



**New Features Summary**

# **SAP Sybase Event Stream Processor 5.1 SP03**

DOCUMENT ID: DC01616-01-0513-02

LAST REVISED: August 2013

Copyright © 2013 by SAP AG or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries. Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notices.

# Contents

- New Features .....1**
  - New Features in 5.1 SP03 .....1
    - ESP Studio .....1
    - Adapters .....1
  - New Features in 5.1 SP02 .....6
    - Installation Changes .....6
    - Server Changes .....6
    - ESP Studio .....7
    - Adapters .....7
    - Monitor and Administer Event Stream Processor .....8
    - HANA Model within PowerDesigner Extensions .....8
    - ODBC Driver for Event Stream Processor .....9
    - SAP BI Authentication .....9
  - New Features in 5.1 SP01 .....9
  - New Features in 5.1 .....10
    - CCL .....10
    - ESP Studio .....12
    - Project Configuration .....13
    - Monitor and Administer Event Stream Processor .....13
    - Adapters .....13
    - Reporting Infrastructure .....14
    - Security .....15
    - PowerDesigner Extensions for Event Stream Processor .....15



# New Features

This New Features Summary provides high-level, overview information on new features introduced in SAP® Sybase® Event Stream Processor 5.1. This is a cumulative summary of all features introduced since the initial release of 5.1 up to and including the current Support Package. Features are listed with the particular release in which they were first introduced, and remain available in all subsequent versions unless otherwise specified.

## New Features in 5.1 SP03

---

SAP Sybase Event Stream Processor 5.1 SP03 introduces an external adapter framework and the ability to run ESP Studio from the HANA Studio.

### ESP Studio

In addition to the SAP Sybase Event Stream Processor Studio functionality provided in version 5.1 SP01 and SP02, Event Stream Processor 5.1 SP03 includes changes to the ESP Studio Help, and the ability to use ESP Studio from within SAP HANA Studio.

ESP Studio Help now has a Welcome page rather than a Learning Perspective. The Welcome page provides access to tutorial cheat sheets, samples, what's new highlights, and links to Web resources including videos and the ESP Developer Center.

SAP Sybase Event Stream Processor Studio can be installed as a feature in SAP HANA Studio. All ESP features in the Authoring and Run/Test perspectives, and the online help are available to HANA Studio users who have installed Event Stream Processor.

For detailed information on using the ESP Studio, see the *Studio Users Guide*.

### Adapters

In addition to the full set of adapters provided in version 5.1 SP02, Event Stream Processor 5.1 SP03 includes support for 13 new external (Java) adapters. These adapters were written using the new ESP adapter toolkit added in this release.

**Table 1. New Adapters in Event Stream Processor 5.1 SP03**

Adapter	Description
File CSV Input and Output Adapters	The File CSV Input adapter obtains CSV data from files on a local hard disk, and publishes it to Event Stream Processor. The File CSV Output adapter reads rows from Event Stream Processor, and writes this data to a specified CSV file.

Adapter	Description
File JSON Input and Output Adapters	The File JSON Input adapter takes JSON messages from JSON files, and publishes them to Event Stream Processor. The File JSON Output adapter takes data from Event Stream Processor, changes it to JSON format, and sends it to a JSON file.
File XML Document Input and Output Adapters	The File XML Document Input adapter loads data from an XML document to a project in Event Stream Processor. The File XML Document Output adapter outputs data from a project in Event Stream Processor to an XML document.
File XML Record Input and Output Adapters	The File XML Record Input adapter reads XML list text files, and inputs this data to Event Stream Processor. The File XML Record Output adapter reads rows from Event Stream Processor, and writes this data to XML list files.
FTP CSV Input and Output Adapters	The FTP CSV Input adapter obtains CSV data from an FTP server, and publishes it to Event Stream Processor. The FTP CSV Output adapter takes data from Event Stream Processor, formats it to CSV format, and saves it to a file on an FTP server.
FTP XML Input and Output Adapters	The FTP XML Input adapter brings data from an XML document on an FTP server to Event Stream Processor. The FTP XML Output adapter reads XML data from an ESP project, writes it to an XML document, and uploads this file to the FTP server.
JDBC Input and Output Adapters	The JDBC Input adapter receives data from tables in a database, and inputs it to Event Stream Processor. The JDBC Output adapter sends data from Event Stream Processor to a database table.
JMS CSV Input and Output Adapters	The JMS CSV Input adapter reads CSV data from a JMS server, and outputs this data to Event Stream Processor. The JMS CSV Output adapter sends CSV data from Event Stream Processor to a JMS server.

Adapter	Description
JMS Object Array Input and Output Adapters	The JMS Object Array Input adapter receives object array data from a JMS server and publishes it to Event Stream Processor. The JMS Object Array Output adapter takes data from Event Stream Processor, formats it in object array format, and sends it to a JMS server.
JMS XML Input and Output Adapters	The JMS XML Input adapter obtains XML list string messages from a JMS server, and publishes them to Event Stream Processor. The JMS XML Output adapter takes XML data from Event Stream Processor, formats it in XML list format, and sends it to a JMS server.
Socket CSV Input and Output Adapters	The Socket CSV Input adapter obtains CSV string data from a socket server, and publishes it to Event Stream Processor. The Socket CSV Output adapter takes data from Event Stream Processor, formats it in CSV format, and outputs it to a socket server.
Socket JSON Input and Output Adapters	The Socket JSON Input adapter obtains streaming data from a socket server, formats data in JSON format, and inputs it to Event Stream Processor. The Socket JSON Output adapter takes JSON data from Event Stream Processor, formats it to ByteBuffer, and transports it to a socket server, in streaming format.
Socket XML Input and Output Adapters	The Socket XML Input adapter obtains XML data from a socket server, and publishes it to Event Stream Processor. The Socket XML Output adapter takes data from Event Stream Processor, formats it to XML list format, and outputs it to a socket server.

For a full list of supported adapters, see *Adapters Supported by Event Stream Processor > Adapter Summary* in the *Adapters Guide*.

Use the new Event Stream Processor adapter toolkit to build custom external adapters. The toolkit consists of several component modules, such as transporters, formatters, and EspConnectors (EspSubscriber and EspPublisher). It also contains numerous standard transporters and formatters that you can re-use when creating a custom external adapter.

**Table 2. Standard Transporters in Event Stream Processor 5.1 SP03**

Transporter	Description
File Input Transporter	The File Input transporter reads data from local files, wraps the data with a <code>string</code> , and sends it to the next module specified in the adapter configuration file.
File Output Transporter	The File Output transporter obtains data from the previous module specified in the adapter configuration file, and writes it to local files.
FTP Input Transporter	The FTP Input transporter reads binary data from files on an FTP server, wraps it up with Byte-Buffer, and sends it to the next configured module in the adapter configuration file.
FTP Output Transporter	The FTP Output transporter obtains data from the previous module configured in the adapter configuration file, and saves it to files on the FTP server.
JDBC Input Transporter	The JDBC Input transporter reads database records using JDBC and sends them to the next module specified in the adapter configuration file.
JDBC Output Transporter	The JDBC Output transporter obtains data from previous modules specified in the adapter configuration file, and writes it to a database table using JDBC.
JMS Input Transporter	The JMS Input transporter receives JMS messages from a JMS server, and sends them to the next module that is configured in the adapter configuration file.
JMS Output Transporter	The JMS Output transport obtains data from the previous module that is configured in the adapter configuration file, wraps it up, and sends it to a JMS server.
Socket Input Transporter	The Socket Input transporter reads binary data from the socket interface, wraps it with Byte-Buffer, and sends it to the next module that is configured in the adapter configuration file.
Socket Output Transporter	The Socket Output transporter obtains data from the previous module configured in the adapter configuration file, and outputs it using the socket interface.



**Table 3. Standard Formatters in Event Stream Processor 5.1 SP03**

Formatter	Description
CSV String to ESP Formatter	The CSV to ESP formatter translates CSV strings to AepRecord objects.
ESP to CSV String Formatter	The ESP to CSV String formatter translates AepRecord objects to CSV strings.
JSON String to ESP Formatter	The JSON String to ESP formatter translates JSON strings to AepRecord objects.
JSON Stream to JSON String Formatter	The JSON Stream to JSON String formatter reads data from InputStream, splits it into standalone JSON message strings, and sends these message strings to the next module that is configured in the adapter configuration file.
ESP to JSON Stream Formatter	The ESP to JSON Stream formatter translates AepRecord objects to JSON strings, and sends the JSON strings to next streaming output transporter that is configured in the adapter configuration file.
Object List to ESP Formatter	The Object List to ESP formatter converts an object list to an ESP row.
ESP to Object List Formatter	The ESP to Object List formatter converts an ESP row to an object list.
Stream to String Formatter	The Stream to String formatter reads streaming data from an input stream, and splits it into Java strings.
String to Stream Formatter	The String to Stream formatter writes Java strings to output streams.
String List to ESP Formatter	The String to ESP formatter converts a string list to an ESP row.
ESP to String List Formatter	The ESP to String List formatter converts an ESP row to a string list.
XML String to ESP Formatter	The XML String to ESP formatter translates ESP XML strings to AepRecord objects.
ESP to XML String Formatter	The ESP to XML String formatter translates AepRecord objects to ESP XML string.

Formatter	Description
XMLDOC Stream to ESP Formatter	The XMLDOC Stream to ESP formatter parses XML format strings, extracts data according to the schema file specified in the adapter configuration file, and translates this data to AepRecord objects.
ESP to XMLDOC String Formatter	The ESP to XMLDOC String formatter translates AepRecord objects to XML format string according to the schema file specified in the adapter configuration file.

For additional information on the adapter toolkit and the standard transporter and formatter modules, see the *Event Stream Processor Adapter Toolkit* section in the *Building Custom Adapters* guide.

## New Features in 5.1 SP02

---

SAP Sybase Event Stream Processor 5.1 SP02 introduces installation and server changes, new schema discovery and mapping functionality in Studio to support two new adapters, new capabilities for administering Event Stream Processor, an ODBC driver for Windows, and a new authentication option.

### Installation Changes

The installation process is updated in two key areas.

First, the SAP Sybase Event Stream Processor Studio is no longer included in the Typical installation on Solaris, nor is it available as an option in the Custom installation. SAP Sybase Event Stream Processor no longer supports SAP Sybase Event Stream Processor Studio on Solaris. However, Solaris users requiring access to SAP Sybase Event Stream Processor Studio can download and install the SAP Sybase Event Stream Processor Studio on one of the other supported platforms, and use that version of SAP Sybase Event Stream Processor Studio with their Solaris server.

Second, the installer now installs a 32-bit ODBC driver on both Windows 32-bit and 64-bit platforms. This driver enables C/C++ programs and third-party applications to send SQL queries or statements to ESP.

### Server Changes

Previously, the ESP server calculated the difference between two date, timestamp, or bigdatetime datatypes as date or time values.

For example, when calculating the difference between the two dates 2010-03-27 00:00:00 and 2010-03-01 00:00:00, the value returned was the date 1970-01-26 00:00:00. In Event Stream

Processor 5.1 SP02, the server has been modified to calculate the difference between two date, timestamp, or bigdatetime datatypes as interval values. For example, when calculating the difference between the two dates 2010-03-27 00:00:00 and 2010-03-01 00:00:00, the value returned is 26 days.

If you have existing projects from previous versions that calculate differences between date, timestamp, or bigdatetime values, your CCL files may produce different results or may fail to compile.

The following sample code illustrates how you can now produce an interval value directly when subtracting two bigdatetime datatypes:

```
CREATE INPUT STREAM NEWSTREAM SCHEMA (Column1 bigdatetime, Column2
bigdatetime);
CREATE OUTPUT STREAM S2 SCHEMA (Column1 interval) as
    select NEWSTREAM.Column1 - NEWSTREAM.Column2 as Column1
from NEWSTREAM;
```

## **ESP Studio**

In addition to the SAP Sybase Event Stream Processor Studio functionality provided in version 5.1 SP01, Event Stream Processor 5.1 SP02 includes new schema discovery and mapping functionality for the RFC and Web Services (SOAP) Input and Output adapters.

The new schema discovery and mapping functionality discovers functions for the RFC output adapter, and functions, tables, or queries for the RFC input adapter, then lets you select which fields to map to Event Stream Processor. Use this new feature to automatically create a mapping file directly from SAP Sybase Event Stream Processor Studio.

This new functionality also discovers schema for the Web Services (SOAP) adapter and creates a mapping file based on the schema you select.

For detailed information on using SAP Sybase Event Stream Processor Studio, including information on using the schema discovery and mapping functionality, see the *Studio Users Guide*.

## **Adapters**

In addition to the full set of adapters provided in version 5.1 SP01, Event Stream Processor 5.1 SP02 includes support for three new adapters.

The new SAP Remote Function Calls (RFC) Input adapter executes RFCs to import data from SAP systems into Event Stream Processor, while the SAP RFC Output adapter exports data from Event Stream Processor into SAP systems. This adapter also publishes statistics to the `_ESP_Adapter_Statistics` metadata stream. Subscribe to this metadata stream to view statistics specific to the SAP RFC adapter.

The new Web Services (SOAP) Input and Output adapters are SOAP Web services clients which connect to a Web service to obtain data to feed into Event Stream Processor and deliver output from Event Stream Processor to a Web service. This adapter also publishes statistics to

## New Features

the `_ESP_Adapter_Statistics` metadata stream. Subscribe to this metadata stream to view statistics specific to the Web Services (SOAP) adapter.

The new ESP Web Services Provider receives data from a Web service client and publishes it to an ESP stream or window within an ESP project.

For a full list of supported adapters, see *Adapters Supported by Event Stream Processor > Adapter Summary* in the *Adapters Guide*.

## **Monitor and Administer Event Stream Processor**

In addition to the SCC functionality provided in version 5.1, Event Stream Processor 5.1 SP02 includes support for several new administration capabilities, including the ability to administer ESP clusters.

Using a manager node, you can now:

- View, add, or remove workspaces.
- Start, stop, add, or remove any project within the cluster to which the manager node belongs. Managing projects from a manager node allows you to see and manage all projects within the cluster rather than only the projects running on a selected controller node.
- Update access control for LDAP and SAP BI authentication without restarting the cluster.

You can also view schema for a stream and view project properties from the ESP Node Monitor or Cluster Monitor view.

For information on using SCC, see the *Sybase Control Center for Event Stream Processor* online help.

## **HANA Model within PowerDesigner Extensions**

In addition to the existing functionality provided with the extensions for SAP Sybase PowerDesigner® 16.5 in version 5.1 SP01, Event Stream Processor 5.1 SP02 includes support for the SAP HANA database.

The PowerDesigner extensions installed with Event Stream Processor provide a sample project which includes:

- A sample ESP schema logical model
- SAP Sybase IQ, SAP HANA, and Sybase ASE physical data models

For more information on using PowerDesigner, see *PowerDesigner for Event Stream Processor* in the *Programmers Guide*.

## **ODBC Driver for Event Stream Processor**

SAP Sybase Event Stream Processor includes a modified version of the Postgres ODBC driver.

Using this driver, C/C++ programs and third-party applications can send SQL queries or statements to ESP. For example, Microsoft Excel® can pull data from ESP into a spreadsheet.

The ODBC driver is 32-bit Windows driver. It is included in the ESP distributions for 32-bit Windows and 64-bit Windows operating systems. It is installed by default. The binaries are placed in the %ESP\_HOME%\odbc folder.

For more details, refer to the *ODBC Driver for ESP* topic in the *Adapters Guide*.

## **SAP BI Authentication**

SAP Sybase Event Stream Processor now allows SAP BI authentication.

If you have SAP BI running, you can have it handle authentication for users connecting to your SAP Sybase Event Stream Processor cluster. You can specify SAP BI authentication when installing if you choose a Custom installation. Or, you can configure SAP BI authentication after you have installed ESP.

For more details, refer to the *Authentication* topics in the *Configuration and Administration Guide*.

## **New Features in 5.1 SP01**

Event Stream Processor 5.1 SP 01 introduces a single, but significant, new feature: a high speed output adapter for SAP HANA.

The new SAP HANA output adapter can use multiple parallel ODBC connections to load data rapidly from Event Stream Processor into an SAP HANA database. The adapter is designed for high-volume, high-speed processing to take full advantage of the SAP HANA real-time, in-memory architecture.

The introduction of the SAP HANA output adapter lets you integrate Event Stream Processor into new or existing SAP HANA environments with virtually no configuration on the SAP HANA side.

This adapter also publishes statistics to the \_ESP\_Adapter\_Statistics metadata stream. Subscribe to this metadata stream to view statistics specific to the SAP HANA adapter.

# New Features in 5.1

---

SAP Sybase Event Stream Processor 5.1 introduces new features, new adapters, a new management and administration tool, and performance enhancements.

## CCL

---

Continuous Computation Language (CCL) is the primary event processing language of SAP Sybase Event Stream Processor.

SAP Sybase Event Stream Processor 5.0 introduced a simplified, streamlined version of CCL based on the CCL offering in SAP CEP R4. Version 5.1 introduces the following additional CCL functionality:

### *Jumping Windows*

Using the **KEEP** clause in CCL, you can define a retention policy for a Named or Unnamed Window. Window retention policies include time-based policies (the time duration for which a window retains rows) and count-based policies (the maximum number of rows that the window can retain). In version 5.0, retention policies always created sliding windows. In a sliding window, individual rows are deleted once a maximum age is reached or the maximum number of rows are retained (depending on the policy type).

Version 5.1 introduces the **EVERY** modifier to the **KEEP** clause. When you use the **EVERY** modifier in your **KEEP** clause, you create a retention policy that uses a jumping window. In a jumping window, all of the retained rows are deleted when the time interval expires or a row arrives that would exceed the maximum number of rows.

For example, if you create a retention policy to keep five rows of data in a sliding window, the window holds the first five rows as they feed into the stream. When the sixth row enters the stream, the first row slides out of the window, and the sixth row slides in. If you define a retention policy to keep five rows of data in a jumping window, the window holds the first five rows as they feed into the stream. When the sixth row enters the stream, all five retained rows jump out of the window, and the sixth row jumps in.

The following sample code, provided only for illustrative purposes, creates an unnamed jumping window on the source Source1 that will retain every row until the end of a 100 second interval.

```
CREATE WINDOW Win1
PRIMARY KEY DEDUCED
AS SELECT * FROM Source1
KEEP EVERY 100 SECONDS;
```

For detailed information on syntax and usage, see the *Programmers Reference*.

### *Content-based Retention*

When working with the **KEEP** clause in version 5.1, you can now use the **PER** sub-clause to retain data based on column content. In a time-based policy, data is retained for a specific time period (specified by **interval**) for each unique column/value combination. In a count-based policy, the specified number of rows (specified by **count**) are retained for each unique column/value combination. In either case, you can specify one or several columns, but you cannot use the same column more than once in the same **PER** sub-clause.

In the following sample, provided only for illustrative purposes, Win1 is a derived window that retains data for each unique value combination of Col1 and Col2. The window is a Jumping window that retains 5 rows. Upon arrival of the sixth unique value combination of Col1 and Col2, the first 5 rows are deleted and the sixth row is stored.

```
CREATE WINDOW Win1
PRIMARY KEY DEDUCED
AS
SELECT * FROM InputWin
KEEP EVERY 5 ROWS PER (Col1, Col2);
```

For detailed information on syntax and usage, see the *Programmers Reference*.

### *Splitting Input into Multiple Outputs*

The **CREATE SPLITTER** statement lets you evaluate an input stream for particular conditions. As the rows in the stream are evaluated, the splitter directs them to different target streams based on which conditions are met.

For example, you can define a splitter statement that evaluates stock symbols and directs them to different streams depending on whether the symbols are for software vendors, hardware vendors, or other vendors. In the following sample code, provided for illustrative purposes only, the splitter evaluates the source stream Trades. When the Symbol column for a row contains "IBM" or "ORCL" the row is added to two different target streams, ProcessHardWareStock and ProcessSoftwareStock. When the Symbol column for a row contains "SAP" or "MSFT" the row is added to the target stream ProcessSoftwareStock. All other rows are sent to the target ProcessOtherStock.

```
CREATE SPLITTER Splitter1 AS
WHEN Trades.Symbol IN ('IBM', 'ORCL' ) THEN ProcessHardWareStock,
ProcessSoftwareStock
WHEN Trades.Symbol IN ('SAP', 'MSFT') THEN ProcessSoftwareStock
ELSE ProcessOtherStock
SELECT * FROM Trades;
```

When a stream is filtered into more than two target streams, a splitter is typically more efficient, both in terms of CPU utilization and throughput, than an equivalent construct composed of two or more streams that does a filter. Unlike other streams in Event Stream Processor, a splitter and all its target streams run in a single thread.

For detailed information on syntax and usage, see the *Programmers Reference*.

### *Automatic Key Generation*

When you are working with input data that does not have a natural primary key, use the **AUTOGENERATE** clause to automatically generate values that can be used, for example, as a primary key for the data. Specify a column that uses the `long` data type. At runtime, values for that column are generated automatically for each insert in the stream. By default, values start at zero, but you can override this using the `FROM` clause and specifying a start value with either a parameter or an explicit long number.

The automatically-generated column is incremented only on an insert, and any preexisting value in an automatically-generated column is, on insert, ignored. On an update, delete, or upsert, a preexisting value in the automatically-generated column is used as-provided in the input row. This rule has the potential to produce duplicate rows in a window, so do not use the **AUTOGENERATE** clause with upserts, especially when the automatically generated column is a primary key.

The following sample code, provided for illustrative purposes only, creates an input stream named `Trades` with a column named `TradeId` for which the values are automatically generated.

```
CREATE INPUT STREAM Trades
SCHEMA (TradeId long, Symbol string, Shares integer, Price money(4))
AUTOGENERATE (TradeId);
```

For detailed information on syntax and usage, see the *Programmers Reference*.

## **ESP Studio**

SAP Sybase Event Stream Processor Studio provides visual and textual authoring capabilities as well as a run-test environment. This enables you to develop, test, and configure projects in a common interface.

The CCL textual editor lets you edit CCL files as text, and provides syntax completion options, syntax checking, and error validation. Experienced programmers can work in the CCL editor exclusively, or use it as a supplement to the visual editor.

The visual editor lets you create and edit projects without learning CCL syntax. The visual editor provides support for all CCL functions, including the new jumping windows, splitter, and automatic key generation features introduced to CCL in version 5.1.

To support the CCL modularity feature, the SAP Sybase Event Stream Processor Studio can be configured with a repository for blocks of CCL code called modules that can be used in multiple projects.

For detailed information on using the Studio, including information on configuring jumping windows, splitter, and automatic key generation in the visual editor, see the *Studio Users Guide*.



## **Project Configuration**

Event Stream Processor project configuration separates physical deployment elements from business logic.

Event Stream Processor 5.1 includes a new project configuration option for handling bad records.

Saves bad records to a file. Select the bad-record-file option for **Project Type**, and indicate the file name of an ESP project for **Value Field**. If a file name is not specified, bad records are discarded. See the *Administrators Guide*.

## **Monitor and Administer Event Stream Processor**

Event Stream Processor includes an operational console, Sybase® Control Center (SCC), that you can use to monitor and administer ESP nodes.

You can use this tool to monitor performance of individual nodes or clusters. It provides statistics for:

- Projects
- Streams
- Connections
- Adapters
- Publishers
- Subscribers

You can also use SCC to:

- Start and stop nodes, projects, and adapters
- Configure alerts on various statistics
- View the file activity report for the SAP Sybase IQ Output Adapter
- View ESP project and cluster node log files
- View a graphical representation (topology) of the ESP cluster

For information on using SCC, see the *Sybase Control Center for Event Stream Processor* online help.

## **Adapters**

In addition to the full set of adapters provided in version 5.0, Event Stream Processor 5.1 includes support for two new adapters.

- The new Adaptive Server Enterprise (ASE) Output adapter reads data from an Event Stream Processor stream and writes it to the Adaptive Server Enterprise using the high speed Open Client Bulk Loading API. This adapter also publishes statistics to the new

## New Features

\_ESP\_Adapter\_Statistics metadata stream. Subscribe to this metadata stream to view statistics specific to the ASE adapter.

- The new SAP Sybase IQ Output adapter reads data from Event Stream Processor and writes it to SAP Sybase IQ load files in the native SAP Sybase IQ binary format. This adapter also publishes statistics to the new \_ESP\_Adapter\_Statistics metadata stream. Subscribe to this metadata stream to view statistics specific to the SAP Sybase IQ adapter.
- **Note:** The new SAP Sybase IQ adapter replaces the adapter shipped in version 5.0, including the **esp\_iqloader** utility. To continue using the 5.0 version of the SAP Sybase IQ adapter, do not install Event Stream Processor 5.1 in the same directory as 5.0.

The generic Database input and output adapters, `db_in` and `db_out`, have also been updated:

- For ODBC connections, performance for batch inserts is enhanced.
- For JDBC connections, the generic JDBC driver is improved for better connectivity with some databases.

For a full list of supported adapters, see *Adapters Supported by Event Stream Processor > Adapter Summary* in the *Adapters Guide*.

## Reporting Infrastructure

Event Stream Processor infrastructure enhancements make it possible to read adapter statistics, adapter latency, CPU usage and more from metadata streams. SDK enhancements also make it possible to determine TCP packet sizes to determine how much room is available when sending messages from the SDK.

### *Metadata Stream Enhancements*

In version 5.1, the new metadata stream \_ESP\_Adapter\_Statistics makes it possible for both internal and external adapters to report custom statistics. To read the statistics, subscribe to this metadata stream using standard SDK subscription calls.

The \_ESP\_Connectors metadata stream now includes an additional column, latency, through which an internal adapter can report on any latency it introduces. This latency information is then reported through SCC.

The \_ESP\_Project\_Monitor metadata stream reports on CPU utilization, memory consumption, and number of threads on a per-project basis. As with other monitor-related streams such as \_ESP\_Streams\_Monitor and \_ESP\_Clients\_Monitor, this metadata stream is only updated when the time-granularity option is set to a value greater than zero.

The \_ESP\_Streams\_Monitor and \_ESP\_Clients\_Monitor metadata streams have been extended to report on system and user CPU percentage and CPU time.

For information on metadata streams, see the *Configuration and Administration Guide*.

### *SDK Support for Network Parameters*

The Java, C/C++, and .NET SDKs now provide a new API that lets you determine how much room there is in the TCP packet when sending from the SDK. Information provided includes the MTU size, IP header size, TCP header size, and ESP header size.

For a connected publisher (single socket connection), the SDK now retrieves:

- ESP binary data block header size (all SDKs, all platforms).
- TCP Maximum Segment Size (MSS) for the connected socket. The MSS is computed by the system as MTU – IP Header Size – TCP header size and denotes the space available for data in a single packet that can be sent without IP fragmentation (C/C++ SDK, Linux and Solaris only).
- Maximum transmission unit (MTU) of the connected interface and default IP and TCP header sizes (Java SDK, Linux and Solaris only).

For a message writer (the component that corresponds to one particular stream on the engine), the SDK now retrieves the row header size for the stream.

## **Security**

Security in Event Stream Processor is improved by requiring authentication for all connections to clusters.

Users connecting to a cluster from Sybase Control Center, SAP Sybase Event Stream Processor Studio, command-line utilities, or the SDK, must authenticate using any of the existing methods: Kerberos, LDAP, RSA, or by specifying user name/password credentials (native OS).

Users connecting to the local cluster in SAP Sybase Event Stream Processor Studio authenticate as the studio user, with a user-defined password that is valid for the current session only.

## **PowerDesigner Extensions for Event Stream Processor**

SAP Sybase PowerDesigner is a tool for creating and manipulating schema definitions which can be used within SAP Sybase Event Stream Processor. Integrated modeling supports efficient schema definition and database design, and consistent production deployments.

PowerDesigner extensions are used for customizing and extending PowerDesigner metaclasses, parameters, and generation. With the extended model setup, you can create and modify schema definitions in the ESP schema model, convert SAP Sybase IQ and SAP Adaptive Server Enterprise physical models into ESP schema logical models, and import and export schema definitions to and from CCL files. You can also use PowerDesigner for impact and lineage analysis.

PowerDesigner extensions can only be used on Windows platforms. You must purchase PowerDesigner separately to use this feature.

## New Features

The PowerDesigner extensions installed with Event Stream Processor provide a sample project which includes:

- A sample ESP schema logical model
- SAP Sybase IQ and ASE physical data models

You can use the sample project to modify or extend existing models, or as a guideline for creating new models within the ESP model categories.

In addition, PowerDesigner can be used to produce a set of data definition language (DDL) statements directly from the physical data models. These DDL scripts can be used to create database objects for your SAP Sybase IQ and ASE databases.

For more information on using PowerDesigner, see *PowerDesigner for Event Stream Processor* in the *Programmers Guide*.