THE POWER TO KNOW.

SAS® Event Stream Processing 4.3 on Linux: Deployment Guide

SAS® Documentation
July 26, 2017

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *SAS® Event Stream Processing 4.3 on Linux: Deployment Guide*. Cary, NC: SAS Institute Inc.

SAS® Event Stream Processing 4.3 on Linux: Deployment Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

May 2017

4.3-P1:dplyesp0phy0lax

Contents

1

Introduction

About This Guide

SAS Event Stream Processing enables developers to build applications that can quickly process and analyze a large number of continuously flowing events in real time. The deployment installs the programming tools that are required to build and execute event stream processing applications.

SAS Event Stream Processing 4.1 was the first release to include compatibility with the SAS Viya platform and to use the same deployment tools and process. SAS Event Stream Processing 4.3 enhances the deployment experience and adds multiple features. When installed along with the Cloud Analytic Services (CAS) components, SAS Event Stream Processing can provide data for analytic processing in SAS Viya.

However, SAS Event Stream Processing can still be installed as a standalone product without SAS Viya. It can also be installed with Base SAS® 9.4.

Use this guide to deploy SAS Event Stream Processing in your environment.

To use this guide successfully, you should have a working knowledge of the Linux operating system and basic commands.

What's New in SAS Deployment

SAS Repositories

To ensure that you deploy the latest software, SAS provides the SAS Event Stream Processing software in repository packages that are maintained by SAS. Specifically, the software is packaged in the RPM Package Manager (RPM) format, which simplifies installation, uninstallation, and upgrade tasks. Each time you deploy or update your software, you automatically receive the latest RPM packages that are available.

Note: The RPM-based deployment model works with repositories that are native to your operating system. As a result, a SAS Software Depot is not required in your environment.

Industry Standard Tools

You can now deploy SAS Event Stream Processing with tools that are designed for deploying and updating software on Linux operating systems. SAS Viya deployment takes advantage of yum, a software package manager for Linux operating systems. Yum commands are used for secure access to RPM packages and for deploying and updating software in your environment.

Note: The SAS Deployment Wizard and the SAS Deployment Manager that support SAS 9.4 are not used to install and configure SAS Event Stream Processing 4.3.

Upgrading Models and Data

Upgrading SAS Event Stream Processing software is not supported. Instead, you must uninstall the older version of the software and then install the newer version.

Note: The term *upgrade* is used to refer to a type of software update that introduces new functionality. At SAS, an upgrade generally involves a new release number. By contrast, an *update* refers to minor changes to the software such as fixes. For more information about updating the software, see "Updating SAS Event Stream Processing" on page 25.

Migrating models and data that you generated from a previous release of SAS Event Stream Processing is supported on a limited basis. You can import files from SAS Event Stream Processing 3.2, 4.1, or 4.2. However, if you plan to import files that you created with SAS Event Stream Processing 3.2, be aware of the following issues:

- Multiple XML elements in SAS Event Stream Processing 4.*x* have changed since 3.2. You must replace the elements that differ. Opening a legacy project in SAS Event Stream Processing Studio does not automatically upgrade your XML code to a valid format.

  You can use the dfesp_xml_migrate script to migrate your XML code to the 4.*x* XML schema.
- Review your C++ code that was used with SAS Event Stream Processing 3.2. You must replace the registerMethod_ds2 function with the registerMethod_DS2TS function.
- The default date format of %Y-%m-%d %H:%M:%S for CSV timestamp and datetime fields is no longer valid. The new ESP_DATETIME fields contain a 64-bit integer that represents seconds since UNIX epoch. The new ESP_TIMESTAMP fields contain a 64-bit integer that represents microseconds since UNIX epoch.
- In addition, you can no longer specify an alternative date format when initializing a SAS Event Stream Processing engine. To pass CSV events using an alternative date format, that format must now be specified on the connector or adapter that is the source or sink of CSV data. All connectors and adapters that support CSV include an optional DateFormat parameter for this purpose.

To upgrade models that you created in SAS Event Stream Processing 4.2 to 4.3, follow these steps:

1. In SAS Event Stream Processing Studio 4.2, export the 4.2 models that you want to use in the newer version of SAS Event Stream Processing.

2. Install SAS Event Stream Processing 4.3, as instructed in the *SAS Event Stream Processing: Deployment Guide*.

3. Use SAS Event Stream Processing Studio to import the 4.2 models that you previously exported. For more information, see *SAS Event Stream Processing: Using SAS Event Stream Processing Studio*.

To import models that you created in SAS Event Stream Processing Studio 3.2, a separate migration step is required. As noted, you must run the dfesp_xml_migrate script to migrate your XML code to the 4.*x* XML schema. For more information about the migration script, contact SAS Technical Support.

Contact SAS Technical Support

Technical support is available to all customers who license SAS software. However, we encourage you to engage your designated on-site SAS support personnel as your first support contact. If your on-site SAS support personnel cannot resolve your issue, have them contact SAS Technical Support to report your problem.

Before you call, explore the SAS Support website at support.sas.com/techsup/. This site offers access to the SAS Knowledge Base, as well as SAS communities, Technical Support contact options, and other support materials that might answer your questions.

When you contact SAS Technical Support, you are required to provide information, such as your SAS site number, company name, email address, and phone number, that identifies you as a licensed SAS software customer.

2

System Requirements

Hardware Requirements

SAS Event Stream Processing can be installed as a stand-alone product. It can also coexist with either SAS Viya or with SAS 9.4. Do not install it on the same machine as another SAS Viya component.

A single machine for the SAS Event Stream Processing components (SAS Event Stream Processing Engine, the web application server, and SAS Event Stream Processing Studio) is the minimum requirement. SAS Event Stream Processing can be deployed on a redundant machine for failover, or it can be distributed across multiple machines. On-premises deployments as well as cloud deployments are supported. You can also deploy the software on the compute layer of a Hadoop cluster, or even at the edge (on a gateway node) of a Hadoop cluster.

The following table describes a standard set of specifications for a machine where SAS Event Stream Processing is deployed:

Item	Recommended Level*
CPU	4 cores (x86 architecture) Intel Xeon chip set with a minimum speed of 2.6 GHz
Memory	8 - 16 GB of RAM Memory clock speed of 1600 MHz
Disk Space and Speed	10 GB 10,000 RPM

*The bare minimum requirements for an installation of SAS Event Stream Processing are 4 cores, 4 GB of memory, and 1 GB of disk space. However, a minimum configuration is not recommended.

An additional machine can be used as a thin client from which end users can access the user interface for SAS Event Stream Processing Studio. This machine requires minimal processing power and storage space and can run on Windows or UNIX.

Operating System Requirements

Supported Operating Systems

For a list of supported operating systems, see https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-operating-systems.html.

Note: SAS Event Stream Processing can also be installed on Microsoft Windows, but a separate package, based on your software order, is required.

Linux Prerequisites

The typical Linux installation includes all of the packages and libraries that SAS requires. Problems can occur if default packages were removed from the base operating system (for example, X11 libraries and system utilities). The following libraries are required:

- glibc 2.12
- libpng (on Red Hat Enterprise Linux 6.*x* or the equivalent)

  libpng12 (on Red Hat Enterprise Linux 7.*x* or the equivalent)
- libXp
- libXmu
- net-tools
- the numactl package
- the X11/Xmotif (GUI) packages
- xterm

On Linux 7.*x*, verify that the systemd package on each machine is at version 219-30 or later. Run the following command:

```
rpm -qa | grep systemd
```

If the version that is returned is not at least 219-30, run the following command to retrieve the most recent package from Red Hat:

```
yum update systemd
```

Additional Linux Requirements

The SAS Event Stream Processing Engine libraries were built using gcc-4.4.7-16 and the Boost library 1.58. The Boost library 1.58 is automatically installed with SAS Event Stream Processing. The libraries where compiled using the following compiler options:

```
-D_REENTRANT
```

```
-D_THREAD_SAFE
```

All the SAS Event Stream Processing applications that you build with SAS Event Stream Processing Studio must also use the same compiler options.

The SAS Event Stream Processing 4.x libraries have been built using gcc-4.4.7-16 on Red Hat Enterprise Linux Server 6.7 using libc-2.12.so, libstdc++.so.6.0.13, and libgcc_s-4.4.7-20120601.so.1.

SAS Support for Alternative Operating Systems

SAS provides support on a limited basis for alternative operating system distributions that customers might select. For more information, see the official support policy statement at http://support.sas.com/techsup/pcn/altopsys.html.

Software Requirements

Java Requirements

The Java Runtime Environment (JRE) must be installed on each machine where you install SAS Event Stream Processing components. Only the JRE is required. The full JDK is not required. For a list of supported JRE distributions, see https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-jre.html.

Web Browsers

SAS Event Stream Processing Studio and Streamviewer include some advanced user interface features, which require a newer web browser. For information about supported browsers, see https://support.sas.com/en/documentation/third-party-software-reference/viya/support-for-web-browsers.html.

If you cannot install one of the supported web browsers for use with SAS Event Stream Processing, be aware of possible unexpected user interface behavior. Because session cookies are required in order to maintain session state, be sure to enable cookies in your browser.

User Accounts

Verify the following prerequisites before you start the deployment:

- Administrator privileges are required for the Linux machine where you are launching the SAS software deployment.
- The user account that you are using for the deployment must have super user (sudo) access. To verify that the user ID is included in the sudoers file, run the following command:

  ```
  sudo -v
  ```

  To verify your sudoers privileges, run the following command:

  ```
  sudo -l
  ```

  Note: The ability to start a shell (with the `!SHELL` entry in some sudoers files) as root is not required.

During the software deployment, one required user account (sas) and one group (also named sas) are created for you unless they already exist. Because the sas account is required for the SAS Event Stream Processing Studio component to run during normal product operation, you must not delete it or change its name. It does not run as root. If you must log on to this account, use sudo to access it.

The following table describes the predefined sas user account:

Account Name and Group	Parameters	Purpose
sas; member of sas group	Non-login service account without user restrictions. No password; can add password after installation if desired. Password does not expire. Default user name is required until the installation is complete. Any post-installation changes to this account do not prevent future software updates.	Required for the installation. The installation process sets user and group ownership permissions on all of the installation files. This user must exist to enable ownership. After the installation has completed, this user account enables required components to run, including the web application server for SAS Event Stream Processing Studio.

Administrator privileges are not required after the installation to run SAS Event Stream Processing. The installation directory path enables write access per user group, and it is owned by the sas user. To grant permission to edit the configuration files, the administrator must add any user requiring write access to these files to the sas group.

(Optional) Encryption and Authentication

SAS Event Stream Processing provides optional encryption and authentication features. Both of these features require encryption libraries that are included in the SAS Event Stream Processing Encryption and Authentication Overlay package, which is installed automatically when you install SAS Event Stream Processing. You can then enable encryption with OpenSSL on TCP/IP connections within an event stream processing engine.

OpenSSL is included with the SAS Event Stream Processing Encryption and Authentication Overlay package. You can also configure SAS Event Stream Processing engines to require client authentication for SAS TCP/IP clients. Authentication and encryption apply to the following Event Stream Processing Engine APIs:

- C or Java Publish/Subscribe API
  - Connections that are created by a client that uses the C or Java Publish/Subscribe API to communicate with a SAS Event Stream Processing engine
  - Connections that are created by an adapter to communicate with a SAS Event Stream Processing engine
- Event Stream Processing XML Server HTTPS API
  - Connections that are created by the Event Stream Processing XML Client (dfesp_xml_client) to communicate with the Event Stream Processing XML server using the HTTPS protocol
  - Connections that are created by the Streamviewer component (streamviewer.html) to communicate with the Event Stream Processing XML server using the HTTPS protocol

If you set up authentication for a SAS Event Stream Processing server, you must then provide authentication tokens or credentials in Streamviewer. You can copy and paste the token directly into an appropriate dialog box in Streamviewer. Alternatively, you can specify a URL that supplies the token. Authentication tokens and credentials are cached for the duration of a Streamviewer session. For more information, see *SAS Event Stream Processing 4.3: Security*.

3

Installing SAS Event Stream Processing

Deploy with Yum

Use the procedures in this section to deploy your SAS software using yum. You can deploy all SAS Event Stream Processing components using yum, with the exception of the optional SAS Event Stream Manager component. If your order includes SAS Event Stream Manager, an Ansible playbook is required for installation. The Ansible playbook is included as an attachment in your SOE.

An upgrade of the SAS Event Stream Processing software is not supported. You must uninstall any previous versions of the software that you purchased before you install the newer version on the same machine.

Set the Environment Variables

You must set several environment variables before you start SAS Event Stream Processing. For a shell that will invoke only SAS Event Stream Processing, run the following commands:

```
export DFESP_HOME=/opt/sas/viya/home/SASEventStreamProcessingEngine/4.3.0
export LD_LIBRARY_PATH=$DFESP_HOME/lib:/opt/sas/viya/home/SASFoundation/sasexe
export PATH=$PATH:$DFESP_HOME/bin
```

If you need to maintain your LD_LIBRARY_PATH setting for another SAS product (the second command in the preceding series of commands), use the following command instead:

```
export LD_LIBRARY_PATH=$DFESP_HOME/lib:/opt/sas/viya/home/SASFoundation/sasexe:$LD_LIBRARY_PATH
```

(Optional) To enable SSL on connections between SAS Event Stream Processing Studio and the SAS Event Stream Processing engines, set the following environment variable:

```
DFESP_SSLPATH=path-to-OpenSSL-shared-object
```

This setting assumes that you installed the OpenSSL libraries on all computer systems that run the client and server. When you install SAS Event Stream Processing, OpenSSL is automatically installed as part of the SAS Event Stream Processing Encryption and Authentication Overlay. The DFESP_SSLPATH environment variable should be set to the path that contains libssl.so and libcrypto.so. By default, when the Encryption and Authentication Overlay package is installed, the path can be represented in either form:

- **`/opt/sas/viya/home/SASEventStreamProcessingEngine/4.3.0/lib`**
- **`$DFESP_HOME/lib`**

SAS Event Stream Processing includes the internal component SAS Micro Analytic Service. To use the Anaconda Python support in SAS Micro Analytic Service, you need to set additional variables for your version of

Python. For instructions, see *SAS Micro Analytic Service: Programming and Administration Guide*, which is available on the SAS Event Stream Processing product page.

Depending on the shell environment that you use, you can also add these export commands to your .bashrc file or .profile file, which updates the settings automatically. Another option is to create a configuration shell script and to copy it to your **/etc/profile.d** directory.

Run the Deployment Script

When you order SAS software, SAS sends a Software Order Email (SOE) to your business or organization. Your SOE includes information about the software order, including several file attachments. The following files are required for deployment:

- the license file
- certificates that enable access to your software
- the customized_deployment_script.sh file, which contains customized commands that are required for accessing and downloading software from SAS repositories

Perform the following steps to install all SAS Event Stream Processing components on the same machine:

1. As instructed in the SOE, save the attached files to a directory on the machine where you plan to launch your deployment.

2. (Optional) If you use FTP to move the files from the SOE to their final location, the customized_deployment_script.sh file might lose its Execute bit. To ensure that the file has the required Execute bit, run the following command:

   ```
   sudo chmod +x customized_deployment_script.sh
   ```

3. Run the script:

   ```
   sudo ./customized_deployment_script.sh
   ```

4. (Optional) When prompted, enter **y** to install the product user interface component, SAS Event Stream Processing Studio (with filename sas-espvm).

   Note: SAS Event Stream Processing Studio can be installed on a separate machine from the SAS Event Stream Processing server.

   Additional steps are required to configure SAS Event Stream Processing Studio. For more information, see "(Optional) Configure SAS Event Stream Processing Studio " on page 13.

5. (Optional) When prompted, enter **y** to install the Streamviewer component (with filename sas-espstrmvwr).

   Note: Streamviewer can be installed on a separate machine from SAS Event Stream Processing.

   Additional steps are required to configure Streamviewer. For more information, see Setting Up and Running Streamviewer.

6. (Optional) When prompted, enter **y** to install the SAS Text Analytics component (with filename sas-txtmineng).

7. (Optional) If you purchased the optional SAS Event Stream Manager product, you should install an agent. When prompted, enter **y** to install the SAS Event Stream Manager agent component (with filename sas-esm-agent).

   When the script completes, a message is displayed to indicate that the software has been installed. For more information, see "Installing the SAS Event Stream Manager Agent" on page 27.

   A valid license file is required in order to run any applications that use SAS Event Stream Processing. The installation copies the license file to the required location on the target machine. If this step fails to complete, the following message is displayed:

```
License file could not be located and will have to be copied manually.
```

8 To copy the license file manually, locate the license file that is in TXT format. Your SOE included this file as an attachment and instructed you to save it.

9 Copy the license file to the default license directory. For *license*.txt, substitute the filename of the license file that you received from SAS:

```
sudo cp license.txt /opt/sas/viya/home/SASEventStreamProcessingEngine/4.3.0/etc/license/license.txt
```

4

Post-Installation Configuration

(Optional) Configure SAS Event Stream Processing Studio

SAS Event Stream Processing Studio is an optional user interface that generates XML code based on the visual models that you create. It is not automatically started during the installation. This section describes the additional steps that are required to use SAS Event Stream Processing Studio.

Start SAS Event Stream Processing Studio

SAS Event Stream Processing Studio provides a visual interface for creating models. It is not automatically started during the installation.

1 SAS Event Stream Processing Studio requires Java 1.8. If Java 1.8 is not the default version of Java on your system, update the following script to set the SAS_JAVA_HOME environment variable:

```
/opt/sas/viya/config/etc/sysconfig/sas-javaesntl/sas-java
```

Here is an example:

```
SAS_JAVA_HOME=/usr/java/jdk1.8.0_101/jre
```

Or supply the location of the JDK, if applicable. For example:

```
SAS_JAVA_HOME=/usr/java/jdk1.8.0_101
```

Note: Do not include the **/bin/java** portion of the path for the definition of SAS_JAVA_HOME.

2 To start SAS Event Stream Processing Studio, run the following command:

```
sudo service sas-viya-espvm-default start
```

3 After you have started the service, you can access SAS Event Stream Processing Studio using a web browser that is running on Windows or Linux. Open SAS Event Stream Processing Studio from a URL with the following format:

http://*esp-studio-hostname*:*port*/SASEventStreamProcessingStudio

Note: For *esp-studio-hostname* and *port*, specify values that are appropriate for your deployment. The default port is 8080. For information about changing the default port, see "(Optional) Change the Default Port" on page 14.

4 Before you can open or create a model in SAS Event Stream Processing Studio, you must start the Event Stream Processing XML server. Change directories to the following location:

```
cd /opt/sas/viya/home/SASEventStreamProcessingEngine/4.3.0/bin
```

5 Run the following command:

```
dfesp_xml_server -pubsub n -http-admin adminport -http-pubsub pubsubport &
```

The `-pubsub` argument specifies a port for publish and subscribe actions. Replace *n* with the appropriate port number.

The `-http-admin` argument runs the Event Stream Processing server as a factory server that supports the creation of projects. For *adminport*, specify the port that you want to use for HTTP administration requests.

The ampersand (&) enables additional commands to be entered in the same window that started the server.

Note: If you have a project that is predefined, use the `-model` argument to run the project as a stand-alone engine.

The `-http-pubsub` argument sets up a publish/subscribe HTTP server that uses the specified port *pubsubport*. For more information about the Event Stream Processing server, see *SAS Event Stream Processing: Using the XML Layer*.

6 The following INFO message is displayed:

```
Access control disabled (could not open permissions.yml, error: file not found)
```

The file that is referenced is required only to enable access control on the Event Stream Processing server. You can ignore this message.

7 (Optional) To check the status of SAS Event Stream Processing Studio, run the following command:

```
sudo service sas-viya-espvm-default status
```

(Optional) Change the Default Port

You can change the port settings for SAS Event Stream Processing Studio. The default port, 8080, is appropriate for most environments.

1 Use your preferred text editor to open and edit the following file:

```
sudo vi /opt/sas/viya/home/bin/sas-espvm
```

2 Locate the following line in the file:

```
export java_option_server_port="-Dserver.port=8080"
```

3 Change the default port, 8080, to the appropriate port.

4 Save and close the file.

5 Restart the espvm service:

```
sudo service sas-viya-espvm-default stop
sudo service sas-viya-espvm-default start
```

Directory Structure and Permissions

After you install SAS Event Stream Processing, the files for the engine, the user interface components, and the authentication package are located in the following directory:

```
/opt/sas/viya/home/SASEventStreamProcessingEngine/
```

Configuration files for adapters and logs are located in the following directory:

```
/opt/sas/viya/config/etc/SASEventStreamProcessingEngine/default
```

Later, if you update your deployment, the configuration files are not altered.

The basic directory path enables write access per user group, and it is owned by the sas user. To grant permission to users to edit the configuration files, the administrator must add them to the sas group.

(Optional) Enable Database Connectivity

To enable database connectivity, perform these configuration tasks after installation has been completed:

1. Use your preferred text editor to open and edit the following file:

   ```
   /opt/sas/viya/home/SASEventStreamProcessingEngine/4.3.0/etc/odbc.ini.template
   ```

2. Search for and replace all instances of $DFESP_HOME with the actual installation location of the SAS Event Stream Processing engine.

   This step is required to enable database drivers to locate the connector software. Here is an example of the pathname that is used to replace $DFESP_HOME:

   ```
   /opt/sas/viya/home/SASEventStreamProcessingEngine/4.3.0
   ```

3. Save the file with filename odbc.ini. Close the file.

5

Validating the Deployment

Verify the RPM Packages

To obtain a list of all SAS Event Stream Processing RPM packages that are deployed on your system, run the following command:

```
rpm -qa sas-esp*
```

Then you can run this basic command to verify an individual RPM package from the list that is returned:

```
rpm -Vv package-name
```

The full name of each RPM is not required. For example, to verify the contents of the sas-espexam-4.3.0-20170109.060004533214.x86_64 package, run the following command:

```
rpm -Vv sas-espbase
```

Note: Run the preceding commands for each host on which you have deployed SAS Event Stream Processing and its optional web application components.

You can also create a for loop command for verifying multiple packages that share a common naming convention. For example, to verify all packages whose names begin with `sas-`, use the following query:

```
for i in $(rpm -qa | grep -e "^sas-");do rpm -Vv $i;done
```

A successful verification shows the list of files that make up the RPM and with no error indicators, as follows:

```
rpm -Vv sas-espexam
......... /opt/sas/viya/home/lib/esp/sas-init-functions
```

An unsuccessful verification provides error indicators beside the filename. Here is an example:

```
rpm -Vv sas-espexam
package sas-espexam is not installed
```

The error indicators are shown in the following format:

```
SM5DLUGT c
```

In addition, if a file is missing, the error message contains the word “missing”:

```
missing  /opt/sas/viya/home/lib/esp/sas-init-functions
```

The meaning of each error indicator is described as follows:

- S - file size

RPM keeps track of file sizes. A difference of even one byte triggers a verification error.

- M - file mode

  The permissions mode is a set of bits that specifies access for the file's owner, group members, and others. Even more important are two additional bits that determine whether a user's group or user ID should be changed if they execute the program that is contained in the file. Since these bits permit any user to become root for the duration of the program, you must be cautious with a file's permissions.

- 5 - MD5 checksum

  The MD5 checksum of a file is a 128-bit number that is mathematically derived from the contents of the file. The MD5 checksum conveys no information about the contents of the original file, but, any change to the file results in a change to the MD5 checksum. RPM creates MD5 checksums for all files that it manipulates, and stores the checksums in its database. If one of these files is changed, the MD5 checksum changes and the change is detected by RPM.

- D - major and minor numbers

  Device character and block files contain a major number. The major number is used to communicate information to the device driver that is associated with the special file. For example, under Linux, the special files for SCSI disk drives should have a major number of 8, and the major number for an IDE disk drive's special file should be 3. Any change to a file's major number could produce disastrous effects. RPM tracks such changes.

  A file's minor number is similar to the major number, but conveys different information to the device driver. For disk drives, this information can consist of a unit identifier.

- L - symbolic link

  If a file is a symbolic link, RPM checks the text string that contains the name of the symbolically linked file.

- U - file owner

  Most operating systems keep track of each file's creator, primarily for resource accounting. Linux and UNIX also use file ownership to help determine access rights to the file. In addition, some files, when executed by a user, can temporarily change the user's ID, normally to a more privileged ID. Therefore, any change of file ownership might have significant effects on data security and system availability.

- G - file group

  Similar to file ownership, a group specification is attached to each file. Primarily used for determining access rights, a file's group specification can also become a user's group ID if that user executes the file's contents. Therefore, any changes in a file's group specification are important and should be monitored.

- T - modification time

  Most operating systems keep track of the date and time that a file was last modified. RPM keeps modification times in its database.

- c - configuration file

  This is useful for quickly identifying configuration files, since they are likely to change and therefore are unlikely to verify successfully.

Error Indicators

RPM verification produces various error indicators. Some of them are innocuous.

You might see the following error indicators when you perform RPM verification:

- S - file size

RPM keeps track of file sizes. A difference of even one byte triggers a verification error.

- M - file mode

  The permissions mode specifies access for the file's owner, group members, and others. Two additional bits determine whether a user's group or user ID should be changed if they execute the program that is contained in the file. These bits permit any user to become root for the duration of the program.

- 5 - MD5 checksum

  The MD5 checksum of a file is a 128-bit number that is mathematically derived from the contents of the file. The MD5 checksum conveys no information about the contents of the original file, but any change to the file results in a change to the checksum. RPM creates MD5 checksums for all files that it manipulates and stores them in its database. If one of these files is changed, the checksum changes and the change is detected by RPM.

- D - major and minor numbers

  Device character and block files contain a major number. The major number is used to communicate information to the device driver that is associated with the special file. For example, under Linux, the special files for SCSI disk drives should have a major number of 8. The major number for an IDE disk drive's special file should be 3.

  CAUTION! Any change to a file's major number could produce unwanted results. RPM tracks these changes.

  A file's minor number is similar to the major number, but conveys different information to the device driver. For disk drives, this information can consist of a unit identifier.

- L - symbolic link

  If a file is a symbolic link, RPM checks the text string that contains the name of the symbolically linked file.

- U - file owner

  Most operating systems keep track of each file's creator, primarily for resource accounting. Linux and UNIX also use file ownership to help determine access rights to the file. In addition, some files, when executed by a user, can temporarily change the user's ID, normally to a more privileged ID. Therefore, any change of file ownership might have significant effects on data security and system availability.

- G - file group

  Similar to file ownership, a group specification is attached to each file. Primarily used for determining access rights, a file's group specification can also become a user's group ID if that user executes the file's contents. Therefore, any changes in a file's group specification are important and should be monitored.

- M - modification time

  Most operating systems keep track of the date and time that a file was last modified. RPM keeps modification times in its database.

- c - configuration file

  This error indicator is useful for quickly identifying the specified configuration file.

View Deployment Logs

To view the logs of your yum deployment, run the following commands:

```
sudo yum history
sudo less /var/log/yum.log
```

6

Next Steps

Code Examples

Code examples to help you write programs are installed along with the software. You can find the examples in the following directory after the deployment has completed:

```
/opt/sas/viya/home/SASEventStreamProcessingEngine/4.3.0/examples/
```

The examples directory includes files for C++, XML, Python, and Java. It also includes a readme_examples.txt file, which briefly describes each example and its usage.

SAS recommends that you copy the examples that you require to a writable directory on the local computer so that you can run them.

Two documents are helpful in understanding the examples. You can find links on the SAS Event Stream Processing product page to the following user guides:

- *DataFlux® Expression Language Reference Guide*
- *SAS® Micro Analytic Service Programming and Administration Guide*

Product Documentation

After you install, configure, and verify the deployment, you are ready to begin writing applications that capture and analyze streaming event data in real time.

The next step is to consult the product documentation. The product documentation is included in SAS Help Center. A link to all SAS Event Stream Processing documentation is available on the SAS Event Stream Processing product page. All product user documentation is also available via single sign-on from the SAS Event Stream Processing user interfaces (SAS Event Stream Processing Studio and Streamviewer).

SAS recommends starting with *SAS Event Stream Processing 4.3: Overview*, which provides an introduction to product features and explains how to proceed with creating event stream processing models and incorporating them into applications.

If you have set up the optional Streamviewer component, you can find more information about it in a separate guide. For a full set of instructions about using Streamviewer, see Visualizing Event Streams with Streamviewer.

7

Uninstalling SAS Event Stream Processing

Uninstall SAS Event Stream Processing

Use yum to uninstall your SAS software:

1 Create a backup copy of the SAS Event Stream Processing Studio database in order to preserve project files. Follow these steps:

   a Stop the SAS Event Stream Processing Studio (espvm) service by running the following command:

   ```
   sudo service sas-viya-espvm-default stop
   ```

   b Create a backup copy of the database, which is a single binary file (studio.mv.db). You can copy it to any directory location outside the SAS Event Stream Processing installation directory structure.

   The location and filename of the database are determined by the environment variable ESP_STUDIO_DB. By default, it is stored in **`/opt/sas/viya/config/data/espvm/`**.

   To create the backup, run the following command:

   ```
   cp studio.mv.db directory-name
   ```

2 (Optional) If you installed Streamviewer, find the process ID so that you can kill the Streamviewer service:

```
ps -ef
```

3 (Optional) Kill the Streamviewer process, substituting the process ID that was returned in the previous step:

```
kill -9 process-id
```

4 Uninstall the SAS Event Stream Processing software:

```
sudo yum groupremove 'SAS Event Stream Processing'
```

5 Remove any additional products that were included in your SAS Event Stream Processing order:

```
sudo yum groupremove 'SAS Event Stream Processing Studio'

sudo yum groupremove 'SAS Event Stream Processing Streamviewer'

sudo yum groupremove 'SAS Event Stream Processing Analytics'

sudo yum groupremove 'SAS Text Analytics for English'
```

No files that are required for other SAS software deployments are removed unless you take additional steps. As an alternative, you can run the following command to remove all SAS products from the machine:

```
sudo yum remove sas-*
```

6 Remove the main repository definition:

```
sudo yum erase sas-meta-repo-1-1
```

7 Remove any remaining components:

```
   sudo rpm -e $(rpm -qg SAS)
```

8 Remove the entitlement certificate:

```
   sudo rm /etc/pki/sas/private/entitlement_certificate.pem
```

If you have questions about removing a specific file, you might want to consult a professional. For more information, see “Contact SAS Technical Support” on page 3.

8

Updating SAS Event Stream Processing

About Updates

A software update makes your deployed software up-to-date with the latest software. Updates are performed by running the same tools that you ran during the initial deployment. You might determine that your software needs to be updated, or you might be notified by SAS that updates are available.

The term *upgrade* is used to refer to a type of software update that introduces new functionality. At SAS, an upgrade generally involves a new release number. By contrast, an *update* refers to minor changes to the software such as fixes. A new Software Order Email (SOE) is not required in order to retrieve the updated software packages.

Applying Updates

You apply updates to the deployed software environment in order to bring the software to the latest version. For SAS Event Stream Processing, you can update with yum.

Update SAS Event Stream Processing on Linux

You can use yum to apply all available updates to SAS Event Stream Processing software on a selected machine.

Note: Optional components, such as SAS Event Stream Processing Studio, Streamviewer, and the SAS Event Stream Manager agent, are not affected by the update.

1 To update all SAS Viya software on the machine, run the following command:

```
sudo yum update "@SAS*" "sas-*"
```

2 At the prompt `Is this ok`, review the available updates and then enter `y`.

3 Repeat these steps on each machine where you have installed SAS Event Stream Processing.

Appendix 1

Installing the SAS Event Stream Manager Agent

(Optional) SAS Event Stream Manager Agent Configuration

The SAS Event Stream Manager Agent is a small executable program that is installed along with SAS Event Stream Processing. Agents relay operational metrics from engines to SAS Event Stream Manager and perform actions on the engines in response to commands that they receive from SAS Event Stream Manager. Agents are optional, but configuring agents is a recommended best practice for SAS Event Stream Manager users, and they are required to support a few features, such as starting and stopping adapters.

Agent Restrictions

The installation of a SAS Event Stream Manager agent is restricted to one agent per machine. If you are running multiple SAS Event Stream Processing engines on a single machine, install a single agent on that machine. You can then select one engine instance to manage on that machine.

You can also change the agent configuration to select another engine instance at a later time. Perform the steps that are listed in "Configure and Start an Agent" on page 28. For the ESM_ESP_HTTP_PUBSUB and ESM_ESP_HTTP_ADMIN variables, substitute values that correspond to the engine that you want to manage.

Install an Agent

The typical SAS Event Stream Processing installation on Linux includes a prompt to install an agent. The agent must be installed on the same machine where SAS Event Stream Processing has been installed. First, determine whether the agent has been installed on the target machine.

The agent is contained in an RPM called sas-esmagent. You can verify that the RPM is installed by running the following command:

```
sudo service sas-viya-esmagent-default status
```

If the agent is already installed, a message is displayed stating that `sas-viya-esmagent-default is stopped` or `sas-viya-esmagent-default is started`.

If the agent is not installed, a `Not found` error message is displayed. Take the following steps to install it:

1 Using an account with sudoers privileges, log on to the Linux machine where you have installed SAS Event Stream Processing.

2 Start the installation again by running the following command:

```
sudo ./customized_deployment_script.sh
```

3 When prompted, enter y to install the SAS Event Stream Manager agent component (with filename sas-esm-agent).

When the script completes, a message states that the software has been installed.

Configure and Start an Agent

Take the following steps to modify agent parameters:

1 Stop the agent. Run the following command:

```
sudo service sas-viya-esmagent-default stop
```

2 Edit the start-up script to set the correct values for some environment variables. Use your preferred text editor to open the following file for editing: **/opt/sas/viya/home/bin/sas-esmagent**.

3 Locate the following environment variables within the start-up script. Set their values to environment-specific values, as specified in the following table:

Variable	Environment-Specific Value
ESM_DISCOVERY_HOST	Host name of the machine where you have installed SAS Event Stream Manager.
ESM_DISCOVERY_PORT	The port where SAS Event Stream Manager is listening for communications from the agent. The default is Port 80.
ESM_HOSTNAME	The host name of the machine where you have installed SAS Event Stream Manager Agent and SAS Event Stream Processing server. (These components must be installed on the same machine.)
ESM_PORT	The port where the agent listens. The default setting is Port 2552.
ESM_FRIENDLY_NAME	The name of the agent that appears in the user interface of SAS Event Stream Manager. The default setting is “ESM Agent.”
ESM_ESP_HTTP_PUBSUB	The HTTP port that has been configured for publish and subscribe actions by the Event Stream Processing XML server.
ESM_ESP_HTTP_ADMIN	The HTTP port that has been configured for HTTP administration requests to the Event Stream Processing XML server.

For more information about XML server parameters, see *SAS Event Stream Processing: Using the XML Layer*.

4 Save your changes to the start-up script.

5 Verify that SAS Event Stream Processing is running. Run the following command:

```
   ps -fu$USER | grep dfexp_xml_server
```

The status that is returned indicates whether the server process is running.

6 Start the agent. Run the following command:

```
sudo service sas-viya-esmagent-default start
```

The following message indicates success: `sas-viya-esmagent-default is running`.

Appendix 2

Creating and Using a Yum Mirror Repository

Overview

By default, SAS downloads and installs the latest software available from the applicable software repositories. If your deployment does not have access to the Internet or if you must always deploy the same version of software (such as for regulatory reasons, or to preserve a testing environment), you can create and use a mirror repository for the deployment.

Prepare the Yum Repository

The topics in this section explain how to deploy on a Red Hat Enterprise Linux platform using a yum repository. Follow these steps to create a local copy of the secure SAS repositories.

Create and Use the setup_repos.sh File

To prepare the target host where you will create a local copy of the SAS software repository, copy the custom shell script file that was attached to your SOE to a directory where you can edit it.

Note: Some of the steps in this procedure require you to log on as the root user.

1 Copy the customized deployment script (customized_deployment_script.sh) and change its name to setup_repos.sh:

```
cp customized_deployment_script.sh setup_repos.sh
```

You now have two shell scripts. Preserve the original script, customized_deployment_script.sh, in case you perform a conventional installation of the software at a later time.

2 Use your preferred text editor to open setup_repos.sh.

3 Comment out the `Install` statement in setup_repos.sh by adding a number sign (#) to the beginning of the line. Here is an example:

```
# Install the software
# yum groupinstall "SAS Event Stream Processing"
```

4 Save and close setup_repos.sh.

5 As root, run setup_repos.sh to configure the required certificates and to locate the SAS hosted repositories. Running the script enables the repository mirror host to mirror the content.

Complete the Setup of the Local Repository

To create the local copy of the SAS repository:

1 Install the required yum utilities package by running the following command:

```
sudo yum install createrepo yum-utils
```

2 Using a text editor, create a new file named createrepos.sh that contains the following content:

```
#!/bin/bash
rpm --import /etc/pki/sas/rpm-gpg/RPM-GPG-KEY-sas
mkdir -p /var/www/html/pulp/repos
for f in $(ls /etc/yum.repos.d/sas-*.repo | cut -f4 -d/ | sed s/.repo//g | grep -v sas-meta)
do
  reposync --gpgcheck -l -n --repoid=${f} \
           --download_path=/var/www/html/pulp/repos --downloadcomps --download-metadata
  cd /var/www/html/pulp/repos/${f}
  createrepo -v /var/www/html/pulp/repos/${f}/ -g comps.xml
done
```

3 Set the Execute bit for createrepos.sh:

```
sudo chmod +x createrepos.sh
```

4 Run createrepos.sh:

```
sudo ./createrepos.sh
```

Deploy the Software

Perform these steps to install the SAS Event Stream Processing components on the machine where you have created a mirror repository.

The software order email (SOE) that you received from SAS directs you to save the attached files to a directory on the host from which you will perform the installation. You must save those files before performing any steps in this section.

1 Follow the internal processes at your organization to set up a hosted repository from the local copy of the SAS repository that you have downloaded. For example, some organizations configure a web server using Apache httpd to serve this content.

2 After you have set up a hosted repository, use the procedures that are appropriate for your environment to install all of the RPMs that are available in your hosted repository.

3 Apply the license file. For more information, see “Apply the License ” on page 33.

Complete the Installation

The procedures in the following topics are required to complete the installation on all platforms.

Apply the License

A valid license file is required in order to run any applications that use SAS Event Stream Processing.

Your SOE contained a license file that you were instructed to save. Now you must apply the license file to the local machine by saving it to the default license directory.

1 Locate the license file that you previously saved.

2 Copy the license file to the default license directory.

   Substitute the actual name of the license file in the following command:

   ```
   sudo cp license-filename /opt/sas/viya/home/SASEventStreamProcessingEngine/4.3.0/etc/license
   ```

Start the Software

Before you try to start SAS Event Stream Processing, be sure to set the required environment variables. For more information, see “Set the Environment Variables” on page 9.

Take a few more steps to start the SAS processes and verify that the software is running.

1 On the machine where you installed SAS Event Stream Processing, run the following command to start the Event Stream Processing XML server:

   ```
   $DFESP_HOME/bin/dfesp_xml_server -http-admin 9900
   ```

   The `-http-admin` argument runs the Event Stream Processing server as a factory server that supports the creation of projects. Port 9900 in the example corresponds to the adminport parameter, which designates the port that is used for HTTP administration requests.

2 Validate the deployment. For more information, see “Validating the Deployment” on page 17.

3 Perform any required steps that apply to your environment in the chapter titled “Post-Installation Configuration” on page 13.