

# **Optional: NiFi Error Handling**

# Error handling

## Dead Letter Queue

Add a new PutFile processor to store failed events to disk in order to create a Dead Letter Queue.

Double click the PutFile processor and set the following property:

- Directory: `/opt/nifi/nifi-current/state/dlq`

Now you need to configure your existing processors to send error events to the dead letter queue.

Hover over a ConsumeKafkaRecord\_2\_6 (doesn't matter which one), then drag its arrow to the PutFile processor, and specify the `parse.failure` relationship for the connection.

Repeat for all other ConsumeKafkaRecord\_2\_6 processors.

## Dead Letter Queue (2)

Now, hover over a PutElasticsearchRecord (doesn't matter which one), then drag its arrow to the PutFile processor, and specify the `failure` relationship for the connection.

Also, drag the same PutElasticsearchRecord's arrow to the PutFile processor again, and specify the `errors` relationship for the connection.

Repeat for all other processors.

This creates a Dead Letter Queue that will store all events that the processor could not handle for some reason or another.

# Error Handling (2)

## Retry Configuration

Double click the PutElasticsearchRecord, navigate to the relationships tab, and select the "retry" option for the `retry` relationship. Leave the retry algorithm as-is, and press apply.

This will automatically configure 10 retries with exponential backoff.

The screenshot shows the 'RELATIONSHIPS' tab of a configuration interface. It features three sections for relationship types: 'errors', 'failure', and 'retry'. Each section has checkboxes for 'terminate' and 'retry'. The 'retry' section is selected, and its configuration is shown on the right. The 'Number Of Retry Attempts' is set to 10, the 'Retry Back Off Policy' is set to 'Penalize', and the 'Retry Maximum Back Off Period' is set to 10 mins.

SETTINGS	SCHEDULING	PROPERTIES	RELATIONSHIPS	COMMENTS
<b>Automatically Terminate / Retry Relationships</b>				
<b>errors</b>				
<input checked="" type="checkbox"/> terminate <input type="checkbox"/> retry				
If a "Result Record Writer" is set, any Record(s) corresponding to Elasticsearch document(s) that resulted in an "error" (within Elasticsearch) will be routed here.				
<b>failure</b>				
<input checked="" type="checkbox"/> terminate <input type="checkbox"/> retry				
All flowfiles that fail for reasons unrelated to server availability go to this relationship.				
<b>retry</b>				
<input checked="" type="checkbox"/> terminate <input checked="" type="checkbox"/> retry				
All flowfiles that fail due to server/cluster availability go to this relationship.				
<b>Number Of Retry Attempts</b>				
10				
<b>Retry Back Off Policy</b>				
<input checked="" type="radio"/> Penalize <input type="radio"/> Yield				
<b>Retry Maximum Back Off Period</b>				
10 mins				

# Next Steps

## Congratulations!

You have successfully completed the following:

- Learned how to handle failure relationships
- Deployed a Dead Letter Queue
- Configured retry for a processor

## What's Next?

You can go back to the main session [here](#).