SAS® Event Stream Processing 4.3: 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 to 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 and 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 H2 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 a remote SAS Event Stream Processing installation.

Note: The following instructions assume that you have already set the environment variables `DFESP_HOME` and `LD_LIBRARY_PATH`. For more information, see the *SAS Event Stream Processing on Linux: Deployment Guide*.

Setting Up and Running Streamviewer

Planning for Streamviewer

Streamviewer installation steps are described in the installation chapter of the *SAS Event Stream Processing: Deployment Guide*.

Before you start Streamviewer, determine how you want to persistently store the Streamviewer configuration.

Streamviewer is shipped with H2, which is a file-based database engine. Using H2, the Streamviewer configuration is stored in a file that is created when Streamviewer is invoked from the command line. This allows Streamviewer databases to be easily created, copied, and shared. For more information about H2, see http://www.h2database.com/.

Streamviewer also supports the following remote database management systems:

- PostgreSQL
- MariaDB
- MySQL
- SQLServer
- Sybase
- Oracle
- Teradata

If you choose not to use the H2 database, the remote database that you intend to use must already be set up and running before you start Streamviewer. The database version that you use is not significant. Streamviewer supports any version of these databases.

Streamviewer is packaged in the streamviewer-4.3.jar file, which is located in the **/lib** directory of your ESP installation (**$DFESP_HOME/lib**). SAS recommends that you run the JAR file from another location or relocate it to another server. Be sure to include the following scripts that are used to start Streamviewer:

- **$DFESP_HOME/bin/streamviewer.sh**
- **$DFESP_HOME/bin/streamviewer.bat**

Streamviewer requires a Java Virtual Machine (JVM) of 1.7 or later.

Starting Streamviewer

Use one of the following commands to start Streamviewer. You can also start Streamviewer from a Java command line. For more information, see "Starting Streamviewer from the Java Command Line" in SAS Event Stream Processing: Advanced Topics.

Linux

```
streamviewer.sh -product database -http http_port <additional arguments>
```

Windows

```
streamviewer.bat -product database -http http_port <additional arguments>
```

For the `-product` argument, use one of the following values, depending on the database that you want to use for the Streamviewer configuration:

- **h2**
- **postgresql**
- **mariadb**
- **mysql**
- **sqlserver**
- **sybase**
- **oracle**
- **teradata**

You can specify any available port number for the `-http` argument.

Use these additional arguments, depending on the database that you want to use.

Table A.1 *Command Arguments for Streamviewer*

Argument	Description
`-h2file` *database_file*	Specifies the name of a file where you want H2 to store the configuration. The file is created if it does not already exist. This argument is valid only for the H2 database. It is required if you are using H2.
`-host` *host*	Specifies the host name for the remote database. Depending on network access, you might need to specify the fully qualified domain name as well. This is required for all database products except H2.
`-port` *port*	Specifies the port number used by the remote database system. This is required for all database products except H2.
`-db` *db_name*	Specifies the name of the remote database or Oracle service to use. This argument is not used for H2, SQLServer, or Sybase.
`-user` *username*	Specifies the user name required to access the remote database. This is required for all database products except H2.
`-password` *password*	Specifies the password required to access the remote database. This is required for all database products except H2.

Note: Use the following arguments to configure authentication in Streamviewer. Not all installations of SAS Event Stream Processing support these arguments. To verify whether your installation supports authentication in Streamviewer, see *SAS Event Stream Processing: What's New*.

Argument	Description
`-noauth`	Runs Streamviewer without authentication enabled. **Note:** Streamviewer uses OAuth 2.0 for user authentication and runs with authentication enabled by default.
`-auth-client` *client*	Specifies the value of the OAuth client. If no value is specified, the default is `sas`.
`-auth-secret` *secret*	Specifies the value of the OAuth secret. If this argument is not specified, the command line requests the OAuth client secret.
`-auth-user` *user*	Specifies the value of the OAuth user. If no value is specified, the default is `sv`.
`-auth-password` *password*	Specifies the value of the OAuth password. If this argument is not specified, the command line requests the OAuth password.

Argument	Description
`-ssl-cert` *file*	Specifies the SSL certificate file. This argument is required if authentication is not disabled.
`-ssl-password` *password*	Specifies the SSL certificate password. This argument is required if authentication is not disabled.
`-ssl-alias` *alias*	Specifies the SSL certificate alias. This argument is required if authentication is not disabled.

Authentication in Streamviewer

Note: Not all installations of SAS Event Stream Processing support authentication in Streamviewer. To verify whether your installation supports authentication in Streamviewer, see *SAS Event Stream Processing: What's New*.

Streamviewer uses OAuth 2.0 for user authentication and runs with authentication enabled by default. To disable authentication, use the `-noauth` parameter when starting the Streamviewer server. If `-noauth` is not specified, the SSL arguments `-ssl-cert`, `-ssl-password`, and `-ssl-alias` are required. For more information about specifying the authentication parameters when starting Streamviewer, see Starting Streamviewer

To create a self-signed SSL certificate, you can use `keytool`:

```
keytool -storetype PKCS12 -storepass password -alias alias -genkey -keyalg RSA -keystore full-keystore-name
 -validity 3650 -keysize 2048
```

`keytool` creates a KeyStore at *full-keystore-name*.

To generate a pseudo-random secret for the OAuth secret, you can use OpenSSL:

```
openssl rand -base64 32
```

The command line returns randomly generated Base64 output.

The Streamviewer server functions as an OAuth server. Send an HTTP request to a running Streamviewer server to retrieve an access token for OAuth authentication. You can send an HTTP request from the command line with the `curl` command-line utility:

```
curl -s -u client:secret -X POST
"https://server:port/oauth/token?grant_type=password&username=user&password=password"
```

Note: If you are retrieving a token from a server using a self-signed certificate, set the `--insecure` flag for the `curl` utility.

The command line returns a JSON object with a field called `access_token`:

```
{"access_token":"13c8edf9-4a05-41af-ba52-418c0d5f6587","token_type":"bearer",
"refresh_token":"af429a12-caa9-492b-a83d-0c21c1b28874","expires_in":43199,"scope":"user_info"}
```

In this case, the access token is `13c8edf9-4a05-41af-ba52-418c0d5f6587`. To make subsequent HTTP requests to the Streamviewer server, set your Authorization header as follows:

```
Authorization:Bearer 13c8edf9-4a05-41af-ba52-418c0d5f6587
```

Access tokens expire after one day. When a token expires, Streamviewer sends an expired token error and prompts you for a new access token.

Note: The Streamviewer UI prompts the user for an access token if a token is not provided or the token provided has expired.

Examples for Starting Streamviewer

The following examples demonstrate how to start Streamviewer using all the required parameters for each supported database product:

Note: Not all installations of SAS Event Stream Processing support the `-noauth` argument. To verify whether your installation supports authentication in Streamviewer, see *SAS Event Stream Processing: What's New*.

H2

Start Streamviewer on a Linux host. The H2 database file has the name **`config.db`**.

```
streamviewer.sh -product h2 -http 5990 -h2file config.db -noauth
```

PostgreSQL

Start Streamviewer on a Windows host. The PostgreSQL database with the name **`config`** is running on host stream.example.com:5980.

```
streamviewer.bat -product postgresql -http 5990 host stream.example.com -port 5980 -db config
-user sasadmin -password saspw -noauth
```

MariaDB

Start Streamviewer on a Linux host. The MariaDB database with the name **`config`** is running on host stream.example.com:5980.

```
streamviewer.sh -product mariadb -http 5990 host stream.example.com -port 5980 -db config
-user sasadmin -password saspw -noauth
```

MySQL

Start Streamviewer on a Windows host. The MySQL database with the name **`config`** is running on host stream.example.com:5980.

```
streamviewer.bat -product mysql -http 5990 host stream.example.com -port 5980 -db config
-user sasadmin -password saspw -noauth
```

SQLServer

Start Streamviewer on a Linux host. The SQLServer database is running on host stream.example.com:5980.

```
streamviewer.sh -product sqlserver -http 5990 host stream.example.com -port 5980
-user sasadmin -password saspw -noauth
```

Sybase

Start Streamviewer on a Windows host. The Sybase database is running on host stream.example.com:5980.

```
streamviewer.bat  -product sybase -http 5990 host stream.example.com -port 5980
-user sasadmin -password saspw -noauth
```

Oracle

Start Streamviewer on a Linux host. The Oracle database with the service name **`config`** is running on host stream.example.com:5980.

```
streamviewer.sh -product oracle -http 5990 host stream.example.com -port 5980 -db config
-user sasadmin -password saspw -noauth
```

Teradata

Start Streamviewer on a Windows host. The Teradata database with the name **`config`** is running on host stream.example.com:5980.

```
streamviewer.bat -product teradata -http 5990 host stream.example.com -port 5980 -db config
-user sasadmin -password saspw -noauth
```

Accessing the Streamviewer URL

After Streamviewer is successfully started, direct your browser to the Streamviewer URL:

```
http://<hostname>:<http_port>
```

Replace *<hostname>* with the host name of the server where the Streamviewer files are installed and running.

Replace *<http_port>* with the port number provided with the launch of the Streamviewer script.

Streamviewer loads the ESP Model Viewer page.

Authenticating ESP 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.

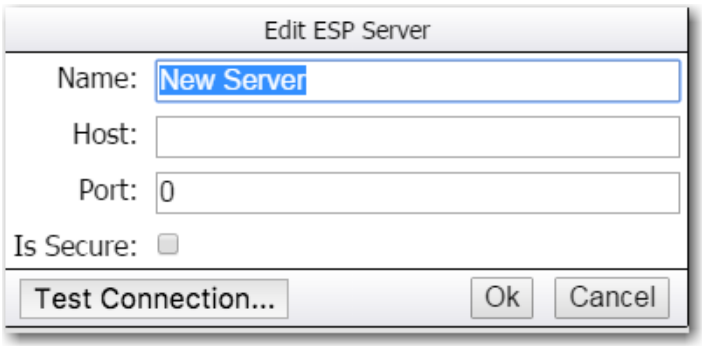
Using Streamviewer

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. You can select running HTTP Publish/Subscribe servers to publish models and events into your dashboard.

Follow these steps to connect to servers in Streamviewer:

1. At the bottom of the ESP Model Viewer page, click **Manage Servers**. The ESP Servers page is displayed. This page shows a list of all ESP servers that have been defined for Streamviewer.
2. To add an ESP server to the list, click **New**. The Edit ESP Server window is displayed.



3. Enter your ESP server location for **Host** and your HTTP Publish/Subscribe port for **Port**. You can use any unique string for **Name**.

   Note: Streamviewer displays events for ESP servers at the same release level as Streamviewer, which is 4.3. Connecting Streamviewer to a server running a previous version of SAS Event Stream Processing is not supported and can display unpredictable results.

4 If the ESP server uses a secure connection, check **Is Secure**.

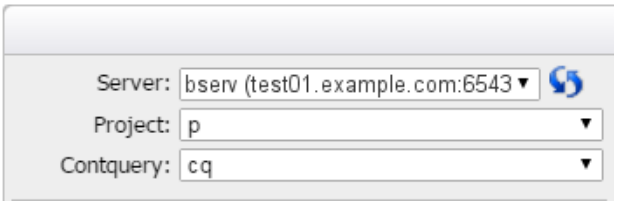
5 If you want to verify that the information that you entered is correct and that the server is running, click **Test Connection**. This test can help verify the information that you entered only if the server is active. If the server is not active, you can still add the server information to the configuration for Streamviewer.

Repeat these steps to add more ESP servers to Streamviewer. 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.

Using the ESP Model Viewer

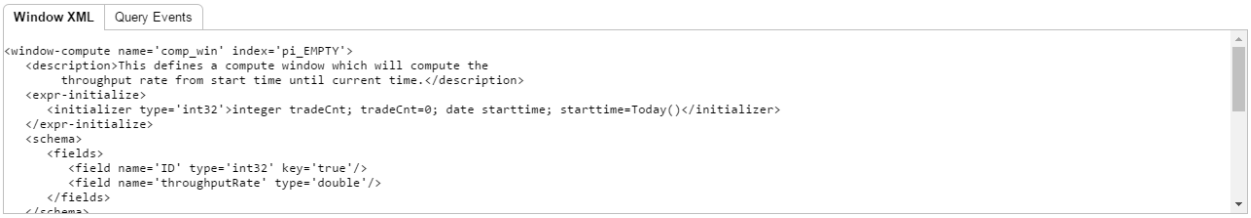
Use the ESP Model Viewer to arrange and display a project's windows and their connectors:

1 Select a **Project** and **Contquery**.



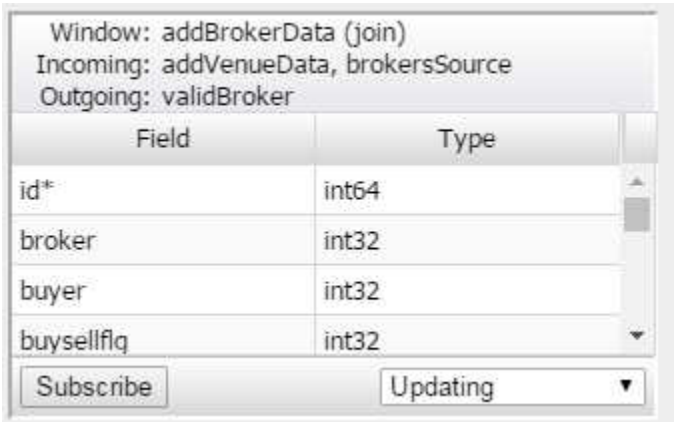
The windows for the project are listed in the left pane along with the window type. The windows are also diagrammed in the content area at right along with their edges. You can change the orientation of the model and the type of arrows between connected windows with the **Link Type** and **Orientation** options at the bottom of the left pane.

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**.



2 Click a window to display details about it. These details include a window's incoming and outgoing data sources and its schema.

Figure A.1 *ESP Window Details*



3 To place a window in the dashboard, select one of the following subscription options from the drop-down list in the ESP Window details:

Updating
The opcode of the event is used either to 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
This option is the same as Streaming mode except that the subscription is populated with a current snapshot of the window contents.

4 After selecting the subscription option, click **Subscribe**. You can subscribe to other windows to view them in the same dashboard.

5 When you are finished with your subscriptions, click **Done**. The Streamviewer dashboard appears.

Using the Dashboard Interface

After you subscribe to windows in the ESP Model Viewer and click **Done**, 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 by 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 resize 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 a window that is subscribed in updating mode, the opcode is checked. If 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 window that is subscribed in streaming mode, 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 tables in streaming mode scroll to the bottom of the table to show the most recent events.

You can pause and restart streaming and updating windows at any time. Pausing any visual component 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 repopulates 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.

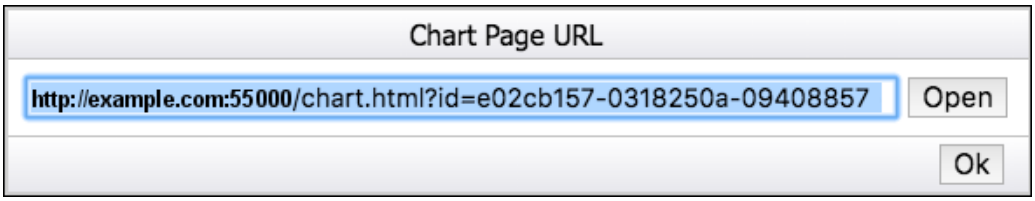
Embedding Charts

You can embed charts from the dashboard into any web page. You can only embed charts that are currently saved within a Streamviewer dashboard. When you delete a chart, it can no longer be embedded.

The simplest way to embed a chart into another web page is to use the <iframe/> tag. Follow these instructions to embed a chart using an <iframe/> tag.

1. Place your pointer over the chart that you want to embed. A toolbar appears for that chart's window.
2. Click the **Edit Chart** icon: 

   The Modify Chart window is displayed.
3. Click **Chart Page**. The chart page URL is displayed:



   Note: You can click **Open** to view the selected chart in its own window.
4. Copy the URL in this field to your clipboard. You might want to save this URL to disk for safekeeping.
5. On the web page where you want to embed the chart, add an <iframe/> tag and use the chart page URL for the value of the `src` attribute:

```
<iframe src="http://example.com:55000/chart.html?id=e02cb157-0318250a-09408857" />
```

If you want to provide more visual control of the components of the chart on a web page, you can embed the chart using Javascript. The Javascript provides methods that allow you to change the appearance of embedded charts or gauges, which display key performance indicators (KPIs). Follow these instructions to embed a chart using Javascript:

1. Copy the following file from the ESP installation directory:

```
$DFESP_HOME/tools/svchart.tar.gz
```

2 Unpack the archive file in the directory of the page that will display the chart:

```
tar -xcf svchart.tar.gz
```

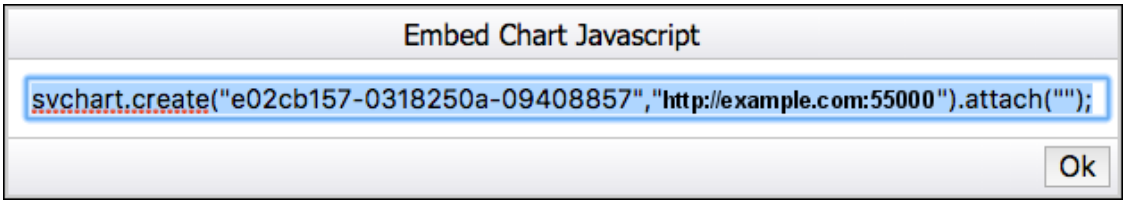
Verify that the files are extracted to an **/esp** subdirectory relative to the page that will display the chart.

3 On the dashboard page in Streamviewer, place your pointer over the chart that you want to embed. A toolbar appears for that chart's window.

4 Click the **Edit Chart** icon: 

The Modify Chart window is displayed.

5 Click **Javascript**. The Embed Chart Javascript window is displayed.



6 Copy the text in this field to your clipboard. You might want to save this text to disk for safekeeping.

7 On the web page where you want to embed the chart, add the following code to the <head/> tag:

```
<script data-main="esp/js/svchartApp" src="esp/js/libs/require.js"></script>
<link rel="stylesheet" href="esp/style/svchart.css" type="text/css" />

<script type="text/javascript">

function
setupEsp(svchart)
{
   /*
      change this if you want to change the appearance of your embedded charts
      valid values are pressed, sheen, crisp, gloss, matte, flat, and none
   */

   svchart.getServerSupport().setGraphSkin("sheen");    // 1

   /*
      change this if you want to change the appearance of your embedded gauges (KPIs)
      valid values are basic, charcoal, modern, onyx, and satin
   */

   svchart.getServerSupport().setKpiSkin("modern");     // 2

   /* add your embedded chart code here */
   svchart.create("e020cb157-0318250a-09408857","http://example.com:55000").attach("mychart"); // 3
}

</script>
```

a You can replace the argument for setGraphSkin() with one of these values: `sheen`, `crisp`, `gloss`, `matte`, `flat`, or `none`.

b You can replace the argument for setKpiSkin() with one of these values: `basic`, `charcoal`, `modern`, `onyx`, or `satin`.

c Replace the svchart.create() method in this code with the code that you copied from the Embed Chart Javascript window. Add a unique page ID to the attach( ) method. In this example, the ID '`mychart`' was used.

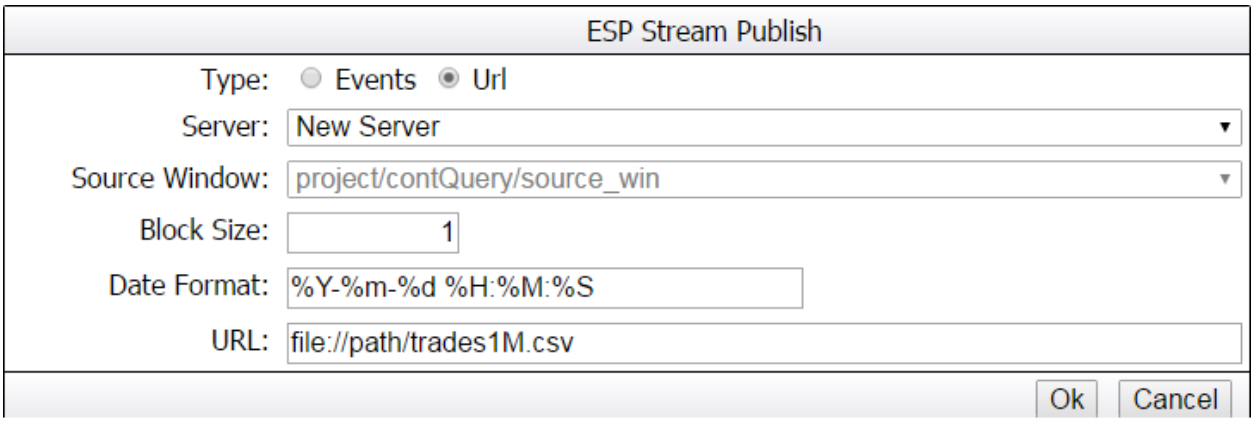
8 Add the chart to the page using the ID from the previous step. Typically, you should use a tag to contain the embedded chart, for example:

```
<div id="mychart" />
```

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.



Sharing Streamviewer Data with Other Servers

You can import and export the Streamviewer configuration data with other servers. You can also use this feature to import SQLite configuration from a previous release of SAS Event Stream Processing.

Exporting Streamviewer Data

Follow these steps to export Streamviewer data to a file that can be imported to an instance of Streamviewer running on another server:

1 Click the **Export** icon ( ) from the toolbar of the dashboard.

The Export Data window is displayed showing the XML configuration.

2 Copy the XML configuration from the Export Data window and save it to a local file.

Use the local file with the XML export data to import the data to another Streamviewer server.

Importing Streamviewer Data

You can import data directly from another running Streamviewer server or from a file:

1 Click the **Import** icon ( ) from the toolbar of the dashboard.

The Import Data dialog box is displayed.

2 Select whether to import data from a running Streamviewer **Server** or from a **File**.

- If you select **Server**, complete these fields and click **Import**:

  Source Config URL
  The configuration URL of the server to import from. If the remote Streamviewer is using versions of SAS Event Stream Processing earlier than 4.3, this is the host name of the ESP server and the -http-sql port. For later releases, this URL is the host name and port of a Streamviewer instance.

  Datasource
  This field is used only if the remote Streamviewer instance is running on a version of SAS Evenet Stream Processing earlier than 4.3. Use the drop-down menu to select a data source from the remote server.

  Change host name to
  Enter the host name of the server that will be using the imported configuration. This enables you to keep a reusable set of dashboards that you can import and direct to a new server.

- If you select **File**, click **Choose File** to browse for the file to upload and enter the new host name for the imported data.

  Note: The imported file should have an XML structure similar to the following. This is the output from following the instructions in Exporting Streamviewer Data on page 11.

```
<rows continue-on-error='false'>
   <insert table='streamviewer_server'>
      <values>
         <value column='id'>e01b157a0318250a09408857a472898e</value>
         <value column='name'>bserv</value>
         ...
         </values>
   </insert>
   <insert table='streamviewer_dashboard'>
      <values>
         <value column='id'>e01ab91c-0318250a-09408857-455cf343</value>
         <value column='name'>sample</value>
         <value column='creation_time'>1485523728055</value>
         ...
         </values>
   </insert>
   ...
</rows>
```

Viewing ESP Models

The ESP Model Visualizer enables you to view SAS Event Stream Processing models that exist either in a server or in a static file. To open the ESP Model Visualizer, click the **Show Model Visualizer** icon from the dashboard toolbar:  .

- To view a model in a local XML file, drag and drop the file into the content portion of the window.
- To view a model in a running ESP server, enter the host name and port of the ESP server in the **Server** field and click **Load**. The **Project** and **Contquery** fields are populated with the default values. You can select other continuous queries to load if they are available in the project.

After you load a model in the visualizer, the windows in the model are shown in the content area. Use the **Orientation** field to display the model with a horizontal or vertical layout. You can change the type of links between the windows to curved, orthogonal, or direct.

For each window, an icon at the bottom indicates whether the window is stateless (has an index of pi_EMPTY) or stateful. For more information about indexes, see “Understanding Primary and Specialized Indexes” in SAS Event Stream Processing: Creating and Using Windows.