

# Dead-letter queue

A dead-letter queue is a connector to which Conduit redirects records that can't be delivered or processed successfully. Any existing Conduit destination connector can be used for the purpose of a dead-letter queue.

For a record to be considered processed successfully it needs to be acked. A record is acked (acknowledged) when it reaches all destinations in a pipeline or it is filtered out by a processor. Likewise, a record is nacked (negatively acknowledged) when a processor or destination return an error while processing the record. In other words, an ack indicates success and a nack indicates failure.

When a record can't be delivered or processed, it is sent to the dead-letter queue connector so that it can be handled separately from the other records in the pipeline. This can be useful for debugging, since it allows you to isolate records that are causing problems and figure out why they are failing. Additionally, it can help to prevent the rest of the records in the pipeline from being blocked or delayed due to the failure of a single record.

## Nack window and threshold

Rerouting records to a dead-letter queue causes the pipeline to keep running despite of nacked (negatively acknowledged) records. However, experiencing a lot of nacked records can indicate that a pipeline is corrupted or malfunctioning. In such a case it might be better to stop the pipeline to prevent further errors and investigate the problem before resuming.

Conduit can monitor a fixed number of last records and track which ones are acked or nacked. As new records are processed, they are added into the window while the oldest ones are discarded to make room for them. This allows the window to "slide" through records, always containing the latest records. Conduit keeps track of the number of nacks in the sliding window and stops the pipeline if the threshold gets exceeded. This can be useful for ensuring that the pipeline is operating correctly and effectively.

Below is a visual representation of a sliding window monitoring 5 records with the nack threshold set to 2. Notice that once the threshold is exceeded, the pipeline stops.

## Configuring a dead-letter queue

By default, a Conduit pipeline will stop immediately when a record gets nacked. You can make the pipeline more fault-tolerant by enabling a dead-letter queue. The first step is to change the window size and nack threshold:

- If the window size is greater than 1 and the nack threshold is 0, then the first nacked record will cause the pipeline to stop running immediately. Such a configuration essentially disables the dead-letter queue. This is the default configuration.
- If the window size is set to 0, then no records are monitored and all nacked records will be rerouted to the dead-letter queue without any limits. Such a configuration essentially disables the nack window and enables the dead-letter queue.
- If the window size is greater than 1 and the nack threshold is greater than 0, then nacked records will be sent to the dead-letter queue. Additionally, Conduit will keep track of the number of nacks in the current window and stop the pipeline if the threshold gets exceeded.

If the window settings are adjusted accordingly, then Conduit will route nacked records to the builtin log connector by default, causing nacked records to show up in Conduit logs. You can adjust the connector used as a dead-letter queue to any Conduit connector you wish.

Below is an example of a pipeline config file that uses the file connector as a dead-letter queue and stops the pipeline if more than 2 out of the last 5 records are nacked.

version :

2.2 pipelines : -

id : dlq - example connectors :

## define source and destination connectors

...

dead-letter-queue : plugin :

"builtin:file"

## **use builtin file plugin as DLQ connector**

settings : path :

"./dlq.out"

## **route records to file ./dlq.out**

window-size :

5

## **monitor last 5 records**

window-nack-threshold :

2

## **tolerate up to 2 negatively acknowledged records**

Please check the [pipeline config file documentation](#) for more information about configuring a pipeline. [Edit this page](#) [Previous API](#) [Next Metrics](#)