Pipeline Configuration

Contents

Pipeline Configuration

What is a Pipeline?

A pipeline describes the flow of data for the Data Collector.

Configure a pipeline on the *pipeline canvas*. To configure a pipeline, you add and configure a single origin stage to represent source data, processor stages to perform processing and routing, and one or more destination stages to represent target data. Connect the stages to create the flow of data from origin to destinations.

The canvas automatically saves and validates your work. The pipeline canvas provides warnings for parts of the pipeline that are not yet complete or valid. You can view a full list of issues to help you configure the pipeline. You can also perform a data preview to view how the pipeline transforms source data.

When you start a pipeline, the Data Collector runs the pipeline until you stop the pipeline or shut down the Data Collector. While the pipeline runs, you can monitor the pipeline and configure alerts to verify that the pipeline performs as expected.

Data Collector Console - Edit Mode

The following image shows the Data Collector console when you configure a pipeline:

Area / Icon	Name	Description
1	Pipeline canvas	Displays the pipeline. Use to configure the pipeline data flow.
2	Stage list	Lists the stages in the pipeline. Use to select a stage to edit. You can also select the stage in the canvas.
3	Properties panel	Displays the properties of the pipeline or selected stage when you configure a pipeline.

Area / Icon	Name	Description
4	Stage library	List of available stages. Use to add stages to the pipeline. You can drag a stage to a location on canvas or click a stage to add it to the end of the pipeline. You can view all stages, stages by type, or stages by library.
	Stage Library icon	Toggles the display of the stage library.
	Issues icon	Displays the number of validation issues in the pipeline. Click to view a detailed list of issues.
	Preview icon	Starts a data preview.
	More icon	Provides additional actions for the pipeline. Use to reset the origin for the pipeline.
	Stream link icon	Indicates the flow of data through the pipeline. Select to configure data rules and alerts.

Replicating Streams

When you connect a stage to multiple stages, the stage passes all data to all connected stages. You can configure required fields for connected stages to discard records before they enter the stage, but by default all records are passed.

For example, in the following pipeline, all of the data from the Directory origin passes to both branches of the pipeline for different types of processing. But you might optionally configure required fields for the Field Splitter or Value Replacer to discard any records that are not needed.

To route data based on more complex conditions, use a Stream Selector.

Merging Streams

You can merge streams of data in a pipeline by connecting two or more stages to the same downstream stage. When you merge streams of data, the Data Collector channels the data from all streams to the same stage, but does not perform a join of records in the stream.

For example, in the following pipeline, the Routes Data Stream Selector sends data with null values to Replaces Null Values Value Replacer.

The data from stream 2 of Routes Data and all data from Replaces Null Values pass to Field-Level Expressions for further processing, but in no particular order and with no record merging.

Note: The pipeline validation does not prevent duplicate data. To avoid writing duplicate data to destinations, configure pipeline logic to remove duplicate data or to prevent the generation of duplicate data.

Error Handling

You can configure the default error handling for a pipeline. Some stages include error handling properties that override the pipeline default. When a Data Collector encounters an unexpected error, it stops the pipeline.

You can configure a pipeline to discard error records or to save error records. When you save error records, you define the directory to use and the parameter for creating additional files. Error files are written to the directory with the following naming convention: `records-<file number>.json.`

Each stage in the pipeline includes built-in resilience. Some stages also include configurable error handling. When a stage without explicit error handling options encounters an error record, the Data Collector uses the default error handling for the pipeline.

For example, when a Kafka Consumer origin stage reads JSON data with a maximum object length of 4096 characters and the stage encounters an object with 5000 characters, the stage discards or saves the object based on the pipeline error handling configuration.

When a stage includes an error handling option, the stage configuration overrides the pipeline configuration.

For example, a Field Splitter splits field values into three parts and is configured to discard records that cannot be split. Even if the pipeline is configured to save error records, the Field Splitter stage discards any record that cannot be split as requested.

Send to Error Stage Option

Any stage with an error handling property includes the **Send to Error** option. The Send to Error option passes the error record to the pipeline for error handling.

The record is either discarded or saved to an error record file based on how the pipeline is configured.

Reprocessing Error Records

You can use a Directory origin in a error pipeline to reprocess error record files. When you reprocess error record files, do not edit or rename the files. The Directory origin expects the files as generated by the original pipeline.

In the error pipeline, include the Directory origin and configure it to use the SDC Records data format. The SDC Records data format provides information for reprocessing the original records with the necessary error handling.

The SDC Records format provides the following information:

- The original source record, as read by the pipeline that generated the error. Changes to the record that might have occurred within the original pipeline are not preserved.
- Additional metadata that includes:
  - The path the record took through the pipeline, including the stage that discarded the record.
  - Information about the error, including the error code and message.

When you create the error pipeline, you can use error and record functions provided by the expression language to route different types of error records through different transformation logic to resolve the error and write the corrected record to destinations.

For example, your error files contain records with invalid product IDs discarded as a required field from the first processor in the pipeline, a Field Filter. They also include records discarded by a Field Converter for attempting an invalid data type conversion.

In the error pipeline, you can use a Stream Selector to route the records discarded from the Field Filter to a branch that corrects product IDs, and the records discarded by the Field Converter to a branch that performs a valid data type conversion.

Delivery Guarantee

When you configure a pipeline, you define how you want data to be treated: Do you want to prevent the loss of data, or do you want to prevent the duplication of data?

The Delivery Guarantee pipeline property offers the following choices:

At least once

The Data Collector ensures that the pipeline processes all data.

If a failure causes the Data Collector to stop while processing a batch of data, when restarts, it reprocesses the batch.

This option ensures that no data is lost. But if the failure occurs as the Data Collector writes data to destinations, up to one batch data might be duplicated in the target.

At most once

The Data Collector ensures that data is not processed more than once.

If a failure causes the Data Collector to stop while processing a batch of data, when it starts up, it begins processing with the next batch of data.

This option ensures duplicate data is not written to the target due to reprocessing. But if the failure occurs as the Data Collector writes data to destinations, up to one batch of data might not be processed.

Required Fields

You can configure required fields for any processor and most destination stages. A required field must exist in a record to allow the record to enter the stage for processing. When a record does not include a required field, the stage passes the record to the pipeline for error handling.

You can configure different required fields in different stages. Configure required fields based on your business requirements or in order to minimize processing errors.

For example, a Field Hasher encodes social security data in the /SSN field. To ensure processing only records that include social security numbers, make /SSN a required field.

SDC Record Data Format

SDC Record is a data format used by the Data Collector to generate error record files for the pipeline. The Data Collector can also use the SDC Record format to generate output files.

Use the SDC Record format to process files produced by another Data Collector pipeline, or to produce files that you might pass to another Data Collector pipeline.

The SDC Record Files destination writes all records using the SDC Record format. You can also use Hadoop FS and the Kafka Producer to write files that use the SDC Record format.

You can use the Directory or Kafka Consumer origins to read SDC Record files.

Validation

The Data Collector console provides two types of validation:

Implicit validation

Implicit validation occurs by default as the console saves your changes. Implicit validation lists missing or incomplete configuration, such as an unconnected stage or a required property that has not been configured.

Errors found by implicit validation display in the Issues list.

Explicit validation

Explicit validation occurs on demand when you click the Validate icon. Explicit validation becomes available when implicit validation passes.

Configuring a Pipeline

Configure a pipeline to define the stream of data. After you configure the pipeline, you can start the pipeline.

A pipeline can include the following components:

- A single origin stage
- Multiple processor stages

- Multiple destination stages

1. If this is the first pipeline for the Data Collector, click **Create Pipeline**.

   If you have created other pipelines, if necessary, use the **Toggle Library** icon to display the Library, and click the **Create New Pipeline** icon.
2. In the **New Pipeline** window, enter a pipeline name and optional description, and click **Save**.
   The pipeline canvas displays the new name. The Properties panel displays the pipeline properties.
3. In the Properties panel, on the **General** tab, configure the **Delivery Guarantee** property:

Pipeline Property	Description
Delivery Guarantee	Determines how the Data Collector handles data after an unexpected event causes the pipeline to stop running: • At Least Once - Ensures all data is processed and written to the destination. Might result in duplicate rows. • At Most Once - Ensures that data is not reprocessed to prevent writing duplicate data to the destination. Might result in missing rows. Default is **At least once**.

4. Click the **Error Records** tab and configure the following error handling option:

Error Records Property	Description
Error Record Handling	Determines how to handle records that cannot be processed as expected. Use one of the following options: • Trash - Discards error records. • Records to File - Saves error records to a file.

5. On the **Bad Records - Records to File** tab, configure error file properties:

Error File Property	Description
Directory	Local directory for error record files.
File Wait Time (secs)	Number of seconds the Data Collector waits for error records. After that time, it creates a new error record file. You can enter a number of seconds or use the default expression to enter the time in minutes.
Max File Size (MB)	Maximum size for error files. Exceeding this size creates a new error file. Use 0 to write to avoid using this property.

6. Use the **Stage Library** to add an origin stage. In the Properties panel, configure the stage properties.

   For configuration details about origin stages, see *Origins*.
7. Use the **Stage Library** to add the next stage that you want to use, connect the origin to the new stage, and configure the new stage.

   For configuration details about processors, see *Processors*.

   For configuration details about destinations, see *Destinations*.
8. Add additional stages as necessary.
9. When the pipeline is valid, you can use the **Preview** icon to preview data to fine-tune pipeline configuration.

   For more information, see *Data Preview*.
10. When the pipeline is complete, use the **Start** icon to run the pipeline.

When the Data Collector starts the pipeline, the Monitor panel displays real-time statistics for the pipeline.

Index