What’s New for SAS Event Stream Processing 5.2

Changes to Product Deployment of SAS Event Stream Processing for Edge Computing

You now use SAS Mirror Manager, a command-line utility for synchronizing a collection of SAS software repositories, to install SAS Event Stream Processing for Edge Computing. Using this utility, you choose packages to install to control the size of the deployment on the edge device.

You must install the contents of the `/basic` directory. The following directories provide optional additions to functionality:

- `/analytics`
- `/astore`
- `/gpu`

For more information, see SAS Event Stream Processing for Edge Computing: Deployment Guide.

New Python Modeling Interface

The SAS Event Stream Processing Python modeling interface enables you to create SAS Event Stream Processing models programmatically in Python. Using the Python modeling interface, you can connect to an ESP server and interact with projects and their components as Python objects. These objects include projects, continuous queries, windows, events, loggers, SAS Micro Analytic Service modules, routers, and analytical algorithms.

The ESPPy package is open source and is expected to be available at https://github.com/sassoftware in September 2018.

For more information, see the SAS Event Stream Processing: Python Modeling Interface.
New JavaScript API

SAS Event Stream Processing 5.2 provides a set of JavaScript objects and methods that communicate with running ESP servers. Using the ESPJS API, you can create models and publish/subscribe operations in SAS Event Stream Processing from within web pages and across other platforms that support JavaScript.

For more information, see *SAS Event Stream Processing: JavaScript API*.

Changes to Streaming Analytics

Beginning with release 5.2, SAS Event Stream Processing Analytics is supported on Microsoft Windows. New analytics algorithms are available. You can use the following online models:

- Team Frequency — Inverse Document Frequency
- Lag Monitoring
- Cepstrum Transform
- Video Encoding
- Subspace Tracking

You can use the following offline models:

- Recommender Scoring
- Stability Monitoring Scoring
- Butterworth Filter

New Support for Using GPUs to Score Data with Deep Learning Models

SAS Event Stream Processing 5.2 supports the use of graphical processing units (GPUs) to score streaming data with the following deep learning models:

- Deep Neural Networks (DNN)
- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)

For more information, see “Using Deep Learning Models” in *SAS Event Stream Processing: Using SAS Event Stream Processing Analytics*.

GPUs are supported only on systems that run Linux. For more information, see *SAS Event Stream Processing on Linux: Deployment Guide*.

New and Improved Connectors and Adapters

SAS Event Stream Processing 5.2 provides the following new connectors and adapters:

- Pylon connector and adapter
- Timer connector and adapter
- Modbus connector and adapter
- Teradata Listener adapter
Note: DataDirect drivers are no longer supported. Generic ODBC drivers are now used. For more information, see “Using the Database Connector” in SAS Event Stream Processing: Connectors and Adapters.

The following improvements have been made to connectors and adapters:

- A new ACTIVE attribute of the connector element enables you to mark connectors inactive in an XML model. Using this attribute can simplify model development and testing.
- Support is now available for publisher max events.
- Support is now available for float precision in ASCII representation of doubles.
- Password encryption for LASR and SAS Cloud Analytic Services (CAS) adapters is now available.
- Support is now available for ESP binary and rstring blob types in the CAS adapter.
- Password encryption now uses AES-256 instead of AES-128.
- Encryption on the connection to the Rabbit MQ server by the Rabbit MQ connector and adapter and client transports is now available.
- Configurable HTTP headers in the REST adapter POST request is now supported.
- You now can configure the metadata topic that is used by Kafka client transports.
- Support is now available for the Kafka connector and adapter and client transports to read multiple Kafka partitions. The new kafkapartition parameter of the Kafka connector and adapter enables you to use all partitions in the Kafka topic.
- The Twitter adapter now requires that you authorize with Twitter using official SAS credentials. Use a new utility, dfesp_twitter_auth, to supply these credentials.

For more information, see SAS Event Stream Processing: Connectors and Adapters.

Changes to the Procedural Window

Beginning with SAS Event Stream Processing 5.2, the Procedural window provides input handlers only for C++ and the DATA step.

Support for SAS Micro Analytic Service (MAS) modules and stores has moved from the Procedural window to the Calculate window.

Note: Support for the Procedural window input handler that used DS2 code run in table server mode is deprecated.

For more information, see “Creating Procedural Windows” in SAS Event Stream Processing: Creating and Using Windows.

SAS Event Stream Processing provides a utility to convert a Procedural window to a Calculate window.

Enhanced Control of Connectors and Patterns

You can turn connectors and patterns on and off in your model code. Use the ACTIVE attribute of the relevant XML element. This might be helpful when you want to use a model in both development and production environments. You could activate one set of connectors for development, deactivate them for production, and activate a different set for production.

For more information, see SAS Event Stream Processing: XML Language Dictionary.

Enhancements to Security

- SSL certificate specification is now optional.
OAuth2 support is now available for cloud deployments. For more information, see “OAuth 2.0 Cloud Authentication and Authorization” in SAS Event Stream Processing: Security.

When setting up access control, you now can specify "*" (including the quotation marks) instead of the name of a specific engine, project, query, or window within the permissions file. For more information, see “Using Access Control” in SAS Event Stream Processing: Security.

The ESP Server can now handle both encrypted and unencrypted models. You can use the ESP server to encrypt models for production so that applications do not expose the model logic. When encrypted models are used in the ESP server, the server blocks queries to model components through the publish/subscribe APIs and the RESTful API. For more information, see “Encrypt a Model” in SAS Event Stream Processing: Using the ESP Server.

New calls to the RESTful API enable you to encrypt a project model or to retrieve a currently loaded model that is encrypted. For more information, see “Encrypting Models” in SAS Event Stream Processing: Using the ESP Server.

ECDH support is now available for encryption ciphers used in encrypted connections. For more information, see “Enabling Encryption on TCP/IP Connections” in SAS Event Stream Processing: Security.

Enhancements to SAS Event Stream Processing Studio
Several enhancements have been made to SAS Event Stream Processing Studio for release 5.2:

- The look and feel of the user interface has been enhanced to improve the user experience.
- You can use SAS Event Stream Processing Studio to create and manage multiple versions of a project. This enables you to edit a working copy of a project without affecting any project versions that you have previously published.
- Projects and engine definitions that are being edited in SAS Event Stream Processing Studio are now automatically locked. This ensures that changes to a project definition or an engine definition cannot be unintentionally overwritten by another user.
- Test mode has been improved to enable you to view test results in paged format. This prevents an excessive number of results from being simultaneously returned from the ESP server. It lets you more easily browse through a large number of results. Also, you can now control whether dates appear in Coordinated Universal Time (UTC) (for example, 2018-10-11T13:33:26.000Z) or whether they appear in an epoch format.
- Changes to the Procedural window are properly reflected in SAS Event Stream Processing Studio.

The following enhancements have been made to the SAS Event Stream Processing Modeler.

<table>
<thead>
<tr>
<th>Enhancement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge display types</td>
<td>Connecting edges now appear differently depending on the type of connection between windows.</td>
</tr>
<tr>
<td>Edge roles</td>
<td>You can now use the user interface to configure specific edge roles in your model based on the edge's target window.</td>
</tr>
<tr>
<td>Model windows</td>
<td>The look and feel of model windows has been enhanced. Window icons have been redesigned to indicate the window's current state.</td>
</tr>
<tr>
<td>Pattern window</td>
<td>The Pattern window has been redesigned to make it easier to create and configure patterns in your model.</td>
</tr>
<tr>
<td>Enhancement</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Score window</td>
<td>The Score window has been modified so that you can specify only one input/output map per Score window for offline model types. For ASTORE models, the input/output map values are generated from the referenced ASTORE file. For recommender models, input/output maps are the same and are retrieved from the ESP Server. You can manually enter additional properties for recommender models. Each Score window can contain its own unique input/output map. It is not necessary to use the same input/output map for all Score windows.</td>
</tr>
<tr>
<td>Model Reader window</td>
<td>The Model Reader window has been modified so that you specify a model type of ASTORE or recommender. For ASTORE models, you must provide a valid file path containing the relevant ASTORE file that you want to use. For recommender models, you can manually enter values for the model's properties. The model type used by the Score window, and any connecting Model Reader windows, must match.</td>
</tr>
</tbody>
</table>

For more information, see [SAS Event Stream Processing: Using SAS Event Stream Processing Studio](https://support.sas.com/documentation/onlinedoc/eatsp/index.htm).

**Enhancements to Streamviewer**

- Various improvements have been made to Streamviewer. The user interface has been updated for improved usability. For more information, see [SAS Event Stream Processing: Visualizing Event Streams with Streamviewer](https://support.sas.com/documentation/onlinedoc/eatsp/index.htm).

**General Product Enhancements**

- You now can reduce memory consumption by not indexing events in models. For more information, see “Reducing Model Memory Consumption” in [SAS Event Stream Processing: Advanced Topics](https://support.sas.com/documentation/onlinedoc/eatsp/index.htm).

- You now can explicitly assign a role to the edge to define which window connects to the Join window: **left** or **right**. For more information, see “Overview to Join Windows” in [SAS Event Stream Processing: Creating and Using Windows](https://support.sas.com/documentation/onlinedoc/eatsp/index.htm).

- The caching store has been extended to provide resilience across ESP server restarts. This enables large lookup databases to load just once. Thus, a restarted server instantaneously has access to the previously loaded lookup data. For more information, see “Using the pi_HLEVELDB_NC Primary Index for Event Data Persistence” in [SAS Event Stream Processing: Creating and Using Windows](https://support.sas.com/documentation/onlinedoc/eatsp/index.htm).