Micro Focus Security
ArcSight Kafka FlexConnector

FlexConnector for Kafka

Configuration Guide

Document Release Date: July 2022

Software Release Date: July 2022

Legal Notices

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK

https://www.microfocus.com

Copyright Notice

Confidential computer software. Valid license from Micro Focus required for possession, use or copying. The information contained herein is subject to change without notice.

The only warranties for Micro Focus products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein.

No portion of this product's documentation may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's internal use, without the express written permission of Micro Focus.

Notwithstanding anything to the contrary in your license agreement for Micro Focus ArcSight software, you may reverse engineer and modify certain open source components of the software in accordance with the license terms for those particular components. See below for the applicable terms.

U.S. Governmental Rights. For purposes of your license to Micro Focus ArcSight software, “commercial computer software” is defined at FAR 2.101. If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation and other technical data subject to the terms of the Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulation (“FAR”) and its successors. If acquired by or on behalf of any agency within the Department of Defense (“DOD”), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of the Agreement as specified in 48 C.F.R. 227.7202-3 of the DOD FAR Supplement (“DFARS”) and its successors. This U.S. Government Rights Section 18.11 is in lieu of, and supersedes, any other FAR, DFARS, or other clause or provision that addresses government rights in computer software or technical data.

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S.Government rights, patent policy, and FIPS compliance, see https://www.microfocus.com/about/legal/.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated
- Software Release Date, which indicates the release date of this version of the software

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

https://www.microfocus.com/support-and-services/documentation

Support

Contact Information

Phone	A list of phone numbers is available on the Technical Support Page: https://softwaresupport.softwaregrp.com/support-contact-information
Support Web Site	https://softwaresupport.softwaregrp.com/
ArcSight Product Documentation	https://www.microfocus.com/documentation/arcsight/

Revision History

Date	Description
11/23/2020	Added support to Azure Monitor Event Hub.
07/24/2019	First edition of this guide

Contents

Configuration Guide for ArcSight Kafka FlexConnector

This guide provides information about installing the ArcSight Kafka FlexConnector and configuring the device for event collection.

The Arcsight Kafka FlexConnector helps you subscribe and collect events from a topic of a Kafka server or Azure Event Hubs for Kafka. Topics only contain a specific event type.

Intended Audience

This guide provides information for IT administrators who are responsible for managing the ArcSight SmartConnectors.

Additional Documentation

The ArcSight SmartConnectors documentation library includes the following resources:

- *Installation Guide for ArcSight SmartConnectors*, which provides detailed information about installing SmartConnectors.
- *Configuration Guides for ArcSight SmartConnectors*, which provides information about configuring SmartConnectors to collect events from different sources.
- *Release Notes for ArcSight SmartConnectors*, which provides information about the latest release

For the most recent version of this guide and other ArcSight SmartConnector documentation resources, visit the documentation site for ArcSight SmartConnectors.

Contact Information

We want to hear your comments and suggestions about this book and the other documentation included with this product. You can use the comment on this topic link at the bottom of each page of the online documentation, or send an email to Documentation-Feedback@microfocus.com.

For specific product issues, contact Micro Focus Customer Care.

Product Overview

Apache Kafka is a community distributed event streaming platform capable of handling trillions of events a day. Apache Kafka is a distributed data store optimized for ingesting and processing streaming data in real-time. Streaming data is data that is continuously generated by thousands of data sources, which typically send the data records in simultaneously. A streaming platform needs to handle this constant influx of data, and process the data sequentially and incrementally.

Kafka is suitable for both offline and online message consumption.

Kafka messages are persisted on the disk and replicated within the cluster to prevent data loss. it is built on top of the ZooKeeper synchronization service and it integrates very well with Apache \Storm and Spark for real-time streaming data analysis.

For more information about Apache Kafka, refer to the Kafka documentaion.

Understanding Kafka Architecture

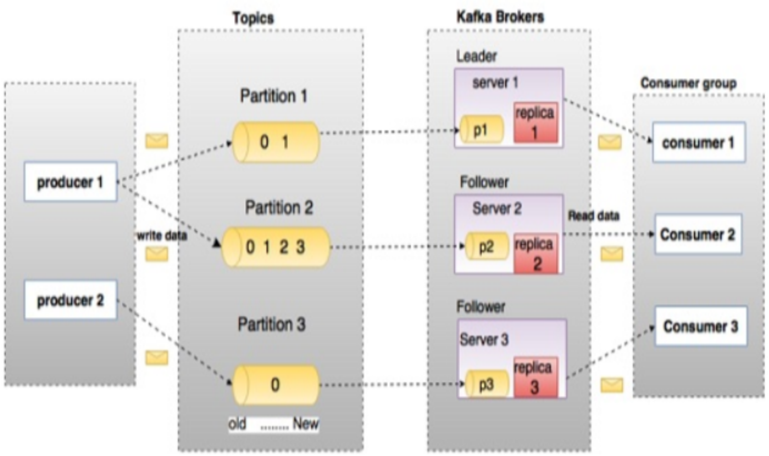
Apache Kafka is a distributed publish-subscribe messaging system and a robust queue that handles a high volume of data and enables you to pass messages from one end-point to another.

The following diagram illustrates the main terminologies and the table describes the diagram components in detail. A topic is configured into three partitions:

- Partition 1 has two offset factors 0 and 1.
- Partition 2 has four offset factors 0, 1, 2, and 3.
- Partition 3 has one offset factor 0.

The Id of the replica is same as the Id of the server that hosts it.

Assume, you want to install 3 Kafka Flex Connectors to parse data of a topic, you need to increase the partitions of your Kafka server.



For more information about Apache Kafka, see Configuring Apache Kafka in Windows or Linux Platforms

Supported Event Sources

This version supports the following event types:

- JSON
- CEF
- REGEX
- SYSLOG
- KEY-VALUE
- AVRO

Because this is a FlexConnector, you need to create your individual parsers before setting up the connector.

Note: The connector consumes only data that is serialized using the correct serializers, because the corresponding deserializers are not configurable.

- **To serialize data, use the following serializers:**
  - Key serializer: `org.apache.kafka.common.serialization.StringSerializer`
  - Value serializer: `org.apache.kafka.common.serialization.BytesSerializer`
- **To deserialize data, the Kafka FlexConnector uses the following deserializers:**
  - Key deserializer: `org.apache.kafka.common.serialization.StringDeserializer`
  - Value deserializer: `org.apache.kafka.common.serialization.BytesDeserializer`

Understanding Azure Event Hubs for Kafka

Azure Event Hubs provides a Kafka endpoint that can be used by your Kafka based applications as an alternative to run Kafka clusters. Azure Event Hubs supports Apache Kafka protocol 1.0 and later; however, it does not support the AMQP and HTTPS protocols.

The Kafka endpoint can be used from applications with just a minimal configuration change:

- Update the connection string in the configurations to point to the Kafka endpoint exposed by your event hub instead of pointing to a Kafka cluster and start streaming events from the applications that use the Kafka protocol into Event Hubs.
- This integration also supports frameworks like Kafka Connect, which is currently in preview.

Every time events are published or consumed from Event Hubs for Kafka, your clients are trying to access the Event Hubs resources. Ensure that the resources are accessed with an authorized entity. When using Apache Kafka protocol with your clients, set the configuration for authentication and encryption using the SASL mechanisms. Event Hubs for Kafka require the TLS-encryption (as all data in transit with Event Hubs is TLS encrypted). This can be done by specifying the SASL_SSL option in the configuration file.

The Arcsight Kafka FlexConnector uses Shared Access Signature (SAS) to authorize access to secure resources.

For more information, see the Azure Documentation.

Configuring Authentication

The Arcsight Kafka FlexConnector provides secure connection to Kafka servers.

This section contains the following information:

Enabling SSL Encryption and Authentication

1. Configure the truststore, keystore, and password in the server.properties file of every broker.
2. Passwords are directly stored in the broker configuration file, so restrict the access to these files via file system permissions

```
ssl.truststore.location=/var/private/ssl/kafka.server.truststore.jks
ssl.truststore.password=test1234
ssl.keystore.location=/var/private/ssl/kafka.server.keystore.jks
ssl.keystore.password=test1234
ssl.key.password=test1234
```

> **Note**: `ssl.truststore`.password is optional but highly recommended. If a password is not set, access to the truststore is still available, but integrity checking is disabled.

Enabling SSL Inter Broker Communication Configuring

1. Add the following property to the broker properties file (it is `PLAINTEXT` by default).

   ```
   security.inter.broker.protocol=SSL
   ```

2. Configure the Apache Kafka® broker ports which listen to client and inter-broker `SSL` connections. Configure the `listeners` and the `advertised.listeners`, in case the value is different.

   ```
   listeners=SSL://kafka1:9093
   advertised.listeners=SSL://0.0.0.0:9093
   ```

3. Configure the `PLAINTEXT` ports if:

   - SSL is not enabled for inter-broker communication.
   - Some clients connecting to the cluster do not use SSL.

     ```
     listeners=PLAINTEXT://kafka1:9092,SSL://kafka1:9093
     advertised.listeners=PLAINTEXT://0.0.0.0:9092,SSL://0.0.0.0:9093
     ```

     > **Note**: `advertised.host.name` and `advertised.port` configure a single `PLAINTEXT` port are incompatible with secure protocols. Use `advertised.listeners` instead.

4. To enable the broker to authenticate clients (2-way authentication), you need to configure all the brokers for client authentication. We recommend setting this value to `required`.

   ```
   ssl.client.auth=required
   ```

   > **Note**: Do not use `requested` as it creates a false sense of security.

> **Important**: If any of the SASL authentication mechanisms are enabled on a given listener, the SSL client authentication is disabled, even if `ssl.client.auth=required` is previously configured. The broker will only authenticate clients via SASL on that listener.

Shared Access Signature

Event Hubs also provide the Shared Access Signatures (SAS) for delegated access to Event Hubs for Kafka resources. Authorizing access with an OAuth 2.0 token-based mechanism provides superior security and ease of use over SAS. The built-in roles can also eliminate the need for ACL-based authorization, which has to be maintained and managed by the user.

This feature can be used with your Kafka clients by specifying the SASL_SSL for the protocol and PLAIN for the mechanism, as shown in the following example:

```
bootstrap.servers=NAMESPACENAME.servicebus.windows.net:9093

security.protocol=SASL_SSL

sasl.mechanism=PLAIN

sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule
required username="$ConnectionString" password="
{YOUR.EVENTHUBS.CONNECTION.STRING}";
```

> **Note** : When using SAS authentication with Kafka clients, established connections are not disconnected after the SAS key is regenerated.

Creating Flex Parsers

To create a flex Parser, see the ArcSight FlexConnector Developer's Guide

Installing the FlexConnector

This topic includes the following sections:

Preparing to Install the FlexConnector

Before you install any SmartConnectors, make sure that the Micro Focus ArcSight products with which the connectors will communicate have already been installed correctly (such as ArcSight ESM or ArcSight Logger).

For complete product information, refer to the *Administrator's Guide to ArcSight Platform*, available on ArcSight Documentation.

If you are adding a connector to the ArcSight Management Center, see the *ArcSight Management Center Administrator's Guide* available on ArcSight Documentation for instructions.

Before installing the FlexConnector, make sure that the following are available:

- Local access to the machine where the FlexConnector is to be installed.
- Vendor login credentials (user name and password). During the configuration, you are redirected to the vendor’s login page, where you will log into the vendor’s application using your vendor credentials. After you log into the vendor application, the connector can access and collect vendor log data.

Unless specified otherwise at the beginning of this guide, this connector can be installed on all ArcSight supported platforms; for the complete list, see the Technical Requirements for SmartConnectors document.

> **Note:** On the Linux platform, if you are logged in as `root` and use Firefox, some versions of the browser can cause the browser launched by the connector during configuration not to open. If you see this issue, try configuring the connector as a non-root user. If you configure the connector as a non-root user, however, you cannot run the connector as a service.

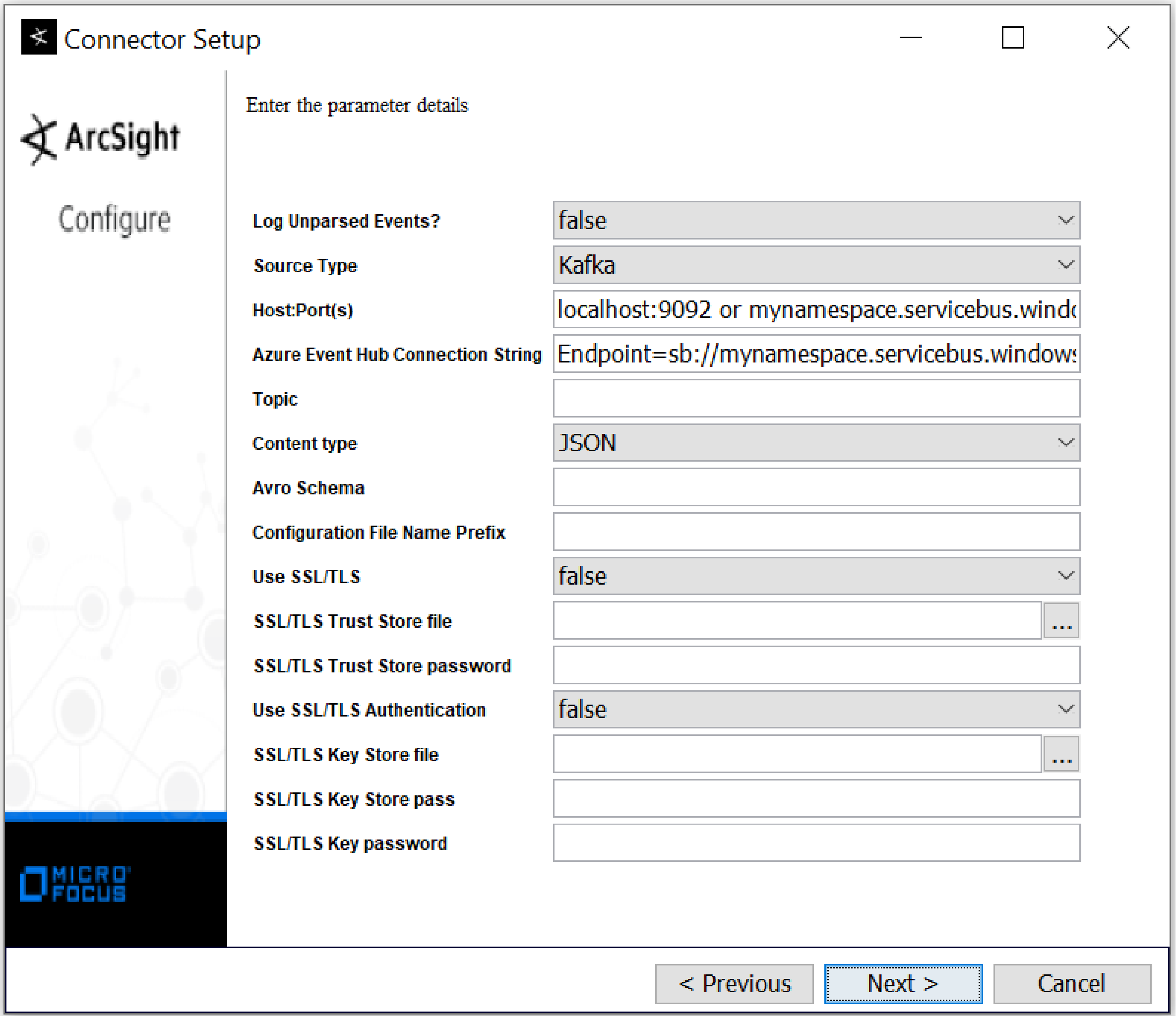
Adding JSON Parser

1. Before configuring the connector, you must exit the wizard to make your JSON parser available to the connector. Click **Cancel** to exit the wizard.
2. Copy your JSON parser file into the `$ARCSIGHT_HOME\current\user\agent\flexagent` directory. See "Create a JSON Parser File" for details on creating the JSON parser file.
3. Execute `runagentsetup` from `$ARCSIGHT_HOME\current\bin` to return to the wizard.

Installing and Configuring the FlexConnector

Complete the following steps to install the connector:

1. Start the installation wizard.
2. Follow the instructions in the wizard to install the core software.
3. Exit the installation wizard.
4. Copy the parser configuration file to the `ARCSIGHT_HOME\user\agent\flexagent` folder. For more information about the specific parser file locations, see the Parser File Locations and Names section in Developer's Guide to FlexConnectors.
5. Browse to `$ARCSIGHT_HOME/current/bin`, then double-click **runagentsetup.bat** file to start the SmartConnector Configuration Wizard.
6. Specify the relevant Global Parameters, when prompted.
7. From the **Type** drop-down menu, select **ArcSight FlexConnector Kafka** and click **Next**.
8. Enter the parameter details and click **Next**.



Connector Setup

Enter the parameter details

Log Unparsed Events?	false
Source Type	Kafka
Host:Port(s)	localhost:9092 or mynamespace.servicebus.windo
Azure Event Hub Connection String	Endpoint=sb://mynamespace.servicebus.window:
Topic	
Content type	JSON
Avro Schema	
Configuration File Name Prefix	
Use SSL/TLS	false
SSL/TLS Trust Store file	...
SSL/TLS Trust Store password	
Use SSL/TLS Authentication	false
SSL/TLS Key Store file	...
SSL/TLS Key Store pass	
SSL/TLS Key password	

To enable SASL plain authentication, set the parameters as follows:

Parameter	Setting
Log Unparsed Events	Select the value as required.
Source Type	Select **Azure Event Hub** to read data from an Azure Event hub.
Host:Port(s)	Enter host and port of the Kafka server.
Azure Event Hub Connection String	Enter connection string in the Azure Event Hub.
Topic	Enter topic name of the Kafka server as **Namespace**.
Content Type	The supported event types are: **JSON**, **CEF**, **SYSLOG**, **REGEX**, **KEY-VALUE**, and **AVRO**. If you select the **AVRO** content type, then you must specify a JSON configuration file name.
Avro Schema	This parameter is applicable only for the **AVRO** content type.

Parameter	Setting
Configuration File Name Prefix	Enter the name of the parser file once you have ensured the parser file has been copied into the `$ARCSIGHT_ HOME\current\user\agent\flexagent\syslog` directory for SYSLOG parser or `$ARCSIGHT_HOME\current\user\agent\flexagent` directory for the remaining event types. For example: for `$ARCSIGHT_HOME\current\user\agent\flexagent\ google.jsonparser.properties`. You can enter the prefix google, and the connector assumes the file name is `google.jsonparser.properties` and resides in `$ARCSIGHT_ HOME\current\user\agent\flexagent`. For more information, see Developer's Guide to FlexConnectors.
Use SSL/TLS	Select **false** from the drop-down list.
SSL/TLS Trust Store file	Leave it blank
SSL/TLS Trust Store password	Leave it blank
Use SSL/TLS Authentication	Select **false** from the drop-down list.
SSL/TLS Key Store file	Leave it blank
SSL/TLS Key Store pass	Leave it blank
SSL/TLS Key password	Leave it blank

Default parameters:

Parameter	Setting
Log Unparsed Events	Log unparsed events on the log folder and only use it for regex and syslog content types.
Source Type	Select **Kafka** to read data from a Kafka topic. Select **Azure Event Hub** to read data from an Azure Event hub.
Host:Port(s)	Enter host and port of the Kafka server.
Azure Event Hub Connection String	Enter connection string in the Azure Event Hub.
Topic	Enter topic name of the Kafka server.
Content Type	Select content type from the drop-down list. The supported content types are: **JSON**, **CEF**, **SYSLOG**, **REGEX**, **KEY-VALUE**, and **AVRO**. If you select the **AVRO** content type, then you must specify a JSON configuration file name.
Avro Schema	Enter a schema file name with full path and file extension (For example: `/opt/TestSchema.avsc`). This parameter is applicable only for the **AVRO** content type. Note that this must be the same schema that was used while writing the security events to Kafka topic.

Parameter	Setting
Configuration File Name Prefix	Enter the name of the parser file once you have ensured the parser file has been copied into the `$ARCSIGHT_HOME\current\user\agent\flexagent\syslog` directory for SYSLOG parser or `$ARCSIGHT_HOME\current\user\agent\flexagent` directory for the remaining event types. For example: for `$ARCSIGHT_HOME\current\user\agent\flexagent\google.jsonparser.properties`. You can enter the prefix google, and the connector assumes the file name is `google.jsonparser.properties` and resides in `$ARCSIGHT_ HOME\current\user\agent\flexagent`. For more information, see Developer's Guide to FlexConnectors.
Use SSL/TLS	Select **true** from the drop-down list if the Kafka server requires it for encrypted data.
SSL/TLS Trust Store file	Enter file path of the SSL/TLS Trust Store file.
SSL/TLS Trust Store password	Enter the **SSL/TLS Trust Store** password of the store file above.
Use SSL/TLS Authentication	Select **true** from the drop-down list if the Kafka server requires it for authentication. You also need to enable the **Use SSL/TLS** parameter.
SSL/TLS Key Store file	Enter the file path of theSSL/TLS Key Store file.
SSL/TLS Key Store pass	Enter the SSL/TLS Key Store password.
SSL/TLS Key password	Enter the SSL/TLS Key password.

9. Select a destination and configure parameters.
10. Specify a name for the connector.
11. If you have selected ArcSight Manager as the destination, the certificate import window for the ArcSight Manager is displayed. Select **Import the certificate to the connector from destination** and click **Next**. (If you select **Do not import the certificate to connector from destination**, the connector installation will end.) The certificate is imported and the **Add connector Summary** window is displayed.
12. Select whether you want to run the connector as a service or in the standalone mode.
13. Complete the installation.
14. Run the SmartConnector.

For instructions about upgrading the connector or modifying parameters, see SmartConnector Installation and User Guide.

Configuring Advanced Parameters

If you choose to perform any of the operations shown in the following table, do so before adding your connector. After installing core software, you can set the following parameters in the **agent.properties** file, as required:

Parameter	Setting
`bootstrap.servers`	Host-IP.
`group.id`	Use for multiple connectors in a Kafka topic.
`max.poll.records`	The maximum number of records returned in a single call to a poll( ). Default value is 500 (maximum).
`auto.commit.interval.ms`	The frequency in milliseconds in which the consumer offsets are auto-committed to Kafka if the `enable.auto.commit` value is set to **True**: 5000 miliseconds.
`reconnect.backoff.ms`	The base waiting time, before attempting to reconnect to a given host. It avoids repeatedly connecting to a host in a tight loop. This backoff applies to all client connection attempts to a broker: 50 times
`retry.backoff.ms`	The amount of waiting time before attempting to retry a failed request to a given topic partition. It avoids repeatedly sending requests in a tight loop under some failure scenarios: 100 times.
`request.timeout.ms`	It controls the maximum amount of waiting time for a request response. If the response is not received before the timeout elapses, the client resends or fails the request (if the connection attempts have reached the limit: 30000 milliseconds.
`client.id`	An id string to pass to the server when making requests. It tracks the request source beyond just ip/port, by allowing a logical application name to be included on the server-side login request. For tracking: arcsight
`heartbeat.interval.ms`	The expected time between heartbeats to the consumer coordinator when using Kafka's group management facilities. Heartbeats are used to ensure that the consumer's session stays active and facilitates rebalancing when new consumers join or leave the group. The value must be set lower than session.timeout.ms and higher than 1/3 of that value. It can be adjusted even lower to control the expected time for normal rebalances.
`connections.max.idle.ms` (Idle connections timeout)	The server socket processor threads close the connections that appear idle for more than 600000 ms.
`auto.offset.reset`	It can be executed when there is not an initial offset in Kafka or if the current offset does not exist in the server anymore.

`disable.activemq`	The values can be:
	`True: Disable ActiveMQ`
	`False: Enable ActiveMQ`

Overriding Parser Files

To override parser files:

1. Stop the connector and navigate to the path

   `<connector_home>/current/users/agent/fcp/connectorname_log>`,

   for example

   `<connector_home>/current/users/agent/fcp/cisco_syslog>`

   The following files should be found under the location:

   `cisco_syslog.subagent.sdkrfilereader.properties`

   `cisco_sdsyslog.subagent.sdkrfilereader.properties`

   As well as an "extra processor" parser required for main-level REGEX type agents:

   `cisco_sdsyslog.sdkkeyvaluefilereader.properties`
2. In order to override these files, create the sub-folder structure and the required file(s) under

   `<connector_home>/current/users/agent/fcp/cisco_syslog`
3. Make sure the override only includes the changes or additions to the base /shipped parser.
4. Start the connector.
5. To confirm the override was successful, go to the `agent.out.wrapper.log` file look for the **"An over-ride file was found and loaded"** note.

   > **Note**: The Override file should be created with the same file name and under the same folder location and replaced without affecting or making changes in the `agent.properties` file.

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this computer, click the link above and an email window opens with the following information in the subject line:

Feedback on Configuration Guide (Kafka FlexConnector 8.3 Patch 3)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to Documentation-Feedback@microfocus.com.

We appreciate your feedback!