Hortonworks Streaming Analytics Manager 3

# Source, Sink, and Processor Configuration Values

**Date of Publish:** 2019-05-15


**https://docs.hortonworks.com/**

# Contents

# Source, Processor, and Sink Configuration Values

As you build your streaming applications, use this reference material to help configure the source, processor, and sink Stream Builder components.

## Source Configuration Values

### Table 1: Kafka

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| Cluster Name | Mandatory. Service pool defined in SAM to get metadata information about Kafka cluster |
| Security Protocol | Mandatory. Protocol to be used to communicate with kafka brokers. E.g. PLAINTEXT. Auto suggest with a list of protocols supported by Kafka service based on cluster name selected. If you select a protocol with SSL or SASL make sure to fill out the related config fields |
| Bootstrap Servers | Mandatory. A comma separated string of host:port representing Kafka broker listeners. Auto suggest with a list of options based on security protocol selected above |
| Kafka topic | Mandatory. Kafka topic to read data from. Make sure that corresponding schema for topic is defined in Schema Registry |
| Consumer Group Id | Mandatory. A unique string that identifies the consumer group it belongs to. Used to keep track of consumer offsets |
| Reader schema version | Optional. Version of schema for topic to read from. Default value is the version used by producer to write data to topic |
| Kerberos client principal | Optional(Mandatory for SASL). Client principal to use to connect to brokers while using SASL GSSAPI mechanism for Kerberos(used in case of security protocol being SASL_PLAINTEXT or SASL_SSL) |
| Kerberos keytab file | Optional(Mandatory for SASL). Keytab file location on worker node containing the secret key for client principal while using SASL GSSAPI mechanism for Kerberos(used in case of security protocol being SASL_PLAINTEXT or SASL_SSL) |
| Kafka service name | Optional(Mandatory for SASL). Service name that Kafka broker is running as(used in case of security protocol being SASL_PLAINTEXT or SASL_SSL) |
| Fetch minimum bytes | Optional. The minimum number of bytes the broker should return for a fetch request. Default value is 1 |
| Maximum fetch bytes per partition | Optional. The maximum amount of data per-partition the broker will return. Default value is 1048576 |
| Maximum records per poll | Optional. The maximum number of records a poll will return. Default value is 500 |
| Poll timeout(ms) | Optional. Time in milliseconds spent waiting in poll if data is not available. Default value is 200 |
| Offset commit period(ms) | Optional. Period in milliseconds at which offsets are committed. Default value is 30000 |
| Maximum uncommitted offsets | Optional.Defines the max number of polled records that can be pending commit, before another poll can take place. Default value is 10000000. This value should depend on the size of each message in Kafka and the memory available to the worker jvm process |

| | |
|---|---|
| First poll offset strategy | Optional. Offset used by the Kafka spout in the first poll to Kafka broker. Pick one from enum values. ["EARLIEST", "LATEST", "UNCOMMITTED_EARLIEST", "UNCOMMITTED_LATEST"]. Default value is EARLIEST_UNCOMMITTED. It means that by default it will start from the earliest uncommitted offset for the consumer group id provided above |
| Partition refresh period(ms) | Optional. Period in milliseconds at which Kafka will be polled for new topics and/or partitions. Default value is 2000 |
| Emit null tuples? | Optional. A flag to indicate if null tuples should be emitted to downstream components or not. Default value is false |
| First retry delay(ms) | Optional. Interval delay in milliseconds for first retry for a failed Kafka spout message. Default value is 0 |
| Retry delay period(ms) | Optional. Retry delay period(geometric progression) in milliseconds for second and subsequent retries for a failed Kafka spout message. Default value is 2 |
| Maximum retries | Optional. Maximum number of times a failed message is retried before it is acked and committed. Default value is 2147483647 |
| Maximum retry delay(ms) | Optional. Maximum interval in milliseconds to wait before successive retries for a failed Kafka spout message. Default value is 10000 |
| Consumer startup delay(ms) | Optional. Delay in milliseconds after which Kafka will be polled for records. This value is to make sure all executors come up before first poll from each executor happens so that partitions are well balanced among executors and onPartitionsRevoked and onPartitionsAssigned is not called later causing duplicate tuples to be emitted. Default value is 60000 |
| SSL keystore location | Optional. The location of the key store file. Used when Kafka client connectivity is over SSL |
| SSL keystore location | Optional. The store password for the key store file |
| SSL key password | Optional. The password of the private key in the key store file |
| SSL truststore location | Optional(Mandatory for SSL). The location of the trust store file |
| SSL truststore password | Optional(Mandatory for SSL). The password for the trust store file |
| SSL enabled protocols | Optional. Comma separated list of protocols enabled for SSL connections |
| SSL keystore type | Optional. File format of keystore file. Default value is JKS |
| SSL truststore type | Optional. File format of truststore file. Default value is JKS |
| SSL protocol | Optional. SSL protocol used to generate SSLContext. Default value is TLS |
| SSL provider | Optional. Security provider used for SSL connections. Default value is default security provider for JVM |
| SSL cipher suites | Optional. Comma separated list of cipher suites. This is a named combination of authentication, encryption, MAC and key exchange algorithm used to negotiate the security settings for a network connection using TLS or SSL network protocol. By default all the available cipher suites are supported |
| SSL endpoint identification algorithm | Optional. The endpoint identification algorithm to validate server hostname using server certificate |
| SSL key manager algorithm | Optional. The algorithm used by key manager factory for SSL connections. Default value is SunX509 |
| SSL secure random implementation | Optional. The SecureRandom PRNG implementation to use for SSL cryptographic operations |
| SSL trust manager algorithm | Optional. The algorithm used by trust manager factory for SSL connections. Default value is the trust manager factory algorithm configured for the Java Virtual Machine. Default value is PKIX |

### Table 2: Event Hubs

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| Username | The Event Hubs user name (policy name in Event Hubs Portal) |
| Password | The Event Hubs password (shared access key in Event Hubs Portal) |
| Namespace | The Event Hubs namespace |
| Entity Path | The Event Hubs entity path |
| Partition Count | The number of partitions in the Event Hubs |
| ZooKeeper Connection String | The ZooKeeper connection string |
| Checkpoint Interval | The frequency at which offsets are checkpointed |
| Receiver Credits | Receiver credits |
| Max Pending Messages Per Partition | The max pending messages per partition |
| Enqueue Time Filter | The enqueue time filter |
| Consumer Group Name | The consumer group name |

### Table 3: HDFS

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| Cluster Name | Service pool defined in SAM to get metadata information about HDFS cluster |
| HDFS URL | HDFS namenode URL |
| Input File Format | The format of the file being consumed dictates the type of reader used to read the file. Currently only 'com.hortonworks.streamline.streams.runtime.storm.spout.JsonFileReader' is supported |
| Source Dir | The HDFS directory from which to read the files. |
| Archive Dir | Files from source dir will be moved to this HDFS location after being completely read. |
| Bad Files Dir | Files from Source Dir will be moved to this HDFS location if there is an error encountered while processing them. |
| Lock Dir | Lock files (used to synchronize multiple reader instances) will be created in this location. Defaults to a '.lock' subdirectory under the source directory. |
| Commit Frequency Count | Records progress in the lock file after specified number of records are processed. Setting it to 0 disables this. |
| Commit Frequency Secs | Records progress in the lock file after specified secs have elapsed. Must be greater than 0. |
| Max Outstanding | Limits the number of unACKed tuples by pausing tuple generation (if ACKers are used in the topology). |
| Lock Timeout Seconds | Duration of inactivity after which a lock file is considered to be abandoned and ready for another spout to take ownership. |
| Ignore Suffix | File names with this suffix in the source dir will not be processed. |

## Processor Configuration Values

### Table 4: Aggregate

| Configuration Field | Description, requirements, tips for configuration |
|---|---|

| | |
|---|---|
| General Processor description | Performs aggregate operations on a stream of events within a window. |
| Select Keys | These are the keys to “group by” for computing the aggregate. |
| Window Interval Type | Time - for time based windows.<br>Count - for count based windows. |
| Window Interval | The length or duration of the window |
| Sliding Interval | The interval at which the window slides |
| Timestamp Field | A field in the event that represents the event timestamp as a long value. If specified the timestamp at which the event occurred will be used for the window computations. |
| Output Fields -- Input | The field on which to apply aggregate function |
| Output Fields -- Aggregate Function | The aggregate function to apply |
| Output Fields -- Output | The output field name |

### Table 5: Branch

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| General processor description | Conditionally redirects tuples from one incoming stream to one or more outbound streams. |
| Process all checkbox | If disabled, stops processing further rules after a rule evaluates successfully. |
| Rule Name | Rule name. Must be unique within the Branch processor. |
| Rule Description | Description of rule |
| Field Name | Field name used in the condition for the rule |
| Rule Operation | The comparison operator for the condition |

### Table 6: Join

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| General Processor Description | Joins one or more event streams into one output stream, based on user defined join criteria |
| Select Stream | Name of stream to join |
| Select Field | Name of field to use for join |
| Window Interval Type | Determines the type of windowing (count/time based) to use for buffering streams to be joined |
| Window Interval | The window size. |
| Sliding Interval | The interval between the start of two consecutive windows |
| Output Fields | Select which of the fields to include in the resulting event |

### Table 7: PMML

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| General Processor Description | Allows users to score tuples according to a choice of PMML model registered in the model registry. The scored results are put in the predicted fields as defined in the PMML XML descriptor file. Predicted fields are available to send downstream, in addition to input fields |
| Model Name | Name of the PPML model in model registry to use |

### Table 8: Projection Bolt

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| General Processor Description | This allows user to choose specific fields from the input events to be passed to output event and apply a transformation using UDF on chosen fields and add result as a field in the output event. |
| Projection Fields | Input event fields to be projected into output event. |
| Function | UDF to be applied on the given input fields and output is added as a new field in the output event. |
| Arguments | Field names to be passed as arguments to the chosen function |
| Fields Name | Name of the inu |
| Plus icon | Add a new transformation |

### Table 9: Rule

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| General Processor Description | Design time definition of a rule whose scope is the input fields. The condition of the rule is defined in the Create Query section. Only runtime values whose rule condition evaluates to true will be sent downstream. |
| Rule Name | Name of the rule. It must be unique only within a Rule processor. Can be reused across rule processors. |
| Description | Documentation detailing the purpose of the rule. For user reference only. |
| Create Query | The condition of the rule is a composition of boolean expressions built with operators on input fields. These boolean expressions are parsed as SQL like query. |

## Sink Configuration Values

### Table 10: Cassandra

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| General Sink Description | This allows users to send events into given cassandra table. |
| Table Name | Name of the table into which events should be written to. |
| Column Name | Column name to which a respective field is mapped. |
| Field Name | Field name to be mapped as respective column name. |
| Cassandra Configurations- User Name | User name to connect to Cassandra cluster. |
| Password | Password to connect to Cassandra cluster. |
| Keyspace | Keyspace in which table exists |
| Nodes | Cassandra nodes configuration to be passed |
| Port | Port number for Cassandra cluster |
| Row Batch Size | Maximum number of rows to be taken in a batch |
| Retry Policy | Class name of the retry policy to be applied. Default value is “DefaultRetryPolicy”. Valid options are "DowngradingConsistencyRetryPolicy", "FallthroughRetryPolicy" and "DefaultRetryPolicy" |

| | |
|---|---|
| Consistency Level | Consistency level at which data is inserted. Default value is: QUORUM, valid values are ["ANY", "ONE", "TWO", "THREE", "QUORUM", "ALL", "LOCAL_QUORUM", "EACH_QUORUM", "SERIAL", "LOCAL_SERIAL", "LOCAL_ONE" ] |
| Reconnection Base Delay | Base delay (in milliseconds) while reconnecting to target. |
| Reconnection Maximum Delay | Maximum delay (in milliseconds) while reconnecting to target. |

### Table 11: Druid

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| General Sink Description | Druid sink is used to push data Druid data store. This sink uses Druid's Tranquility library to push data. More details : http://druid.io/docs/latest/ingestion/stream-push.html |
| Name of the Indexing Service | The druid.service name of the indexing service overlord node.<br>It is mandatory parameter. |
| Service Discovery path | Curator service discovery path. It is mandatory parameter. |
| ZooKeeper Connect String | ZooKeeper connect string. It is mandatory parameter. |
| Datasource name | The name of the ingested data source. Datasources can be thought of as tables. It is mandatory parameter. |
| Dimensions | Specifies the dimensions(columns) of the data. It is mandatory parameter. |
| TimeStamp Field Name | Specifies the column and format of the timestamp.It is mandatory parameter. |
| Window Period | Window Period takes ISO 8601 Period format (https://en.wikipedia.org/wiki/ISO_8601). It is mandatory parameter. |
| Index Retry Period | If an indexing service overlord call fails for some apparently-transient reason, retry for this long before giving up. It takes ISO 8601 Period format (https://en.wikipedia.org/wiki/ISO_8601). It is mandatory parameter. |
| Segment Granularity | The granularity to create segments. |
| Query Granularity | The minimum granularity to be able to query results at and the granularity of the data inside the segment. |
| Batch Size | Maximum number of messages to send at once |
| Max Pending Batches | Maximum number of batches that may be in flight |
| Linger millis | Wait this long for batches to collect more messages (up to maxBatchSize) before sending them. |
| Block On Full | Whether send will block (true) or throw an exception (false) when called while the outgoing queue is full |
| Druid partitions | Number of Druid partitions to create. |
| Partition Replication | Number of instances of each Druid partition to create. |
| Aggregator Info | A list of aggregators. Currently we support Count Aggregator, Double Sum Aggregator, Double Max Aggregator, Double Min Aggregator, Long Sum Aggregator, Long Max Aggregator, Long Min Aggregators. |

### Table 12: Hive

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| General Sink Description | Hive sink is used to write data to Hive tables |
| Metastore URI | URI of the metastore to connect to e.g.: thrift://localhost:9083 |

| | |
|---|---|
| Database Name | Name of the Hive database |
| Table name | Name of table to stream to |
| Fields | The event fields to stream to hive |
| Partition fields | The event fields on which to partition the data |
| Flush Interval | The interval (in seconds) at which a transaction batch is committed |
| Transactions per batch | The number of transactions per batch |
| Max open connections | The maximum number of open connections to Hive |
| Batch size | The number of events per batch |
| Idle timeout | The idle timeout |
| Call timeout | The call timeout |
| Heartbeat Interval | The heart beat interval |
| Auto create partitions | If true, the partition specified in the endpoint will be auto created if it does not exist |
| Kerberos keytab | Kerberos keytab file path |
| Kerberos principal | Kerberos principal name |

### Table 13: HBase

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| General Sink Description | Writes to events to HBase |
| HBase table | Hbase table to write to |
| Column Family | Hbase table column family |
| Batch Size | Number of records in the batch to trigger flushing. Note that every batch needs to be full before it can be flushed as tick tuple is not supported currently due to the fact that all bolts in topology receive a tick tuple if enabled |
| Row Key Field | Field to be used as row key for table |

### Table 14: HDFS

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| General Sink Description | Writes events to HDFS |
| Hdfs URL | Hdfs Namenode URL |
| Path | Directory to which the files will be written |
| Flush Count | Number of records to wait for before flushing to Hdfs |
| Rotation Policy | Strategy to rotate files in Hdfs |
| Rotation Interval Multiplier | Rotation interval multiplier for timed rotation policy |
| Rotation Interval Unit | Rotation interval unit for timed rotation policy |
| Output fields | Specify the output fields, in the desired order |
| Prefix | Prefix for default file name format |
| Extension | Extension for default file name format |

### Table 15: JDBC

| Configuration Field | Description, requirements, tips for configuration |
|---|---|

| | |
|---|---|
| General Sink Description | Writes events to a database using JDBC. |
| Driver Class Name | The driver class name. E.g. com.mysql.jdbc.Driver |
| JDBC URL | JDBC Url, E.g. jdbc:mysql://localhost:3306/test |
| User Name | Database username. |
| Password | Database password. |
| Table Name | Table to write to. |
| Column Names | Names of the database columns |

### Table 16: Kafka

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| General Sink Description | Kafka sink to write SAM events to a kafka topic |
| Cluster Name | Mandatory. Service pool defined in SAM to get metadata information about Kafka cluster |
| Kafka Topic | Mandatory. Kafka topic to write data to. Make sure that the schema for the corresponding topic exists in SR. The incoming SAM event into Kafka sink should adhere to the version of schema selected |
| Security Protocol | Mandatory. Protocol to be used to communicate with kafka brokers. E.g. PLAINTEXT. Auto suggest with a list of protocols supported by Kafka service based on cluster name selected. If you select a protocol with SSL or SASL make sure to fill out the related config fields |
| Bootstrap Servers | Mandatory. A comma separated string of host:port representing Kafka broker listeners. Auto suggest with a list of options based on security protocol selected above |
| Fire And Forget? | Optional. A flag to indicate if kafka producer should wait for ack or not. Default value is false |
| Async? | Optional. A flag to indicate whether to use async kafka producer or not. Default value is true |
| Key serializer | Optional. Type of key serializer to use. Options are ["String", "Integer", "Long", "ByteArray"]. Default value is ByteArray. Note that this field does not save any key in the kafka message. Incoming SAM event is stored as value in Kafka message with key being null |
| Key field | Optional. Name of the key field. One of the fields from incoming event schema |
| Writer schema version | Optional. Version of schema for topic to use for serializing the message. Default is the latest version for the schema |
| Ack mode | Optional. Ack mode used in producer request for a record sent to server(None\|Leader\|Min in-sync replicas). Options are [“None”, “Leader”, “All”]. Default value is “Leader” |
| Buffer memory | Optional. The total bytes of memory the producer can use to buffer records waiting to be sent to the server. Default value is 33554432 |
| Compression type | Optional. The compression type for all data generated by the producer. Options are ["none", "gzip", "snappy", "lz4"]. Default value is “none” |
| Retries | Optional. Number of retry attempts for a record send failure. Default value is 0 |
| Batch size | Optional. Producer batch size in bytes for records sent to same partition. Default value is 16384 |
| Client id | Optional. Id sent to server in producer request for tracking in server logs |
| Max connection idle | Optional. Time in milliseconds for which connections can be idle before getting closed. Default value is 540000 |

| | |
|---|---|
| Linger time | Optional. Time in milliseconds to wait before sending a record out when batch is not full. Default value is 0 |
| Max block | Optional. Time in milliseconds that send and partitionsFor methods will block for. Default value is 60000 |
| Max request size | Optional. Maximum size of a request in bytes. Default value is 1048576 |
| Receive buffer size | Optional. Size in bytes of TCP receive buffer (SO_RCVBUF) to use when reading data. Default value is 32768 |
| Request timeout | Optional. Maximum amount of time in milliseconds the producer will wait for the response of a request. Default value is 30000 |
| Kerberos client principal | Optional(Mandatory for SASL). Client principal to use to connect to brokers while using SASL GSSAPI mechanism for Kerberos(used in case of security protocol being SASL_PLAINTEXT or SASL_SSL) |
| Kerberos keytab file | Optional(Mandatory for SASL). Keytab file location on worker node containing the secret key for client principal while using SASL GSSAPI mechanism for Kerberos(used in case of security protocol being SASL_PLAINTEXT or SASL_SSL) |
| Kafka service name | Optional(Mandatory for SASL). Service name that Kafka broker is running as(used in case of security protocol being SASL_PLAINTEXT or SASL_SSL) |
| Send buffer size | Optional. Size in bytes of TCP send buffer (SO_SNDBUF) to use when sending data. Default value is 131072 |
| Timeout | Optional. Maximum amount of time in milliseconds server will wait for acks from followers. Default value is 30000 |
| Block on buffer full? | Optional. Boolean to indicate whether to block on a full buffer or throw an exception.Default value is true |
| Max in-flight requests | Optional. Maximum number of unacknowledged requests producer will send per connection before blocking. Default value is 5 |
| Metadata fetch timeout | Optional. Timeout in milliseconds for a topic metadata fetch request. Default value is 60000 |
| Metadata max age | Optional. Time in milliseconds after which a metadata fetch request is forced. Default value is 300000 |
| Reconnect backoff | Optional. Amount of time in milliseconds to wait before attempting to reconnect to a host. Default value is 50 |
| Retry backoff | Optional. Amount of time in milliseconds to wait before attempting to retry a failed fetch request. Default value is 100 |
| SSL keystore location | Optional. The location of the key store file. Used when Kafka client connectivity is over SSL |
| SSL keystore location | Optional. The store password for the key store file |
| SSL key password | Optional. The password of the private key in the key store file |
| SSL truststore location | Optional(Mandatory for SSL). The location of the trust store file |
| SSL truststore password | Optional(Mandatory for SSL). The password for the trust store file |
| SSL enabled protocols | Optional. Comma separated list of protocols enabled for SSL connections |
| SSL keystore type | Optional. File format of keystore file. Default value is JKS |
| SSL truststore type | Optional. File format of truststore file. Default value is JKS |
| SSL protocol | Optional. SSL protocol used to generate SSLContext. Default value is TLS |
| SSL provider | Optional. Security provider used for SSL connections. Default value is default security provider for JVM |

| | |
|---|---|
| SSL cipher suites | Optional. Comma separated list of cipher suites. This is a named combination of authentication, encryption, MAC and key exchange algorithm used to negotiate the security settings for a network connection using TLS or SSL network protocol. By default all the available cipher suites are supported |
| SSL endpoint identification algorithm | Optional. The endpoint identification algorithm to validate server hostname using server certificate |
| SSL key manager algorithm | Optional. The algorithm used by key manager factory for SSL connections. Default value is SunX509 |
| SSL secure random implementation | Optional. The SecureRandom PRNG implementation to use for SSL cryptographic operations |
| SSL trust manager algorithm | Optional. The algorithm used by trust manager factory for SSL connections. Default value is the trust manager factory algorithm configured for the Java Virtual Machine. Default value is PKIX |

### Table 17: Notification

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| General Sink Description | Can be used to send out notifications (currently supports email) |
| Username | The username for the mail server |
| Password | The password for the mail server |
| Host | Mail server host name |
| Port | Mail server port |
| SSL? | If the connection should be over SSL |
| Start TLS | Flag to indicate the TLS setting |
| Debug? | Whether to log debug messages |
| Email Server Protocol | The email server protocol. E.g. smtp |
| Authenticate | Flag to indicate if authentication is to be performed |

### Table 18: Open TSDB

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| General Sink Description | Sink to which events can be written given OpenTSDB cluster. |
| REST API URL | The URL of the REST API (ex: http://localhost:4242) |
| Metric Field Name | Field name of the metric |
| Timestamp Field Name | Field name of the timestamp |
| Tags Field Name | Field name of tags. |
| Value Field Name | Field name of the value |
| Fail Tuple for Failed Metrics? | Whether to fail tuple for any failed metrics to OpenTSDB |
| Sync? | Flag to indicate whether to sync or not. |
| Sync Timeout | Sync timeout in (milliseconds), this is taken into account only when Sync is true. |
| Return Summary? | Whether to return summary or not |
| Return Details? | Whether to return details or not. |
| Enable Chunked Encoding? | Whether to enable chunked encoding or not for REST API calls to OpenTSDB |

### Table 19: Solr

| Configuration Field | Description, requirements, tips for configuration |
|---|---|
| General Sink Description | Enables indexing of live input data into Apache Solr collections |
| Apache Solr ZooKeeper Host String | Info about the zookeeper ensemble used to coordinate the Solr cluster. This string is specified in a comma separated value as follows: zk1.host.com:2181,zk2.host.com:2181,zk3.example.com:2181 |
| Apache Solr Collection Name | The name of the Apache Solr collection where to index live data |
| Commit Batch Size | Defines how often the indexed data is committed into Apache Solr. It is specified using an integral number. For instance, if set to 100, every 100 tuples Apache Solr will commit the data |