StreamSets Data Collector

Cumulative 3.0.x.x Release Notes

This document contains release information for the following versions of StreamSets Data Collector:

* * * * * * * *

StreamSets Data Collector 3.0.2.0 Release Notes

January 10, 2017

We're happy to announce a new version of StreamSets Data Collector. This version contains an enhancement and some important bug fixes.

This document contains important information about the following topics for this release:

Upgrading to Version 3.0.2.0

You can upgrade previous versions of Data Collector to version 3.0.2.0. For complete instructions on upgrading, see the Upgrade Documentation.

Update Control Hub On-premises

If you use StreamSets Control Hub (SCH) on-premises and you upgrade registered Data Collectors to a version higher than your current version of Control Hub, you must modify the Data Collector version range within your Control Hub installation.

By default, Control Hub can work with registered Data Collectors from version 2.1.0.0 to the current version of Control Hub. You can customize the Data Collector version range. For example, if you use Control Hub on-premises version 2.7.2 and you upgrade registered Data Collectors to version 3.0.2.0, you must configure the maximum Data Collector version that can work with Control Hub to version 3.0.2.0.

To modify the Data Collector version range:

1. Log in to Control Hub as the default system administrator - the admin@admin user account.
2. In the Navigation panel, click **Administration > Data Collectors**.
3. Click the **Component Version Range** icon.

4. Enter 3.0.2.0 as the maximum Data Collector version that can work with Control Hub.

Update Pipelines using Legacy Stage Libraries

With version 3.0.0.0, a set of older stage libraries are no longer included with Data Collector. Pipelines that use these legacy stage libraries will not run until you perform one of the following tasks:

Use a current stage library

We strongly recommend that you upgrade your system and use a current stage library in the pipeline:

1. Upgrade the system to a more current version.

2. Install the stage library for the upgraded system.

3. In the pipeline, edit the stage and select the appropriate stage library.

Install the legacy library

Though not recommended, you can still download and install the older stage libraries as custom stage libraries. For more information, see Legacy Stage Libraries.

Disable Cloudera Navigator Integration

With version 3.0.0.0, the beta version of Cloudera Navigator integration is no longer available with Data Collector.

Cloudera Navigator integration is now released as part of the StreamSets Commercial Subscription. For information about the StreamSets Commercial Subscription, contact us.

When upgrading from a previous version with Cloudera Navigator integration enabled, do not include the Cloudera Navigator properties when you configure the 3.0.1.0 Data Collector configuration file, sdc.properties. The properties to omit are:

- lineage.publishers
- lineage.publisher.navigator.def
- All other properties with the lineage.publisher.navigator prefix

JDBC Multitable Consumer Query Interval Change

With version 3.0.0.0, the Query Interval property is replaced by the new Queries per Second property.

Upgraded pipelines with the Query Interval specified using a constant or the default format and unit of time, ${10 * SECONDS}, have the new Queries per Second property calculated and defined as follows:

```
Queries per Second = Number of Threads / Query Interval (in seconds)
```

For example, say the origin uses three threads and Query Interval is configured for ${15 * SECONDS}. Then, the upgraded origin sets Queries per Seconds to 3 divided by 15, which is .2. This means the origin will run a maximum of two queries every 10 seconds.

The upgrade would occur the same way if Query Interval were set to 15.

Pipelines with a Query Interval configured to use other units of time, such as ${.1 *MINUTES}, or configured with a different expression format, such as ${SECONDS * 5}, are upgraded to use the default for Queries per Second, which is 10. This means the pipeline will run a maximum of 10 queries per second. The fact that these expressions are not upgraded correctly is noted in the Data Collector log.

If necessary, update the Queries per Second property as needed after the upgrade.

Update JDBC Query Consumer Pipelines used for SQL Server CDC Data

With version 3.0.0.0, the Microsoft SQL Server CDC functionality in the JDBC Query Consumer origin has been deprecated and will be removed in a future release.

For pipelines that use the JDBC Query Consumer to process Microsoft SQL Server CDC data, replace the JDBC Query Consumer origin with another origin:

- To read data from Microsoft SQL Server CDC tables, use the SQL Server CDC Client origin.
- To read data from Microsoft SQL Server change tracking tables, use the SQL Server Change Tracking origin.

Update MongoDB Destination Upsert Pipelines

With version 3.0.0.0, the MongoDB destination supports the replace and update operation codes, and no longer supports the upsert operation code. You can use a new Upsert flag in conjunction with Replace and Update.

After upgrading from a version earlier than 3.0.0.0, update the pipeline as needed to ensure that records passed to the destination do not use the upsert operation code (sdc.operation.type = 4). Records that use the upsert operation code will be sent to error.

In previous releases, records flagged for upsert were treated in the MongoDB system as Replace operations with the Upsert flag set.

If you want to replicate the upsert behavior from earlier releases, perform the following steps:

1. Configure the pipeline to use the Replace operation code.

Make sure that the sdc.operation.type is set to 7 for Replace instead of 4 for Upsert.

2. In the MongoDB destination, enable the new Upsert property.

Time Zones in Stages

With version 3.0.0.0, time zones have been organized and updated to use JDK 8 names. This should make it easier to select time zones in stage properties.

In the rare case that an upgraded pipeline uses a format not supported by JDK 8, edit the pipeline to select a compatible time zone.

Update Kudu Pipelines

With version 3.0.0.0, if the destination receives a change data capture log from the following source systems, you must specify the source system so that the destination can determine the format of the log: Microsoft SQL Server, Oracle CDC Client, MySQL Binary Log, or MongoDB Oplog.

Previously, the Kudu destination could not directly receive changed data from these source systems. You either had to include a scripting processor in the pipeline to modify the field paths in the record to a format that the destination could read. Or, you had to add multiple Kudu destinations to the pipeline - one per operation type - and include a Stream Selector processor to send records to the destination by operation type.

If you implemented one of these workarounds, then after upgrading, modify the pipeline to remove the scripting processor or the Stream Selector processor and the multiple destinations. In the Kudu destination, set the Change Log Format to the appropriate format of the log: Microsoft SQL Server, Oracle CDC Client, MySQL Binary Log, or MongoDB Oplog.

New Enhancements in 3.0.2.0

This version includes the following new enhancement:

- **SFTP/FTP Client origin enhancement** - The origin can now generate events when starting and completing processing for a file and when all available files have been processed.

Fixed Issues in 3.0.2.0

Several known issues are fixed with this release. For the full list, click here.

Known Issues in 3.0.2.0

Please note the following known issues with this release.

For a full list of known issues, check out our JIRA.

JIRA	Description

SDC-8176
SDC-8174
	Origins and destinations in HDP stage libraries cannot process protobuf data at this time.
	Workaround: Copy the protobuf jar file, protobuf-java-2.5.0.jar, from another stage library into the HDP stage library that you want to use.
	For example, to enable processing protobuf data with the HDP 2.6 stage library, you can copy the file from the basic stage library in the following location:
	`$SDC_DIST/streamsets-libs/streamsets-datacollector-basic-lib/lib/`
	to the HDP 2.6 stage library in the following location:
	`$SDC_DIST/streamsets-libs/streamsets-datacollector-hdp_2_6-lib/lib/`

SDC-8095	The JMS Consumer origin fails to process SDC Record data with the following error:

	`PARSER_03 - Cannot parse record from message 'JMSTestQueue::0': com.streamsets.pipeline.lib.parser.DataParserException: SDC_RECORD_PARSER_00 - Could advance reader 'JMSTestQueue::0' to '0' offset` Workaround: Use the JSON data format instead of the SDC record data format for both the JMS Consumer and JMS Producer stages. If information in the record header attributes are needed in the pipeline, use an Expression Evaluator processor to include the data in record fields.
SDC-8078	The HTTP Server origin does not release the ports that it uses after the pipeline stops. Releasing the ports requires restarting Data Collector.

SDC-8069	After upgrading to Data Collector 3.0.0.0 that is not enabled to work with Control Hub, cluster batch pipelines fail validation with the following message:
	
	`VALIDATION_0071 - Stage 'Write to DPM directly' from 'Basic' library does not support 'Cluster Batch' execution mode`
	
	Workaround: Modify the pipeline JSON files to configure the StatsAggregatorStage to discard statistics. Locate the following lines in the file and set them to the values in bold:
	
	`...`
	`{`
	`"name" : "statsAggregatorStage",`
	`"value" : "streamsets-datacollector-basic-lib::com_streamsets_pipeline_stage_destination_devnull_StatsNullDTarget::1"`
	`}`
	`...`
	`"statsAggregatorStage" : {`
	`"instanceName" : "Discard_StatsAggregatorStage",`
	`"library" : "streamsets-datacollector-basic-lib",`
	`"stageName" : "com_streamsets_pipeline_stage_destination_devnull_StatsNullDTarget",`
	`...`

SDC-7986	A Data Collector registered with StreamSets Control Hub becomes unusable when the Data Collector is configured to use LDAP authentication. Workaround: Configure the Data Collector to use file-based authentication.
SDC-7903	Cluster mode pipelines that read from a MapR Streams Consumer origin fail to run on MapR 6.0.
SDC-6438	Do not use Cluster Streaming pipelines with MapR and Spark 2.x.

SDC-7872	Due to Oracle LogMiner returning an unsupported SQL format, Data Collector cannot parse changes to tables containing the XML type.
SDC-7761	The Java keystore credential store implementation fails to work for a Data Collector installed through Cloudera Manager. The jks-cs command creates the Java keystore file in the Data Collector configuration directory defined for the parcel. However, for Data Collector to access the Java keystore file, the file must be outside of the parcel directory. The CyberArk and Vault credential store implementations do work with a Data Collector installed through Cloudera Manager.
SDC-7645	The Data Collector Docker image does not support processing data using another locale. Workaround: Install Data Collector from the tarball or RPM package.
SDC-7448	You cannot run cluster streaming pipelines on MapR MEP 3 clusters at this time.
SDC-6554	When converting Avro to Parquet on Impala, Decimal fields seem to be unreadable. Data Collector writes the Decimal data as variable-length byte arrays. And due to Impala issue IMPALA-2494, Impala cannot read the data.
SDC-6509	When using the Write to Another Pipeline pipeline error handling option in cluster batch pipeline, if the error handling pipeline encounters a problem and stops, the original pipeline stops with the message “Job has been finished” instead of indicating there was a problem with the error handling pipeline.
SDC-6438	The MapR distribution for Spark 2.x is not supported by cluster streaming pipelines.
SDC-6210	The show-vault-id command returns a NPE when Data Collector is installed on EC2 with IPv6 enabled. Workaround: If you can run Data Collector without IPv6, in the /etc/sysctl.conf file, disable IPv6 by configuring the disable IPv6 property as follows: `net.ipv6.conf.all.disable_ipv6 = 1`
SDC-6077	The Field Remover processor does not remove list fields from list-map data.
SDC-5758	Due to expected Kudu behavior, if Data Collector is configured to use Kerberos, but your Kudu installation does not, Data Collector cannot connect. This applies to Kudu version 1.2 and later. Workaround: If necessary, install a separate Data Collector that does not have Kerberos enabled.
SDC-5521	Kerberos-enabled pipelines that are valid using an Apache Kafka stage library can fail validation when using a Cloudera distribution of Kafka.

	The Apache Kafka client libraries have been modified by StreamSets to allow connectivity to Kerberized Kafka without requiring a JAAS configuration file. The Apache Kafka stage libraries (version 0.9 and greater) do not require a JAAS configuration file when enabling Kerberos. For CDH Kafka versions, a JAAS configuration is still required. Workaround: Include a JAAS configuration file on the classpath by modifying the `sdc-env.sh` or `sdcd-env.sh` file to include the following option in the SDC_JAVA_OPTS environment variable: `-Djava.security.auth.login.config=<path-to-jaas-config>` Or, you can use the matching version Apache Kafka stage library, instead of the Cloudera stage libraries.
SDC-5357	The Jython Evaluator and stages that can access the file system - such as the Directory and File Tail origins - can access all pipeline JSON files stored in the $SDC_DATA directory. This allows users to access pipelines that they might not have permission to access within Data Collector. Workaround: To secure your pipelines, complete the following tasks: 1. Remove the Jython stage library and use the Groovy Evaluator or JavaScript Evaluator processor instead of the Jython Evaluator. 2. Update the Data Collector security policy file, `$SDC_CONF/sdc-security.policy`, so that Data Collector stages do not have `AllPermission` access to the file system. Update the security policy for the following code bases: streamsets-libs-extras, streamsets-libs, and streamsets-datacollector-dev-lib. Use the policy file syntax to set the security policies.
SDC-5325	Cluster mode pipelines that read from a MapR cluster fail when the MapR cluster uses username/password login authentication. Workaround: On the Cluster tab for the pipeline, add the following Java property to the Worker Java Options property: `-Dmaprlogin.password.enabled=true`
SDC-5141	Due to a limitation in the Javascript engine, the Javascript Evaluator issues a null pointer exception when unable to compile a script.
SDC-5039	When you use the Hadoop FS origin to read files from all subdirectories, the origin cannot use the configured Hadoop FS User as a proxy user to read from HDFS. Workaround: If you need to use a proxy user to read from all subdirectories of the specified directories, set the HADOOP_PROXY_USER environment variable to the proxy user in libexec/_cluster-manager script, as follows: `export HADOOP_PROXY_USER = <proxy-user>`
SDC-4212	If you configure a UDP Source or UDP to Kafka origin to enable multithreading after you have already run the pipeline with the option disabled, the following validation error displays: `Multithreaded UDP server is not available on your platform.`

	Workaround: Restart Data Collector.
SDC-3944	The Hive Streaming destination using the MapR library cannot connect to a MapR cluster that uses Kerberos or username/password login authentication.
SDC-3133	When you upgrade Data Collector from the RPM package, the environment configuration file `$SDC_DIST/libexec/sdcd-env.sh` is overwritten. Workaround: Back up the `sdcd-env.sh` file before you upgrade.
SDC-2374	A cluster mode pipeline can hang with a CONNECT_ERROR status. This can be a temporary connection problem that resolves, returning the pipeline to the RUNNING status. If the problem is not temporary, you might need to manually edit the pipeline state file to set the pipeline to STOPPED. Edit the file only after you confirm that the pipeline is no longer running on the cluster or that the cluster has been decommissioned. To manually change the pipeline state, edit the following file: `$SDC_DATA/runInfo/<cluster pipeline name>/<revision>/pipelineState.json` In the file, change CONNECT_ERROR to STOPPED and save the file.

Contact Information

For more information about StreamSets, visit our website: https://streamsets.com/.

Check out our Documentation page for doc highlights, what's new, and tutorials: streamsets.com/docs

Or you can go straight to our latest documentation here:
https://streamsets.com/documentation/datacollector/latest/help

To report an issue, ask for help from our Google group or Slack channel, or find out about our next meetup, check out our Community page: https://streamsets.com/community/.

For general inquiries, email us at info@streamsets.com.

* * * * * * * *

StreamSets Data Collector 3.0.1.0 Release Notes

December 27, 2017

We're happy to announce a new version of StreamSets Data Collector. This version contains several enhancements and some important bug fixes.

This document contains important information about the following topics for this release:

Upgrading to Version 3.0.1.0

You can upgrade previous versions of Data Collector to version 3.0.1.0. For complete instructions on upgrading, see the Upgrade Documentation.

Update Control Hub On-premises

If you use StreamSets Control Hub (SCH) on-premises and you upgrade registered Data Collectors to a version higher than your current version of Control Hub, you must modify the Data Collector version range within your Control Hub installation.

By default, Control Hub can work with registered Data Collectors from version 2.1.0.0 to the current version of Control Hub. You can customize the Data Collector version range. For example, if you use Control Hub on-premises version 2.7.2 and you upgrade registered Data Collectors to version 3.0.1.0, you must configure the maximum Data Collector version that can work with Control Hub to version 3.0.1.0.

To modify the Data Collector version range:

5. Log in to Control Hub as the default system administrator - the admin@admin user account.
6. In the Navigation panel, click **Administration > Data Collectors**.
7. Click the **Component Version Range** icon.
8. Enter 3.0.1.0 as the maximum Data Collector version that can work with Control Hub.

Update Pipelines using Legacy Stage Libraries

With version 3.0.0.0, a set of older stage libraries are no longer included with Data Collector. Pipelines that use these legacy stage libraries will not run until you perform one of the following tasks:

Use a current stage library

We strongly recommend that you upgrade your system and use a current stage library in the pipeline:

1. Upgrade the system to a more current version.

2. Install the stage library for the upgraded system.

3. In the pipeline, edit the stage and select the appropriate stage library.

Install the legacy library

Though not recommended, you can still download and install the older stage libraries as custom stage libraries. For more information, see Legacy Stage Libraries.

Disable Cloudera Navigator Integration

The beta version of Cloudera Navigator integration is no longer available with Data Collector.

Cloudera Navigator integration is now released as part of the StreamSets Commercial Subscription. For information about the StreamSets Commercial Subscription, contact us.

When upgrading from a previous version with Cloudera Navigator integration enabled, do not include the Cloudera Navigator properties when you configure the 3.0.1.0 Data Collector configuration file, sdc.properties. The properties to omit are:

- lineage.publishers
- lineage.publisher.navigator.def
- All other properties with the lineage.publisher.navigator prefix

JDBC Multitable Consumer Query Interval Change

With version 3.0.0.0, the Query Interval property is replaced by the new Queries per Second property.

Upgraded pipelines with the Query Interval specified using a constant or the default format and unit of time, ${10 * SECONDS}, have the new Queries per Second property calculated and defined as follows:

```
Queries per Second = Number of Threads / Query Interval (in seconds)
```

For example, say the origin uses three threads and Query Interval is configured for ${15 * SECONDS}. Then, the upgraded origin sets Queries per Seconds to 3 divided by 15, which is .2. This means the origin will run a maximum of two queries every 10 seconds.

The upgrade would occur the same way if Query Interval were set to 15.

Pipelines with a Query Interval configured to use other units of time, such as ${.1 *MINUTES}, or configured with a different expression format, such as ${SECONDS * 5}, are upgraded to use the default for Queries per Second, which is 10. This means the pipeline will run a maximum of 10 queries per second. The fact that these expressions are not upgraded correctly is noted in the Data Collector log.

If necessary, update the Queries per Second property as needed after the upgrade.

Update JDBC Query Consumer Pipelines used for SQL Server CDC Data

The Microsoft SQL Server CDC functionality in the JDBC Query Consumer origin has been deprecated and will be removed in a future release.

For pipelines that use the JDBC Query Consumer to process Microsoft SQL Server CDC data, replace the JDBC Query Consumer origin with another origin:

- To read data from Microsoft SQL Server CDC tables, use the SQL Server CDC Client origin.
- To read data from Microsoft SQL Server change tracking tables, use the SQL Server Change Tracking origin.

Update MongoDB Destination Upsert Pipelines

With version 3.0.0.0, the MongoDB destination supports the replace and update operation codes, and no longer supports the upsert operation code. You can use a new Upsert flag in conjunction with Replace and Update.

After upgrading from a version earlier than 3.0.0.0, update the pipeline as needed to ensure that records passed to the destination do not use the upsert operation code (sdc.operation.type = 4). Records that use the upsert operation code will be sent to error.

In previous releases, records flagged for upsert were treated in the MongoDB system as Replace operations with the Upsert flag set.

If you want to replicate the upsert behavior from earlier releases, perform the following steps:

1. Configure the pipeline to use the Replace operation code.

Make sure that the sdc.operation.type is set to 7 for Replace instead of 4 for Upsert.

2. In the MongoDB destination, enable the new Upsert property.

Time Zones in Stages

With version 3.0.0.0, time zones have been organized and updated to use JDK 8 names. This should make it easier to select time zones in stage properties.

In the rare case that an upgraded pipeline uses a format not supported by JDK 8, edit the pipeline to select a compatible time zone.

Update Kudu Pipelines

With version 3.0.0.0, if the destination receives a change data capture log from the following source systems, you must specify the source system so that the destination can determine the format of the log: Microsoft SQL Server, Oracle CDC Client, MySQL Binary Log, or MongoDB Oplog.

Previously, the Kudu destination could not directly receive changed data from these source systems. You either had to include a scripting processor in the pipeline to modify the field paths in the record to a format that the destination could read. Or, you had to add multiple Kudu destinations to the pipeline - one per operation type - and include a Stream Selector processor to send records to the destination by operation type.

If you implemented one of these workarounds, then after upgrading, modify the pipeline to remove the scripting processor or the Stream Selector processor and the multiple destinations. In the Kudu destination, set the Change Log Format to the appropriate format of the log: Microsoft SQL Server, Oracle CDC Client, MySQL Binary Log, or MongoDB Oplog.

New Enhancements in 3.0.1.0

This version includes the following new enhancements in the following areas.

- **Azure IoT/Event Hub Consumer origin enhancement** - The Azure Event Hub Consumer origin has been renamed to the Azure IoT/Event Hub Consumer origin.
- **HTTP Server origin enhancement** - The HTTP Server origin now includes path and queryString record header attributes, as well as any other HTTP header attributes included in the request.
- **MongoDB origins enhancement** - Both the MongoDB origin and the MongoDB Oplog origin now support delegated authentication and BSON data type for binary data.
- **SQL Server CDC origin enhancement** - The SQL Server CDC origin now includes information from the SQL Server CDC __$command_id column in a record header attribute named `jdbc. __$command_id`.
- **Mongo DB destination enhancement** - The MongoDB destination now supports delegated authentication.

Fixed Issues in 3.0.1.0

The following table lists some of the known issues that are fixed with this release.

For the full list, click here.

JIRA	Description
SDC-8070	The JDBC Multitable Consumer origin upgrader fails when the Number of Threads is set to an expression.
SDC-8058	Stopping a pipeline with the OPC UA Client origin results in repetitive Container errors.
SDC-7990	The SQL Server CDC origin fails to get CDC tables during validation.
SDC-7290	Using incorrect offset column conditions in the JDBC Multitable Consumer origin generates an endless cycle of exceptions that requires restarting Data Collector to stop the exceptions.

Known Issues in 3.0.1.0

Please note the following known issues with this release.

For a full list of known issues, check out our JIRA.

JIRA	Description

SDC-8095	The JMS Consumer origin fails to process SDC Record data with the following error: `PARSER_03 - Cannot parse record from message 'JMSTestQueue::0': com.streamsets.pipeline.lib.parser.DataParserException: SDC_RECORD_PARSER_00 - Could advance reader 'JMSTestQueue::0' to '0' offset` Workaround: Use the JSON data format instead of the SDC record data format for both the JMS Consumer and JMS Producer stages. If information in the record header attributes are needed in the pipeline, use an Expression Evaluator processor to include the data in record fields.
SDC-8078	The HTTP Server origin does not release the ports that it uses after the pipeline stops. Releasing the ports requires restarting Data Collector.

SDC-8069	After upgrading to Data Collector 3.0.0.0 that is not enabled to work with Control Hub, cluster batch pipelines fail validation with the following message:
	`VALIDATION_0071 - Stage 'Write to DPM directly' from 'Basic' library does not support 'Cluster Batch' execution mode`
	Workaround: Modify the pipeline JSON files to configure the StatsAggregatorStage to discard statistics. Locate the following lines in the file and set them to the values in bold:
	`...`
	`{`
	`"name" : "statsAggregatorStage",`
	`"value" : "streamsets-datacollector-basic-lib::com_streamsets_pipeline_stage_destination_devnull_StatsNullDTarget::1"`
	`}`
	`...`
	`"statsAggregatorStage" : {`
	`"instanceName" : "Discard_StatsAggregatorStage",`
	`"library" : "streamsets-datacollector-basic-lib",`
	`"stageName" : "com_streamsets_pipeline_stage_destination_devnull_StatsNullDTarget",`
	`...`

SDC-7986	A Data Collector registered with StreamSets Control Hub becomes unusable when the Data Collector is configured to use LDAP authentication. Workaround: Configure the Data Collector to use file-based authentication.
SDC-7903	Cluster mode pipelines that read from a MapR Streams Consumer origin fail to run on MapR 6.0.

SDC-6438	Do not use Cluster Streaming pipelines with MapR and Spark 2.x.
SDC-7872	Due to Oracle LogMiner returning an unsupported SQL format, Data Collector cannot parse changes to tables containing the XML type.
SDC-7761	The Java keystore credential store implementation fails to work for a Data Collector installed through Cloudera Manager. The jks-cs command creates the Java keystore file in the Data Collector configuration directory defined for the parcel. However, for Data Collector to access the Java keystore file, the file must be outside of the parcel directory. The CyberArk and Vault credential store implementations do work with a Data Collector installed through Cloudera Manager.
SDC-7645	The Data Collector Docker image does not support processing data using another locale. Workaround: Install Data Collector from the tarball or RPM package.
SDC-7448	You cannot run cluster streaming pipelines on MapR MEP 3 clusters at this time.
SDC-6554	When converting Avro to Parquet on Impala, Decimal fields seem to be unreadable. Data Collector writes the Decimal data as variable-length byte arrays. And due to Impala issue IMPALA-2494, Impala cannot read the data.
SDC-6509	When using the Write to Another Pipeline pipeline error handling option in cluster batch pipeline, if the error handling pipeline encounters a problem and stops, the original pipeline stops with the message “Job has been finished” instead of indicating there was a problem with the error handling pipeline.
SDC-6438	The MapR distribution for Spark 2.x is not supported by cluster streaming pipelines.
SDC-6210	The show-vault-id command returns a NPE when Data Collector is installed on EC2 with IPv6 enabled. Workaround: If you can run Data Collector without IPv6, in the /etc/sysctl.conf file, disable IPv6 by configuring the disable IPv6 property as follows: `net.ipv6.conf.all.disable_ipv6 = 1`
SDC-6077	The Field Remover processor does not remove list fields from list-map data.
SDC-5758	Due to expected Kudu behavior, if Data Collector is configured to use Kerberos, but your Kudu installation does not, Data Collector cannot connect. This applies to Kudu version 1.2 and later. Workaround: If necessary, install a separate Data Collector that does not have Kerberos enabled.

SDC-5521	Kerberos-enabled pipelines that are valid using an Apache Kafka stage library can fail validation when using a Cloudera distribution of Kafka. The Apache Kafka client libraries have been modified by StreamSets to allow connectivity to Kerberized Kafka without requiring a JAAS configuration file. The Apache Kafka stage libraries (version 0.9 and greater) do not require a JAAS configuration file when enabling Kerberos. For CDH Kafka versions, a JAAS configuration is still required. Workaround: Include a JAAS configuration file on the classpath by modifying the `sdc-env.sh` or `sdcd-env.sh` file to include the following option in the SDC_JAVA_OPTS environment variable: `-Djava.security.auth.login.config=<path-to-jaas-config>` Or, you can use the matching version Apache Kafka stage library, instead of the Cloudera stage libraries.
SDC-5357	The Jython Evaluator and stages that can access the file system - such as the Directory and File Tail origins - can access all pipeline JSON files stored in the $SDC_DATA directory. This allows users to access pipelines that they might not have permission to access within Data Collector. Workaround: To secure your pipelines, complete the following tasks: 3. Remove the Jython stage library and use the Groovy Evaluator or JavaScript Evaluator processor instead of the Jython Evaluator. 4. Update the Data Collector security policy file, `$SDC_CONF/sdc-security.policy`, so that Data Collector stages do not have `AllPermission` access to the file system. Update the security policy for the following code bases: streamsets-libs-extras, streamsets-libs, and streamsets-datacollector-dev-lib. Use the policy file syntax to set the security policies.
SDC-5325	Cluster mode pipelines that read from a MapR cluster fail when the MapR cluster uses username/password login authentication. Workaround: On the Cluster tab for the pipeline, add the following Java property to the Worker Java Options property: `-Dmaprlogin.password.enabled=true`
SDC-5141	Due to a limitation in the Javascript engine, the Javascript Evaluator issues a null pointer exception when unable to compile a script.
SDC-5039	When you use the Hadoop FS origin to read files from all subdirectories, the origin cannot use the configured Hadoop FS User as a proxy user to read from HDFS. Workaround: If you need to use a proxy user to read from all subdirectories of the specified directories, set the HADOOP_PROXY_USER environment variable to the proxy user in libexec/_cluster-manager script, as follows: `export HADOOP_PROXY_USER = <proxy-user>`
SDC-4212	If you configure a UDP Source or UDP to Kafka origin to enable multithreading after you have already run the pipeline with the option disabled, the following validation

	error displays: `Multithreaded UDP server is not available on your platform.` Workaround: Restart Data Collector.
SDC-3944	The Hive Streaming destination using the MapR library cannot connect to a MapR cluster that uses Kerberos or username/password login authentication.
SDC-3133	When you upgrade Data Collector from the RPM package, the environment configuration file `$SDC_DIST/libexec/sdcd-env.sh` is overwritten. Workaround: Back up the `sdcd-env.sh` file before you upgrade.
SDC-2374	A cluster mode pipeline can hang with a CONNECT_ERROR status. This can be a temporary connection problem that resolves, returning the pipeline to the RUNNING status. If the problem is not temporary, you might need to manually edit the pipeline state file to set the pipeline to STOPPED. Edit the file only after you confirm that the pipeline is no longer running on the cluster or that the cluster has been decommissioned. To manually change the pipeline state, edit the following file: `$SDC_DATA/runInfo/<cluster pipeline name>/<revision>/pipelineState.json` In the file, change CONNECT_ERROR to STOPPED and save the file.

Contact Information

For more information about StreamSets, visit our website: https://streamsets.com/.

Check out our Documentation page for doc highlights, what's new, and tutorials: streamsets.com/docs

Or you can go straight to our latest documentation here:
https://streamsets.com/documentation/datacollector/latest/help

To report an issue, ask for help from our Google group or Slack channel, or find out about our next meetup, check out our Community page: https://streamsets.com/community/.

For general inquiries, email us at info@streamsets.com.

StreamSets Data Collector 3.0.0.0 Release Notes

November 27, 2017

We're happy to announce a new version of StreamSets Data Collector. This version contains many new features and enhancements, and some important bug fixes.

This document contains important information about the following topics for this release:

Upgrading to Version 3.0.0.0

You can upgrade previous versions of Data Collector to version 3.0.0.0. For complete instructions on upgrading, see the Upgrade Documentation.

Update DPM On-premises

If you use DPM on-premises and you upgrade registered Data Collectors to a version higher than your current version of DPM, you must modify the Data Collector version range within your DPM installation.

By default, DPM can work with registered Data Collectors from version 2.1.0.0 to the current version of DPM. You can customize the Data Collector version range. For example, if you use DPM on-premises version 2.7.2 and you upgrade registered Data Collectors to version 3.0.0.0, you must configure the maximum Data Collector version that can work with DPM to version 3.0.0.0.

To modify the Data Collector version range:

9. Log in to DPM as the default system administrator - the admin@admin user account.
10. In the Navigation panel, click **Administration > Data Collectors**.
11. Click the **Component Version Range** icon.
12. Enter 3.0.0.0 as the maximum Data Collector version that can work with DPM.

Update Pipelines using Legacy Stage Libraries

With this release, a set of older stage libraries are no longer included with Data Collector. Pipelines that use these legacy stage libraries will not run until you perform one of the following tasks:

Use a current stage library

We strongly recommend that you upgrade your system and use a current stage library in the pipeline:

1. Upgrade the system to a more current version.

2. Install the stage library for the upgraded system.

3. In the pipeline, edit the stage and select the appropriate stage library.

Install the legacy library

Though not recommended, you can still download and install the older stage libraries as custom stage libraries. For more information, see Legacy Stage Libraries.

Disable Cloudera Navigator Integration

The beta version of Cloudera Navigator integration is no longer available with Data Collector.

Cloudera Navigator integration is now released as part of the StreamSets Commercial Subscription. For information about the StreamSets Commercial Subscription, contact us.

When upgrading from a previous version with Cloudera Navigator integration enabled, do not include the Cloudera Navigator properties when you configure the 3.0.0.0 Data Collector configuration file, sdc.properties. The properties to omit are:

- lineage.publishers
- lineage.publisher.navigator.def
- All other properties with the lineage.publisher.navigator prefix

JDBC Multitable Consumer Query Interval Change

With version 3.0.0.0, the Query Interval property is replaced by the new Queries per Second property.

Upgraded pipelines with the Query Interval specified using a constant or the default format and unit of time, ${10 * SECONDS}, have the new Queries per Second property calculated and defined as follows:

```
Queries per Second = Number of Threads / Query Interval (in seconds)
```

For example, say the origin uses three threads and Query Interval is configured for ${15 * SECONDS}. Then, the upgraded origin sets Queries per Seconds to 3 divided by 15, which is .2. This means the origin will run a maximum of two queries every 10 seconds.

The upgrade would occur the same way if Query Interval were set to 15.

Pipelines with a Query Interval configured to use other units of time, such as ${.1 *MINUTES}, or configured with a different expression format, such as ${SECONDS * 5}, are upgraded to use the default for Queries per Second, which is 10. This means the pipeline will run a maximum of 10 queries per second. The fact that these expressions are not upgraded correctly is noted in the Data Collector log.

If necessary, update the Queries per Second property as needed after the upgrade.

Update JDBC Query Consumer Pipelines used for SQL Server CDC Data

The Microsoft SQL Server CDC functionality in the JDBC Query Consumer origin has been deprecated and will be removed in a future release.

For pipelines that use the JDBC Query Consumer to process Microsoft SQL Server CDC data, replace the JDBC Query Consumer origin with another origin:

- To read data from Microsoft SQL Server CDC tables, use the SQL Server CDC Client origin.
- To read data from Microsoft SQL Server change tracking tables, use the SQL Server Change Tracking origin.

Update MongoDB Destination Upsert Pipelines

With version 3.0.0.0, the MongoDB destination supports the replace and update operation codes, and no longer supports the upsert operation code. You can use a new Upsert flag in conjunction with Replace and Update.

After upgrading to version 3.0.0.0 or later, update the pipeline as needed to ensure that records passed to the destination do not use the upsert operation code (sdc.operation.type = 4). Records that use the upsert operation code will be sent to error.

In previous releases, records flagged for upsert were treated in the MongoDB system as Replace operations with the Upsert flag set.

If you want to replicate the upsert behavior from earlier releases, perform the following steps:

1. Configure the pipeline to use the Replace operation code.

Make sure that the sdc.operation.type is set to 7 for Replace instead of 4 for Upsert.

2. In the MongoDB destination, enable the new Upsert property.

Time Zones in Stages

With version 3.0.0.0, time zones have been organized and updated to use JDK 8 names. This should make it easier to select time zones in stage properties.

In the rare case that an upgraded pipeline uses a format not supported by JDK 8, edit the pipeline to select a compatible time zone.

Update Kudu Pipelines

With version 3.0.0.0, if the destination receives a change data capture log from the following source systems, you must specify the source system so that the destination can determine the format of the log: Microsoft SQL Server, Oracle CDC Client, MySQL Binary Log, or MongoDB Oplog.

Previously, the Kudu destination could not directly receive changed data from these source systems. You either had to include a scripting processor in the pipeline to modify the field paths in the record to a format that the destination could read. Or, you had to add multiple Kudu destinations to the pipeline - one per operation type - and include a Stream Selector processor to send records to the destination by operation type.

If you implemented one of these workarounds, then after upgrading, modify the pipeline to remove the scripting processor or the Stream Selector processor and the multiple destinations. In the Kudu destination, set the Change Log Format to the appropriate format of the log: Microsoft SQL Server, Oracle CDC Client, MySQL Binary Log, or MongoDB Oplog.

New Features and Enhancements in 3.0.0.0

This version includes the following new features and enhancements in the following areas.

Installation

- **Java requirement** - Data Collector now supports both Oracle Java 8 and OpenJDK 8.
- **RPM packages** - StreamSets now provides the following Data Collector RPM packages:
  - EL6 - Use to install Data Collector on CentOS 6 or Red Hat Enterprise Linux 6.
  - EL7 - Use to install Data Collector on CentOS 7 or Red Hat Enterprise Linux 7.

Previously, StreamSets provided a single RPM package used to install Data Collector on any of these operating systems.

Edge Pipelines

You can now design and run edge pipelines to read data from or send data to an edge device. Edge pipelines are bidirectional. They can send edge data to other Data Collector pipelines for further processing. Or, they can receive data from other pipelines and then act on that data to control the edge device.

Edge pipelines run in edge execution mode on StreamSets Data Collector Edge (SDC Edge). SDC Edge is a lightweight agent without a UI that runs pipelines on edge devices. Install SDC Edge on each edge device where you want to run edge pipelines.

You design edge pipelines in Data Collector, export the edge pipelines, and then use commands to run the edge pipelines on an SDC Edge installed on an edge device.

Origins

- **New Amazon SQS Consumer origin** - An origin that reads messages from Amazon Simple Queue Service (SQS). Can create multiple threads to enable parallel processing in a multithreaded pipeline.
- **New Google Cloud Storage origin** - An origin that reads fully written objects in Google Cloud Storage.
- **New MapR DB CDC origin** - An origin that reads changed MapR DB data that has been written to MapR Streams. Can create multiple threads to enable parallel processing in a multithreaded pipeline.
- **New MapR Multitopic Streams Consumer origin** - An origin that reads messages from multiple MapR Streams topics. It can create multiple threads to enable parallel processing in a multithreaded pipeline.
- **New UDP Multithreaded Source origin** - The origin listens for UDP messages on one or more ports and queues incoming packets on an intermediate queue for processing. It can create multiple threads to enable parallel processing in a multithreaded pipeline.
- **New WebSocket Client origin** - An origin that reads data from a WebSocket server endpoint.
- **New Windows Event Log origin** - An origin that reads data from Microsoft Windows event logs. You can use this origin only in pipelines configured for edge execution mode.
- **New Sensor Reader development origin** - A development origin that generates sample atmospheric data for edge pipelines.
- **Amazon S3 origin enhancements:**
  - The origin now produces no-more-data events and includes a new socket timeout property.
  - You can now specify the number of times the origin retries a query. The default is three.

- **Directory origin enhancement** - The origin can now use multiple threads to perform parallel processing of files.

- **JDBC Multitable Consumer origin enhancements:**

    - The origin can now use non-incremental processing for tables with no primary key or offset column.

    - You can now specify a query to be executed after establishing a connection to the database, before performing other tasks. This can be used, for example, to modify session attributes.

    - A new Queries Per Second property determines how many queries can be run every second.

      This property replaces the Query Interval property. For information about possible upgrade impact, see JDBC Multitable Consumer Query Interval Change.

- **JDBC Query Consumer origin enhancements:**

    - You can now specify a query to be executed after establishing a connection to the database, before performing other tasks. This can be used, for example, to modify session attributes.

    - The Microsoft SQL Server CDC functionality in the JDBC Query Consumer origin is now deprecated and will be removed from the origin in a future release. For upgrade information, see Update JDBC Query Consumer Pipelines used for SQL Server CDC Data.

- **Kafka Multitopic Consumer origin enhancement** - The origin is now available in the following stage libraries:
    - Apache Kafka 0.9, 0.10, 0.11, and 1.0
    - CDH Kafka 2.0 (0.9.0), 2.1 (0.9.0), and 3.0 (0.11.0)
    - HDP 2.5 and 2.6

- **Kinesis Consumer origin enhancement** - You can now specify the number of times the origin retries a query. The default is three.

- **Oracle CDC Client origin enhancements:**

    - When using SCNs for the initial change, the origin now treats the specified SCN as a starting point rather than looking for an exact match.

    - The origin now passes raw data to the pipeline as a byte array.

    - The origin can now include unparsed strings from the parsed SQL query for unsupported data types in records.

    - The origin now uses local buffering instead of Oracle LogMiner buffering by default. Upgraded pipelines require no changes.

    - The origin now supports reading the Timestamp with Timezone data type. When reading Timestamp with Timezone data, the origin includes the offset with the datetime data in the Data Collector Zoned Datetime data type. It does not include the

time zone ID.

- **SQL Server CDC Client origin enhancements** - You can now perform the following tasks with the SQL Server CDC Client origin:
    - Process CDC tables that appear after the pipeline starts.
    - Check for schema changes and generate events when they are found.
    - A new Capture Instance Name property replaces the Schema and Table Name Pattern properties from earlier releases.

      You can simply use the schema name and table name pattern for the capture instance name. Or, you can specify the schema name and a capture instance name pattern, which allows you to specify specific CDC tables to process when you have multiple CDC tables for a single data table.

      Upgraded pipelines require no changes.
- **UDP Source origin enhancement** - The Enable Multithreading property that enabled using multiple epoll receiver threads is now named Use Native Transports (epoll).

Processors

- **New Aggregator processor** - A processor that aggregates data within a window of time. Displays the results in Monitor mode and can write the results to events.
- **New Delay processor** - A processor that can delay processing a batch of records for a specified amount of time.
- **Field Type Converter processor enhancement** - You can now convert strings to the Zoned Datetime data type, and vice versa. You can also specify the format to use.
- **Hive Metadata processor enhancement** - You can now configure additional JDBC configuration properties to pass to the JDBC driver.
- **HTTP Client processor enhancement** - The Rate Limit now defines the minimum amount of time between requests in milliseconds. Previously, it defined the time between requests in seconds. Upgraded pipelines require no changes.
- **JDBC Lookup and JDBC Tee processor enhancements** - You can now specify a query to be executed after establishing a connection to the database, before performing other tasks. This can be used, for example, to modify session attributes.
- **Kudu Lookup processor enhancement** - The Cache Kudu Table property is now named Enable Table Caching. The Maximum Entries to Cache Table Objects property is now named Maximum Table Entries to Cache.
- **Salesforce Lookup processor enhancement** - You can use a new Retrieve lookup mode to look up data for a batch of records instead of record-by-record. The mode provided in previous releases is now named SOQL Query. Upgraded pipelines require no changes.

Destinations

- **New Google Cloud Storage destination** - A new destination that writes data to objects in Google Cloud Storage. The destination can generate events for use as dataflow triggers.

- **New KineticaDB destination** **-** A new destination that writes data to a Kinetica table.

- **Amazon S3 destination enhancement** - You can now specify the number of times the destination retries a query. The default is three.

- **Hive Metastore destination enhancement** - You can now configure additional JDBC configuration properties to pass to the JDBC driver.

- **HTTP Client destination enhancement** - You can now use the HTTP Client destination to write Avro, Delimited, and Protobuf data in addition to the previous data formats.

- **JDBC Producer destination enhancement** - You can now specify a query to be executed after establishing a connection to the database, before performing other tasks. This can be used, for example, to modify session attributes.

- **Kudu destination enhancement** - If the destination receives a change data capture log from the following source systems, you now must specify the source system in the Change Log Format property so that the destination can determine the format of the log: Microsoft SQL Server, Oracle CDC Client, MySQL Binary Log, or MongoDB Oplog.

- **MapR DB JSON destination enhancement** - The destination now supports writing to MapR DB based on the CRUD operation in record header attributes and the Insert and Set API properties.

- **MongoDB destination enhancements** - With this release, the Upsert operation is no longer supported by the destination. Instead, the destination includes the following enhancements:

    - Support for the Replace and Update operations.
    - Support for an Upsert flag that, when enabled, is used with both the Replace and Update operations.

  For information about upgrading existing upsert pipelines, see Update MongoDB Destination Upsert Pipelines.

- **Redis destination enhancement** - The destination now supports processing data using CRUD operations stored in record header attributes.

- **Salesforce destination enhancement** - When using the Salesforce Bulk API to update, insert, or upsert data, you can now use a colon (:) or period (.) as a field separator when defining the Salesforce field to map the Data Collector field to. For example, `Parent__r:External_Id__c` or `Parent__r.External_Id__c` are both valid Salesforce fields.

- **Wave Analytics destination rename** - With this release, the Wave Analytics destination is now named the Einstein Analytics destination, following the recent Salesforce rebranding. All of the properties and functionality of the destination remain the same.

Executors

- **Hive Query executor enhancement** - You can now configure additional JDBC configuration properties to pass to the JDBC driver.

- **JDBC Query executor enhancement** - You can now specify an Init Query to be executed after establishing a connection to the database, before performing other tasks. This can be used, for example, to modify session attributes.

Cloudera Navigator

Cloudera Navigator integration is now released as part of the StreamSets Commercial Subscription. The beta version included in earlier releases is no longer available with Data Collector. For information about the StreamSets Commercial Subscription, contact us.

For information about upgrading a version of Data Collector with Cloudera Navigator integration enabled, see Disable Cloudera Navigator Integration.

Credential Stores

- **CyberArk** - Data Collector now provides a credential store implementation for CyberArk Application Identity Manager. You can define the credentials required by external systems - user names or passwords - in CyberArk. Then you use credential expression language functions in JDBC stage properties to retrieve those values, instead of directly entering credential values in stage properties.

- **Supported stages** - You can now use the credential functions in all stages that require you to enter sensitive information. Previously, you could only use the credential functions in JDBC stages.

Data Collector Configuration

By default when Data Collector restarts, it automatically restarts all pipelines that were running before Data Collector shut down. You can now disable the automatic restart of pipelines by configuring the `runner.boot.pipeline.restart` property in the `$SDC_CONF/sdc.properties` file.

Dataflow Performance Manager / StreamSets Control Hub

- **StreamSets Control Hub** - With this release, we have created a new product called StreamSets Control Hub™ (SCH) that includes a number of new cloud-based dataflow design, deployment, and scale-up features. Since this release is now our core service for controlling dataflows, we have renamed the StreamSets cloud experience from "Dataflow Performance Manager (DPM)" to "StreamSets Control Hub (SCH)".

  DPM now refers to the performance management functions that reside in the cloud such as live metrics and data SLAs. Customers who have purchased the StreamSets Enterprise Edition will gain access to all SCH functionality and continue to have access to all DPM functionality as before.

  To understand the end-to-end StreamSets Data Operations Platform and how the products fit together, visit https://streamsets.com/products/.

- **Aggregated statistics** - When working with DPM, you can now configure a pipeline to write aggregated statistics to MapR Streams.

Data Formats

- **New NetFlow 9 support** - Data Collector now supports processing NetFlow 9 template-based messages. Stages that previously processed NetFlow 5 data can now process NetFlow 9 data as well.

- **Datagram data format enhancement** - The Datagram Data Format property is now named the Datagram Packet Format.

- **Delimited data format enhancement -** Data Collector can now process data using the Postgres CSV and Postgres Text delimited format types.

Expression Language

This release includes the following enhancements:

- **New field path expressions** - You can use field path expressions in certain stages to specify the fields that you want to use in an expression.

- **New field functions** - You can use the following new field functions in field path expressions:

    - **f:attribute()** - Returns the value of the specified attribute.

    - **f:type()** - Returns the data type of a field.

    - **f:value()** - Returns the value of a field.

- **New string functions** - The release includes the following new functions:

    - **str:isNullOrEmpty()** - Returns true or false based on whether a string is null or is the empty string.

    - **str:splitKV()** - Splits key-value pairs in a string into a map of string values.

Stage Libraries

- **New stage libraries -** This release includes the following new stage libraries:
    - Apache Kafka 1.0
    - Apache Kafka 0.11
    - Apache Kudu 1.5
    - Cloudera CDH 5.13
    - Cloudera Kafka 3.0.0 (0.11.0)
    - Hortonworks 2.6.1, including Hive 1.2
    - Hortonworks 2.6.2, including Hive 1.2 and 2.0
    - MapR version 6.0 (MEP 4)
    - MapR Spark 2.1 (MEP 3)

- **Legacy stage libraries** - Stage libraries that are more than two years old are no longer included with Data Collector. Though not recommended, you can still download and install the older stage libraries as custom stage libraries.

If you have pipelines that use these legacy stage libraries, you will need to update the pipelines to use a more current stage library or install the legacy stage library manually, For more information see Update Pipelines using Legacy Stage Libraries.

- **Statistics stage library enhancement** - The statistics stage library is now included in the core Data Collector installation.

Miscellaneous

- **New data type** - Data Collector now supports the Zoned Datetime data type.
- **New Data Collector metrics** - JVM metrics have been renamed Data Collector Metrics and now include general Data Collector metrics in addition to JVM metrics. The JVM Metrics menu item has also been renamed SDC Metrics.
- **Pipeline error records** - You can now write error records to Google Pub/Sub, Google Cloud Storage, or an MQTT broker.
- **Snapshot enhancements:**
  - Standalone pipelines can now automatically take a snapshot when the pipeline fails due to a data-related exception.
  - You can now download snapshots through the UI and the REST API.
- **Time zone enhancement** - Time zones have been organized and updated to use JDK 8 names. This should make it easier to select time zones in stage properties.

  In the rare case that your pipeline uses a format not supported by JDK 8, edit the pipeline to select a compatible time zone.

Fixed Issues in 3.0.0.0

The following table lists some of the known issues that are fixed with this release.

For the full list, click here.

JIRA	Description
SDC-7881	Support bundle should redact the jmx.json file.
SDC-7813	The Kinesis Firehose destination doesn't log error messages.
SDC-7603	When the JDBC Query Consumer origin performs an incremental query, it does not generate a no-more-data event if the table is empty.
SDC-7548	The Salesforce Lookup processor fails when a record ID includes the string 'FROM'.
SDC-7523	A webhook notification encounters a null pointer exception when the Pipeline Finisher executor stops a pipeline and transitions it to a Finished state.
SDC-7477	The JMS Producer destination doesn't support record functions in the JMS Destination Name property.

SDC-7431	When a stage processes Avro data, it should preserve the Avro field type as 'timestamp-millis' instead of 'long' when the incoming data is null or empty.
SDC-7428	When the polling interval for the OPC UA Client origin is set to greater than 5 minutes, the origin stops reading after the first batch.
SDC-7416	In some cases, the Vault credential store doesn't perform an initial login before running a renewal task.
SDC-7404	The commons-csv library should be upgraded from version 1.4 to version 1.5.
SDC-7333	The JDBC Multitable Consumer origin Query Interval property is applied to queries when switching between partitions as well as between tables. This causes unexpected delays in processing, especially when performing multithreaded partition processing or when processing partitioned tables.
SDC-7262	The Package Manager should not be available to install additional stage libraries for a Data Collector installation with Cloudera Manager.
SDC-7225	The JDBC Query Consumer origin should support an uppercase ${OFFSET} when the SQL query is defined in a resource file and loaded from the file at runtime using the runtime:loadResource function.
SDC-7206	A webhook notification encounters a null pointer exception when a pipeline transitions to an error state and the error message is null.
SDC-7009	To avoid pipelines being stuck in a STOPPING state, create a separate thread pool for stopping pipelines.
SDC-6982	HTTP stages should log the request data.
SDC-6550	When the Redis Consumer origin fails to connect to Redis, the pipeline becomes stuck in a STARTING state and the logs do not indicate the connection timeout error.

Known Issues in 3.0.0.0

Please note the following known issues with this release.

For a full list of known issues, check out our JIRA.

JIRA	Description
SDC-7986	A Data Collector registered with StreamSets Control Hub becomes unusable when the Data Collector is configured to use LDAP authentication. Workaround: Configure the Data Collector to use file-based authentication.
SDC-7903	Cluster mode pipelines that read from a MapR Streams Consumer origin fail to run

	on MapR 6.0.
SDC-6438	Do not use Cluster Streaming pipelines with MapR and Spark 2.x.
SDC-7872	Due to Oracle LogMiner returning an unsupported SQL format, Data Collector cannot parse changes to tables containing the XML type.
SDC-7761	The Java keystore credential store implementation fails to work for a Data Collector installed through Cloudera Manager. The jks-cs command creates the Java keystore file in the Data Collector configuration directory defined for the parcel. However, for Data Collector to access the Java keystore file, the file must be outside of the parcel directory. The CyberArk and Vault credential store implementations do work with a Data Collector installed through Cloudera Manager.
SDC-7645	The Data Collector Docker image does not support processing data using another locale. Workaround: Install Data Collector from the tarball or RPM package.
SDC-7448	You cannot run cluster streaming pipelines on MapR MEP 3 clusters at this time.
SDC-7290	Using incorrect offset column conditions in the JDBC Multitable Consumer origin generates an endless cycle of exceptions that requires restarting Data Collector to stop the exceptions.
SDC-6554	When converting Avro to Parquet on Impala, Decimal fields seem to be unreadable. Data Collector writes the Decimal data as variable-length byte arrays. And due to Impala issue IMPALA-2494, Impala cannot read the data.
SDC-6509	When using the Write to Another Pipeline pipeline error handling option in cluster batch pipeline, if the error handling pipeline encounters a problem and stops, the original pipeline stops with the message “Job has been finished” instead of indicating there was a problem with the error handling pipeline.
SDC-6438	The MapR distribution for Spark 2.x is not supported by cluster streaming pipelines.
SDC-6210	The show-vault-id command returns a NPE when Data Collector is installed on EC2 with IPv6 enabled. Workaround: If you can run Data Collector without IPv6, in the /etc/sysctl.conf file, disable IPv6 by configuring the disable IPv6 property as follows: `net.ipv6.conf.all.disable_ipv6 = 1`
SDC-6077	The Field Remover processor does not remove list fields from list-map data.

SDC-5758	Due to expected Kudu behavior, if Data Collector is configured to use Kerberos, but your Kudu installation does not, Data Collector cannot connect. This applies to Kudu version 1.2 and later. Workaround: If necessary, install a separate Data Collector that does not have Kerberos enabled.
SDC-5521	Kerberos-enabled pipelines that are valid using an Apache Kafka stage library can fail validation when using a Cloudera distribution of Kafka. The Apache Kafka client libraries have been modified by StreamSets to allow connectivity to Kerberized Kafka without requiring a JAAS configuration file. The Apache Kafka stage libraries (version 0.9 and greater) do not require a JAAS configuration file when enabling Kerberos. For CDH Kafka versions, a JAAS configuration is still required. Workaround: Include a JAAS configuration file on the classpath by modifying the `sdc-env.sh` or `sdcd-env.sh` file to include the following option in the SDC_JAVA_OPTS environment variable: `-Djava.security.auth.login.config=<path-to-jaas-config>` Or, you can use the matching version Apache Kafka stage library, instead of the Cloudera stage libraries.
SDC-5357	The Jython Evaluator and stages that can access the file system - such as the Directory and File Tail origins - can access all pipeline JSON files stored in the $SDC_DATA directory. This allows users to access pipelines that they might not have permission to access within Data Collector. Workaround: To secure your pipelines, complete the following tasks: 5. Remove the Jython stage library and use the Groovy Evaluator or JavaScript Evaluator processor instead of the Jython Evaluator. 6. Update the Data Collector security policy file, `$SDC_CONF/sdc-security.policy`, so that Data Collector stages do not have `AllPermission` access to the file system. Update the security policy for the following code bases: streamsets-libs-extras, streamsets-libs, and streamsets-datacollector-dev-lib. Use the policy file syntax to set the security policies.
SDC-5325	Cluster mode pipelines that read from a MapR cluster fail when the MapR cluster uses username/password login authentication. Workaround: On the Cluster tab for the pipeline, add the following Java property to the Worker Java Options property: `-Dmaprlogin.password.enabled=true`
SDC-5141	Due to a limitation in the Javascript engine, the Javascript Evaluator issues a null pointer exception when unable to compile a script.
SDC-5039	When you use the Hadoop FS origin to read files from all subdirectories, the origin cannot use the configured Hadoop FS User as a proxy user to read from HDFS.

	Workaround: If you need to use a proxy user to read from all subdirectories of the specified directories, set the HADOOP_PROXY_USER environment variable to the proxy user in libexec/_cluster-manager script, as follows: `export HADOOP_PROXY_USER = <proxy-user>`
SDC-4212	If you configure a UDP Source or UDP to Kafka origin to enable multithreading after you have already run the pipeline with the option disabled, the following validation error displays: `Multithreaded UDP server is not available on your platform.` Workaround: Restart Data Collector.
SDC-3944	The Hive Streaming destination using the MapR library cannot connect to a MapR cluster that uses Kerberos or username/password login authentication.
SDC-3133	When you upgrade Data Collector from the RPM package, the environment configuration file `$SDC_DIST/libexec/sdcd-env.sh` is overwritten. Workaround: Back up the `sdcd-env.sh` file before you upgrade.
SDC-2374	A cluster mode pipeline can hang with a CONNECT_ERROR status. This can be a temporary connection problem that resolves, returning the pipeline to the RUNNING status. If the problem is not temporary, you might need to manually edit the pipeline state file to set the pipeline to STOPPED. Edit the file only after you confirm that the pipeline is no longer running on the cluster or that the cluster has been decommissioned. To manually change the pipeline state, edit the following file: `$SDC_DATA/runInfo/<cluster pipeline name>/<revision>/pipelineState.json` In the file, change CONNECT_ERROR to STOPPED and save the file.

Contact Information

For more information about StreamSets, visit our website: https://streamsets.com/.

Check out our Documentation page for doc highlights, what's new, and tutorials: streamsets.com/docs

Or you can go straight to our latest documentation here:
https://streamsets.com/documentation/datacollector/latest/help

To report an issue, ask for help from our Google group or Slack channel, or find out about our next meetup, check out our Community page: https://streamsets.com/community/.

For general inquiries, email us at info@streamsets.com.