SAS® Event Stream Processing 4.2: Visualizing Event Streams with Streamviewer

Overview

Streamviewer provides a user interface that enables you to subscribe to window event streams from one or more event stream processing engines, and display the events that stream through it. You can display each event as a row in a table or as an element of a graph. Each table row is keyed by the schema key of the corresponding window. You can save and load a collection of tables, graphs, and their customized settings in *dashboards*.

Streamviewer dashboards are stored in a configuration database. For the configuration database storage mechanism, you can use one of several supported enterprise databases or the stand-alone, file-based RDBMS SQLite database that is provided with SAS Event Stream Processing. After you have configured your storage mechanism, you can run the Streamviewer user interface from either a local or from a remote SAS Event Stream Processing installation.

It is recommended that you run Streamviewer from a remote installation using SQLite as the storage mechanism. This helps you avoid browser limitations on local file access. It also makes it easier for more than one user to access Streamviewer at a time.

Note: The following instructions assume that you have already set the environment variables `DFESP_HOME` and `LD_LIBRARY_PATH`.

Setting Up and Running Streamviewer

Setting Up the Configuration Database

Streamviewer requires a relational database to persistently store configuration information. You can use SQLite, which is provided with the product. Alternatively, you can use an enterprise database management system.

Setting Up SQLite as Storage Mechanism

SAS Event Stream Processing includes an empty SQLite database and a script named **sqlite_run.sh** to run a SQLite-based event stream processing configuration server.

To run the Streamviewer configuration server using SQLite as its persistent storage mechanism, do the following:

1. Create a directory for your configuration server: **SVCONF_DIR** .
2. Change your current directory to **SVCONF_DIR** .
3. Copy the database configuration files from the SAS Event Stream Processing installation's **streamviewer** directory to **SVCONF_DIR**:

   ```
   $ cp $DFESP_HOME/share/tools/streamviewer/db/sqlite_run.sh .
   $ chmod +x ./sqlite_run.sh
   $ cp $DFESP_HOME/share/tools/streamviewer/db/streamviewer.db .
   ```

4. Run the SQLite script.

   ```
   $ ./sqlite_run.sh -p <http_sql_port>
    -d <database_file>
   ```

   Note: Replace *<http_sql_port>* with an available port of your choice.

   Note: The `-d <database_file>` option defaults the *database_file* to **streamviewer.db**. You can use this option to run the configuration server using a database file other than **streamviewer.db**. This enables you to keep any number of Streamviewer databases and use different database files to organize your dashboard collections.

For more information about SQLite, visit https://www.sqlite.org/ .

Setting Up Enterprise Database Management Systems as Storage Mechanisms

The following enterprise databases are supported for Streamviewer:

- MySQL
- Oracle Database
- PostreSQL
- Microsoft SQL Server
- Sybase (SAP)
- IBM DB2

To set up Streamviewer to use with these systems, you must create configuration tables in your selected database. The SAS Event Stream Processing installation includes an SQL script that creates the Streamviewer configuration tables: **$DFESP_HOME/share/tools/streamviewer/db/createsv.sql** .

Note: Consider the **createsv.sql** script a template. The script might generate warnings or have issues with different databases. Refer to your database documentation and make changes to the **createsv.sql** script accordingly.

Create an **odbc.ini** file with the appropriate settings for your database. You can find a template named **odbc.ini.template** in the **$DFESP_HOME/etc** .

The template file includes sample settings for each enterprise RDBMS. The template table for the RDBMS in use can be copied to the **odbc.ini** file. Change the necessary drivers, host names, port numbers, and user credentials according to the database that you want to use.

To run the Streamviewer configuration server using an enterprise database as the persistent storage mechanism:

1. Create a directory where you run the configuration server **SVCONF_DIR** .
2. Place the new **odbc.ini** file (created from the template file `odbc.ini.template`) in **SVCONF_DIR** .
3. Change your current directory to **SVCONF_DIR** .
4. Set the ODBCINI environment variable:

   ```
   $ export ODBCINI=odbc.ini
   ```

5. Start the Streamviewer configuration server:

   ```
   $ $DFESP_HOME/bin/dfesp_xml_server -http-sql <http_sql_port>
   ```

Testing the Server-Database Connection

Test your server and database connectivity:

```
$ $DFESP_HOME/bin/dfesp_xml_client -url "http://[host:port]/tables/streamviewer"
```

Where *host* is the XML server location and *port* is the SQL port specified in your server settings.

The output from running **dfesp_xml_client** should look something like this:

```
<tables>
    <table name='streamviewer_chart'/>
    <table name='streamviewer_dashboard'/>
    <table name='streamviewer_server'/>
</tables>
```

Installing and Running Streamviewer

You can run Streamviewer on a local computer system or inside an Apache Tomcat web server.

Installing Streamviewer on a Local Computer System

To install Streamviewer on a laptop or desktop computer system, compress the **streamviewer** directory in the SAS Event Stream Processing installation files. Move the compressed **streamviewer** directory to a local directory and uncompress it at that location. Name this local Streamviewer location ***SV_INSTALL*** .

To do this in a Linux installation, run these commands:

```
$ cd $DFESP_HOME/share/tools
```

4

```
$ zip -r streamviewer.zip streamviewer
```

To do this in a Windows installation:

1. Open Windows Explorer.
2. Navigate to **$DFESP_HOME/share/tools**.
3. Right-click on **streamviewer**.
4. Select **Send to compressed folder**.

These steps create a ZIP file called **streamviewer.zip**. The ZIP file contains the Streamviewer application. If the SAS Event Stream Processing installation is on a server, use SCP, FTP, or some other file transfer utility to get the ZIP file to your local system.

After you have **streamviewer.zip** on your local system, move it to***SV_INSTALL***. Uncompress the ZIP file into this local directory.

Running Streamviewer on a Local Computer System with Firefox

Run Streamviewer in Firefox using a `file:` protocol:

1. Open Firefox.
2. In the address bar, enter **about:config**.
3. In the **Search** field, enter **security.fileuri**.
4. You should see a setting with a **Preference Name** of **security.fileuri.strict_origin_policy**. Change the **Value** of this row to **false** .
5. Run a Finder or Explorer window. Double-click on **index.html** in the ***SV_INSTALL*/streamviewer** directory, or enter **file://*SV_INSTALL*/streamviewer** in the address bar.

   Note: Remember to replace ***SV_INSTALL*** with the full local directory path containing your **streamviewer** files.

Note: You can choose not to change this setting for security purposes and run Streamviewer locally through Firefox with a Python HTTP server. See Running Streamviewer on a Local Computer System with Chrome on page 4 for instructions on setting up a simple HTTP server with Python.

Running Streamviewer on a Local Computer System with Chrome

You can run Streamviewer in Chrome on a simple Python HTTP server. If you are running Windows, Python might need to be installed. If you are running Mac OS X, Python should already be available on your system.

1. Ensure that Python is installed.
2. In a terminal window (Mac OS X) or command prompt (Windows), run the following:

   ```
   $ cd $SV_INSTALL/streamviewer
   ```

   ```
   $ python -m SimpleHTTPServer <http_port>
   ```

   Note: Replace *http_port* with an available port of your choice.
3. Open Chrome.
4. In the address bar, navigate to http://localhost:*http_port*.

Running Streamviewer on a Web Server

To run Streamviewer with SAS Event Stream Processing installed in a remote location, use a web server. If a web server is available, you can deploy Streamviewer to that server for use. We use Apache Tomcat, an open-source web server application, to configure and use Streamviewer as a web application.

Installing and Configuring Apache Tomcat for Streamviewer Use

Note: If you have set up a web server, see Installing Streamviewer in an Apache Tomcat Instance on page 5 .

To install Apache Tomcat for Streamviewer use, download Apache Tomcat. You can find Tomcat binary distributions at http://tomcat.apache.org/ . Follow the instructions provided at that website to install Apache Tomcat.

Note: Obtain an updated version of Java SE Development Kit. You can find this at http://www.oracle.com/technetwork/java/javase/downloads. The Apache Tomcat web page http://tomcat.apache.org/whichversion.html provides the requirements for each version of Tomcat.

Installing Streamviewer in an Apache Tomcat Instance

To install Streamviewer under an Apache Tomcat instance, copy the **`streamviewer`** directory from **`$DFESP_HOME/share/tools`** into the Apache Tomcat **`webapps`** directory. Rename the Apache Tomcat installation root to **`TOMCAT_HOME`**.

On Linux, do the following:

```
$ export TOMCAT_HOME=/path/apache-tomcat-X.x.x
```

```
$ cp -r $DFESP_HOME/share/tools/streamviewer $TOMCAT_HOME/webapps/
```

On Windows:

1. Open Windows Explorer.
2. Navigate to **`$DFESP_HOME/share/tools`**.
3. Right-click **`streamviewer`**, select **Copy**.
4. Navigate to **`TOMCAT_HOME/webapps`**.
5. Right-click and select **Paste**.

Note: If your Apache Tomcat instance is on a machine other than the one where SAS Event Stream Processing is installed, compress the **`streamviewer`** directory for easier transfer between machines.

If your Apache Tomcat is running on host *tomcat_host* and port *tomcat_port*, open a web browser and navigate to http://*tomcat_host*:*tomcat_port*/streamviewer.

Connecting to the Configuration Server

1. After Streamviewer is running, connect to the event stream processing configuration server that you have started. In the ESP Model Viewer window, enter your server information in the **Configuration** field. This is the information provided by your server location and the SQL port that you specified earlier.

Note: Specify the URI that reflects the protocol that your configuration server is using. For example, when you have a remote UNIX server *hostname* with an SQL port *port*, enter http://*hostname*:*port* or https://*hostname*:*port*.

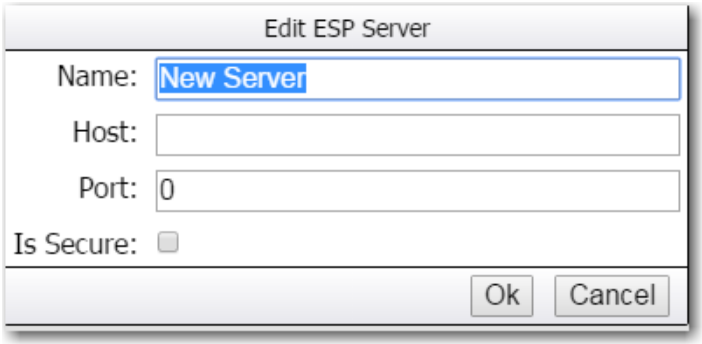
2 Click **Load**. **My Dashboards** appears in the **Datasource** field. You should see any data sources that you have defined in your odbc.ini file used by the configuration server.

Connecting to Event Stream Processing Servers in Streamviewer

Streamviewer enables you to view event streaming models from more than one event stream processing server at a time. After you have specified the configuration server information and have loaded or created a dashboard, you can select running servers to publish models and events into your dashboard.

To connect to a server in Streamviewer,

1 Click the **Manage Servers** button to select an HTTP Publish/Subscribe server. You can click **New** to enter your ESP server location for **Host** and your HTTP Publish/Subscribe port for **Port**.



2 All the server locations that you have created appear in the ESP Servers window. Select a server from the list and click **Done**. You now can build and save dashboards in Streamviewer using the servers that you selected in the ESP Servers window, and the configuration server that you specified in the ESP Model Viewer window.

Authenticating Event Stream Processing Servers in Streamviewer

If you have set up authentication for an event stream processing server, you must provide authentication tokens or credentials in Streamviewer through a dialog box. You can either copy and paste the token directly into the **OAuth Token** text area, or you can specify a URL that supplies the token. Authentication tokens and credentials are cached as you use Streamviewer.

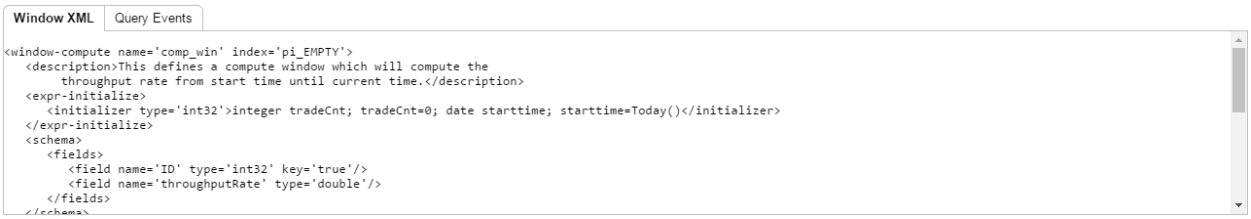
For more information about setting up authentication, see “Enabling Authentication on Socket Connections” in *SAS Event Stream Processing: Security*.

Using Streamviewer

Using ESP Model Viewer

After connecting to a configuration server, choosing a dashboard data source, and connecting to a running server, use the ESP Model Viewer to arrange and display a project's windows and their connectors.

1. Select a **Project** and **Contquery**. You can change the orientation of a model and the type of arrows between connected windows with the **Link Type** and **Orientation** options.
2. Click an **ESP Window** to display details about each window. These details include a window's incoming and outgoing data sources and its schema.
3. To see the XML code for a window or to query the events that are streaming through a window, click a window and select **Show Window Info**.



4. To place windows in a dashboard, click a window in a model and subscribe to it one of three ways:

   Updating
   The opcode of the event is used to either add, modify, or delete the row specified by the event key in the table. The subscription is populated with a current snapshot of the window contents.

   Streaming
   The event is appended directly to the table, and the opcode is displayed with the event. Only events occurring after the creation of the subscription are displayed.

   Streaming with snapshot
   The same as Streaming mode except that the subscription is populated with a current snapshot of the window contents.

5. Click **Done**. The Streamviewer dashboard appears.

Using the Dashboard Interface

After you subscribe to windows in the ESP Model Viewer, tables associated with those windows appear in the dashboard. You can do the following:

- edit settings to determine how events are read into your subscribed windows
- publish data directly into your subscribed windows
- create new tables or graphs for subscribed windows

You can save any changes you make in a dashboard. Use the options at the top of the subscribed window tables and the options that appear at the top of the dashboard interface.

Note: The number of subscribed windows affects Streamviewer performance.

With the **Edit Dashboard** setting, you can set the polling interval to determine how often you want to fetch events to refresh the data. You can also specify the maximum number of events to permit in a given window. The default maximum is 25,000. Use these two settings to regulate processing and to tune stream viewing performance.

Creating Tables, Graphs, and Charts in Streamviewer

After you have created subscriptions from the ESP Model Viewer, tables of the subscribed windows appear in the Streamviewer dashboard. After a table is displayed, you can change its display type or create other visual components such as graphs or charts based on the same subscription. You can organize data in charts by key fields or classes.

Several icons become visible when you hold your pointer over a visual component. Click the new chart icon to create a new table, graph, or chart based on a subscribed window. As data streams through the model, your graphs and charts update accordingly. You can pause data streaming in any element in your dashboard. When you apply a filter to one element, it affects all related elements.

After a visual component is displayed, you can move it by dragging it to the desired position. You can re-size the visual component by grabbing the lower right corner and dragging it. Anytime that you click on a visual component, it is brought in front of the other components on the screen. When you want to remove a visual, click the delete icon in the upper right corner of its window.

Every *n* seconds (as determined by the polling parameter), Streamviewer gathers events from the server that have occurred since the last check. These events are collected and handled by their corresponding visual components. For an *update*, the opcode is checked. When the opcode is *delete*, the event with the event key is deleted. Otherwise, the event is either inserted or updated. Anytime that a polling request is sent and received, any rows in a table that were inserted or updated are highlighted or displayed in bold text. This enables you to see what has changed during the polling interval.

For a *streaming* visual component, any events that arrive during the polling interval are added to the end of the component. After an update, if the number of events in a streaming table exceeds the setting for **Streaming Table Rows**, then events are removed from the beginning of the table. Also, all *streaming* tables scroll to the bottom of the table to show the most recent events.

You can pause and restart *streaming* and *update* windows at any time. Pausing a table unsubscribes the window on the server, conserving resources. When all windows are paused, polling the server for updates ceases.

Window types handle a play or restart differently. The update clears the table contents and re-populates the table with a fresh snapshot of the window from the server. Subsequent updates are applied to that state. The streaming tables start appending the events to the current contents of the table.

All visual components that use the same window and mode (*updating* or *streaming*) are synchronized with one another. When you sort a table, any associated graphs sort themselves accordingly. When you pause or restart a visual component, all associated items are paused or restarted as well.

Embedding Charts

You can embed any chart into a web page.

Note: You only can embed charts that are currently saved within a Streamviewer dashboard. When you delete a chart, it is no longer embeddable.

1. Change directories to the Streamviewer installation directory.
2. Copy the `js` directory from there to the directory where your web page resides, here represented by the environment variable `$PAGEROOT`:

   ```
   $ cp -r js $PAGEROOT
   ```

3. Copy the Streamviewer CSS file to `$PAGEROOT/style`

$ mkdir $PAGEROOT/style (if it does not exist)

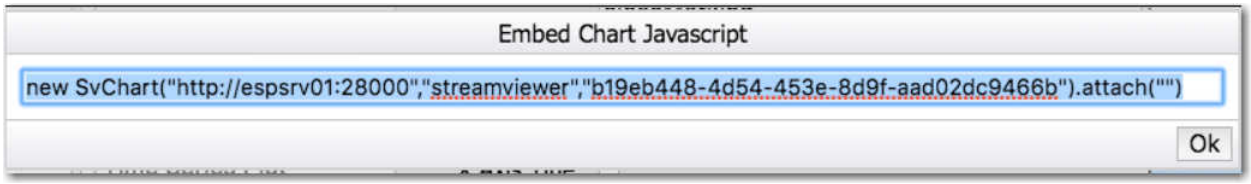
$ cp -r style/streamviewer.css $PAGEROOT/style

4 Add the following lines to your web page:

```
<script src="js/libs/common/collections.js"></script>
<script src="js/libs/common/ajax.js"></script>
<script src="js/libs/common/xpath.js"></script>
<script src="js/libs/common/server.js"></script>
<script src="js/libs/common/table.js"></script>
<script src="js/libs/common/sql.js"></script>
<script src="js/libs/common/svchart.js"></script>
<script data-main="js/app" src="js/libs/require.js"></script>
<link rel="stylesheet" href="style/streamviewer.css" type="text/css" />
<script type="text/javascript">

var   charts = null;
var   sasgraphdata = null;

function
init(a,b)
{
   charts = a;
   sasgraphdata = b;

   /*
        specify value for the appearance of your embedded charts
        valid values are pressed, sheen, crisp, gloss, matte, flat, and none
   */

   //serverSupport._graphSkin = "sheen";

   /*
        specify value for the appearance of your embedded gauges (KPIs)
        valid values are basic, charcoal, modern, onyx, and satin
   */

   //serverSupport._kpiSkin = "modern";

   /* add your embedded chart code here */
   //new SvChart("http://myserver:1234","streamviewer",
                "47dadf39-9c0b-4a5b-a00f-17a54162c7f6").attach("mydiv");
}

</script>
```

5 For any Streamviewer graphic that you want to include in your web page, select the **Javascript** button in the Modify Chart screen.

Copy the highlighted Javascript.



Put the highlighted text in your page. Supply a valid value for your page as the argument to `attach`.

```
new SvChart("http://
espsrv01:28000","streamviewer","436ad966-0345-41ea-8484-2e9e82abf9bf").attach("t
otalViolations");
```

Publishing to Subscribed Windows

You can inject data directly into a source window by specifying a URL or by specifying events.

When you are running an event stream processing HTTP Admin Server, you can specify that URL in the **Server** field. Using an event stream processing HTTP Admin Server might improve event throughput performance.