Standalone Processors

These are processors you'd write yourself in cases theBuilt-in ones don't meet your needs.

How to write one

Thanks to ourWeb Assembly (Wasm) processor you can start writing processors in any language that can be compiled to Web Assembly. As a start, Conduit already provides aconduit-processor-sdk that will let youwrite a processor in Go.

Where to put them

By default, standalone processors are expected to be found in a folder namedprocessors alongside of your pipelines orstandalone connectors :

|

Conduit binary

├── conduit |

Folder with pipeline configurations (yaml files)

├── pipelines |

Folder with standalone connectors (binary files)

├── connectors |

Folder with standalone processors (wasm files)

└── processors However, in case you need to reference processors in a different location, you could use the-processors.path flag when running Conduit:

./conduit -processors.path /my-custom-processors-path

Using the**conduit-processor-sdk**

Assuming you use ourconduit-processor-sdk , this is how a processorplugin written in Go could look like. In the following example, we're going to be adding aprocessed field to each record processed by our pipeline:

```
//go:build wasm

package main

import

( "context"

"github.com/conduitio/conduit-commons/opencdc" sdk "github.com/conduitio/conduit-processor-sdk" )

func

main ( )

{ sdk . Run ( sdk . NewProcessorFunc ( sdk . Specification { Name :

"simple-processor" , Version :

"v1.0.0" } , func ( ctx context . Context , record opencdc . Record )

( opencdc . Record ,
```

error)

{ record . Payload . After . (opencdc . StructuredData) ["processed"]

=

true return record ,

nil } ,)) After that, you'd need to compile it, and locate its.wasm file into the desiredprocessors directory as previously mentioned:

GOARCH

wasm GOOS = wasip1 go build -o simple-processor.wasm main.go

Using it in your pipeline

As mentioned in ourGetting Started page , in order to use a processor in your pipeline, you need to update itsconfiguration file andreference it accordingly:

version :

2.2 pipelines : -

id : example - pipeline connectors :

define source and destination connectors

...

processors : -

id : add - processed - field plugin : standalone : simple - processor When running your pipeline again, you should expect seeing a newprocessed field on every record processed.

info If you end up writing a standalone processor you'd like to share with the community, please let us know! We'd love to hear from you by:

- Joining our Discord
- Posting a comment on GitHub Discussions Edit this page Previous webhook.http Next Build your own