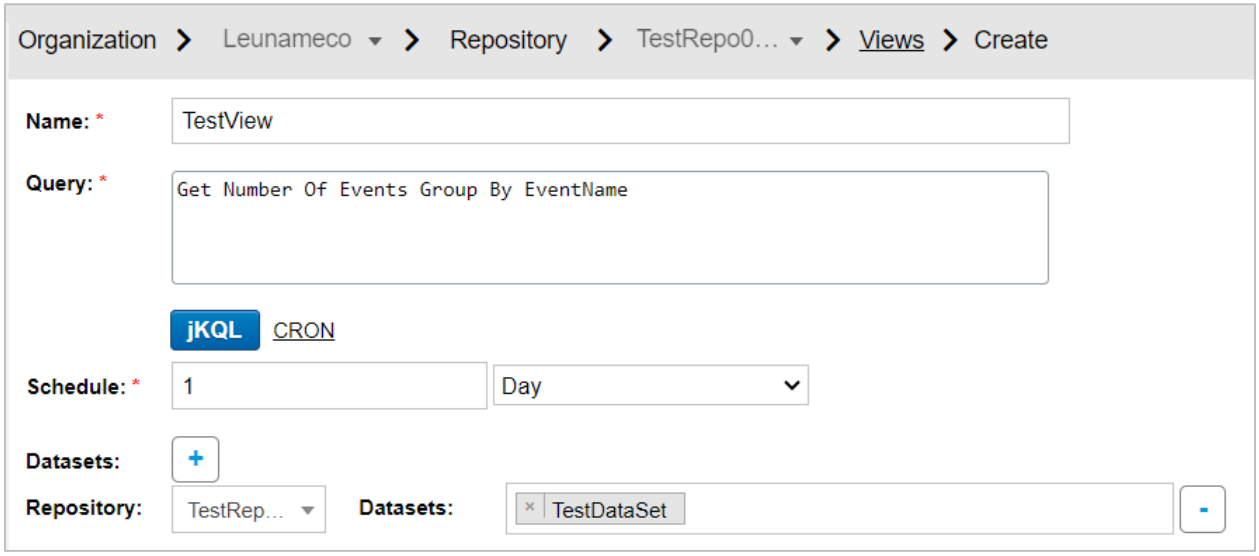


Figure 3.1.11-A. Creating a View

When views are displayed in a viewlet, you can edit, delete, activate, or deactivate Views directly from the query results. Select the view to display an action menu.

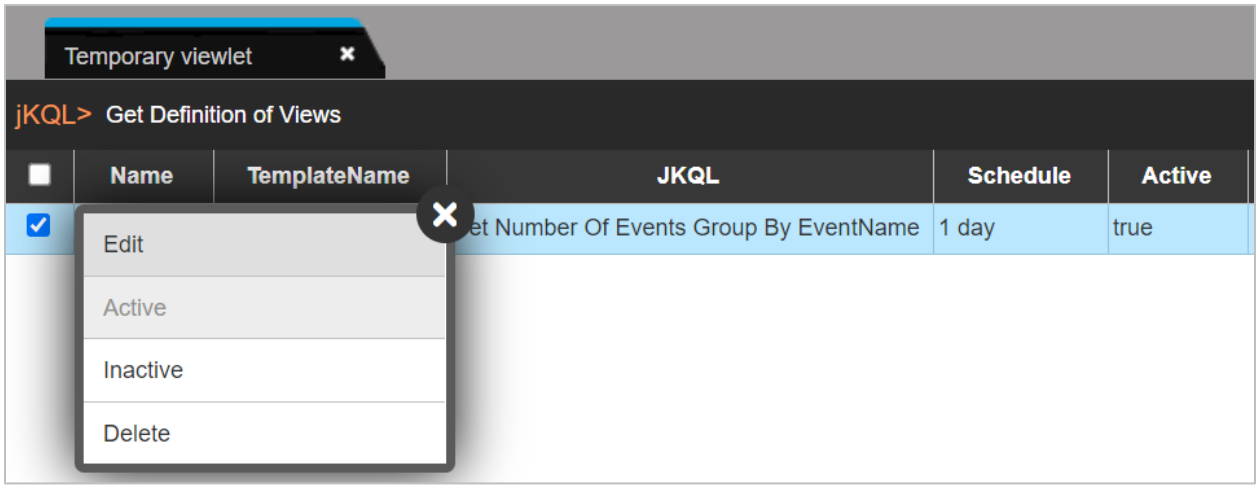


Figure 3.1.11-B. Views Action Menu

Refer to the *jKQL User's Guide* in the XRay section of the Resource Center for more information on views.

3.1.12 jKQL Scripts

jKQL Scripts allow custom processing functionality to be executed. For those familiar with SQL systems, these are analogous to stored procedures/functions. With them, data can be loaded from jKQL data store, processed, and written back out to data store and/or returned for display in UI.

jKQL Script definitions are kept in jKQL data store. The definition contains either the complete text for the script, or a URI from which to retrieve the text.

The GenerateDashboard and GenerateMLDashboard JKQL scripts, if available, are preloaded and can be modified. GenerateDashboard is a helper script that can be reused in other scripts. When these scripts are found, you can automatically build a dashboard containing viewlets of machine learning queries. See *Create Viewlet with a jKQL Query* for more information.

jKQL scripts can be implemented directly using jKQL, or through the user interface. Some examples of the direct implementation of jKQL scripts are shown below. These are defined using the Upsert statement:

```
Upsert Script Name = 'TestScript', Text = 'var rs = executeJKQL(\'Get
number of events for latest year group by eventname\');
setReturnResult(rs);'
```

```
Upsert Script Name = 'TestUrl', Url = 'file:/home/me/example.js',
Properties = ('FilterField'='STRING', 'FilterValue'='STRING',
'GroupField'='STRING'), Options = ('MaxRawRows'=30000)
```

3.1.12.1 Add a script

To add a script through the user interface, start from the jKQL Scripts tab of Admin Settings.

1. Go to Main Menu > Admin Settings.
2. Select the jKQL Scripts tab.
3. Click **Create**.
4. In the **Script editor**, enter the script that you want to save. It will take the form of JavaScript code that interfaces with jKQL. For example:

```
var startTime = getScriptParam('starttime');
var endTime = getScriptParam('endtime');
var interval = getScriptParam('interval');
var query = "get number of activities fields avg(elapsedtime) as
elapsedtime where starttime > '"
    + startTime
    + "'  and starttime <= '"
    + endTime
    + "' group by starttime bucketed by " + interval;
var rsActivities = executeJKQL(query);
for (var arow = 1; arow <= rsActivities.getRowCount(); arow++) {
    var dataset = createDataset();
```



```
    dataset.setField(FieldType.DATASET_NAME, 'TestDataset1');

    dataset.setMapFieldKey(FieldType.PROPERTIES, "time",
rsActivities.getValue(arow, 1).getBegin()); // starttime

    dataset.setMapFieldKey(FieldType.PROPERTIES,
rsActivities.getColumnName(2), rsActivities.getInteger(arow, 2)); //
number of rows

    dataset.setMapFieldKey(FieldType.PROPERTIES,
rsActivities.getColumnName(3), rsActivities.getTimeInterval(arow, 3));
// average elapsedtime

    upsert(dataset);

}

executeJKQL("gt datasets fields all")
```

You can even write an entire JavaScript program (that interacts with JKQL) and save it as a script. To learn how to call jKQL from within the JavaScript, refer to the Script section of the *jKQL User's Guide* in the XRay section of the Resource Center.

5. In the panel on the right side, enter the name of the script in the **jKQL Script Name** field, and select the **Repository** it will use.
6. Use the **Time Period** parameter to define the timeframe for which you want the query and the script to be applied.
7. Click **Save** to save the script for future reuse.
8. Use the **jKQL Script Parameters** area to enter any runtime values for the parameters that you are asking users to enter via the 'getScriptParam' function (such as 'starttime'). These parameters are displayed immediately after a script is saved, and whenever a script is run.
9. Click **Run**. Results are displayed in a Results Viewlet in the Console panel.

NOTE

Before data can be made available to queries, it must be committed to index files in Solr. Therefore, you may experience a delay before results become available. After you run a script, try waiting several minutes, then use a temporary viewlet to query your results (click the Create temporary viewlet button  next to the Console label).

10. Click **Close** to return to the jKQL Scripts list.

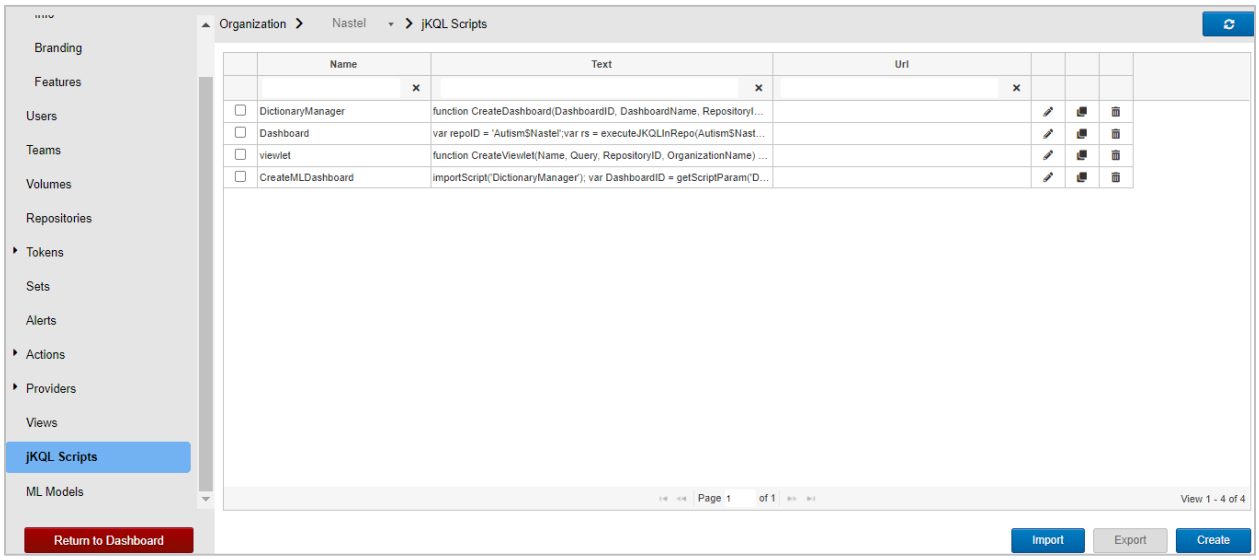


Figure 3.1.12.1-A. jKQL Script List

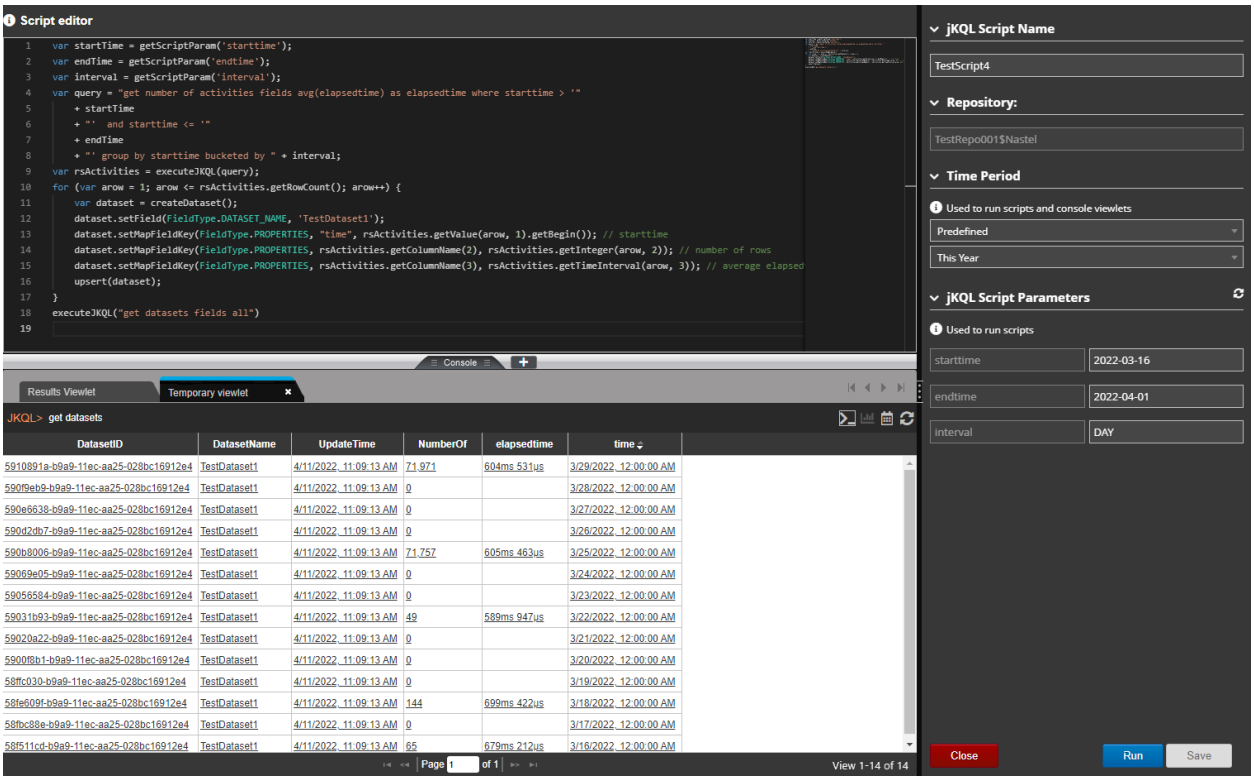


Figure 3.1.12.1-B. jKQL Script

3.1.12.2 Clone a script

To create a script that is based on another script, use the clone feature.

1. Click the clone icon . The Script editor opens. The **jKQL Script Name** of the new script is the name of the cloned script followed by "copy."
2. Update the **jKQL Script Name** field to rename the new script as needed.

3. Make your changes to the new script. For details, see the instructions above for adding a script.
4. Click **Save**.

3.1.12.3 Edit a script

1. Click the Edit icon . The Script editor opens.
2. Make your changes to the script. For details, see the instructions above for adding a script.
3. Click Save.

3.1.12.4 Import a script

1. Click **Import** (in the lower-right corner of the screen).
2. Click **Choose files** to navigate to and select one or more files. Or drag one or more files onto the outlined area.
3. Click **Next** to preview the script. If you already have a script of the same name, you must choose **Yes** to override the existing one.
4. Click **Finish** to view the imported script in the list of jKQL scripts.

3.1.12.5 Export a script

1. Select the checkbox of the jKQL script you would like to export.
2. Click **Export** (in the lower-right corner of the screen). The browser indicates that the file has been downloaded. You can find the file in your default downloads folder.

3.1.13 ML Models (Machine Learning)

You can configure Machine Learning by defining the models to be built. Building a model is also known as "training" a model. Models are trained with learning data and a target variable.

- The target variable is the field that you wish to gain insight into, predict, perform what-if on, detect anomalies on, or forecast.
- The learning data is the data that you suspect has impact on the target field, the interval that you wish to forecast, the fields to input into what-if scenarios.

3.1.13.1 Define a new Machine Learning model

To begin defining a Machine Learning model, do the following:

1. Go to Main Menu > Admin Settings.
2. Select the ML Models tab. Before any models are created, the tab looks like the figure below.
3. Click **Create** to begin defining a new model.

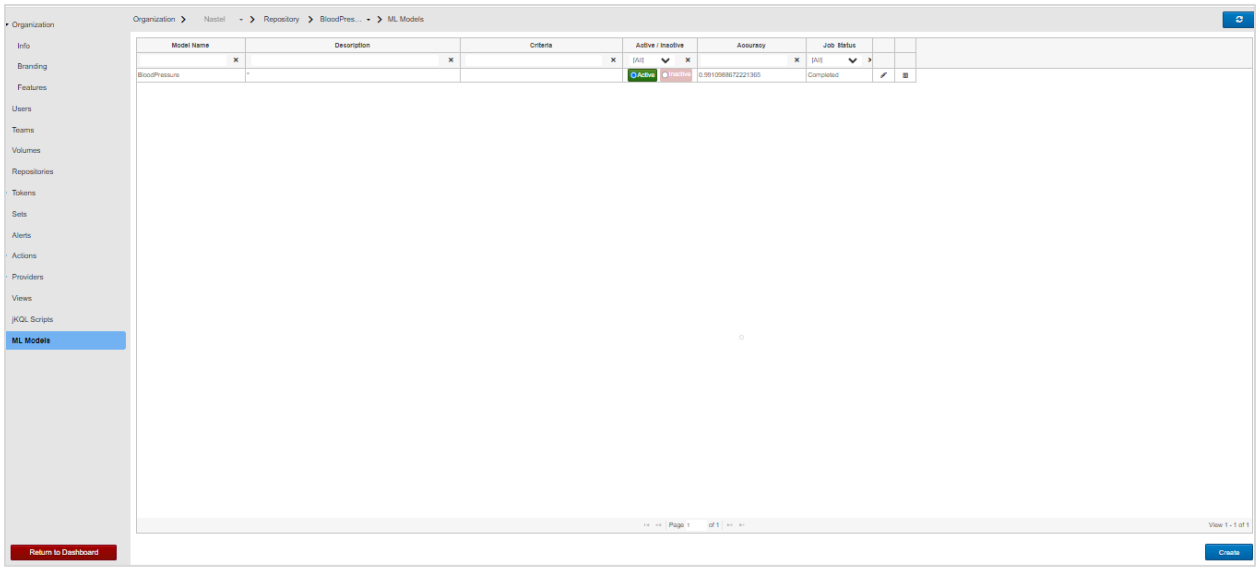


Figure 3.1.13.1-A. ML Models

3.1.13.1.1Dataset

The first step is to identify the data that you want to analyze using Machine Learning.

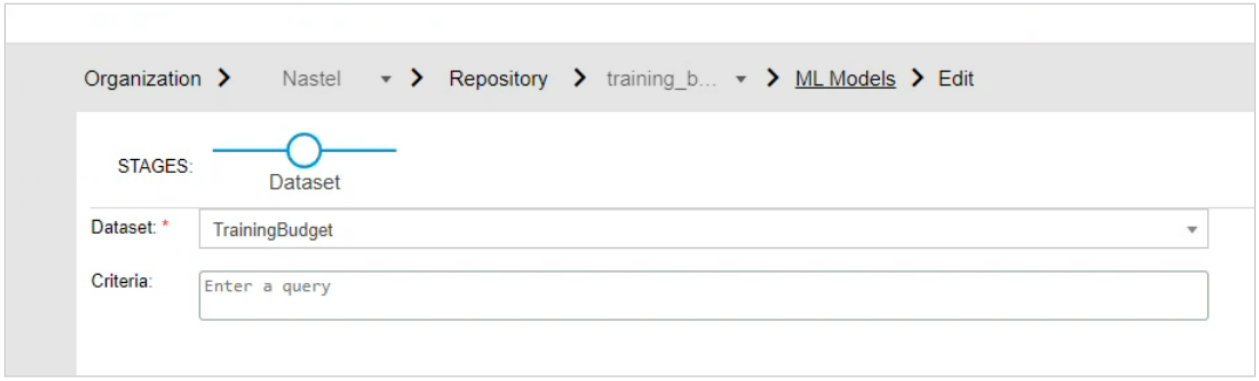


Figure 3.1.13.1.1. Dataset

Select the name of the **Dataset** you want to analyze. You can also enter a where clause in the **Criteria** field to constrain the volume of data that will be analyzed.

Click **Next**.

3.1.13.1.2Target

In the **Target** field, enter the specific field you wish to gain insight into. Note that the example below is not a time series example. For more information on time series, which involves forecasting one or more values out into the future, see the separate Time Series Example below.

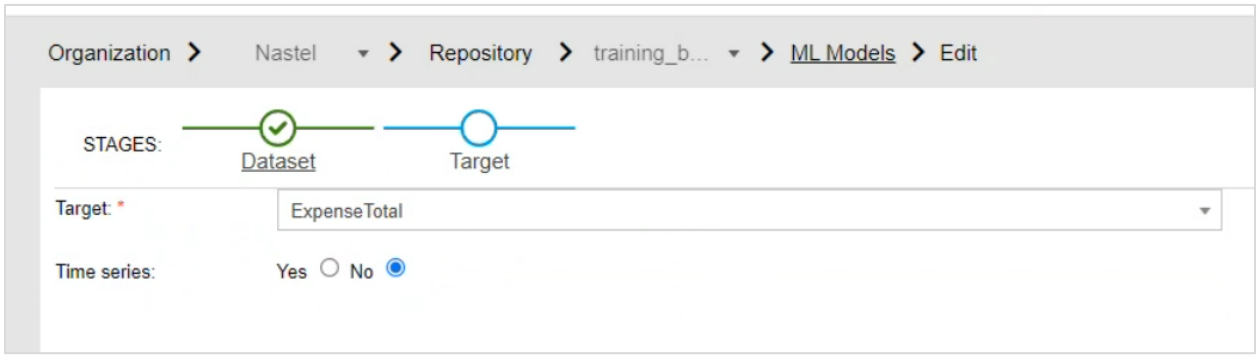


Figure 3.1.13.1.2. Target

Click **Next**.

3.1.13.1.3Variables

Next, help identify the fields that may impact your Target field. What are the potential driving factors behind your target? Select these factors from the **Available Properties** list.

Click **Next**.

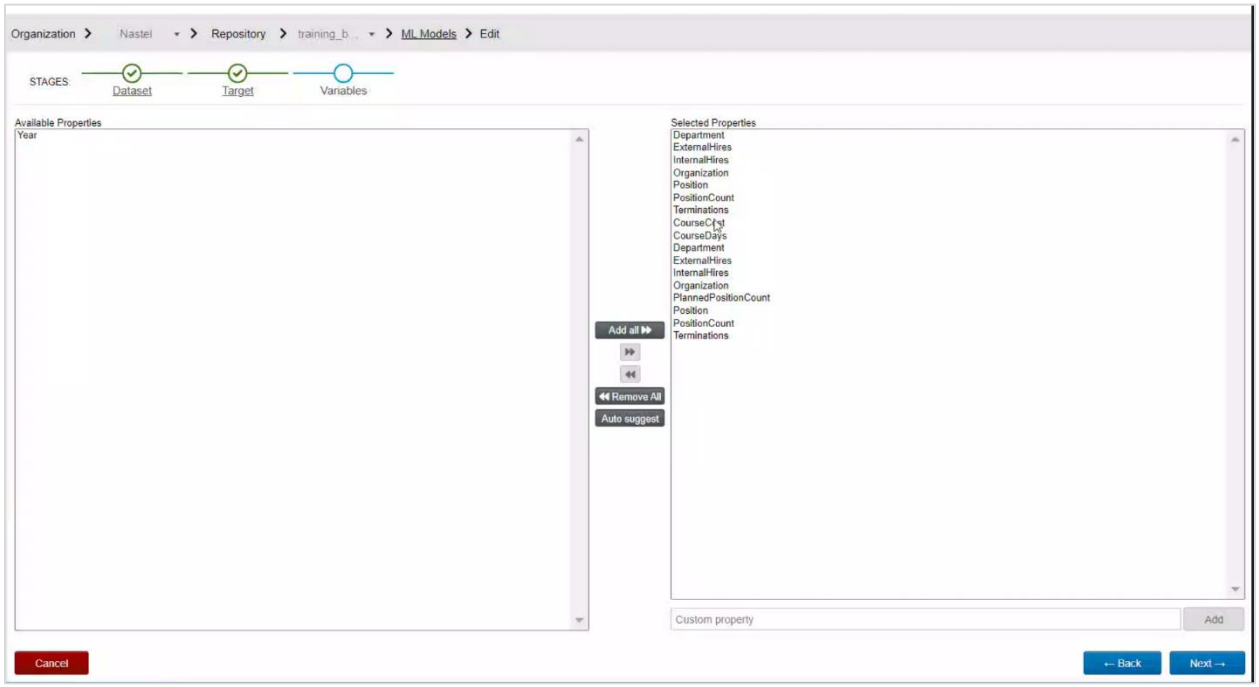


Figure 3.1.13.1.3. Variables

3.1.13.1.4Name

Enter the **Model Name** and a **Description**.

Click **Next**.

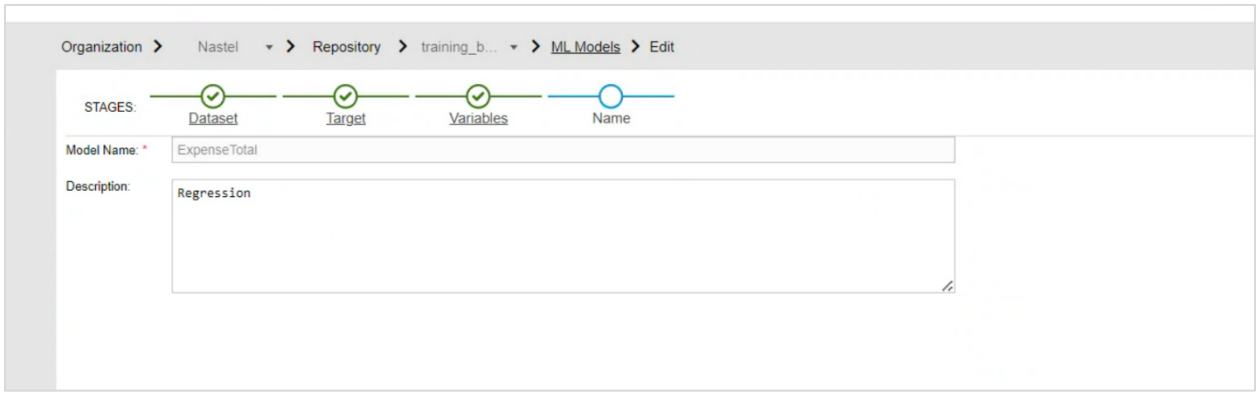


Figure 3.1.13.1.4. Name

3.1.13.1.5Model retraining

After the initial build, models must periodically be retrained. Select the training schedule frequency of the model you just created. You can choose days or hours. Enter the number of days or hours in the space provided.

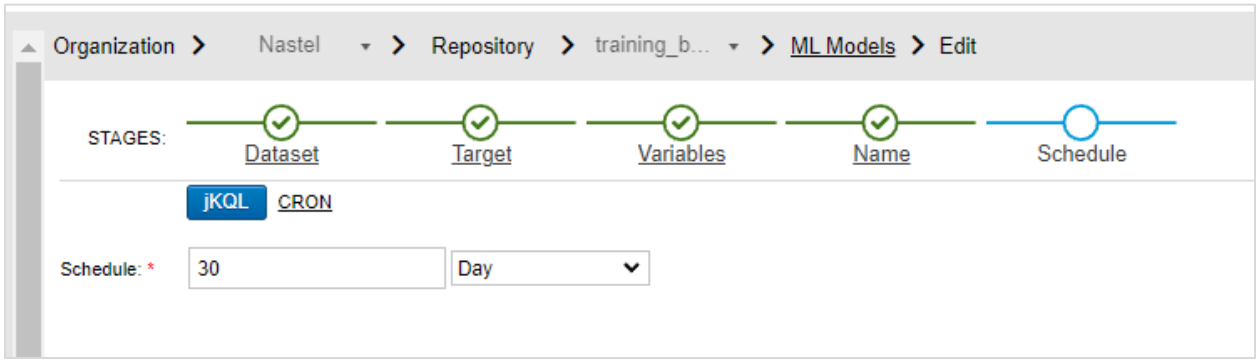


Figure 3.1.13.1.5. Model Retraining

Click **Finish** to save the new model. Model training begins automatically.

3.1.13.1.6After the model is defined

Once the models are built, you can run Machine Learning jKQL queries that perform analytics such as:

- FeatureSelection to determine driving factors
- Forecast for forecasting into the future and detecting anomalies
- Expected for finding out what a target is most likely to be given the driving factors
- Running what-if scenarios

3.1.13.2 Time Series Example

When dealing with time series data, you will need to build time series models. They're different from regular models in that they involve time. To make a model time series, in the **Time Series** field, indicate whether you wish to work with data in a time series manner (*Yes* or *No*). You should specify data as time series if you wish to forecast into the future at a specified interval. In the example below, the pageViewDate is forecast once a day.

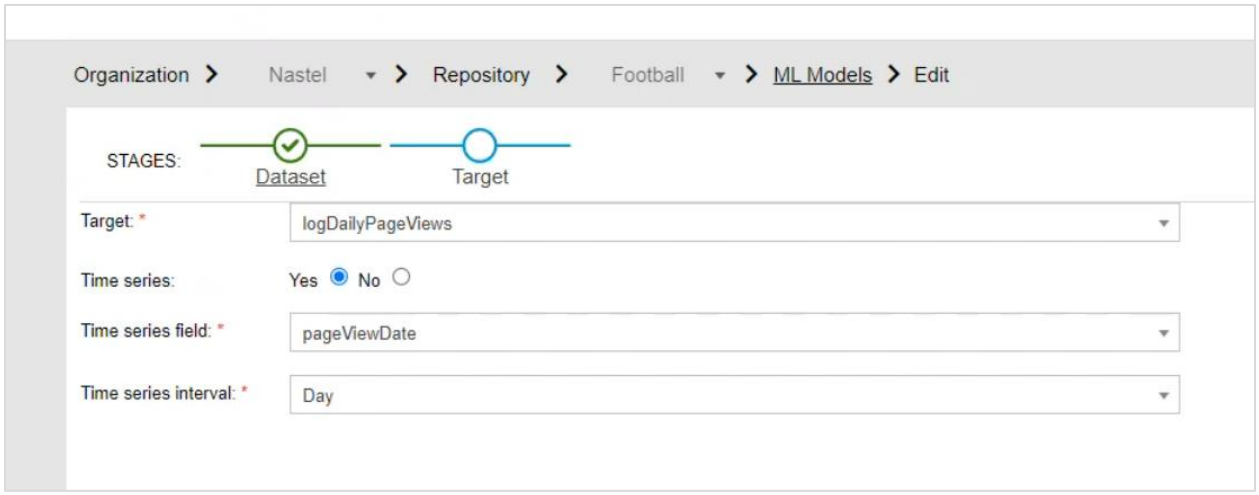


Figure 3.1.13.2-A. Time Series Example

For time series models only, an Anomalies step is presented as part of the wizard. You must configure anomalies detection using the fields provided.

- If you want anomalies to be processed in real time, when they occur, select the **Use Real Time Processing** check box.
- Select the **Anomaly Margin Type**: *Percent* or *Numeric*.
- If you chose Numeric, enter the margin of error for a numeric value to be considered an anomaly in the **Anomaly Margin** field.

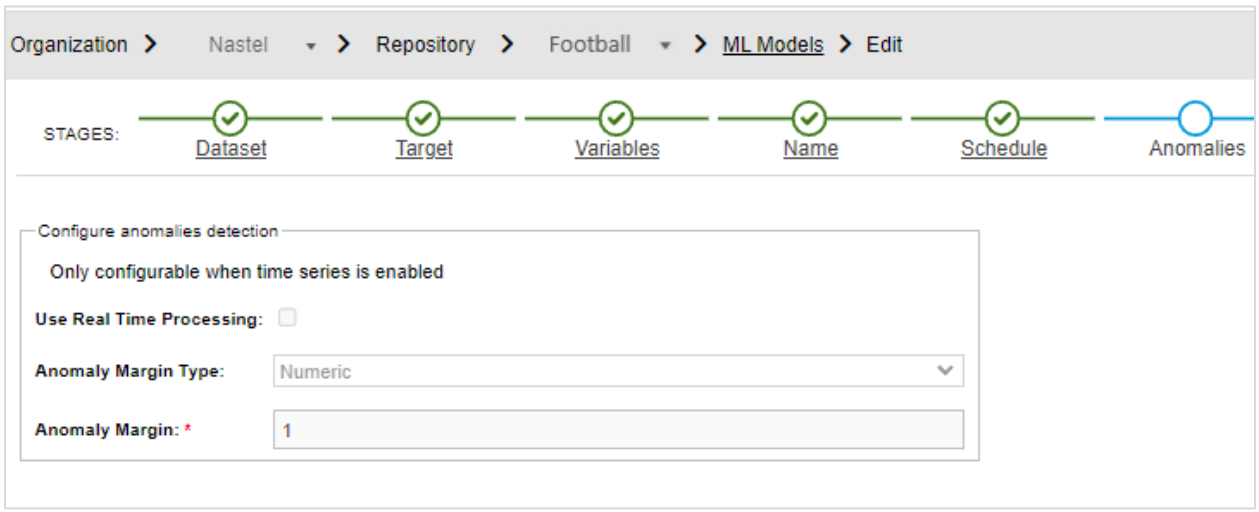


Figure 3.1.13.2-B. Anomalies Configuration

3.1.13.3 Generate a dashboard from a Machine Learning Model

You can generate a Machine Learning dashboard based on a jKQL Script like the one below.

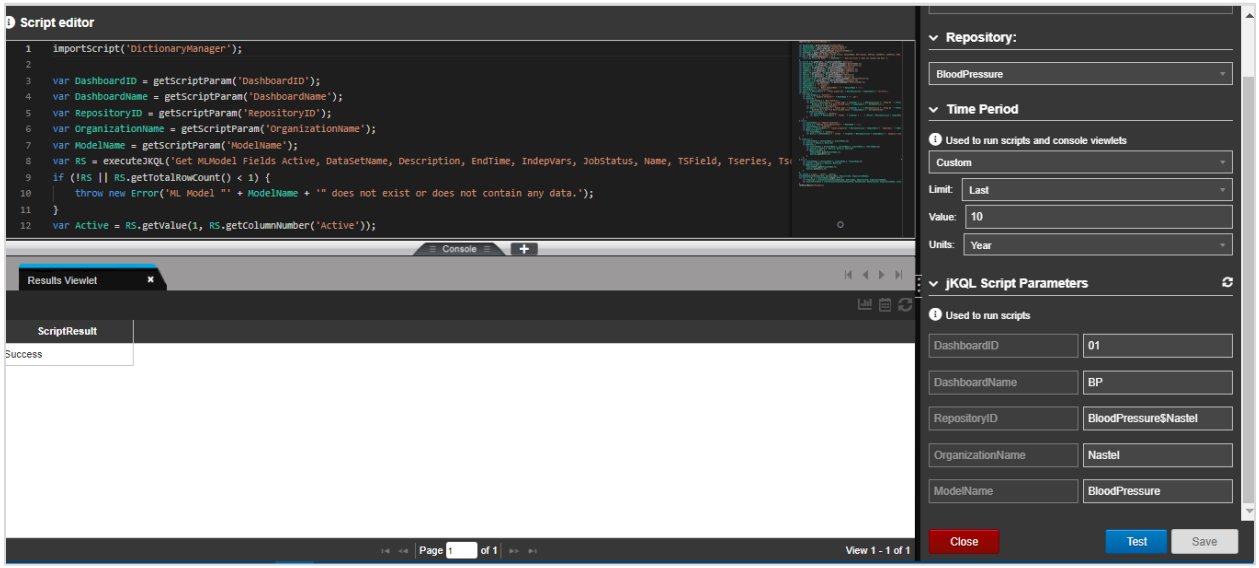


Figure 3.1.13.3-A. jKQL Script for Machine Learning

In the **jKQL Script Parameters** area on the lower right, be sure to specify the dashboard, repository, organization, and ML model information, as shown in the green box below.

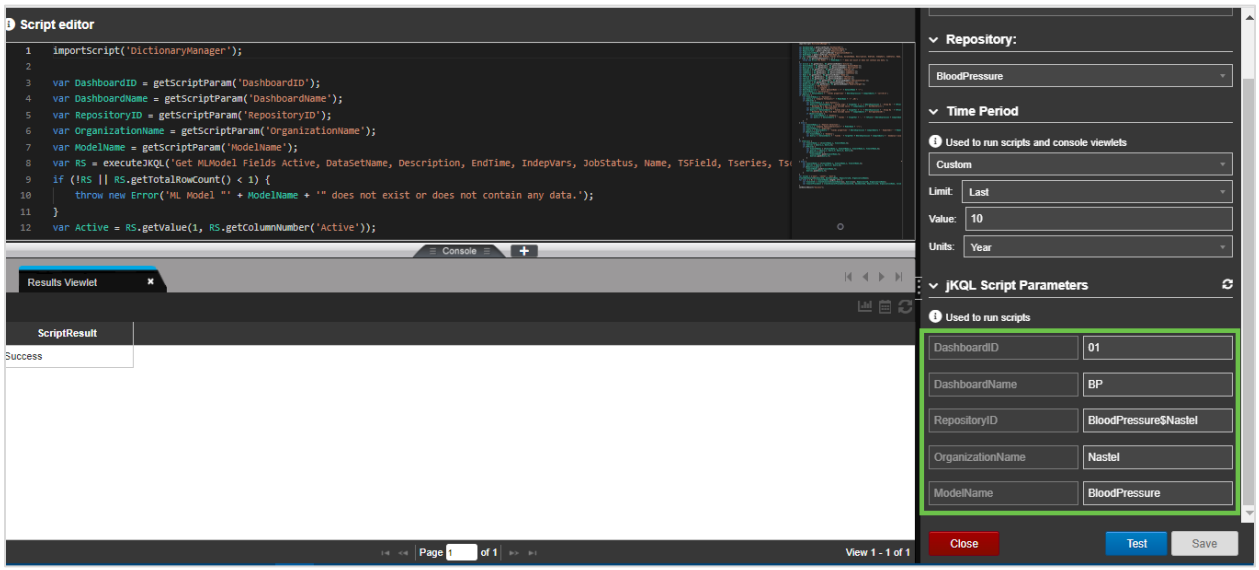


Figure 3.1.13.3-B. jKQL Script Parameters

NOTE

The RepositoryID must be the name of the repository followed by a dollar sign ($) and the Organization name. In the example above, BloodPressure$Nastel refers to the BloodPressure repository for the Nastel organization.

Click **Test** to make sure the ScriptResult is successful.

Click **Save** to save the script. Close the Script editor.

Create a temporary viewlet. At the jKQL prompt, enter `get MLModel` to return the Machine Learning model from which you want to create a dashboard.

Select the check box for the model and select **Generate Dashboard** from the pop-up menu.

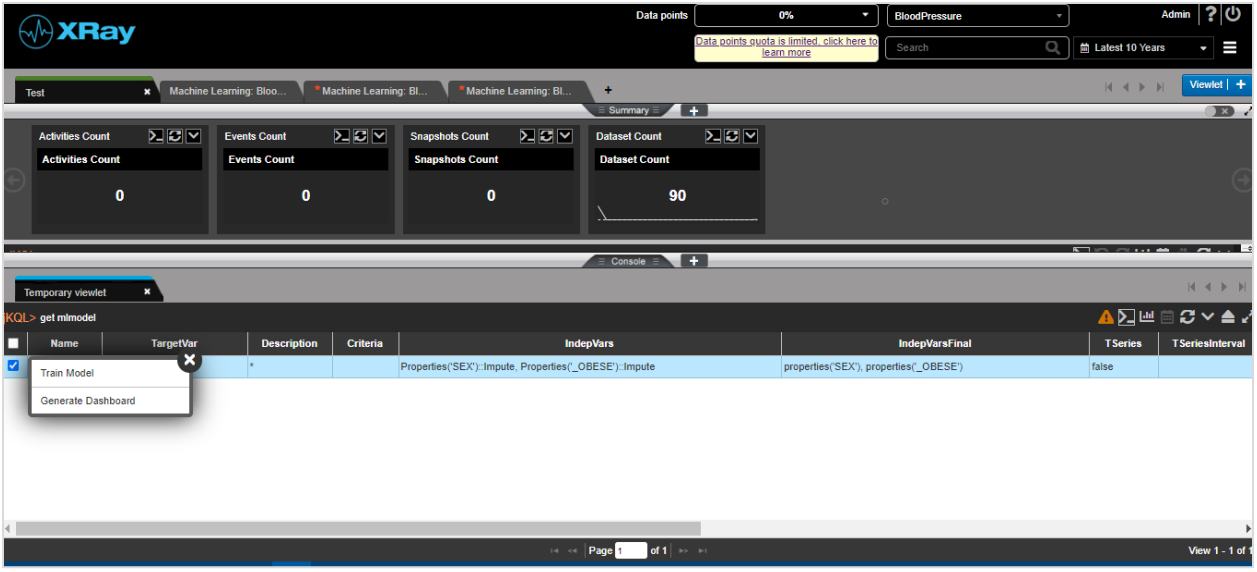


Figure 3.1.13.3-C. Generate Dashboard

A new dashboard is generated based on the model.

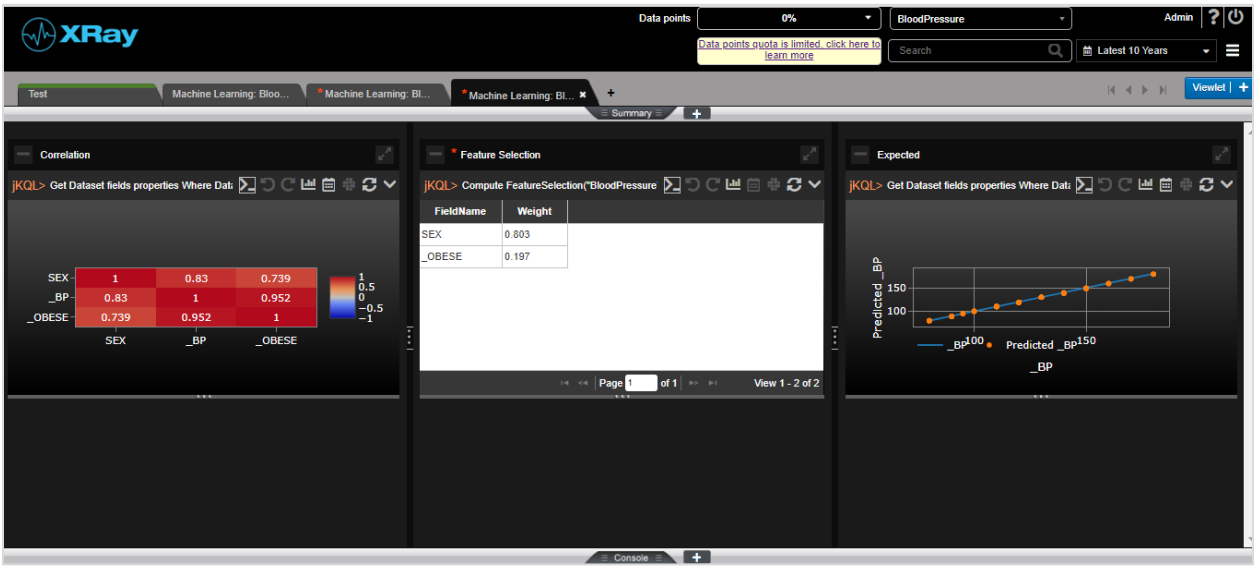


Figure 3.1.13.3-D. Machine Learning Dashboard

When the dashboard is complete, a "Dashboard generated successfully" message is displayed. Click **OK** and save the dashboard.

3.2 User Settings

There are settings that can be set at user level. Clicking **User Settings** from the **Main Menu** gives users the following options. See sections 3.2.1 – 3.2.6 for information on each option.

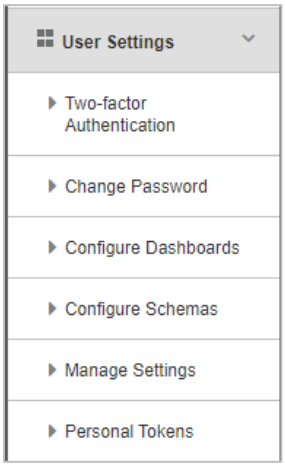


Figure 3.2-A. Main Menu > User Settings

3.2.1 Two-factor Authentication

Two-factor authentication (2FA) is used to ensure a secure login by requiring verification when logging in. A TFA app is required, for example, Google Authentication or FreeOTP.

3.2.1.1 Setup

For each user that will utilize 2FA, perform the following within each user's account to enable this feature:

1. Go to **Main Menu > User Settings > Two-factor Authentication**. The below window opens. Change the **Status** of two-factor authentication to **Enabled** and click **Save**.

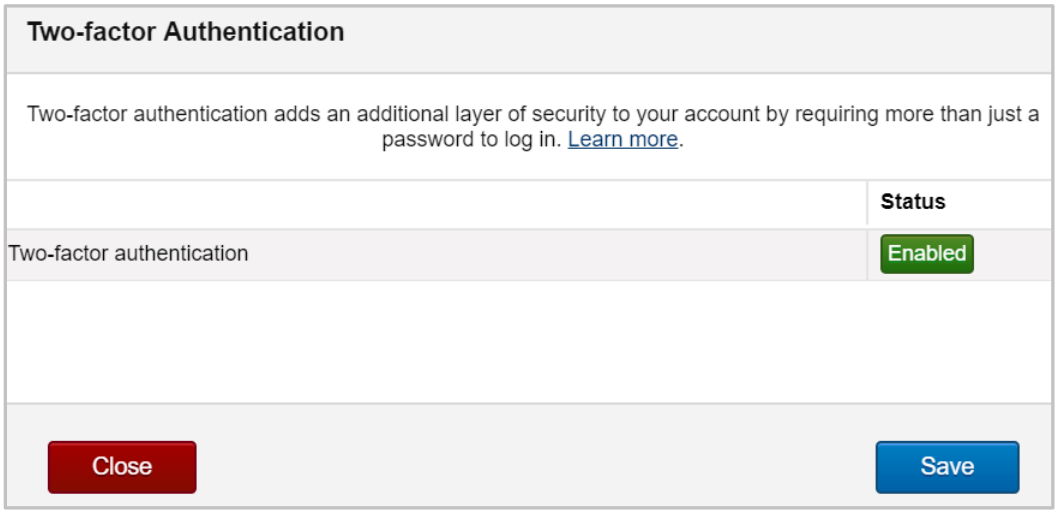


Figure 3.2.1.1-A. Two-factor Authentication

2. You will need to scan your personal token's QR code from your 2FA app. Click the QR icon which displays after saving the 2FA status as enabled. Scan the code with your 2FA app.

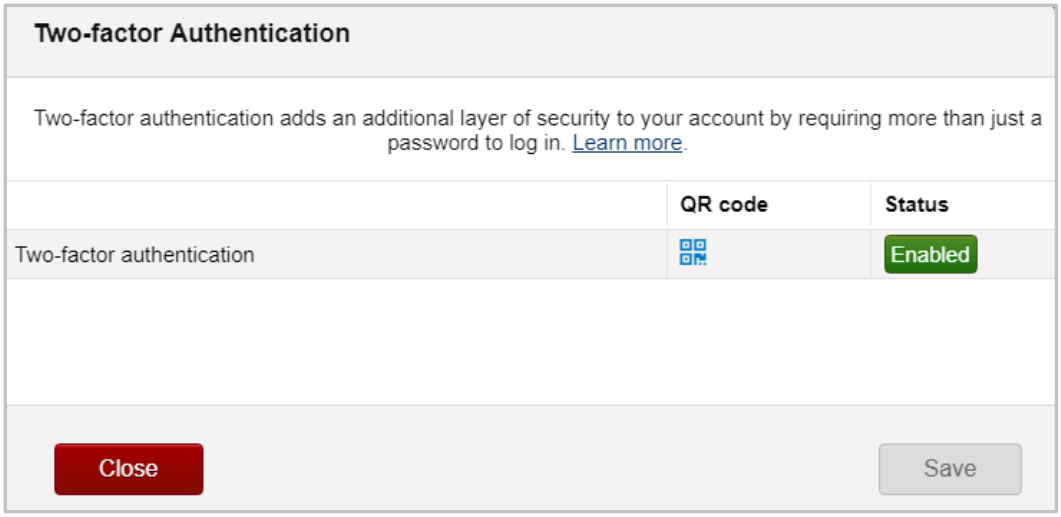


Figure 3.2.1.1-B. Two-factor Authentication – QR Code Generated

Figure 3.2.1.1-C. Two-factor Authentication – QR Code

3. From this point forward, the user will be required to enter a 6-digit code from the 2FA app when logging in.

3.2.1.2 Reset

To disable 2FA for a user, perform the below steps. Only organization users with admin roles have this ability.

1. Go to **Admin Menu** > **Organization** > **Users**.
2. Select the user.
3. Click **Reset two factor authentication**.

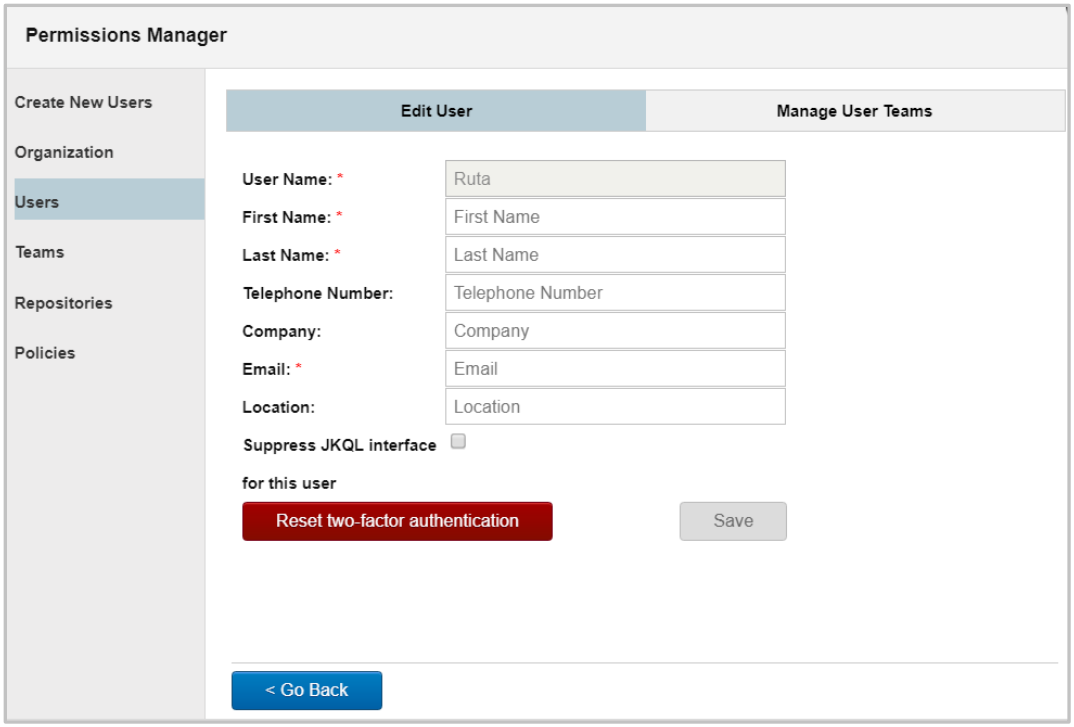


Figure 3.2.1.2-A. Reset Two-factor Authentication

3.2.2 Change Password

To change your user password, go to **Main Menu** > **User Settings** > **Change Password**. The following screen opens. Enter your current password within the **Existing Password** field. Enter your new password within the **New Password** and **Confirm New Password** fields. Click **CHANGE PASSWORD**.

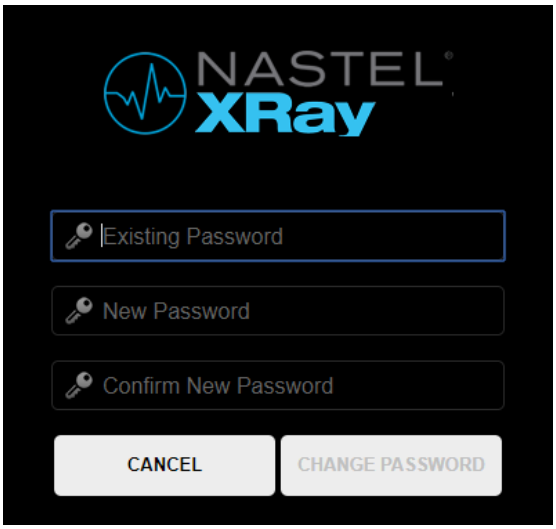


Figure 3.2.2-A. Change Password

3.2.3 Configure Dashboards

The *Configure Dashboards* dialog box is used for customizing a user's dashboards. To open the Configure Dashboard dialog box, go to **Main Menu** > **User Settings** > **Configure Dashboards**. On this dialog box, users have the option to:

- Rename dashboards
- Change dashboard layouts
- Delete dashboards

3.2.3.1 Rename

To rename a dashboard, go to **Main Menu** > **User Settings** > **Configure Dashboards**. Click the pencil icon of the dashboard you would like to rename.

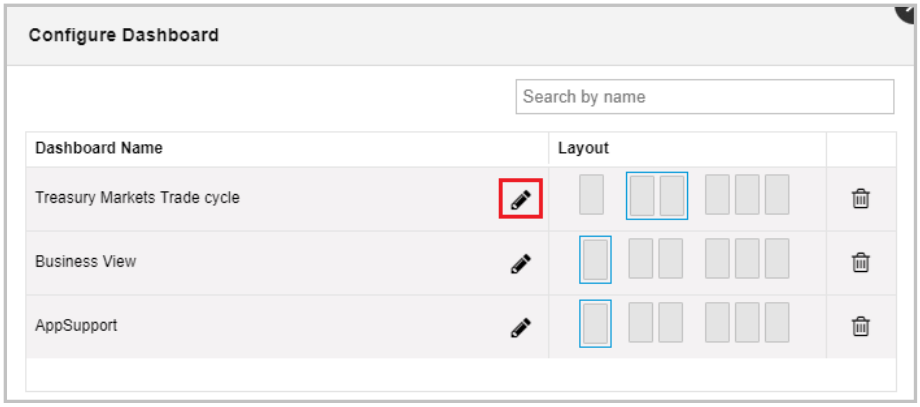


Figure 3.2.3.1-A. Configure Dashboard – Rename Dashboard

Enter a new name and click the check mark . Click **Save**.

3.2.3.2 Change Layout

Dashboard layouts can either be one, two or three columns. To change the layout of a dashboard, go to **Main Menu** > **User Settings** > **Configure Dashboards**.

The **Layout** field of each dashboard will have a blue box around the layout the dashboard is using. To change the layout of a dashboard, simply select the new layout and click **Save**.

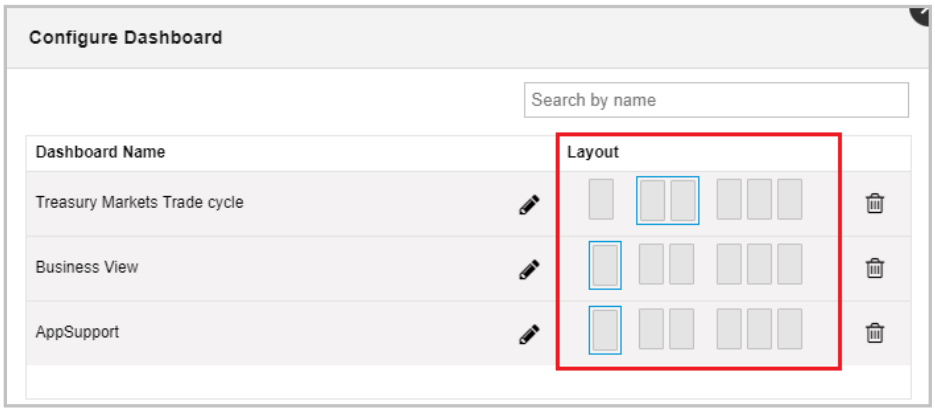


Figure 3.2.3.2-A. Configure Dashboard – Dashboard Layout

3.2.3.3 Delete Dashboard

To delete a dashboard, go to **Main Menu** > **User Settings** > **Configure Dashboards**. Your list of dashboards will display. Scroll through the list to find the dashboard you would like to delete or use the **Search by name** search field. Click the trash button to delete the selected dashboard.

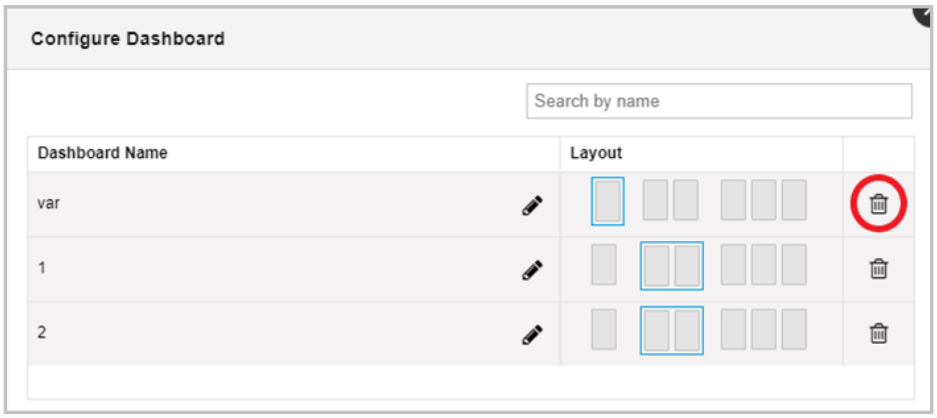


Figure 3.2.3.3-A. Delete Dashboard

After clicking the delete button, an **Undo** button will appear. Click this button if would like to cancel the delete action.

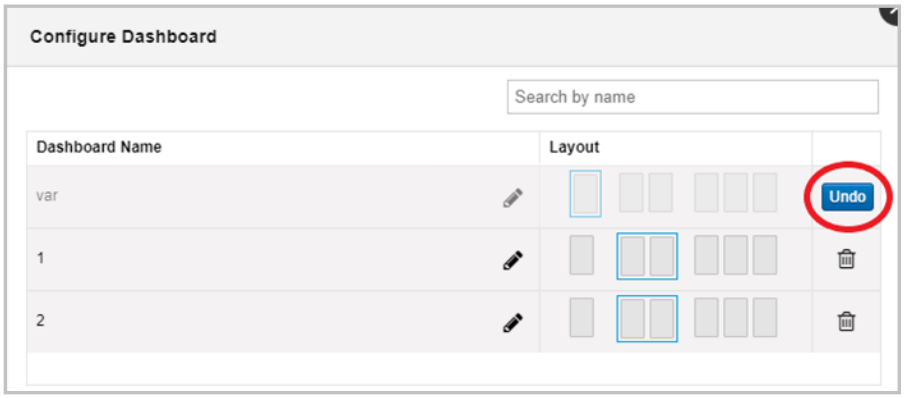


Figure 3.2.3.3-B. Delete Dashboard – Undo

To continue deleting, click **Save**. A confirmation dialog box appears. Click **Yes** to delete.

Click **No** to close the dialog box. You will be brought back to the *Configure Dashboard* screen where you can click **Undo** to cancel the deletion.

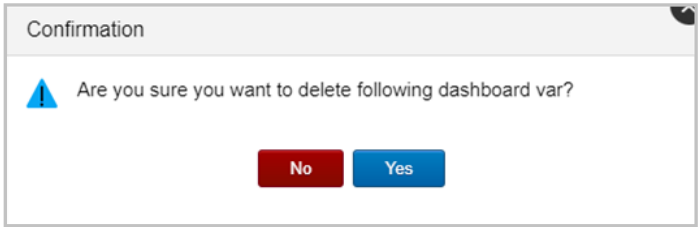


Figure 3.2.3.3-C. Delete Dashboard Confirmation

A confirmation will appear stating that the dashboard has been successfully deleted.

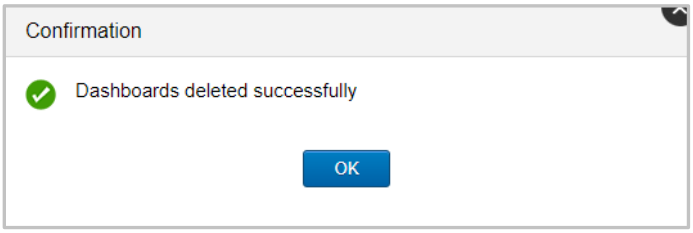


Figure 3.2.3.3-D. Dashboard Successfully Deleted

3.2.4 Configure Schemas

Select **Configure Schemas** from **User Settings** menu to create and maintain data view schemas. Schemas are used to customize how data is displayed in table viewlets using various filters. The **Schemas** window opens.

Click the pencil icon to edit existing schemas. To delete a schema, simply click the trashcan icon. Click the **Create** button to create a new schema (see the instructions in the Create / Edit Schema section below).

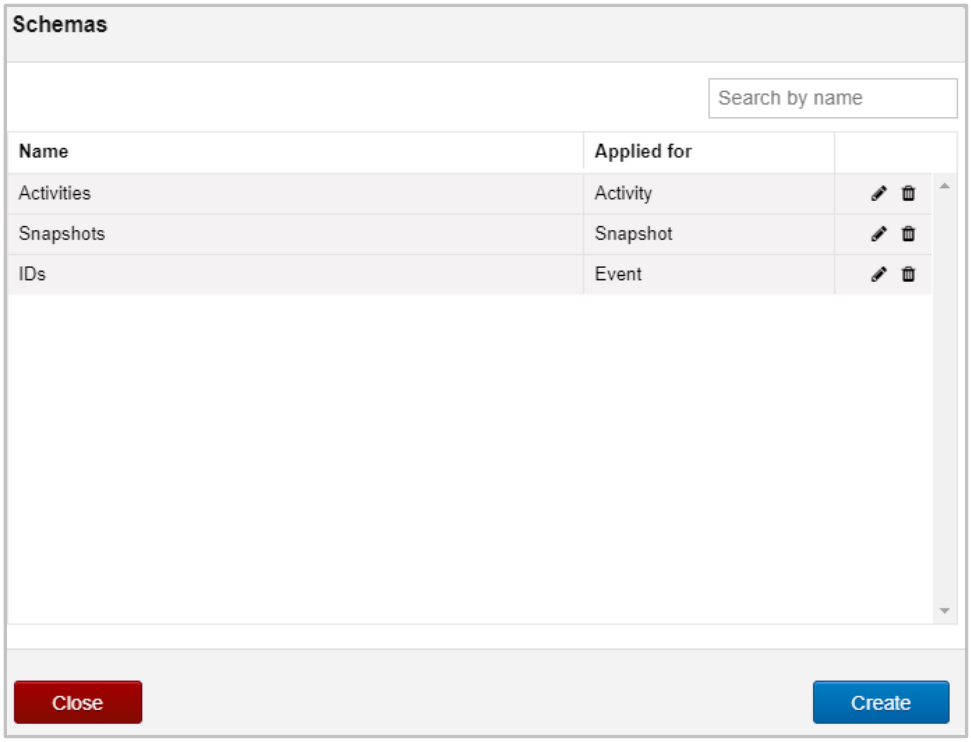


Figure 3.2.4-A. Schemas

3.2.4.1 Create / Edit Schema

To create a schema, click the **Create** button on the *Schemas* window (*Figure 3.2.4-A*). Specify the **Schema Name** (required). Multiple item types can be added to a schema (activity, event, snapshot). To add an item type, select it from the **Item Type** drop-down menu. Add filters to the item type by clicking the **Add fields** button.

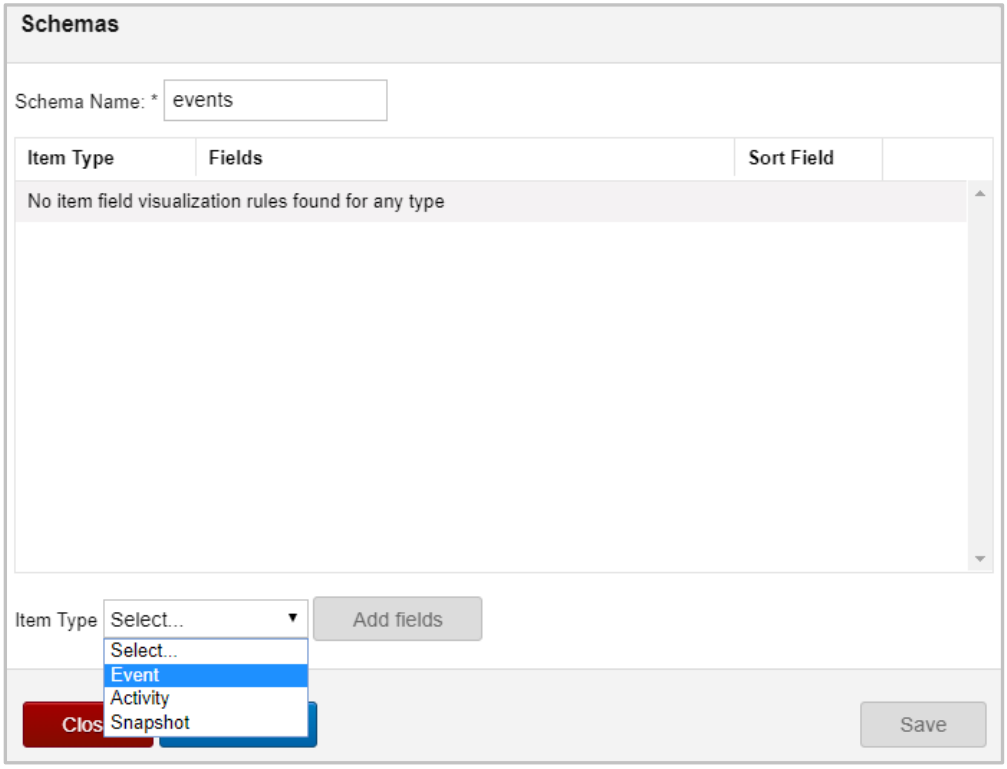


Figure 3.2.4.1-A. Create New Schema

On the window that opens, users specify what fields should be displayed in viewlets and the order in which they should be displayed.

From the **Available Fields** box located on the left side of the screen, select a field, and click the right arrow button » to add the field to the **Selected Fields** section on the right side of the screen. Multiple fields can be selected and added at one time by holding down the **Ctrl** key. Select the **Move all** button with right arrows Move all » to move all available fields.

Use the left arrow button « or the left Move all button « Move all to remove the fields from the **Selected Fields** section.

On the right side of the **Selected Fields** section, use **Move to Top**, **Move to Bottom**, **Move Up** and **Move Down** arrows to change the field sequence. This is the order the fields will appear in viewlets.

In the **Current item type fields sorted by** drop-down menu, all fields within the **Selected Fields** section will appear. Select a field to be used as the main field for sorting viewlets.

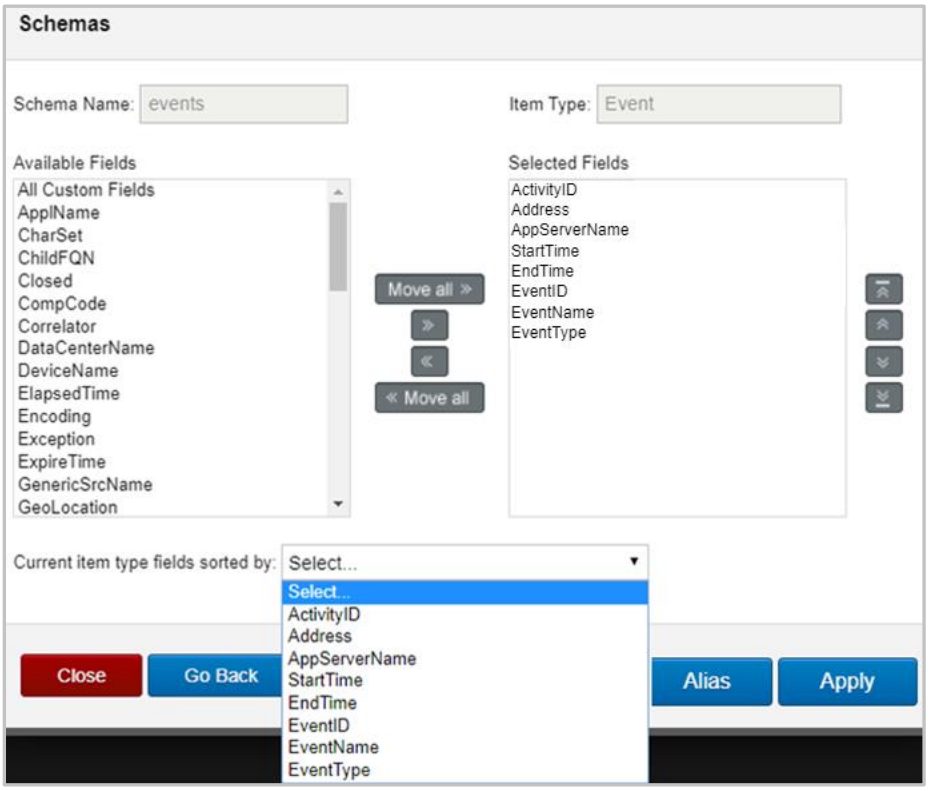


Figure 3.2.4.1-B. Add Fields for Schema

You can specify alias names for the selected fields. These names will be displayed in viewlets instead of the default names of the fields. To do this, click the **Alias** button. A screen opens listing all fields that were selected. Enter the alias name for all desired fields. If you do not want an alias name used for a field, simply leave it blank.

In the below example, EventID will display as "ID" and EventName will display as "Name" in viewlets.

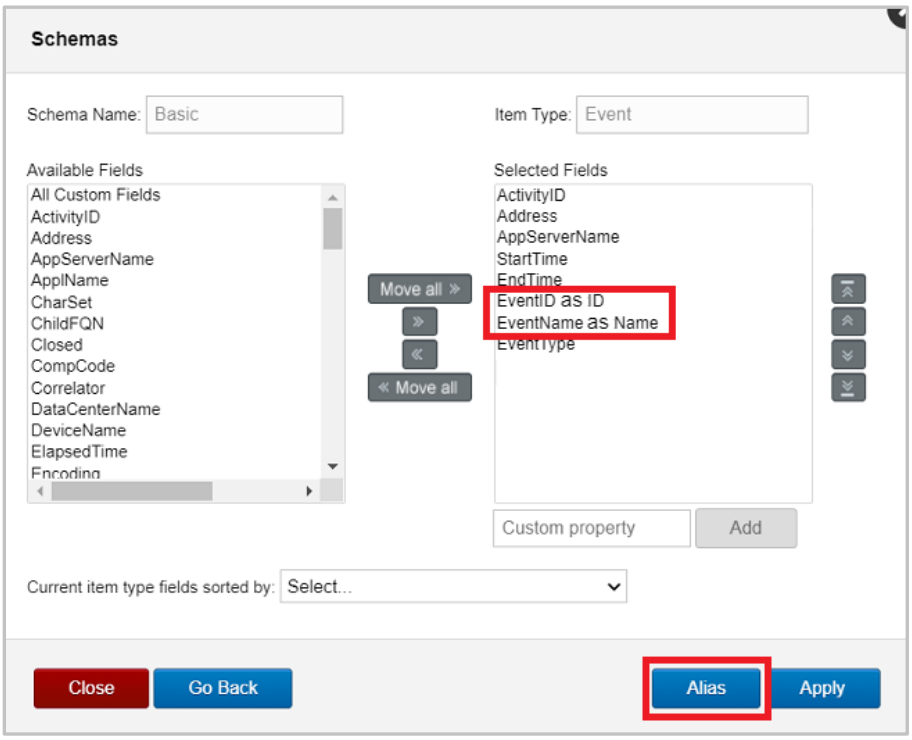


Figure 3.2.4.1-C. Alias Names

Click **Apply** when you are finished.

In the following window, the summary details of your newly created schema appear. Click the pencil icon to edit, or the trashcan icon to delete this schema. Click **Save** to save the configurations done.

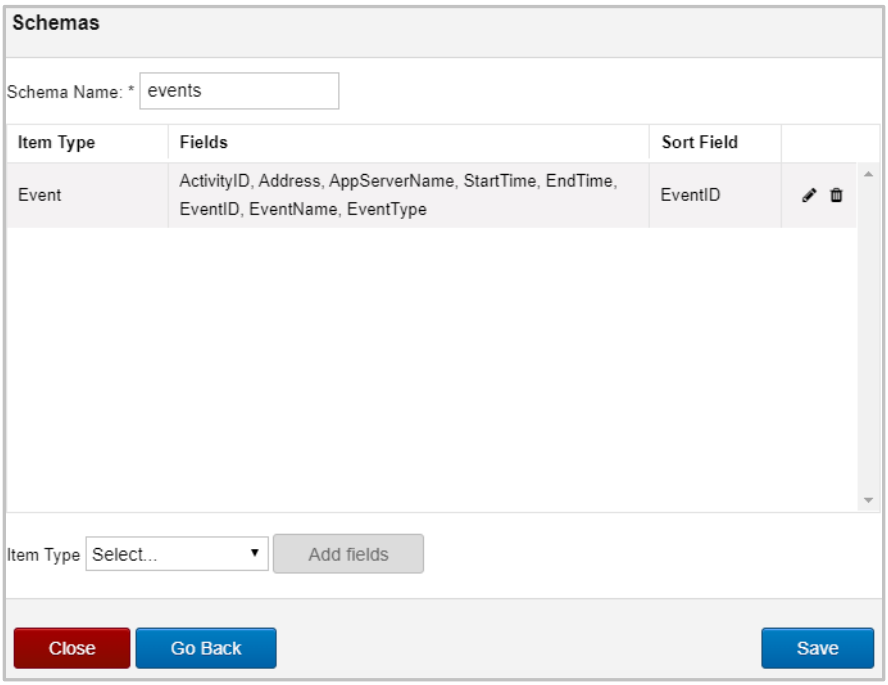


Figure 3.2.4.1-D. Schema Summary

The created schema now appears in a schemas' list. See Section 2.4.2.8, Change Layout, to learn how to apply a schema to a dashboard's viewlets.

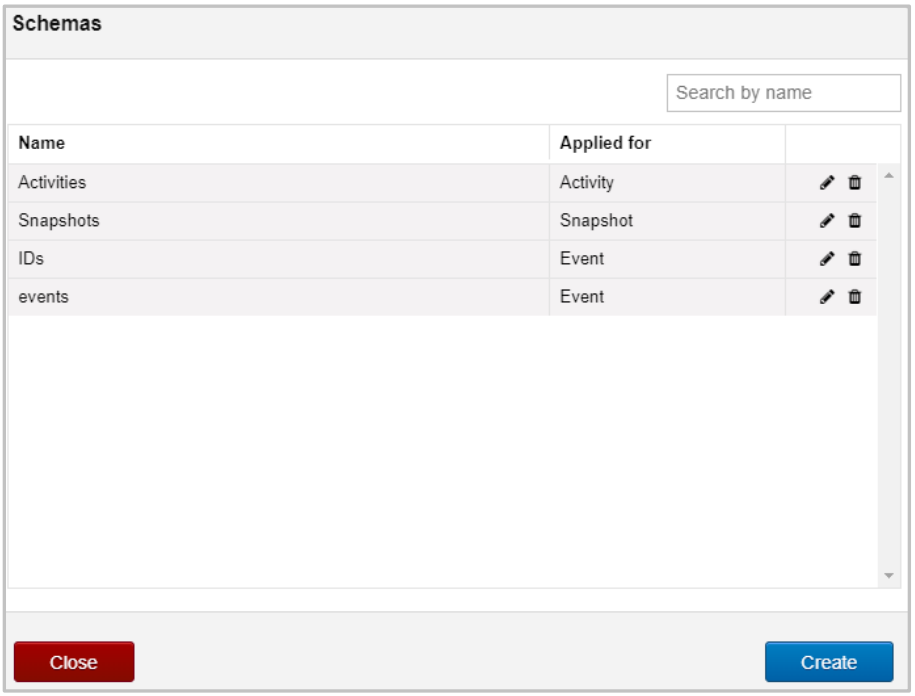


Figure 3.2.4.1-E. Schemas List

3.2.5 Manage Settings

The User Settings > Manage Settings menu allows you to control the default behavior of the application.

The General tab contains general settings. The settings shown reflect the user interface of XRay versions 1.5 and later. In earlier versions of XRay, the settings that follow are found in different locations: the Default Repository ID is found under User Settings > Repository; the Date & Time Range is found under User Settings > Date & Time range; the choices in the Colors tab are found under Admin Settings > Graph; and Summary Console, Landing page, and Auto Refresh are located under User Settings > Configure Dashboards.

The table below describes the settings on the General tab.

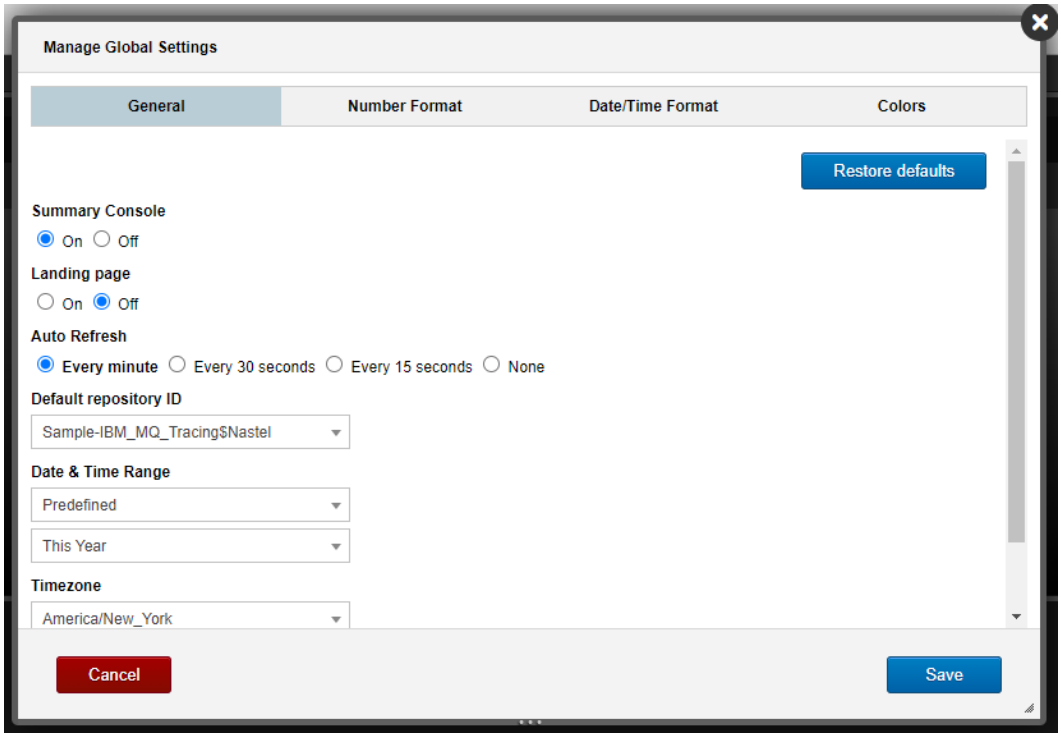


Figure 3.2.5-A. Manage Global Settings: General tab

Table 6. Manage Settings: General tab

Field	Definition
Summary Console	You can configure XRay to automatically have the Summary Console displayed or hidden every time you log in. Select either **On** (to display the Summary Console) or **Off** (to hide it). If you select **Off**, you can still view the Summary Console by clicking the **Summary** tab on the dashboard.
Landing Page	You can choose to show the Landing Page when you log into XRay (**On**) or to skip the landing page (**Off**). If the Landing Page is not displayed, when you log in, you will see your dashboard immediately.
Auto Refresh	The dashboard Auto Refresh interval specifies how often viewlets will be refreshed in all dashboards. It is especially useful for viewlets that display real-time data. To specify the refresh interval, select the desired refresh time.
Default repository ID	You can specify a default repository so that every time you log into XRay, that repository will automatically be loaded. Select a repository from the drop-down menu and click **Save**. From this point forward, the selected repository will be loaded when you log into XRay.

Table 6. Manage Settings: General tab	
Field	**Definition**
Date & Time Range	Set the default date and time range for dashboards and their viewlets for the repository you currently have open. Use the options within the drop-down menus. For more information see Section 2.3.5, Default Date & Time Range). Please note that the date and time range of viewlets and dashboards can still be updated, but the update will only be active within the current session. When logging back into the application, the default date and time range specified within the *Date & Time Range* dialog box above will be applied.
Timezone	Select your local Time zone.
Locale	Select a Locale to determine your language settings. On the following two tabs (Number Format and Date/Time Format), you can choose to base number, date, and time formatting on your Locale selection.

On the Number Format tab, choose the format that you want to use when numbers are displayed. You can choose to use browser settings or to format the numbers based on your Locale selection (on the General tab). Lastly, you can choose to design your own Custom format based on conventions such as which symbols to use for grouping and for the negative sign.

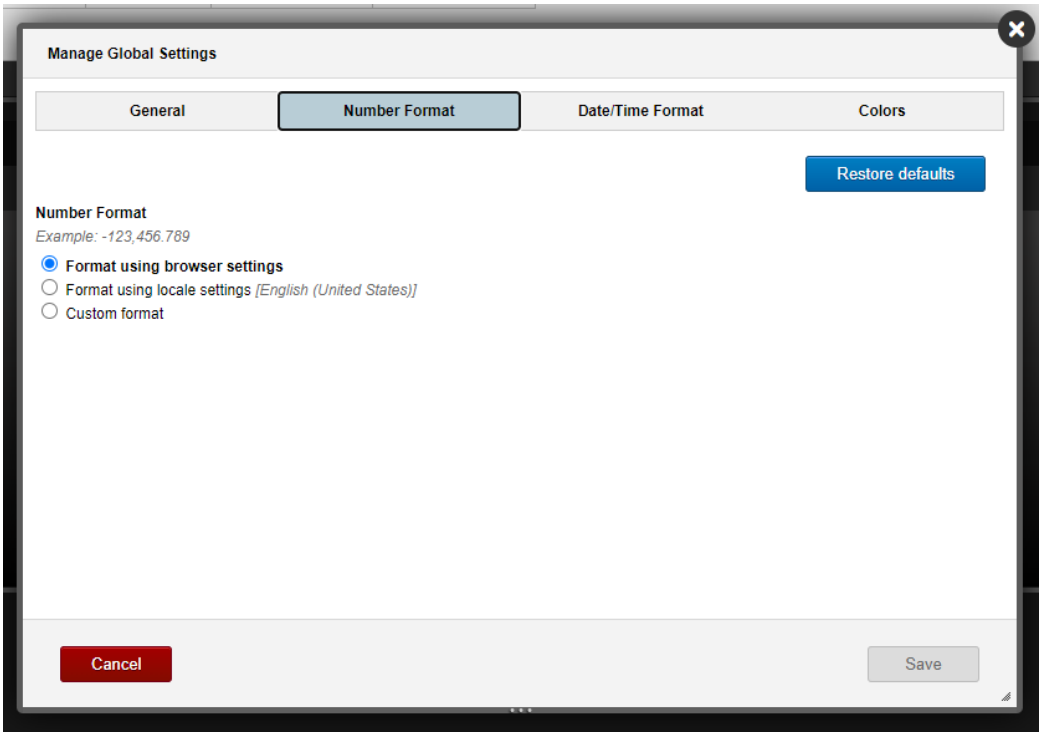


Figure 3.2.5-B. Manage Global Settings: General tab

On the Date/Time Format tab, choose the format that you want to use when dates and times are displayed. You can choose to use browser settings or to format dates based on

your Locale selection (on the General tab). Lastly, you can choose to design your own Custom format using codes to represent date and time elements. Click the information icon for a list of codes that you can use to build your own format.

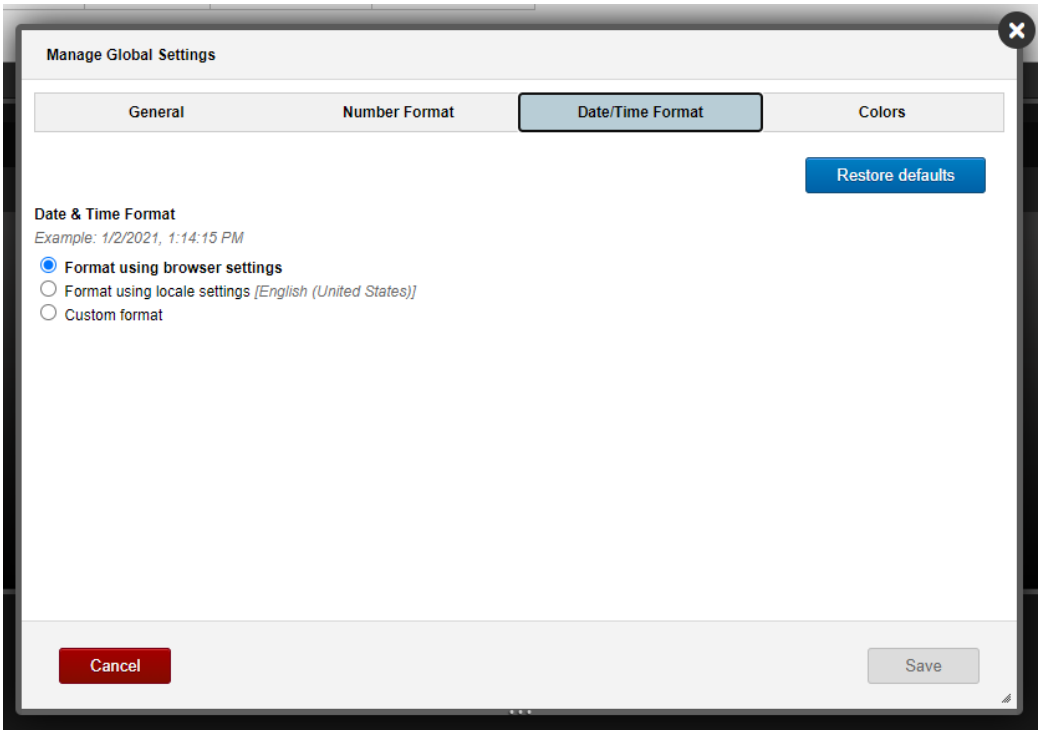


Figure 3.2.5-C. Manage Global Settings: General tab

Select the Colors tab to customize default colors for severity and status fields used in charts.

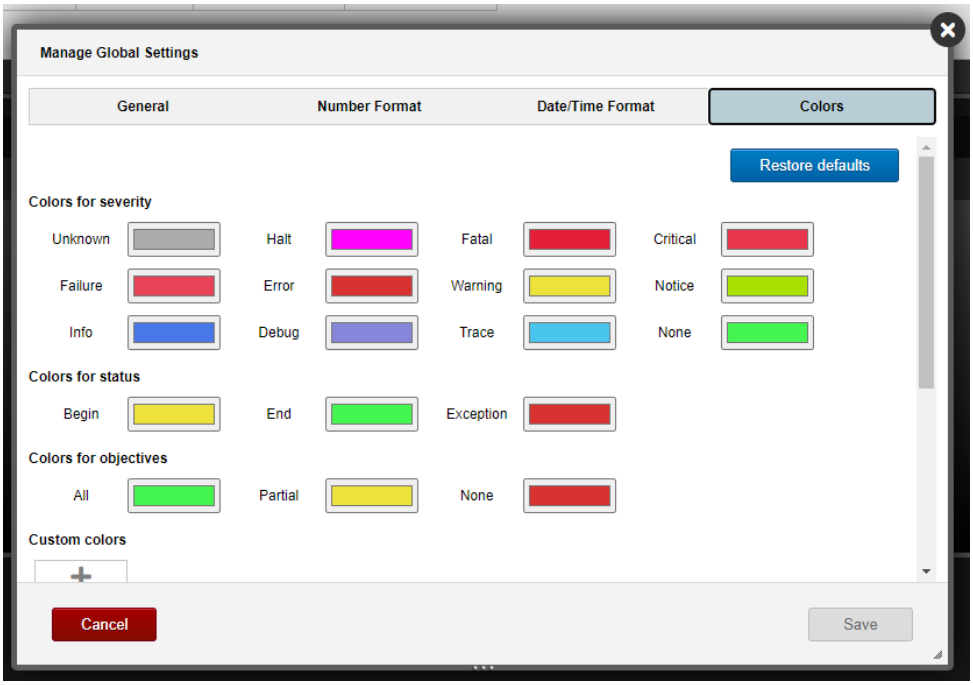


Figure 3.2.5-D. Manage Global Settings: General tab

3.2.6 Personal Tokens

Go to **Main Menu** > **User Settings** > **Personal Tokens** to view all tokens. Please note that the **Personal Tokens** option is only available on the **User Settings** menu when using a non-Global repository.

See section *3.1.6*, Tokens, for information about token setup.

Chapter 4: Functions

See below sections for the library of functions available in XRay.

4.1 General Functions

Table 7. General Functions

Function	Definition

`Cast(`***`expr`***`,`***`type`***`)`	Converts ***`expr`*** to the specified ***`type`***, where `type` is one of the following:
	`BINARY`
	`BOOLEAN`
	`DECIMAL`
	`INTEGER`
	`STRING`
	`TIMESTAMP`
	`TIMEINTERVAL`
	If ***`expr`*** cannot be converted to the specified ***`type`***, then `NULL` is returned.

`Coalesce(`***`expr`***`, ...)`	Returns the first non-`NULL` argument, or `NULL` if all arguments are `NULL`.
`Convert(`***`expr`***`,`***`type`***`)`	Synonym for `Cast`.
`FindIn(`***`item`***`,`***`list`***`)`	Returns the 0-based index of ***`item`*** in ***`list`***. If ***`item`*** is not found, returns -1.
`UUID()`	Returns a newly generated UUID.
`ValueAt(`***`pos`***`,`***`list`***`)`	Returns the item in 0-based position ***`pos`*** in ***`list`***. Returns null if ***`pos`*** is negative or >= ***`list`*** size.

4.2 Numeric Functions

Table 8. Numeric Functions

Function	Definition
`Abs(`***`x`***`)`	Returns the absolute value of ***`x`***.
`Ceil(`***`x`***`)`	Return the smallest integer value not less than ***`x`***.
`Ceiling(`***`x`***`)`	Synonym for `Ceil`.

Table 8. Numeric Functions

Function	Definition
`Exp(`***x***`)`	Returns Euler's number *e* raised to the power ***x*** (e^x) .
`Floor(`***x***`)`	Returns the largest integer value not greater than ***x***.
`Ln(`***x***`)`	Returns the natural logarithm of ***x***.
`Log(`***x***`)`	Synonym for `Ln`.
`Log10(`***x***`)`	Returns the base-10 logarithm of ***x***.
`Pow(`***x***`,`***y***`)`	Synonym for `Power`.
`Power(`***x***`,`***y***`)`	Returns `x` raised to the power ***y*** (x^y) .
`Round(`***x***`)`	Returns the closest integer to ***x***.
`Sqrt(`***x***`)`	Returns the square root of ***x***.

4.3 String Functions

Table 9. String Functions

Function	Definition
`Concat(`*expr*`,`*expr*`,...)`	Returns the string resulting from concatenating the string representation of each *expr* in order. `NULL` values are skipped.
`ConcatWS(`*sep*`,`*expr*`,`*expr*`,...)`	Returns the string resulting from concatenating the string representation of each *expr* in order, with each value being separated by *sep*, which must be a `STRING`. `NULL` values are skipped.
`Lcase(`*expr*`)`	Synonym for `Lower`.
`Left(`*expr*`,`*len*`)`	Returns the left-most *len* characters from string representation of *expr*.
`Len(`*expr*`)`	Synonym for `Length`.
`Length(`*expr*`)`	Returns the length of the specified *expr*. If *expr* is a list, returns the number of items in the list. Otherwise, returns the number of characters in the string representation of *expr*.
`Locate(`*expr*`,`*substr*`,`	Synonym for `Position`.

Table 9. String Functions

Function	Definition
[*pos*,[*occ*]])	
LocateRE(*expr*,*regex*, [*pos*,[*occ*]])	Synonym for PositionRE.
Lower(*expr*)	Returns the lower-case string representation of *expr*.
Position(*expr*,*substr* [,*pos*[,*occ*]])	Returns the 0-based index of the *occ* occurrence (default is 1) of *substr* in string representation of *expr*, starting at 0-based position *pos* (defaults to 0). Returns -1 if number of required occurrences of *substr* are not found.
PositionRE(*expr*,*regex* [,*pos*[,*occ*]])	Returns the 0-based index of the *occ* occurrence (default is 1) of substring matching *regex* in string representation of *expr*, starting at 0-based position *pos* (defaults to 0). Returns -1 if number of required occurrences of *substr* are not found.
Replace(*expr*,*substr* [,*repl*[,*pos*]])	Replaces each occurrence of *substr* in string representation of *expr*, starting at 0-based position *pos* (defaults to 0), with *repl*. If *repl* is not specified, then each occurrence of *substr* is removed.
Right(*expr*,*len*)	Returns the right-most *len* characters from string representation of *expr*.
StrAt(*expr*,*pos*[,*sep*])	Returns the string at 0-based position *pos* from result of splitting string representation of *expr* using *sep* as element separator. If *sep* is not specified, then string representation of *expr* is treated as a simple character array and returns the character at *pos* as a string.
SubStr(*expr*,*start*[,*end*])	Returns the substring from string representation of *expr*, starting at 0-based position *start* inclusive, ending at position *end*, exclusive. If *end* is not specified, then defaults to end of *expr*.
SubStrRE(*expr*,*regex* [,*pos*[,*occ*]])	Returns the *occ* (occurrence), or regex group (default is 1) of the substring from string representation of *expr*, starting at 0-based position *pos* (defaults to 0). Returns NULL if number of required occurrences of substring matching *regex* are not found.
Tokenize(*expr*[,*sep*])	Returns the list of strings formed by splitting the string representation of *expr* with *sep* being the separator between tokens (default is ",").

Table 9. String Functions	
Function	**Definition**
`Ucase(`*`expr`*`)`	Synonym for `Upper`.
`Upper(`*`expr`*`)`	Returns the upper-case string representation of *`expr`*.

4.4 Date and Time Functions

Table 10. Date and Time Functions	
Function	**Definition**
`CurrentTime()`	Synonym for `Now`.
`CurTime()`	Synonym for `Now`.
`DateAdd(`*`tstamp`*`,`*`intvl`*`)`	Adds time interval *`intvl`* to timestamp *`tstamp`*, returning the resulting timestamp.
`DateAdjust(`*`tstamp`*`,`*`cal`*`[,`*`dir`*`])`	Returns the timestamp resulting from adjusting the specified *`tstamp`*, based on the specified calendar component *`cal`* and the adjustment direction *`dir`*. *`cal`* is one of: `YEAR`, `MONTH`, `DAY`, `HOUR`, `MINUTE`, `SECOND`, `MILLISECOND`, `MICROSECOND`, `WEEK` `dir` is one of: `START`, `END` (if omitted, defaults to `START`) Example: `DateAdjust(StartTime, 'DAY', 'START')` returns the start of the day for timestamp in StartTime field.
`DateDiff(`*`tstamp1`*`,`*`tstamp2`*`)`	Returns the difference between the 2 timestamps (*`tstamp1 - tstamp2`*) as a time interval.
`DateExtract(`*`tstamp`*`,`*`cal`*`)`	Returns the value of the specified calendar component *`cal`* from timestamp *`tstamp`*.

Table 10. Date and Time Functions	
Function	**Definition**
	cal is one of: YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, MILLISECOND, MICROSECOND, WEEK
DayOfWeek(*tstamp*)	Returns the day of the week that timestamp *tstamp* falls on.
Now()	Returns current time as a timestamp.

4.5 Built-in Aggregate Functions

Table 11. Built-in Aggregate Functions	
Function	**Definition**
Apdex([DISTINCT] *expr*, *target*[,*tolerable*])	Returns the Apdex (Application Performance Index), which is a measure of satisfaction level, in the range 0.0 – 1.0, of the set of values for *expr* based on target value *target* and tolerable value *tolerable*, where 0.0 means totally unacceptable and 1.0 means totally satisfied. The target value is the value below which all values are satisfactory, or acceptable. The tolerable value is the value at or below which the values are tolerable. This value defaults to 4 times the target value. The Apdex formula is defined as follows: $Apdex = \frac{SatisfiedCount + 0.5(ToleratedCount)}{TotalCount}$ Where: *SatisfiedCount* is the number of *expr* values < *target* *ToleratedCount* is the number of *expr* values >= *target* and <= *tolerable* *TotalCount* is the total number of *expr* values (including those that are > *tolerable*). If DISTINCT is specified, returns the Apdex value from set of distinct values.

Table 11. Built-in Aggregate Functions

Function	Definition
`Average([DISTINCT] expr)`	Synonym for `Avg`.
`Avg([DISTINCT] expr)`	Returns the average of all `expr` values for group. If `DISTINCT` is specified, returns the average of distinct set of values.
`Close([DISTINCT] expr [,basedon])`	Returns the "closing" or "ending" value of *`expr`*, which is the value of *`expr`* having the maximum value of *`basedon`* expression. If *`basedon`* is not specified, then the default date field for item type in statement is used. `DISTINCT` is accepted but is ignored.
`Count([DISTINCT] expr)`	Returns the number of *`expr`* values for group. If `DISTINCT` is specified, returns the number of distinct values.
`List([DISTINCT] expr)`	Returns the comma-separated list of all *`expr`* values. If `DISTINCT` is specified, returns the list of distinct values.
`Max([DISTINCT] expr)`	Returns the maximum of *`expr`* values for group. `DISTINCT` is accepted but is ignored.
`Maximum([DISTINCT] expr)`	Synonym for `Max`.
`Mean([DISTINCT] expr)`	Synonym for `Avg`.
`Median([DISTINCT] expr)`	Returns the "middle" value, based on sorted order of all values for *`expr`*. If `DISTINCT` is specified, returns the middle value from set of sorted distinct values.
`Min([DISTINCT] expr)`	Returns the minimum of *`expr`* values for group. `DISTINCT` is accepted but is ignored.
`Minimum([DISTINCT] expr)`	Synonym for `Min`.
`Open([DISTINCT] expr [,basedon])`	Returns the "opening" or "starting" value of *`expr`*, which is the value of *`expr`* having the minimum value of *`basedon`* expression. If *`basedon`* is not specified, then the default date field for item type in statement is used. `DISTINCT` is accepted but is ignored.
`StdDev([DISTINCT] expr)`	Synonym for `StdDevPop`.
`StdDevPop([DISTINCT] expr)`	Returns the population standard deviation of all values for *`expr`*. If `DISTINCT` is specified, returns population standard deviation of distinct set of values.

Table 11. Built-in Aggregate Functions	
Function	**Definition**
`StdDevSample([DISTINCT] expr)`	Returns the sample standard deviation of all values for *expr*. If `DISTINCT` is specified, returns sample standard deviation of distinct set of values.
`Sum([DISTINCT] expr)`	Returns the sum of all *expr* values for group. If `DISTINCT` is specified, returns the sum of distinct set of values.
`Var([DISTINCT] expr)`	Synonym for `VariancePop`.
`Variance([DISTINCT] expr)`	Synonym for `VariancePop`.
`VariancePop([DISTINCT] expr)`	Returns the population variance of all values for *expr*. If `DISTINCT` is specified, returns population variance of distinct set of values.
`VarianceSample([DISTINCT] expr)`	Returns the sample variance of all values for *expr*. If `DISTINCT` is specified, returns sample variance of distinct set of values.
`VarPop([DISTINCT] expr)`	Synonym for `VariancePop`.
`VarSample([DISTINCT] expr)`	Synonym for `VarianceSample`.

4.6 Built-in Analytic Functions

Table 12. Built-in Analytic Functions	
Function	**Definition**
`Anomaly(expr, season)`	Will detect an anomaly on the value of expr. This function uses Netflix RAD Outlier detection which requires a season. The season will be either 'day/week' or 'hour/day'. Queries using this function must be grouped by a time and bucket by either week or day (depending on the season chosen). For example: `Get activity compute anomaly avg(ElapsedTime),'day/week') where name = 'Orders' and startTime > '2017-01-02' and starttime < '2017-02-01' group by starttime bucketed by day`

Table 12. Built-in Analytic Functions

Function	Definition
`anomalyDeepDiveRogueEdges()`	Once an anomaly is detected, anomalyDeepDiveRogueEdges can provide further insight into why the anomaly occurred. It will return records which when shown as a topology in the XRay UI, will color red and edges that contain numeric values that deviate from the norm for the edge by a specified margin.
`Average(`*`expr`*`)`	Synonym for `Avg`.
`Avg(`*`expr`*`)`	Returns the average of all `expr` values.
`BBands(`*`expr`* `[,`*`window`*`[,`*`stdevs`* `[,`*`useEMA`*`]]])`	Returns the Bollinger Bands based on value of *`expr`*. Bollinger Bands are used to measure the "highness" or "lowness" of a value relative to previous values. They consist of: • a *`window`*-period (default is `20`) moving average (MA). • an upper band at *`stdevs`* (default is `2`) times the N-period standard deviation above the moving average (MA + Kσ) . • a lower band at *`stdevs`* times an N-period standard deviation below the moving average (MA − Kσ) . The moving average is computed as an Exponential Moving Average (EMA) if *`useEMA`* is `true` (the default), or as a Simple Moving Average (SMA) if *`useEMA`* is `false`.
`BollingerBands(`*`expr`* `[,`*`window`*`[,`*`stdevs`*`[,`*`useEMA`*`]]])`	Synonym for `BBands`.
`EMA(`*`expr`* `[,`*`window`*`])`	Returns the Exponential Moving Average (EMA) based on value of *`expr`*. An EMA is a *`window`*-period (default is `20`) type of moving average that is similar to a simple moving average, except that more weight is given to the latest data. The general formula is: *"curEMA = "(("curVal - priorEMA")" * weight")" + priorEMA"* Where: *weight* = 2 / (window + 1)
`Max(`*`expr`*`)`	Returns the maximum of *`expr`* values.
`Maximum(`*`expr`*`)`	Synonym for `Max`.

Table 12. Built-in Analytic Functions

Function	Definition
`Mean(expr)`	Synonym for `Avg`.
`Median(expr)`	Returns the "middle" value, based on sorted order of all values for *expr*.
`Min(expr)`	Returns the minimum of *expr* values for group.
`Minimum(expr)`	Synonym for `Min`.
`SMA(expr[,window])`	Returns the Simple Moving Average (SMA) based on value of *expr*. An SMA is a *window*-period (default is `20`) type of moving average that gives equal weight to each data item. It is essentially the mean of the data items in the window.
`StdDev(expr)`	Synonym for `StdDevPop`.
`StdDevPop(expr)`	Returns the population standard deviation of all values for *expr*.
`StdDevSample(expr)`	Returns the sample standard deviation of all values for *expr*.
`Subanomaly(begin, end, anomaly-begin, anomaly-end, season, expr)`	Will provide further detail if an anomaly was detected when the `Anomaly` function was run from *begin* to *end* with the season and an anomaly was detected between anomaly-begin and anomaly-end. For example: `get activity compute subanomalies('2017-01-02','2017-02-01','2017-01-22','2017-01-23','day/week','avg(elapsedTime)')`
`Sum(expr)`	Returns the sum of all *expr* values for group.
`Var(expr)`	Synonym for `VariancePop`.
`Variance(expr)`	Synonym for `VariancePop`.
`VariancePop(expr)`	Returns the population variance of all values for *expr*.
`VarianceSample(expr)`	Returns the sample variance of all values for *expr*.
`VarPop(expr)`	Synonym for `VariancePop`.
`VarSample(expr)`	Synonym for `VarianceSample`.

Example

The following example is to compute the BollingerBands for events based on the average daily elapsed time based on a 10-day exponential moving average for this month:

```
jKQL> Get Events Compute BBands(Avg(ElapsedTime), 10) For This Month Group By
StartTime Bucketed by Day
```

Chapter 5: Using jKQL

The jKQL Data Query Language allows you to *talk* to your data. Create viewlets and modify them to get visually represented information about your data.

To generate viewlets, queries require certain components at a minimum. The syntax of a jKQL query includes the operation or action to be used for a specific item type, as well as, various date &time, numeric expressions, limiting operators, result grouping modifiers and viewlet chart type names. The syntax of a jKQL query should appear as follows (required elements are bolded):

```
jKQL> <action> <numeric expression> <limiting operator> <item type>
<date/time expression> <comparison operator> <result grouping modifier>
<viewlet chart type>
```

The table below lists basic query elements.

Table 13. Query Syntax Elements

Query Element	Expression
<action>	• Get (*Section 5.1*) • Subscribe to (*Section 5.3*) • Compare (*Section 5.4*) • Find (*Section 5.4*)
<numeric expression>	Specify any number of data items, which should be included in a viewlet – 5, 8, 10, etc.
<limiting expression>	These operators limit the query results. If the number of items to which the limits will be set was not specified, the default number will be "1". • Best • Bottom • Earliest • First • Largest • Last • Latest • Longest • Shortest • Smallest • Top • Worst
<item type>	jKQL recognizes singular or plural form of the expression, that is, Activity and Activities, so both forms are valid to use. • Activity • Event • Snapshot • Source

Table 13. Query Syntax Elements

Query Element	Expression
	• Resource • Set • Relative • Field • Provider • Action • Trigger • Log • Function • Repository • Team • Organization • User • Parameter • License Expressions 'Count of' or 'Number of' can be added before item type names.
`<date/time expression>`	jKQL recognizes singular or plural expression forms, that is, Year and Years, so both forms are valid to use. • Year • Month • Week • Day • Hour • Minute • Second • Millisecond • Microsecond These date and time expressions can be combined with numbers and limiting operators, for example last 5 years, latest 3 weeks and so on. Below is a list of date and time limiting operators: • Last • Next • Latest • Earliest • This • Today • Yesterday • Tomorrow

Table 13. Query Syntax Elements	
Query Element	**Expression**
	An exact time value can be added for certain time expressions, for example, yesterday at 9 am.
`<comparison operator>`	• Starts with all / any of <*listed items*>; Does not start • Matches all / any / none of <*listed items*>; Does not match • Contains all / any / none of <*listed items*>; Does not contain • Ends with all / any / none of <*listed items*>; Does not end • Is (=); Is not (!= or <>) in <*list of items*> • Has all / any / none of <*listed items*>; Does not have • Between; not between • > or >= – greater than or equal to specified *expression* • < or <= – less than or equal to specified *expression*
`<result grouping modifier>`	• Group by • Sort by • Order asc / desc (ascending / descending) • Bucketed by See Section 5.4, Additional Query Options, for more information.
`<viewlet chart type>`	'Show as' expression must be used before specifying the chart type. All chart types are noted in Section 5.2, Show As. 'Show as table' is the default.

For more information on the data model and functions jKQL supports, please see the *jKQL User's Guide*.

5.1 Get

The **Get** statement is used for retrieving items from the database. Please see the table below for examples.

Table 14. Get Examples	
Definition	**Query Statement**
To get default fields for all Activity items	`jKQL> Get Activities`

To get all fields for all Activity items in Set "*Purchasing*"	`jKQL> Get Activity Fields All from 'Purchasing'`
To get the number of Activity items in Set "*Purchasing*"	`jKQL> Get number of Activities from 'Purchasing'`
To get the number of Activity items in Set "*Purchasing*" that started today	`jKQL> Get number of Activities from 'Purchasing' for today`
To get the 10 longest running Activities in Set "*Purchasing*" that started today	`jKQL> Get top 10 Activities from 'Purchasing' for today sort by ElapsedTime desc`
To get the number of Activities in "*Payment*" last week grouped by their start time	`jKQL> Get number of Activities from Payment for last week group by starttime`
To get the number of Activities in Set "*Purchasing*" for each Activity status for the last week	`jKQL> Get number of Activities from 'Purchasing' for last week group by Status`
To get the number of Activities in Set "*Purchasing*" that met all objectives	`jKQL> Get number of Activities from 'Purchasing' that met all objectives`
To get the number of Activities in Set "*Purchasing*" that did not meet some objectives	`jKQL> Get number of Activities from 'Purchasing' that have not met all objectives`
To get the number of Activities in Set "*Purchasing*" that did not meet objectives "*A*" and "*B*"	`jKQL> Get number of Activities from 'Purchasing' that have not met objectives 'A','B'`
To get Activities in Set "*Purchasing*" that did not meet objectives "*A*" and "*B*" from set "*Web Purchases*"	`jKQL> Get Activities from 'Purchasing' that have not met objectives 'A','B' from 'Web Purchases'`

5.2 Show As

Users can specify the format of the displayed results by using **show as** at the end of a query. **Show as Table** is the default. Other **show as** viewlet type options are column chart ("`colchart`"), bar chart ("`barchart`"), line chart ("`linechart`"), pie chart ("`piechart`"),

stack chart ("stackchart"), geo map ("geomap"), scorecard ("scorecard"), area chart ("areachart"), summary ("summary"), topology ("topology"), anomaly chart ("anomalychart"), compare table ("comparetable") and histogram ("histogram").

The following is an example:

```
jKQL> Get relatives show as topology
```

5.3 Subscribe To

The **Subscribe** statement is used for submitting real-time queries, which are queries that are evaluated as the data is streamed in. The following are examples:

- `jKQL> Subscribe to Number Of Event group by Severity output every 5 seconds show as barchart`
- `jKQL> Subscribe to Number Of Event group by Severity output every 5 seconds ORDER BY severity show as piechart`
- `jKQL> Subscribe to number of events where eventname contains 'Order' group by eventname, severity order by severity output every 2 seconds show as colchart`

5.4 Additional Query Options

Enrich your queries with additional items such as:

- **Time ranges –** Month, day, hour
  `jKQL> Get events for this month`
- **Group by –** Creates a row for each unique set of values for columns being grouped on
  jKQL> Get events fields location where eventname contains 'order' group by location show as barchart
- **Buckets –** Bucketing allows multiple "*group by*" result rows to be combined into a single result row. Used when a "*group by*" statement returns too much data. Bucketing can only be applied to `INTEGER`, `DECIMAL`, `TIMESTAMP`, and `TIMEINTERVAL` data types

jKQL> Get number of events group by starttime bucketed by minute show as anomalychart
* this query will use the auto bucketing type

```
jKQL> Get number of activities group by snapshotcount bucketed by size 7 show as histogram
```

* this query will use the "*size*" bucketing type. It displays data, divided in multiple intervals according to the size range specified.

```
jKQL> Get number of activities group by eventcount bucketed by count 3 show as table
```

* this query will use the "*count*" bucketing type. It displays data, divided into a specified number of intervals.

- **Locations –** Geolocation
  jKQL> Get Event for This Month where Location ='London, England'
- **Sort by –** Sorting criteria
  jKQL> Get Activities from 'Purchasing' for today sort by ElapsedTime desc
- **Order by –** Sort data in ascending (asc) or descending (desc) ordering
  jKQL> `Get Events order by eventID desc`
- **Last –** Filter data for a specified time range

jKQL> Get number of events for last 5 days

- **Latest –** Starts filtering from latest time period, which includes displayable data. This differs from "*last*" which could return nothing as there may not be any events in the last 5 days

  jKQL> `Get number of events for latest 5 days group by starttime bucketed by day, severity show as stackchart`

- **Compare** – Compare two or more items (*section 2.5.4.1.1.8*). Use the expression "*only diffs"* to display only differences in the table.

  jKQL> `Compare Activity where ActivityID in ('activity ID of first selected activity', 'activity ID of second selected activity') show as comparetable`

  jKQL> `Compare Event where EventID in ('event ID of first selected event', 'event ID of second selected event') show as comparetable`

  jKQL> `Compare only diffs longest 2 event show as comparetable`

- **Find –** Search through the items (*section 2.3.4*)

jKQL> Find 'order' in Activities
jKQL> Find 'critical' in Events
jKQL> Find 'CPU' in Snapshots

- **Modify** – Filter viewlet data (*Section 2.5.8.1*). Use the expression, "*where <item_type>=${E:<item_type>:<item>:<item_type>}*".

jKQL> get number of Event where EventType = ${E:event type:Event:EventType} group by Severity show as colchart
*Change the values between the curly brackets
jKQL> get relative where RelativeType = ${E:Relative Type:Relative:RelativeType} show as topology get Relative where RelativeType = 'SendTo' show as topology
jKQL> get event containing all of ${SM:<filter_name>} show as table
jKQL> get Event where Severity = ${E:severity:Event:Severity} AND ElapsedTime > ${V:etime} AND CompCode >= ${E:compcode:Event:CompCode} show as table
* *"AND"* expressions can be used to apply multiple variables

Chapter 6: Use Case Examples

6.1 Root Cause Analysis of Application Performance Problems

Figure 6.1-A. Root Cause Analysis

XRay uses machine learning to detect anomalies in time-series data and can automatically determine the probable root cause of this anomaly. It can create a dynamic visualization of application topology and show the chain of causality between the anomaly and the applications that it has impacted. It can also detect if any business objectives or SLAs were impacted by this anomaly.

The sample viewlet above is using machine learning to detect anomalies. This scenario is based on real data representing airport terminals and flights. We have represented an airline at a terminal as an application, a terminal as a server, a data center as an airport, and the sky as a resource.

An anomaly was detected on February 18th with an average delay for the day of 45 minutes. If we click on the anomaly, we are transported to the console for a drill-down showing the topology of that anomaly. The graph shows a US Air flight traveling from Charlotte (CLT) to Phoenix (PHX). The red edges, called rogue edges, represent a problematic relationship between the terminal in Charlotte and the one in Phoenix. Clicking on the rogue edge provides a root-cause analysis of the problem. There was a delay at Charlotte, and it took 8 times longer than average to get into the air. The average delay was about 9 minutes, while the worst actual delay was about 1 hour and 16 minutes.

While this example used airports, it's easy to see how this would be applied to elapsed time for applications in an IT operation use case.

6.2 Real User Monitoring

Figure 6.2-A. Real User Monitoring

The screenshot above shows a real user monitoring scenario focusing on users in North America. The popup on the geographic map shows a full breakdown of the components and elapsed time for the user transaction.

Complete tracking of the end user's experience is provided in real time. Browsers are automatically injected with instrumentation without a need to modify your applications. XRay can find bottlenecks that cause a user to have a negative experience and correlate their problems with issues in the browser itself, applications that the user's session is dependent on such as JVMs or databases. XRay tracks transactions end-to-end starting at the user with a web browser and interacting with application servers, middleware, databases, and local or Cloud mainframes.

6.3 Managed File Transfer (MFT)

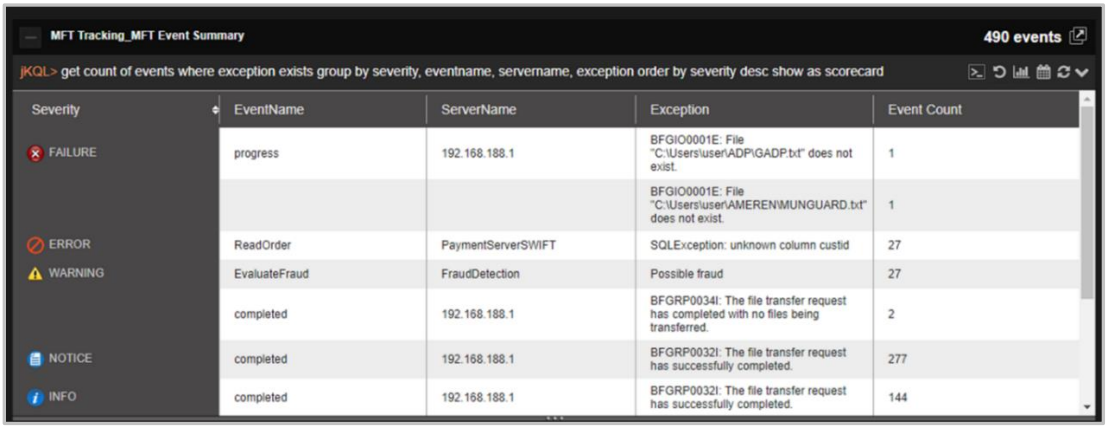


Figure 6.3-A. Managed File Transfers

The XRay dashboard above has been set up to analyze managed file transfers (MFTs). There are various viewlets to track MFTs by application, agent, resource, destination, and status.

XRay tracks all data movement across complex topologies. All MFT transfers relate to downstream events from sources including other MFTs, middleware, brokers, and other business applications. Metrics on MFTs are captured in real-time and evaluated in terms of SLAs and business objectives. Appropriate notifications are sent out for missed objectives. A search capability is provided to review past transfers and their attributes. XRay provides a dynamic topology of all MFT transactions.

6.4 Application Performance Monitoring

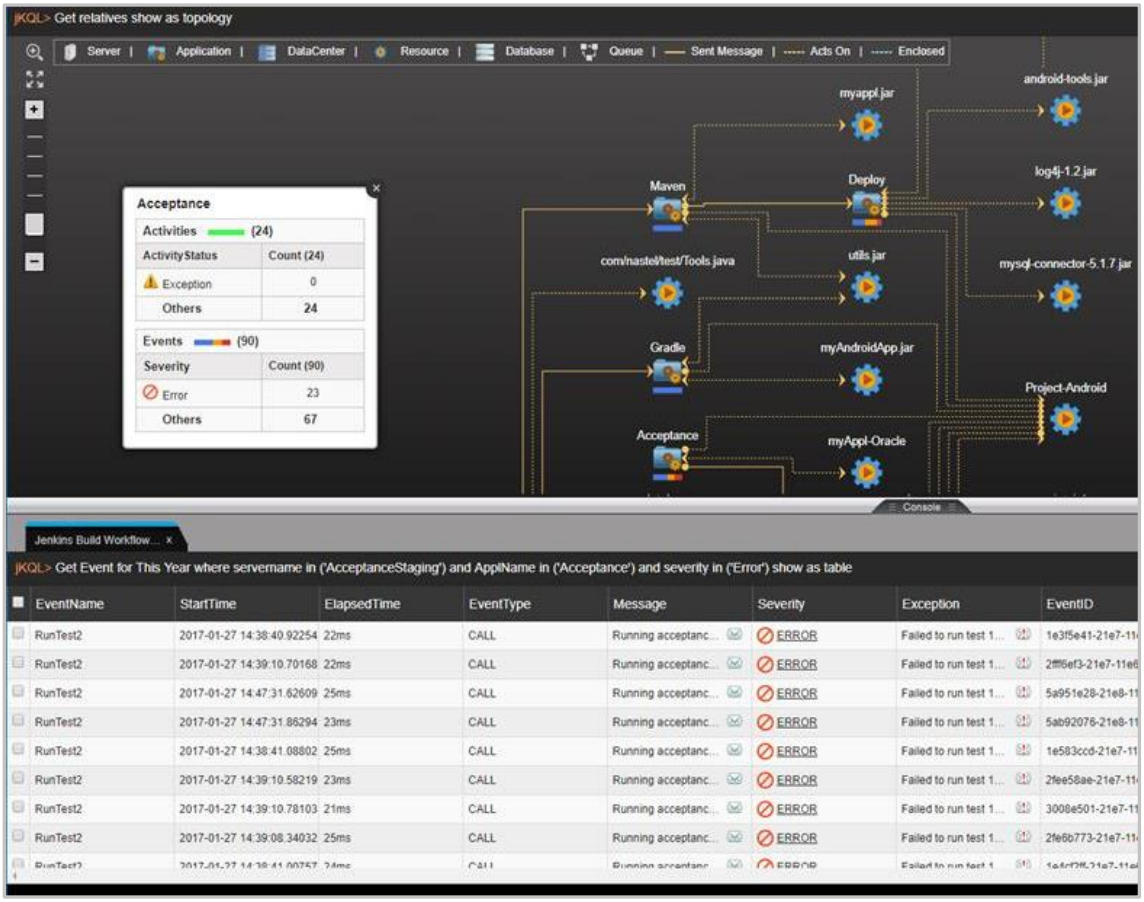


Figure 6.4-A. Application Performance Monitoring

The XRay dashboard example above for application performance monitoring (APM) is illustrating how to monitor the DevOps Jenkins based continuous build-deploy process. The top viewlet is an automatically discovered topology map showing applications and their relationships to other applications such as "Maven" to "Deploy" as well as resources such an Oracle database and a log4j jar file. It shows the flow of a deployment process and any exceptions incurred. The bottom viewlet is called the Console and it opens when a user drills down into an object on a viewlet to get additional details.

XRay provides deep-dive monitoring of the performance and availability of applications end-to-end across Web Services, application servers (Java, .Net), middleware, mainframes, and more. Its automation eliminates the need for constant "eyes-on-screen" monitoring to eradicate false alarms and provide automated notification of real situations that require attention.

6.5 Mobile Analytics

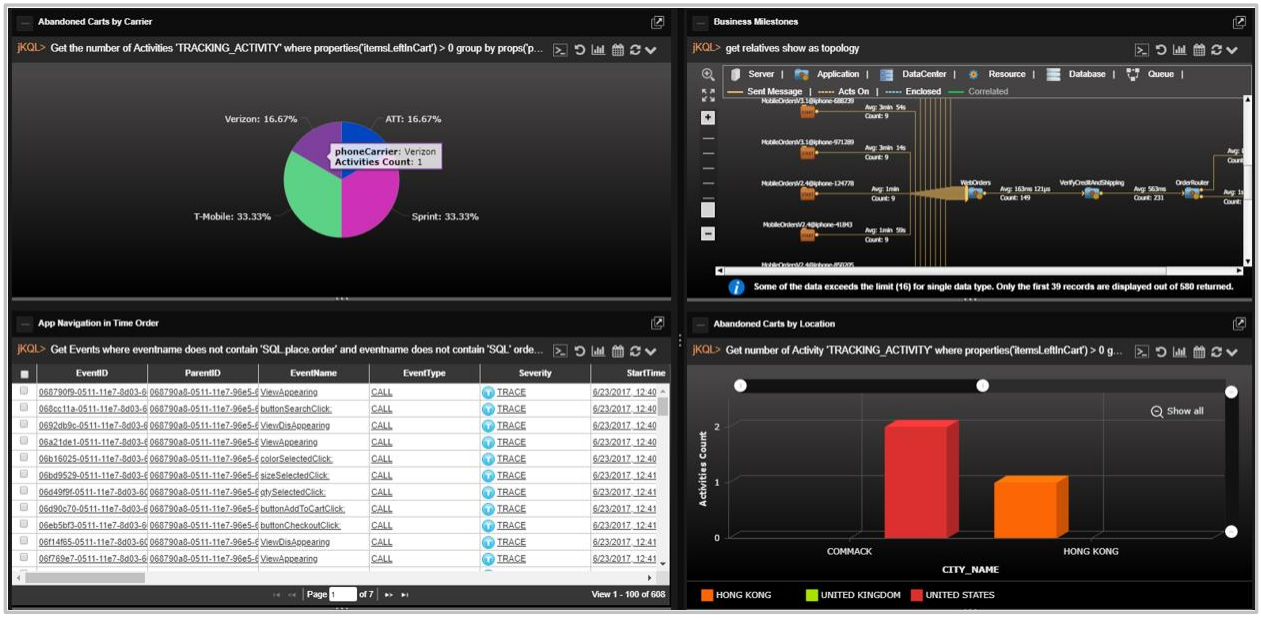


Figure 6.5-A. Mobile Analytics

The sample mobile analytics dashboard above highlights a scenario where performance is compared to mobile app version, carrier, and device. Using our mobile APIs we can track user experience through every mobile app screen, analyze user experience and determine which app versions, devices and carriers deliver the best experience.

XRay provides end-to-end visibility into mobile application behavior and performance for both iOS and Android. RESTful APIs for streaming data and real-time tracking are provided. Mobile apps can stream their data to XRay, submit interactive queries, and subscribe to real-time analytics. Crashes can be captured and analyzed for forensic purposes. The APIs enable complete analysis of a user's interaction with your applications, relating the specific click path through an application correlated with app version, device information, and even business behavior such as purchasing or cart abandonment.

6.6 Kafka Monitoring

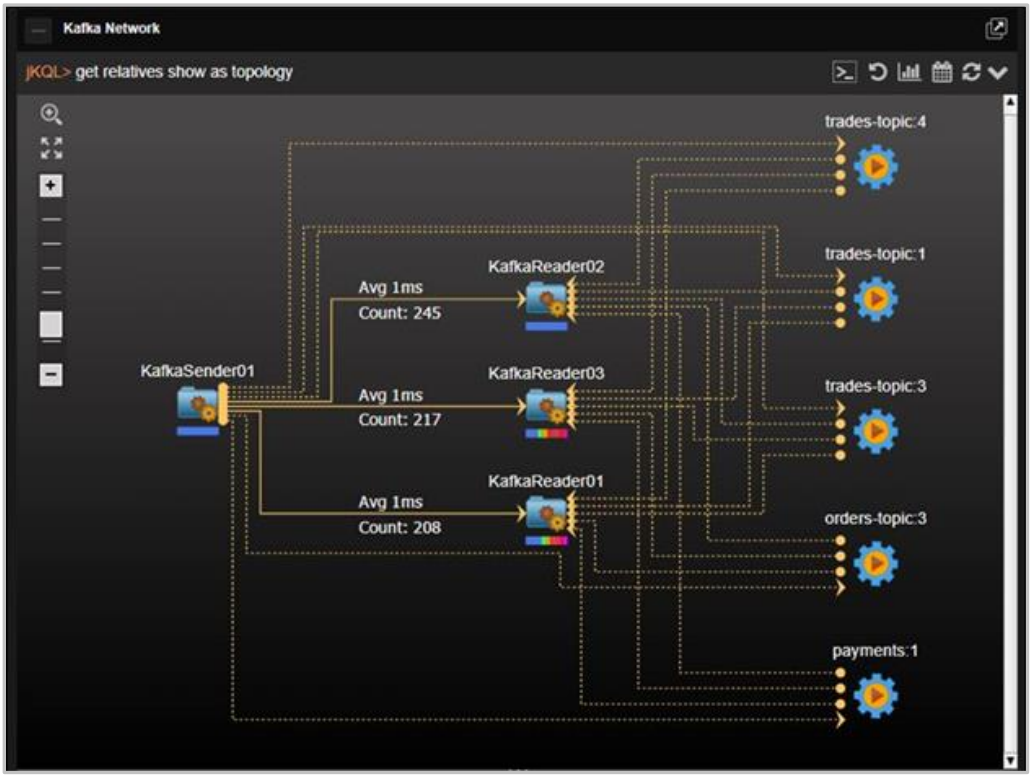


Figure 6.6-A. Kafka Monitoring

The viewlet above shows the auto-discovered, publish-subscribe topology of a Kafka network including senders, readers, and topics. Each edge (the lines between nodes) has statistics showing average elapsed time and count. This image shows the topology of a Kafka sender publishing messages with topics and several Kafka readers subscribed to specific topics.

A single point of truth is provided to track performance, latency, logs, auditing, and content surveillance. XRay provides complete message flow analytics relating applications to the messages they publish to Kafka and the applications that subscribe to them.

Chapter 7: Troubleshooting

The following are examples of jKQL query errors and suggestions on how to resolve them.

Case 1: The message, *No record found*, is displayed in the viewlet.

→ Try to modify the viewlet's date and time range. Confirm that your repository has imported data.

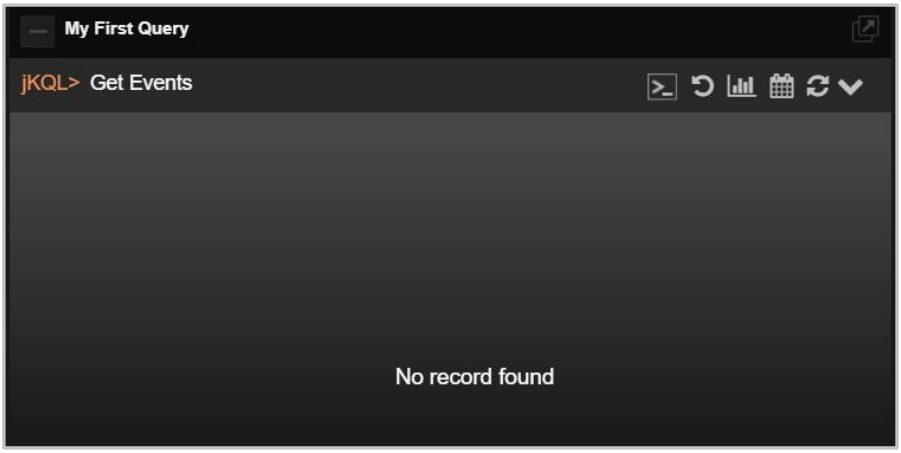


Figure 7-A. No Record Found Message

→ Try to modify the query conditions if the date and time range do not work.

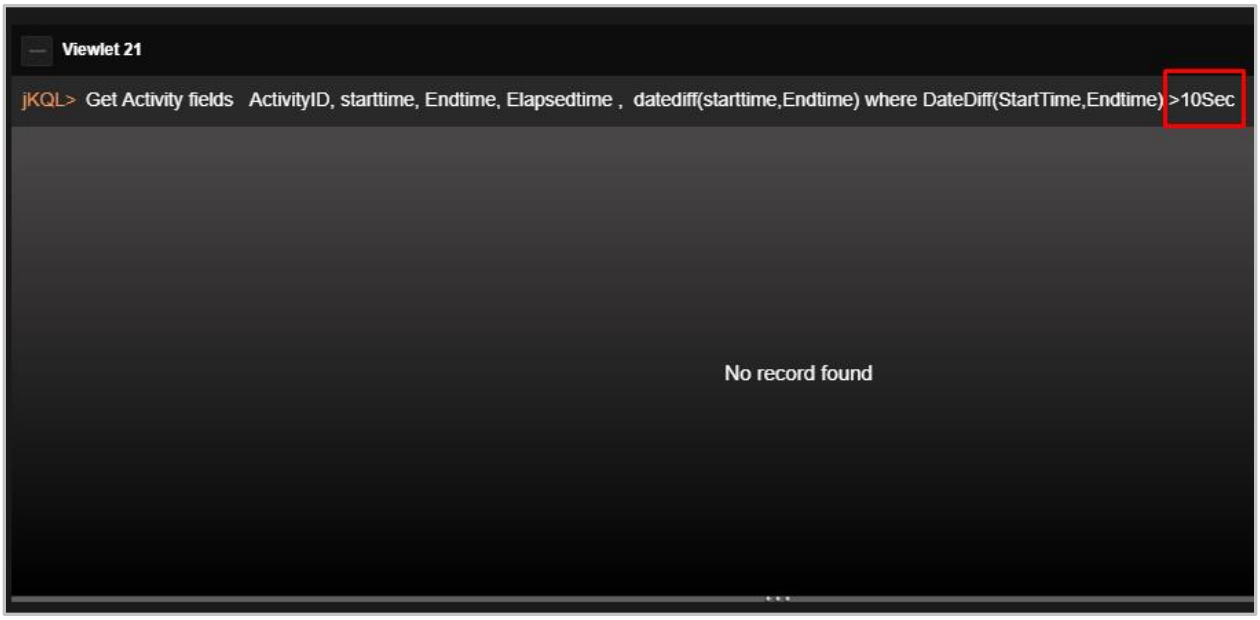


Figure 7-B. No Record Found – Modify Query

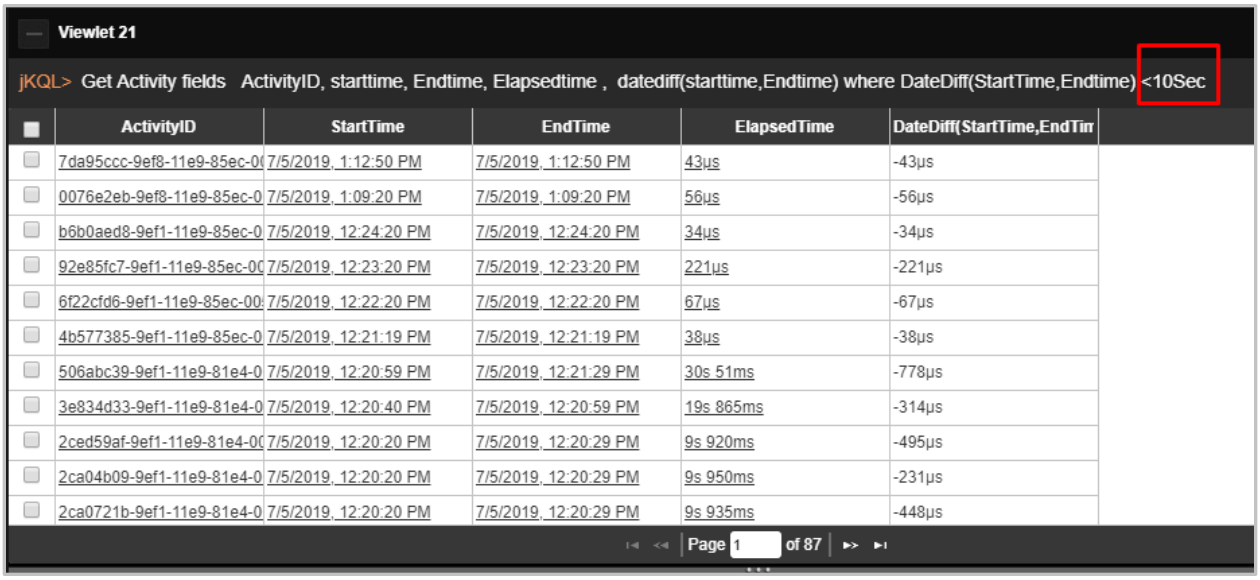


Figure 7-C. No Record Found – Modify Query

Case 2: The viewlet message displays the requirements of the query.

→ Update your query according to the viewlet's message.

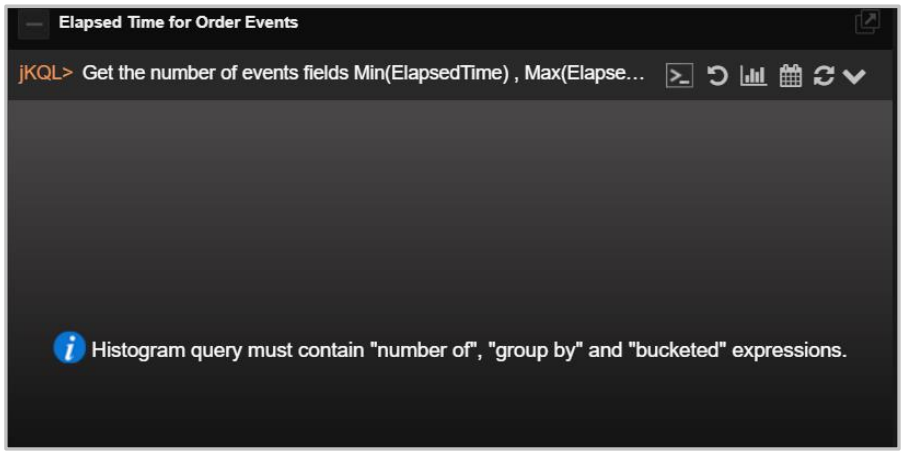


Figure 7-D. Query Requirements Message

Case 3: The viewlet message states that the query needs to be modified.

→ Modify the query's expressions. Confirm that the appropriate data is supplied for the chart axes.

A notification similar to the example below is displayed when a Y axis has incorrect data defined, for example, "String." Another example would be when a histogram's X axis has "Timestamp" defined.

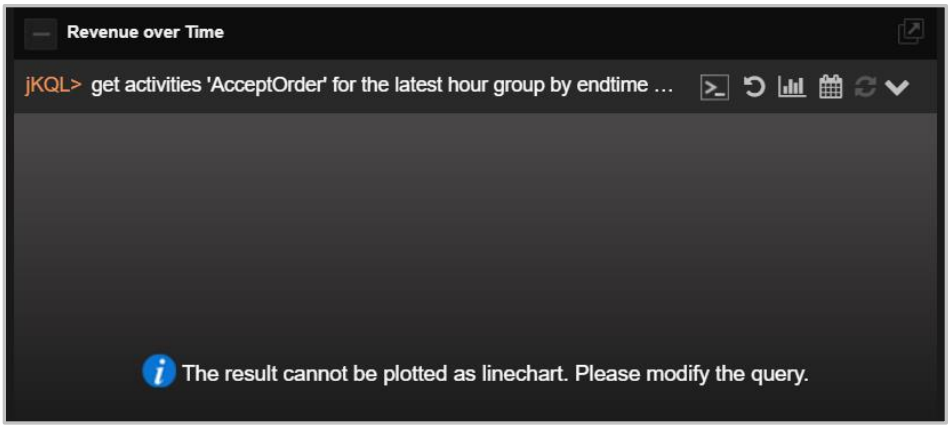


Figure 7-E. Query Requirements Message

Case 4: System displays an error message.

→ Modify the query using the information provided in the error message.

Using the example below, many times the solution is to increase "Bucketed by size."

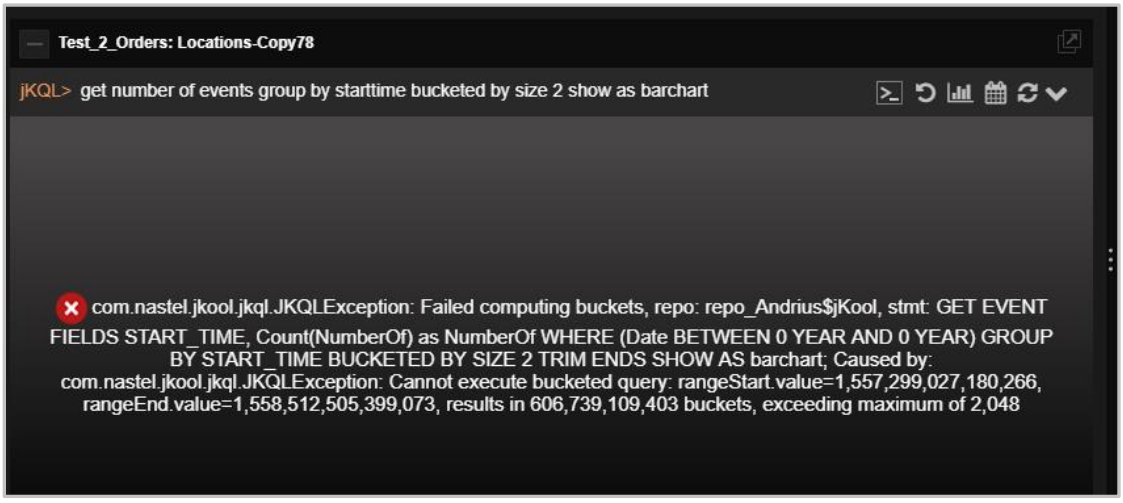


Figure 7-F. Query Error Message

When data of "enum" data type (for example, severity, compcode) are queried to display, the data type values must be used. The severity name can be replaced with the ID from the severity values table, specified by the query.

jKQL> `Get enumeration for severity`

For example: **jKQL>** `Get events where severity is ('3')`

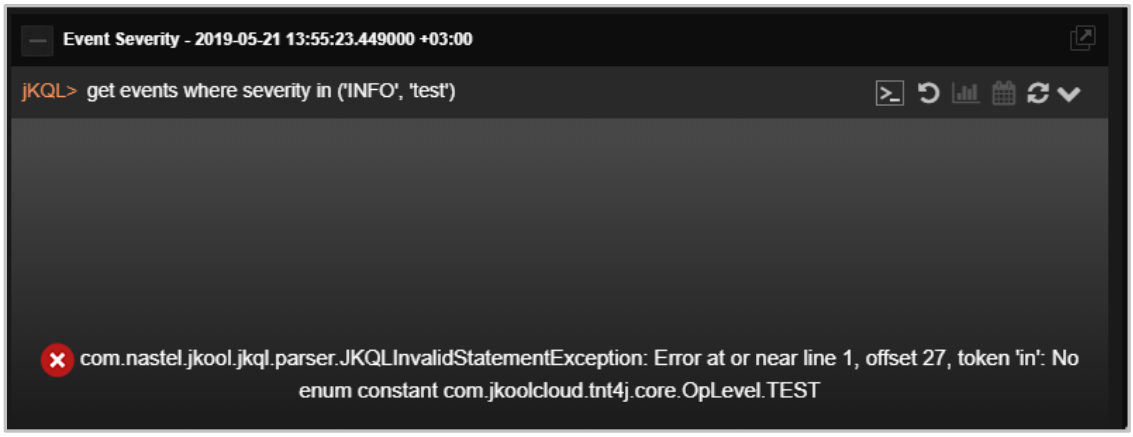


Figure 7-G. Query Error Message

"String" functions such as "Starts With," "Ends With," "Contains," cannot be defined for "Enum" data types.

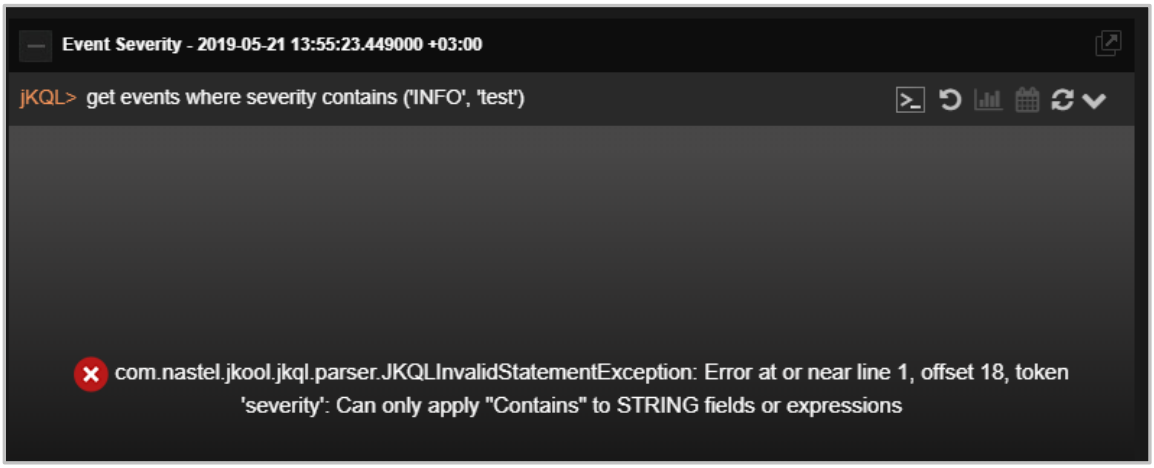


Figure 7-H. Query Error Message

Case 5: System displays invalid field.

→ Modify the query by using the function prop before the field.

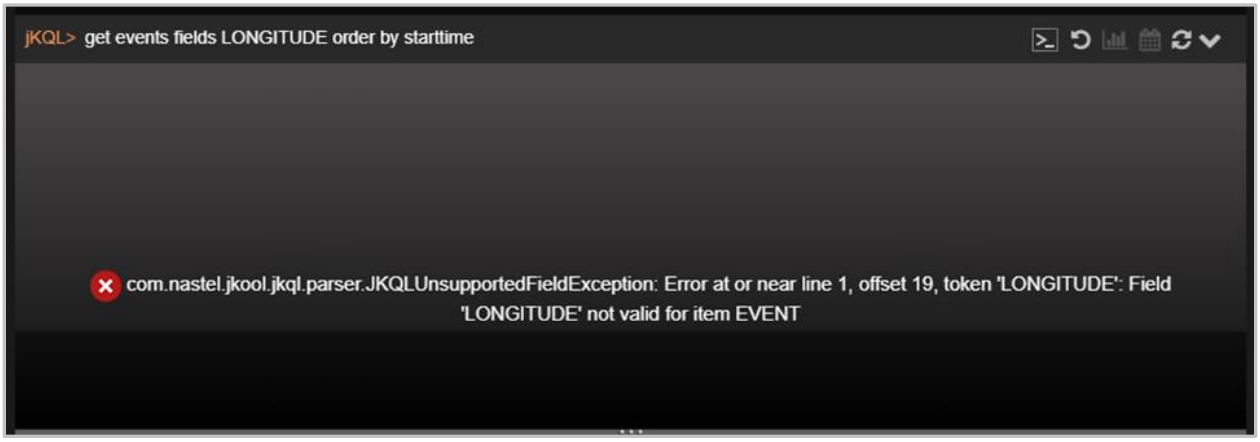


Figure 7-I. Invalid Field

Some fields are property fields which must be used with the function "prop."

jKQL> `get events fields prop('LONGITUDE') order by start time`

Figure 7-J. Prop Function

Case 6: Access required error

An error similar to the one below is displayed when trying to access sets, but this feature is disabled.

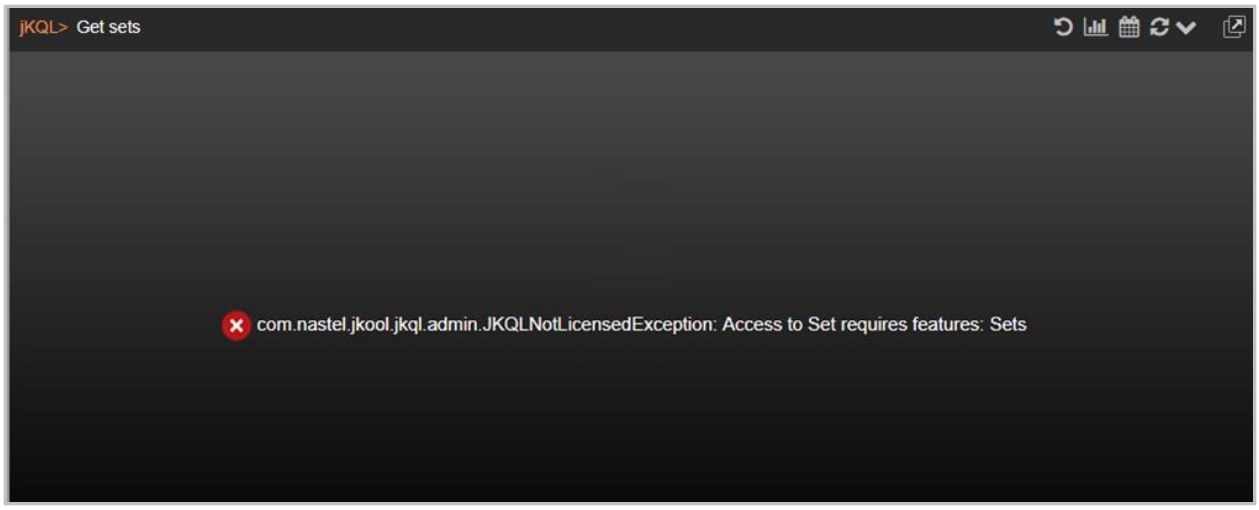


Figure 7-K. Access Required Error

→ Enable sets from the **Features** tab of the *Admin Settings* window.

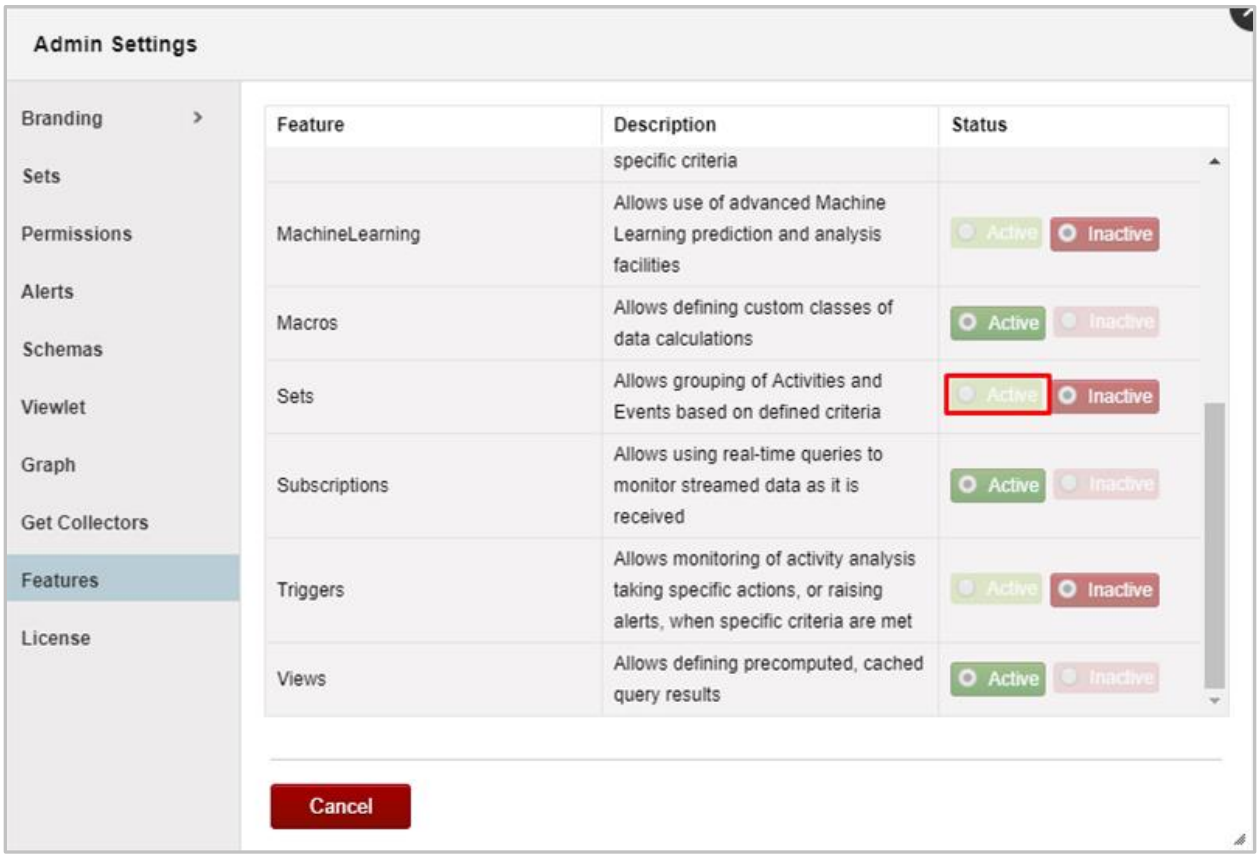


Figure 7-L. Enable Sets

Case 7: Syntax error.

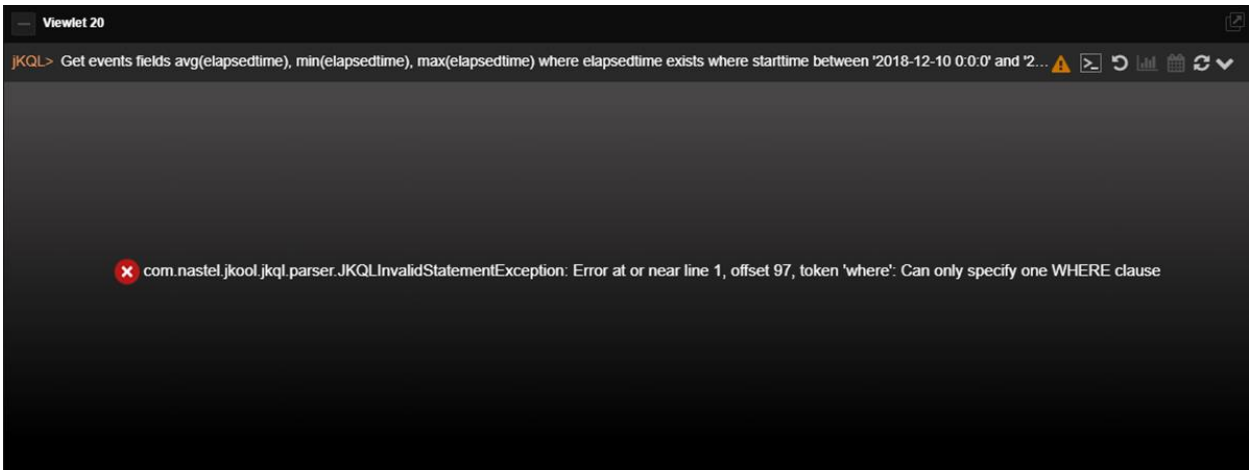


Figure 7-M. Syntax Error

As the error above mentions, only one where clause should be mentioned as seen in the example below.

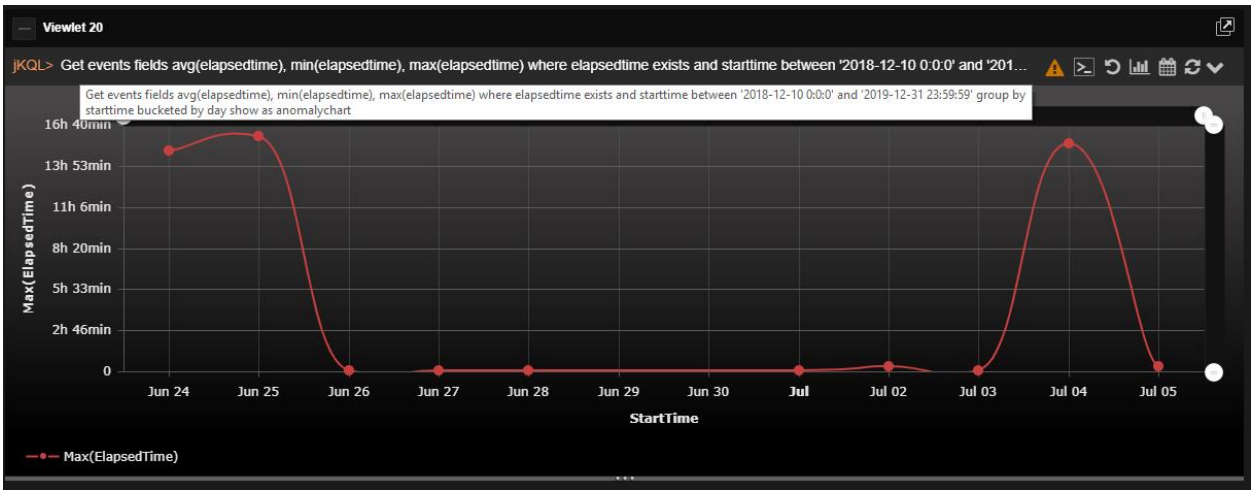


Figure 7-N. One Where Clause

Syntax error query > Get events fields avg(elapsedtime), min(elapsedtime), max(elapsedtime) where elapsedtime exists where starttime between '2018-12-10 0:0:0' and '2019-12-31 23:59:59' group by starttime bucketed by day show as anomalychart

Correct syntax > Get events fields avg(elapsedtime), min(elapsedtime), max(elapsedtime) where elapsedtime exists and starttime between '2018-12-10 0:0:0' and '2019-12-31 23:59:59' group by starttime bucketed by day show as anomalychart

Case 7.1: When passing a field name to a function, do not use the **'** symbol.

Figure 7-O. Syntax Error Due to ' Symbol

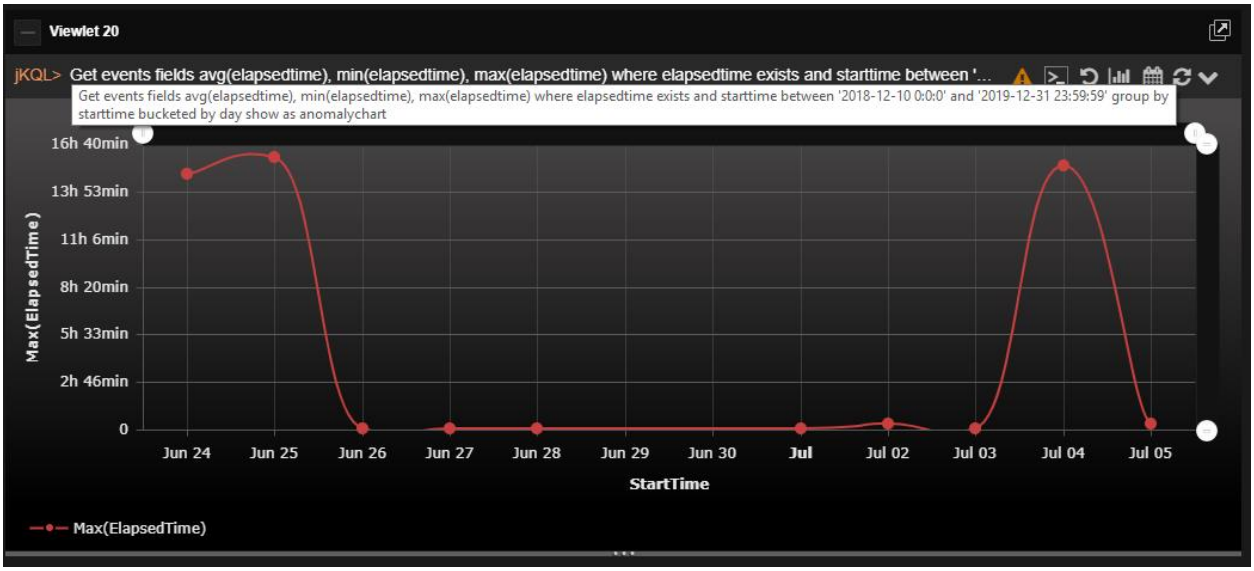


Figure 7-P. Passing Field Name

For example:

jKQL> `Get events fields avg(elapsedtime), min(elapsedtime), max(elapsedtime) where elapsedtime exists and starttime between '2018-12-10 0:0:0' and '2019-12-31 23:59:59' group by starttime bucketed by day show as anomalychart`

Case 8: Request time out is displayed.

→ Try to refresh the viewlet by clicking on the **Refresh** button located at the top right corner of the viewlet.

Sometimes the response can take longer than expected due to various reasons, for example, a network issue.

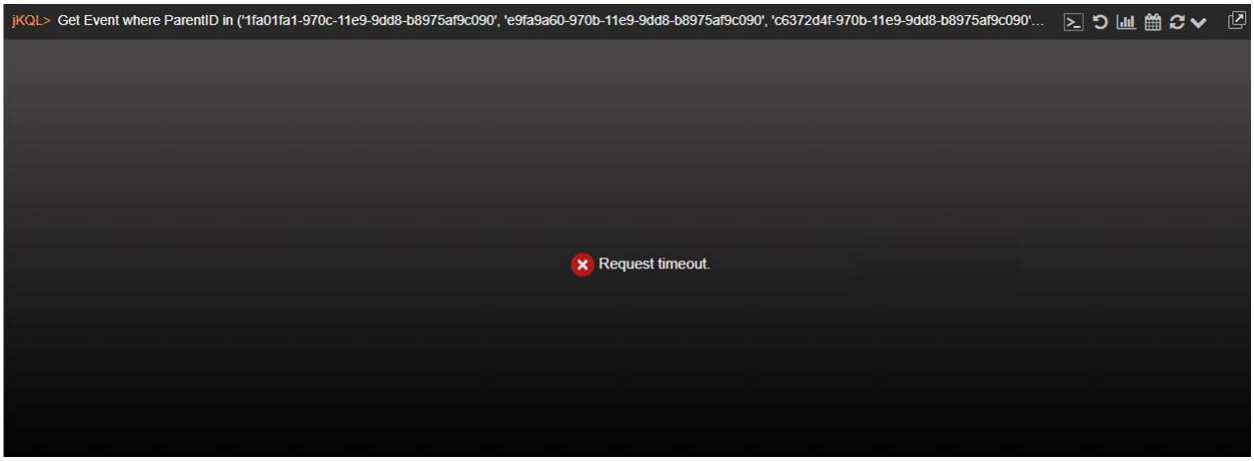


Figure 7-Q. Request Timeout

Case 9: "Show as" type is invalid.

→ An incorrect viewlet chart type was used. Specify a valid chart type to display the data (see the subsections of 2.5.4 for available chart types to use).

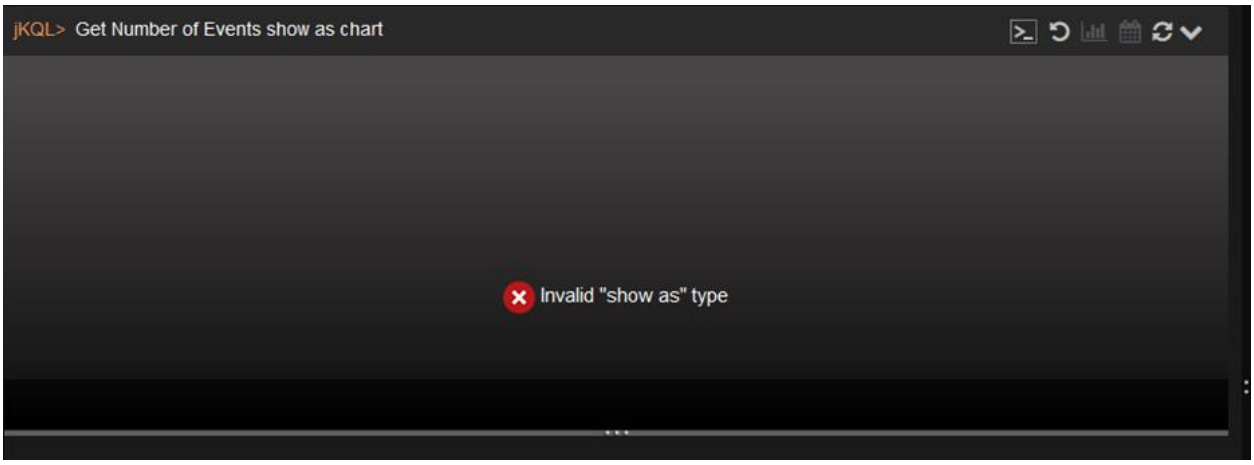


Figure 7-R. Invalid Type

Case 10: Error message when using the Relative function.

→ The Relative function currently only works for activities.

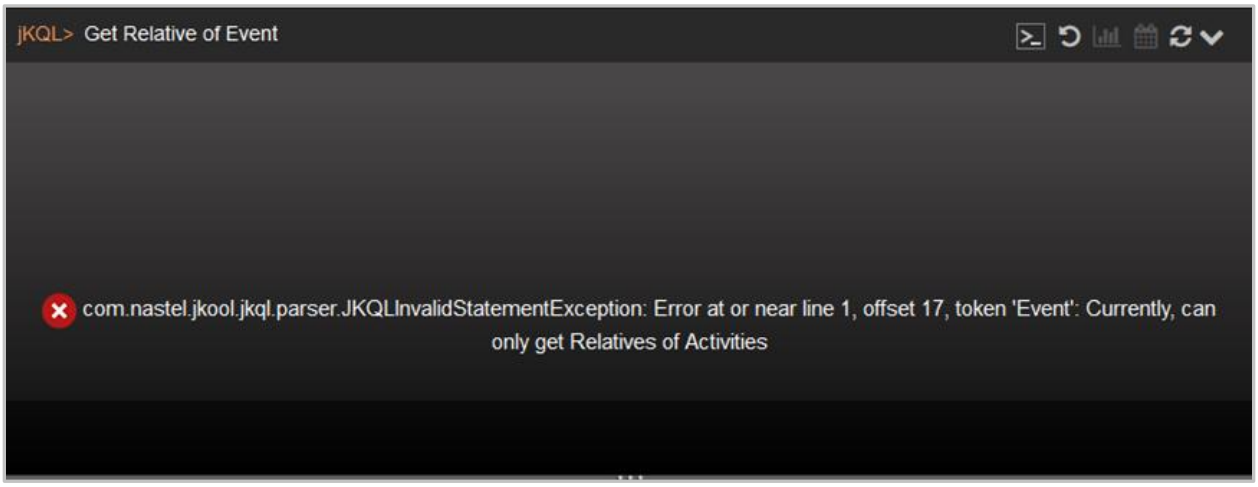


Figure 7-S. Relatives Error

Index