



Configuration Guide

Process Integration Pipeline Extension - Restart via Data Store

June 2025

Table of contents

Prerequisites	3
Documentation	4
Configuration steps on SAP Cloud Integration	8
Configure and Deploy mandatory Integration Artefacts of DS Restart Extension package	8
Assign Data Store Restart Configuration for Configuration Scenarios in Partner Directory	9
Enable Custom Exception Handling in generic IFlows of Process Integration Pipeline package	10
Create Restart Job Profiles in Partner Directory	11
Create a Restart Job Integration Flow from Template, Configure Parameters and Deploy	12
(Optional) Configure and Deploy Integration Flows for On-Demand Restarts or Data Store management via HTTP API	14
Appendix	17

Prerequisites

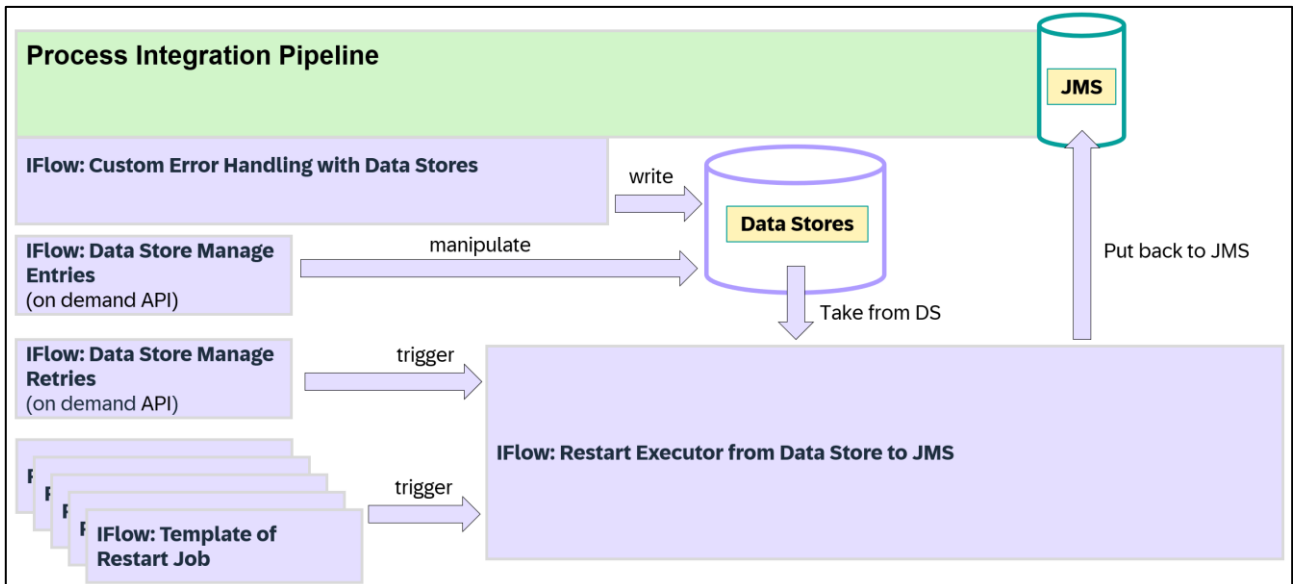
This package extends the [Process Integration Pipeline - Generic Integration Flows and Templates](#) package. To use it effectively, ensure the following prerequisites are met:

- The [Process Integration Pipeline - Generic Integration Flows and Templates](#) package (minimum required version is 1.0.10 or higher) is available in your Cloud Integration tenant. All necessary configurations, such as JMS queue activation, must be completed, and the generic Integration Flows (IFlows) must be deployed. The Restart Extension is compatible with both the [full pipeline](#) (all four generic pipeline stages) and a [shorter version with 2 persistence steps](#) (Integrated Messaging Runtime Async).
- Your asynchronous interfaces are connected to the Process Integration Pipeline, and you require enhanced restart capabilities, including Data Store usage and automatic retries through a configurable Restart Job.

Documentation

The following schema illustrates the logical connections between Integration Flows in the Integration Package. The package consists of five Integration Flows and one central Script Collection. Four of these flows can be deployed with minimal configuration, while a template Integration Flow is designed to be copied into a custom package. This allows for the creation of multiple versions of a Restart Job (represented by an Integration Flow template with a timer start event) with interface-specific configurations.

Integration Package Logical Connections



Schema Component Descriptions:

- **Pipeline DS Retry - Custom Error Handling with Data Stores.** This Integration Flow implements Custom Error Handling for exceptions occurring in the generic steps of the Process Integration Pipeline. It leverages the public Pipeline extension exit, as described in the [Custom Exception Handling documentation](#). Its primary function is to determine whether retries should continue through the JMS persistence layer or, if all configured JMS retries are exhausted, to move the message out of the queue into the Data Store while capturing all necessary restart attributes and forming the Data Store entry body.
- **Pipeline DS Retry - Restart Executor from Data Store to JMS.** This Integration Flow handles message reprocessing by moving messages from the Data Store back to the source JMS processing queue, allowing the JMS layer to manage the actual retry process. It is triggered via Process Direct and requires XML input specifying the source Data Store, maximum retry count, and the retry subject – either a list of Data Store entry names or an Interface PID for dynamically retrying all associated Data Store entries.
- **Pipeline DS Retry - Template - Restart Job.** This Integration Flow represents a Restart Job that automatically reprocesses messages from the Data Store based on assigned restart configuration profiles. Execution is scheduled using a Start Timer event. Provided as a template, this Integration Flow allows for the creation of multiple customized copies in a custom package, with different Job Profiles maintained in the Partner Directory.
- **Pipeline API - Data Store Manage Retries (Optional).** This Integration Flow enables on-demand restart operations via API clients like Postman. It exposes an HTTP endpoint to trigger restarts for specified Data Store entries or based on selection criteria. Additional operations allow users to retrieve

a list of Data Stores in the tenant, fetch entries from a specific Data Store, and move entries between Data Stores (e.g., transferring entries from the “Retry” Data Store to the “NoRetry” Data Store to prevent automatic retriggering).

- **Pipeline API - Data Store Manage Entries** (Optional). This Integration Flow provides API-based management of Data Store entries, separate from the "Manage Retries" API, to enforce role-based access. It allows users to read messages from a selected Data Store, delete messages (individually or based on query conditions), and add messages to the Data Store (this capability can be disabled at the Integration Flow level).

Data Store Entry Name Format

The Data Store entry format determines the available restart query options. The maximum allowed length for an entry name is 255 characters. The format follows a tilde (~)-delimited structure:

PartnerId~PipelineStage(~Receiver)~MplId~UTC-Timestamp(~RetryCount)

- **PartnerId (Mandatory)** – A unique identifier in the Process Integration Pipeline that defines the configuration scenario. It is derived from the sender system name and sender interface name. More details can be found in the Partner ID documentation.
- **PipelineStage (Mandatory)** – Identifies the stage in the Process Integration Pipeline where the exception occurred. It uses predefined two-letter identifiers enclosed in parentheses (e.g., (IB)), resulting in a four-character code:
 - o **(IB)** – Inbound Stage: The message was received by an inbound Integration Flow, stored in the first JMS queue, and encountered an exception during the next processing step.
 - o **(RD)** – Receiver Determination Stage: The inbound conversion was successful, but an error occurred while identifying the receiver system.
 - o **(ID)** – Interface Determination Stage: The receiver system was identified, but an error occurred while determining the target inbound interface for that receiver (i.e., the Process Direct endpoint of the outbound Integration Flow).
 - o **(OB)** – Outbound Stage: The message was successfully routed, but an error occurred during message conversion to the outbound format or while transmitting data to the receiver. This is the most common error-handling stage.
 - o **(NA)** – The error originated from an unknown or custom Integration Flow (e.g., a copy of the generic flow or an interface-specific flow connected to the retry extension).
- **Receiver (Optional)** – The receiver business system or alias to which the message is to be delivered. This field is optional because in the |IB| and |RD| stages, the receiver is still unknown. It is assigned only starting from the |ID| stage.
- **MplId (Mandatory)** – The Message Processing Log ID (Message GUID) of the erroneous message, a 28-character unique identifier assigned to each message in Cloud Integration.
- **UTC-Timestamp (Mandatory)** – A timestamp marking when the message first entered the Custom Error Handling Integration Flow. It follows the ISO 8601 format:


```
yyyy-MM-dd'T'HH:mm:ss.SSS'Z'
```

 (Example: 2025-04-20T06:51:11.123Z)
- **RetryCount (Optional)** – A counter tracking Data Store restart attempts. For a new erroneous message using Data Store-based restarts, this counter starts at 0 and increments with each retry attempt. This prevents exceeding the maximum allowed retries (MaxDataStoreRetries).

Pipeline Stage Determination

The pipeline stage is determined based on the prefix of the source pipeline Integration Flow's technical ID. The logic is as follows:

- If the technical ID starts with `com.sap.integration.cloud.pipeline.generic.step02.inbound.*` → Stage: (IB)
- If the technical ID starts with `com.sap.integration.cloud.pipeline.generic.step02.integrated.*` → Stage: (IB)
- If the technical ID starts with `com.sap.integration.cloud.pipeline.generic.step04.receiver.*` → Stage: (RD)
- If the technical ID starts with `com.sap.integration.cloud.pipeline.generic.step05.interface.*` → Stage: (ID)
- If the technical ID starts with `com.sap.integration.cloud.pipeline.generic.step06.outbound.*` → Stage: (OB)
- For all other cases → Stage: (NA)

Important Note:

When copying generic Integration Flows — for example, creating a receiver-specific Outbound Pipeline Integration Flow — ensure that the technical ID of the copied flow uses the appropriate stage prefix (e.g., for outbound flows, it should begin with the prefix corresponding to stage “(OB)”).

Transaction Handling Remark

The Process Integration Pipeline package utilizes the JMS persistence layer exclusively. This extension package introduces the use of Data Stores, which rely on the database persistence layer and use JDBC-based access.

Since Cloud Integration [does not support distributed transactions, it is not possible to execute JMS and JDBC transactions within the same transactional context](#). As a result, when a message is transferred from a JMS queue to a Data Store, there is a low probability of message duplication. For example, if the JDBC transaction is successfully committed but the subsequent JMS transaction fails, the message remains in the source JMS queue and is retried. This can result in the same message being processed again, with a new Data Store entry ID assigned. A similar scenario may theoretically occur in reverse — when a message is transferred from the retry Data Store to the JMS queue. However, the likelihood of such duplicates is very low, and message loss is not possible in any case.

To mitigate the risk of such rare duplicates — particularly in scenarios where duplicates are unacceptable— you can implement the [Idempotent Process Call](#). This pattern ensures that a unique ID used in a process call is stored in the idempotency repository (retained by default for 90 days), allowing duplicates to be detected and ignored. For implementation details, refer to the [Design Guidelines: "Quality of Service Exactly Once"](#).

Data Store Entry Body Format

When a message is written to the Data Store, its actual message body and optional attachments are Base64-encoded. Additionally, a message header is included, which contains essential metadata for reprocessing:

- The source JMS queue where the message will be placed for reprocessing
- The entry name of the Data Store record
- The Data Store name used for reprocessing
- The maximum retry count assigned to the interface in the Partner Directory (or a default of 3 retries from the Data Store)
- The exception message, also stored in Base64-encoded format

Below is the XML structure of the Data Store entry:

```
<?xml version="1.0" encoding="UTF-8"?>
<Message>
  <MsgHeader>
    <RestartQueue>${property.SourceJMSQueue}</RestartQueue>
    <EntryID>${property.EntryID}</EntryID>
    <RetryDataStore>${property.RetryDataStore}</RetryDataStore>
    <MaxRetryCount>${property.MaxDataStoreRetries}</MaxRetryCount>
    <Error>exception.message.Base64encoded</Error>
  </MsgHeader>
  <Payload>Payload.Base64encoded</Payload>
  <Attachments> (minOccurs="0")
    <Attachment> (maxOccurs="unbounded")
      <AttachmentName>attachmentName</AttachmentName>
      <AttachmentContentType>contentType</AttachmentContentType>
      <AttachmentContent>content.Base64</AttachmentContent>
    </Attachment>
  </Attachments>
</Message>
```

Data Store Options

The extension package utilizes two data store options: RetryStore and NoRetryStore. The Integration Flow “Pipeline DS Retry - Custom Error Handling with Data Stores” determines – based on the number of JMS retries, PartnerId configuration in Partner Directory and the pipeline Stage where the error occurred – into which kind of Data Store the message is placed.

- **RetryStore** – If a message should be automatically restarted in case of an error in inbound or outbound processing or in the integrated runtime, it will be put into this store. The messages in this store option can be restarted via the Restart Job Template Integration Flow. If the maximum number of retries is not reached for restarted messages, they will be put back into this store with an increased retry counter.
- **NoRetryStore** – If a message should not be restarted automatically in case of failing receiver or interface determination or when the maximum number of retries have been reached by an already restarted message, it will be put into this store. The messages in this store option can be restarted explicitly via the Integration Flow “Pipeline API - Data Store Manage Retries”.

Configuration steps on SAP Cloud Integration

1. [Configure and Deploy mandatory Integration Artefacts of DS Restart Extension package.](#)
2. [Assign Data Store Restart Configuration for Configuration Scenarios in Partner Directory.](#)
3. [Enable Custom Exception Handling in generic IFlows of Process Integration Pipeline package.](#)
4. [Create Restart Job Profiles in Partner Directory.](#)
5. [Create a Restart Job Integration Flow from Template, Configure Parameters and Deploy.](#)
6. [\(Optional\) Configure and Deploy Integration Flows for On-Demand Restarts or Data Store management via HTTP API.](#)

Configure and Deploy mandatory Integration Artefacts of DS Restart Extension package

1. Deploy Script Collection *"Pipeline DS Retry - Script Collection"*.
2. Configure Integration Flow *"Pipeline DS Retry - Custom Error Handling with Data Stores"*. Configuration steps below are optional and can be applied if default values need to be customized.
 - Set Process Direct Address in the Sender tab equal to the value of the Process Direct Address under Receiver tab and Receiver name "CustomErrorFlow" in the main Process Integration Pipeline Integration Flows.
 - Default is *"/pip/custom/errorHandling"*.
 - Set Expiration and Alerting periods in days for the entries written to the Data Store. This is a global configuration for all interfaces handled by the Restart via Data Store extension
 - Default values: 2 days for Alerting period and 90 days for Expiration period.
 - Configure text for Custom Statuses which are set by the Integration Flow when the message is stored in the *"Retry"* or *"NoRetry"* Data Stores.
 - Default Custom Status is either *"MaxRetriesExceeded_MovedToRetryStore"* or *"MaxRetriesExceeded_MovedToNoRetryStore"*.
3. Deploy Integration Flow *"Pipeline DS Retry - Custom Error Handling with Data Stores"*.
4. Configure Integration Flow *"Pipeline DS Retry - Restart Executor from Data Store to JMS"*.
 - Default configuration is sufficient for this Integration Flow. Adjustments of Splitter configuration and Process Direct endpoint are possible, but they are optional.
5. Deploy Integration Flow *"Pipeline DS Retry - Restart Executor from Data Store to JMS"*.

Assign Data Store Restart Configuration for Configuration Scenarios in Partner Directory

The Integration Flow “*Pipeline DS Retry - Custom Error Handling with Data Stores*” will put the failed messages into Data Stores based on *PartnerId* configuration defined in the Partner Directory. If the configuration is missing in the Partner Directory for a given *PartnerId*, default values will be applied, as described below.


The Partner Directory configuration for each *PartnerId* can be defined with String Parameters:

- **RetryDataStore** – Defines the name of the Data Store where the erroneous message should be stored in case of exceeding maximum number of retry attempts at JMS layer.
 - o Note that even when set to a specific value, messages destined for the *NoRetryStore* will have the “_NoRetry” postfix appended to the Data Store name.
 - o In the example below, the *RetryDataStore* is “MATMAS_NDLV” and the corresponding *NoRetryStore* will be automatically set to “MATMAS_NDLV_NoRetry”. No explicit configuration is needed in the Partner Directory for the *NoRetryStore*.
 - o If *RetryDataStore* String parameter is not set for the *PartnerId*, then the used data store names will be <PartnerId>_Retry and <PartnerId>_NoRetry.
- **restartMode** – Can be set to either **DS** (Data Store) or **JMS** (JMS queue) to define how the messages should be restarted. Default option is **DS** if not set.
 - o **DS**: putting a message back to the JMS queue, but in case of an error immediately taking it to a Data Store (preferred and default).
 - o **JMS**: putting a message back to the JMS queue and in case of an error executing retries in this JMS queue according to the maximum number of JMS retries.
- **MaxDataStoreRetries** – Defines the number of retries after a restart of a message. If the limit is reached the message is moved into the “_NoRetry” Data Store. Default number is 5.

MATMAS_S4H		
String Parameters	Binary Parameters	Alternative Partners
Authorized Users		
String Parameters (4)		
ID	Value	Last Modified By
MaxDataStoreRetries	5	
MaxJMSRetries	2	
restartMode	DS	
RetryDataStore	MATMAS_NDLV	

Enable Custom Exception Handling in generic IFlows of Process Integration Pipeline package

Custom exception handling can be enabled in the generic IFlows by setting the parameter “CustomXError_Enabled” to true:



Configure "Pipeline Generic Step04 - Receiver Determination"

Sender Receiver **More**

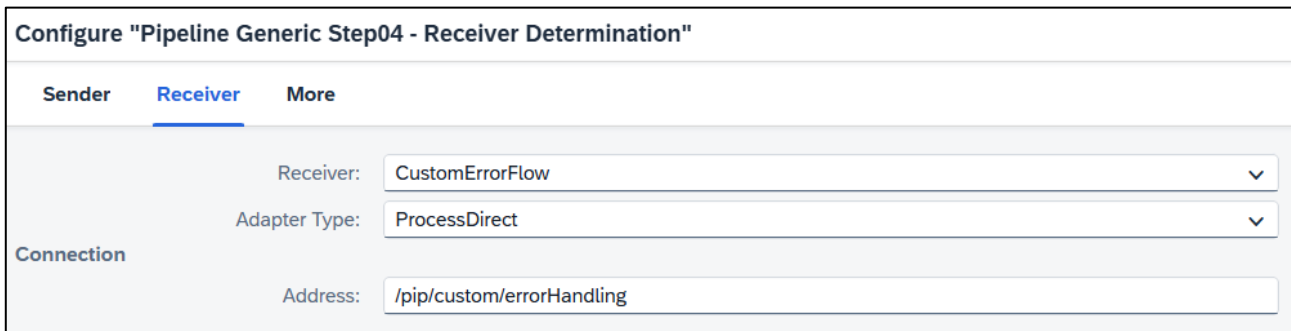
Type: All Parameters

CustomXError_Enabled: true

DLQ_EndEvent: Escalated

PipelineJMSQueuePrefix: PIP

If the Process Direct endpoint of “Custom Error Handling with Data Stores” IFlow is not default, then the Address of the CustomErrorFlow receiver to be maintained:



Configure "Pipeline Generic Step04 - Receiver Determination"

Sender **Receiver** More

Receiver: CustomErrorFlow

Adapter Type: ProcessDirect

Connection

Address: /pip/custom/errorHandling

This needs to be configured in all IFlows in the Fully Decoupled Pipeline and in the Integrated Messaging Runtime (Asynchronous):

- Pipeline Generic Step02 - Inbound Processing
- Pipeline Generic Step02 - Integrated Messaging Runtime Async
- Pipeline Generic Step04 - Receiver Determination
- Pipeline Generic Step05 - Interface Determination
- Pipeline Generic Step06 - Outbound Processing
- Any copies of Pipeline Generic IFlows

Refer to the Process Integration Pipeline error handling options: [Custom Exception Handling](#).

Remark: this configuration is global and applied for all interfaces that are processed via Process Integration Pipeline.

Create Restart Job Profiles in Partner Directory

The Integration Flow “*Pipeline DS Retry - Template - Restart Job*” relies on the configuration of different Restart Job Profiles. Multiple variants can be configured to have different scheduled versions of the Restart Job for specific interfaces.

Note that interfaces must be explicitly specified, no regex pattern is applicable in the profiles. Also, there is no default Job Profile applicable for all integration scenarios, i.e. if there is no defined job profile for the interface – the messages won’t be restarted

- **Status** – Can be set to “*active*” or “*inactive*”.
- **InterfacePidList** – Defines the list of the interfaces (or integration scenarios) for which the message restarts should be triggered. Multiple scenarios can be set by using the character semicolon “;” as a delimiter. The Restart Job will consider the scenario-specific configuration defined in the Partner Directory (assigned Data Store, number of retries and restart mode).
- **Store** – (Optional parameter). If not set, *Retry* stores will be used always (with value *Retry*). In case of emergencies or if the API based restart can’t be used, restart from the *NoRetryStore* can be set for scheduled Restart Jobs.

Filters within the Restart Job Profile. If not set, all entries from the given Data Store will be restarted.

- **Receiver** – (Optional parameter). Receiver name to select all the entries from the Data Store that belong to the specified receiver (for the stages when receiver is unknown (e.g. Inbound), the value “*null*” can be sent and then all entries without receiver will be selected). Example of value: “*S4HCLNT100*”. Note that the parameter setting is global for all interfaces maintained in the *InterfacePidList* parameter, it can’t be set for each interface separately.
- **TimeFrom** – (Optional parameter): Interval start timestamp in UTC format to select all the entries from the Data Store that have timestamp in DS Entry name later than specified timestamp. Example of value: “*2025-06-01T09:00:00.000Z*”.
- **TimeTo** – (Optional parameter): Interval end timestamp in UTC format to select all the entries from the Data Store that have timestamp in DS Entry name earlier than specified timestamp. Example of value: “*2025-06-02T09:00:00.000Z*”.

RestartJobProfile_MATMAS		
String Parameters	Binary Parameters	Alternative Partners
Authorized Users		
String Parameters (2)		
ID	Value	Last Modified By
InterfacePidList	MATMAS_T33;MATMAS_S4H	
Status	active	

Create a Restart Job Integration Flow from Template, Configure Parameters and Deploy

The extension package contains the template Integration Flow “*Pipeline DS Retry - Template - Restart Job*” which is designed to be copied to custom packages to be configurable via the Restart Job Profiles from the Partner Directory. This way multiple Restart Job Profiles and schedules can be configured.

The Restart Job can be configured to use a schedule-based trigger (start Timer event). Default configuration is to run *On Deployment*.

Configure "Pipeline DS Retry - Template - Restart Job"

Timer: Start Timer 1 [StartEvent_51]

☒ Basic ☐ Advanced

Frequency

Enter As: Simple Schedule

Repeat: Hourly

Every: 1 Hours

Time Range

Start Date and Time: 2025-04-02 10:00

End Date and Time: 2026-04-01 10:00

Time Zone: (UTC 0:00) Greenwich Mean Time(Etc/GMT)

☒ Throw exception on schedule expiry

The Restart Job will call the [OData API of SAP Cloud Integration](#) to check whether a Data Store exists and what entries they contain. These APIs require OAuth 2.0 Client Credentials Grant-based authentication for which the service instances and service keys can be generated in the SAP BTP Cockpit described [here](#). The generated service instance must have the following roles:

- DataStoresAndQueuesRead
- DataStorePayloadsRead
- DataStoresAndQueuesConfig

The generated *clientID* and *clientSecret* from the service key must be deployed as *User Credential*, *clientID* set as Username and *clientSecret* set Password.

Create User Credentials

Name: * CI_Internal_API_ClientID_ClientSecret

Description: Contains clientID and clientSecret for CI API calls

Runtimes: * Cloud Integration X

Type: * User Credentials

User: * sb-8ffa4b7-bba5-4523-ac82-dc60f9111f88!b15282|i...

Password:

Repeat Password:

Deploy Cancel

The configuration of the Integration Flow “*Pipeline DS Retry - Template - Restart Job*” contains below parameters:

- **API_OAuth_Client_Credentials_Alias** – The alias of the *User Credential* containing the service key’s *clientID* and *clientSecret* previously deployed.
- **API_TokenURL**– Set the tenant specific OAuth token URL that is maintained in the service key.
- **Hostname** – Set the tenant specific OData API URL. The parameter must contain the path */api/v1*.
- **RestartJobProfile**: Set the maintained profile name from the Partner Directory.
- **Splitter related parameters** (Optional):
 - **Number_Of_Concurrent_Processes**: Set the number of concurrent processes if parallel processing is enabled.
 - **Parallel_Processing**: Enable parallelization for Splitter. Default disabled.
 - **Streaming**: Default enabled. Not recommended to be disabled.
 - **Timeout**: Defines how long the Splitter waits for the split messages to finish execution. Default 300 seconds.

Configure "Pipeline DS Retry - Template - Restart Job"

Timer

Receiver

More

Type:

All Parameters

API_OAuth_Client_Credentials_Alias:

CI_Internal_API_ClientID_ClientSecret

API_TokenURL:

https:// authentication.eu10.hana.ondemand.com/oauth/token

Hostname:

https:// .it-cpi018.cfapps.eu10-003.hana.ondemand.com:443/api/v1

Number_Of_Concurrent_Processes:

10

Parallel_Processing:

☐

RestartJobProfile:

RestartJobProfile_RetryStore

Streaming:

☒

Timeout:

300

(Optional) Configure and Deploy Integration Flows for On-Demand Restarts or Data Store management via HTTP API

There are 2 optional Integration Flows in the package which can be used for on demand operations. They expose 2 HTTP endpoints to manage Retry related operations and to manage Data Store entries. The functionality is divided into 2 Integration Flows for more granular control on authorization level who is allowed to call Restart operations only and who is allowed to Cancel messages (manipulate Data Store entries).

Pipeline API Operations

IFlow: Pipeline API – Data Store Manage Retries (on demand API) User Role <i>DSRestart.manage</i>	GET	If Data Store Name is <u>not</u> provided - View list of Data Stores. If Data Store Name is provided - Get all Entry IDs from the selected data store and selection criteria.
	POST	Execute “Restart from Data Store” for entries from the selected data store (either specified entries in the body or entries satisfying selection criteria).
	PUT	Move DS Entries from source DS to target DS (either specified entries in the body or entries satisfying selection criteria).
	DELETE	Move entries from selected DS to “NoRetry” DS (either specified entries in the body or entries satisfying selection criteria).
IFlow: Pipeline API – Data Store Manage Entries (on demand API) User Role: <i>DSEntry.manage</i>	GET	Read 1 DS Entry from Data Store (DS entry body and message headers) Read 1 message from Data Store (message body and message headers)
	PUT	Put 1 DS Entry to the Data Store (DS entry body and message headers)
	DELETE	Delete DS Entries (messages) from the Data Store (either specified entries in the body or entries satisfying selection criteria).

Configuration of the Integration Flow “**Pipeline API - Data Store Manage Retries**”:

- Sender Adapter configuration:
 - o **Address (HTTPS)** – the endpoint of Restart API that is maintained in HTTPS sender adapter. Default endpoint address is /pipeline/api/v1/dsretry.
 - o **User Role** – the user role to authorize Integration Flow execution request. Default value is “DSRestart.manage”. In order to follow configuration with this default value the User Role needs to be created in the tenant per [Managing User Roles](#) and required [inbound HTTP connection for Integration Flow processing](#) is enabled with this User Role.
 - o **CSRF Protected** – [CSRF protection](#) is activated by default for modifying HTTP requests.
 - o **Body Size (in MB)** – the HTTP request body size is externalized with default value 40 MB, but it is not expected that this value will require adaptation.
- Receiver Adapter configuration:
 - o **Address (ProcessDirect)** – the ProcessDirect endpoint of the Integration Flow “Pipeline DS Retry - Restart Job by each Interface” configured in the step (4) of [Configure and Deploy mandatory Integration Artefacts of DS Restart Extension package](#).
- Integration Flow steps configuration:
 - o **CI_ODataAPI_OAuth_Client_Credentials_Alias** – the alias of the *User Credential* containing the service key’s *clientID* and *clientSecret* (the same as configured and deployed at [Create a Restart Job Integration Flow from Template, Configure Parameters and Deploy](#)).
 - o **CI_ODataAPI_TokenURL** – tenant specific OAuth token URL that is maintained in the service key (refer to the previous step or link above). The parameter must contain the path /oauth/token.
 - o **CI_ODataAPI_URL** – tenant specific OData API URL (refer to the previous step or link above). The parameter must contain the path /api/v1.

- **Splitter related parameters for MOVE** (Optional) – default Splitter configuration for the Move operation (move Data Store entries from the source DS to the target DS): parallel processing with streaming enabled, number of concurrent processes is 10 and timeout 300 seconds applied. Parameters are externalized for potential performance tuning requirements.
- **Splitter related parameters for RESTART** (Optional) – default Splitter configuration for the Restart operation (if entries that are related to multiple interfaces to be restarted, then calling Integration Flow “*Pipeline DS Retry - Restart Job by each Interface*” with a split by Interface): parallel processing with streaming enabled, number of concurrent processes is 10 and timeout 300 seconds applied. Parameters are externalized for potential performance tuning requirements.

Configure "Pipeline API - Data Store Manage Retries"	
Sender	Receiver
<div> <div>Type:</div> <div>All Parameters</div> </div>	
CI_ODataAPI_OAuth_Client_Crede...	CI_Internal_API_ClientID_ClientSecret
CI_ODataAPI_TokenURL:	https://.hana.ondemand.com/oauth/token
CI_ODataAPI_URL:	https://.hana.ondemand.com:443/api/v1
SplitterConcurrentProcessesMove:	10
SplitterConcurrentProcessesRestart:	10
SplitterParallelProcessingMove:	<input checked="" type="checkbox"/>
SplitterParallelProcessingRestart:	<input checked="" type="checkbox"/>
SplitterStreamingMove:	<input checked="" type="checkbox"/>
SplitterStreamingRestart:	<input checked="" type="checkbox"/>
SplitterTimeoutMove:	300
SplitterTimeoutRestart:	300

Configuration of the Integration Flow “**Pipeline API - Data Store Manage Entries**”:

- Sender Adapter configuration:
 - **Address (HTTPS)** – the endpoint of DS Entry API that is maintained in HTTPS sender adapter. Default endpoint address is /pipeline/api/v1/dsentry.
 - **User Role** – the user role to authorize Integration Flow execution request. Default value is “*DSEntry.manage*”. In order to follow configuration with this default value the User Role needs to be created in the tenant per [Managing User Roles](#) and required [inbound HTTP connection for Integration Flow processing](#) is enabled with this User Role.
 - **CSRF Protected** – [CSRF protection](#) is activated by default for modifying HTTP requests.
 - **Body Size (in MB)** – the HTTP request body size is externalized with default value 40 MB, but it is not expected that this value will require adaptation.
- Integration Flow steps configuration:
 - **CI_ODataAPI_OAuth_Client_Credentials_Alias** – the alias of the *User Credential* containing the service key’s *clientID* and *clientSecret* (the same as configured and deployed at [Create a Restart Job Integration Flow from Template, Configure Parameters and Deploy](#)).
 - **CI_ODataAPI_TokenURL** – tenant specific OAuth token URL that is maintained in the service key (refer to the previous step or link above). The parameter must contain the path /oauth/token.

- **CI_ODataAPI_URL** – tenant specific OData API URL (refer to the previous step or link above). The parameter must contain the path /api/v1.
- **DSEntryExpirationPeriodDays** – set in the Data Store Write operation the number of days after which the stored messages are deleted (default value is 90 days). More information about [Write operation](#). This setting is only applicable for PUT operation.
- **DSEntryAlertingThresholdDays** – set in the Data Store Write operation the number of days after which the alert is raised if the message is not fetched (default value is 2 days). More information about [Write operation](#). This setting is only applicable for PUT operation.
- **AllowPutOperation** – this switch allows to restrict PUT operation (insert DS Entry to the Data Store) in the Data Store Manage Entries API Integration Flow. By default, PUT operation is allowed and “AllowPutOperation” is set to *true*.
- **Splitter related parameters for DELETE** (Optional) – default Splitter configuration for the Delete operation (delete DS entries from the Data Store): parallel processing with streaming enabled, number of concurrent processes is 10 and timeout 300 seconds applied. Parameters are externalized for potential performance tuning requirements.

Configure "Pipeline API - Data Store Manage Entries"	
Sender	More
Type:	All Parameters
AllowPutOperation:	true
CI_ODataAPI_OAuth_Client_Crede... :	CI_Internal_API_ClientID_ClientSecret
CI_ODataAPI_TokenURL:	https://[redacted].hana.ondemand.com/oauth/token
CI_ODataAPI_URL:	https://[redacted].hana.ondemand.com:443/api/v1
DSEntryAlertingThresholdDays:	2
DSEntryExpirationPeriodDays:	90
SplitterConcurrentProcesses:	10
SplitterParallelProcessing:	true
SplitterTimeout:	300

As soon as configuration is done, the Integration Flows can be deployed.

For the details how to call these APIs refer to the [Appendix](#) section of this Configuration Guide.

Appendix

This chapter covers how to work with the on-demand Pipeline APIs (Data Store Manage Retries and Data Store Manage Entries). It is explained below how to configure the HTTP requests in the HTTP client. The API endpoint is defined in Integration Flow configuration and can be copied from: Monitor → Integrations and APIs → Manage Integration Content and then copy the endpoints of the deployed Integration Flows. Authentication, Authorization and usage of CSRF protection are done based on your choice and selected externalized configuration in the Integration Flows (like User Roles). Details below are covering which HTTP method to use, which query parameters are mandatory or optional, whether message headers or message body are required. There is a validation of input data implemented in the Integration Flow and in case of wrong request parameters HTTP 400 response is expected with a response header “message” that can provide additional details which expectation is not met in the submitted request.

For HTTP requests where entries are retrieved per selection criteria the query parameters can be combined: for example, you can select messages for certain receiver and interface and provide the time interval (from start time *time-from* until end time *time-to*).

Pipeline API - Data Store Manage Retries

- **Read all Data Stores that exist in the tenant**
 - o HTTP method: GET
 - o Query:
 - **\$select** (Optional): the fields of Data Store API can be selected as comma separated value (possible fields: DataStoreName, IntegrationFlow, Type, Visibility, NumberOfMessages, NumberOfOverdueMessages). Example of value: “DataStoreName,NumberOfMessages,NumberOfOverdueMessages”.
 - o HTTP request headers:
 - **Accept** (Optional): if value “application/json” – the response is JSON, otherwise XML.
 - o HTTP request body: No body.

- **Read all Entries from the Data Store**

- HTTP method: GET
- Query:
 - **dsname** (Mandatory): name of the Data Store. Example of value: "MATMAS_NDLV".
 - **interface-pid** (Optional): Partner ID to select all the entries from the Data Store that belong to the specified interface. Example of value: "MATMAS_S4H".
 - **receiver** (Optional): Receiver name to select all the entries from the Data Store that belong to the specified receiver (for the stages when receiver is unknown (e.g. Inbound), the value "null" can be sent and then all entries without receiver will be selected). Example of value: "S4HCLNT100".
 - **time-from** (Optional): Interval start timestamp in UTC format to select all the entries from the Data Store that have timestamp in DS Entry name later than specified timestamp. Example of value: "2025-01-01T12:34:56.789Z".
 - **time-to** (Optional): Interval end timestamp in UTC format to select all the entries from the Data Store that have timestamp in DS Entry name earlier than specified timestamp. Example of value: "2025-01-01T12:34:56.789Z".
- HTTP request headers:
 - **Accept** (Optional): if value "application/json" – the response is JSON, otherwise XML.
- HTTP request body: No body.

Remark: reading of the Data Store entries is aimed only for the Data Stores that are used for pipeline based retries and such Data Stores are defined with visibility Global. Usage of this API for reading entries from Integration Flow specific Data Stores is not supported.

- **Restart selected Entries from the Data Store.** There are 2 options to execute this request: either to provide list of entries to be restarted or to provide selection criteria to restart matching entries.

- **Restart DS entries provided in the request body**

- HTTP method: POST
 - Query:
 - **dsname** (Mandatory): name of the Data Store whose entries are going to be restarted. Example of value: "MATMAS_NDLV".
 - **retry-count** (Optional): integer value of maximum number of retries that is written to the header maxJMSRetries when the message is placed back to the source JMS queue. Default value is 1.
 - HTTP request headers: No specific headers.
 - HTTP request body: JSON array with DS Entry IDs.
 - Example of body:

```
[
  "zkv04~(OB)~IS_CI_Receiver~AGf-bpTD4JXQxwgjRKoiAmz1JcxS~2025-04-15T14:35:00Z~5",
  "zkv04~(IB)~AGf-vkUnqoRREci7eqxIhdoiV_QX~2025-04-15T20:15:01Z~1"
]
```

- **Restart DS entries matching selection criteria**
 - HTTP method: POST
 - Query:
 - **dsname** (Mandatory): name of the Data Store whose entries are going to be restarted. Example of value: “MATMAS_NDLV”.
 - **retry-count** (Optional): integer value of maximum number of retries that is written to the header maxJMSRetries when the message is placed back to the source JMS queue. Default value is 1.
 - **interface-pid** (Optional): Partner ID to select all the entries from the Data Store that belong to the specified interface. Example of value: “MATMAS_S4H”.
 - **receiver** (Optional): Receiver name to select all the entries from the Data Store that belong to the specified receiver (for the stages when receiver is unknown (e.g. Inbound), the value “null” can be sent and then all entries without receiver will be selected). Example of value: “S4HCLNT100”.
 - **time-from** (Optional): Interval start timestamp in UTC format to select all the entries from the Data Store that have timestamp in DS Entry name later than specified timestamp. Example of value: “2025-01-01T12:34:56.789Z”.
 - **time-to** (Optional): Interval end timestamp in UTC format to select all the entries from the Data Store that have timestamp in DS Entry name earlier than specified timestamp. Example of value: “2025-01-01T12:34:56.789Z”.
 - HTTP request headers: No specific headers.
 - HTTP request body: No body.
- **Move selected Entries to the NoRetry Data Store.** There are 2 options to execute this request: either to provide list of entries to be moved or to provide selection criteria to move matching entries.
 - **Move DS entries provided in the request body to the NoRetry Data Store**
 - HTTP method: DELETE
 - Query:
 - **dsname** (Mandatory): name of the (Retry) Data Store whose entries should be moved to NoRetry Data Store. Example of value: “MATMAS_NDLV”.
NOTE: This operation only allows to move Entry ID from Retry DS to NoRetry DS, the target NoRetry Data Store is determined simply by adding postfix “_NoRetry”.
 - HTTP request headers: No specific headers.
 - HTTP request body: JSON array with DS Entry IDs.
 - Example of body:


```
[
    "zkv04~(OB)~IS_CI_Receiver~AGf-bpTD4JXQxwgjRKoiAmz1JcxS~2025-04-15T14:35:00Z~5",
    "zkv04~(IB)~AGf-vkUnqoRREci7eqxlhdoiV_QX~2025-04-15T20:15:01Z~1"
  ]
```

- **Move DS entries matching selection criteria to the NoRetry Data Store**
 - HTTP method: DELETE
 - Query:
 - **dsname** (Mandatory): name of the (*Retry*) Data Store whose entries should be moved to NoRetry Data Store. Example of value: “MATMAS_NDLV”.
 - **interface-pid** (Optional): Partner ID to select all the entries from the Data Store that belong to the specified interface. Example of value: “MATMAS_S4H”.
 - **receiver** (Optional): Receiver name to select all the entries from the Data Store that belong to the specified receiver (for the stages when receiver is unknown (e.g. Inbound), the value “null” can be sent and then all entries without receiver will be selected). Example of value: “S4HCLNT100”.
 - **time-from** (Optional): Interval start timestamp in UTC format to select all the entries from the Data Store that have timestamp in DS Entry name later than specified timestamp. Example of value: “2025-01-01T12:34:56.789Z”.
 - **time-to** (Optional): Interval end timestamp in UTC format to select all the entries from the Data Store that have timestamp in DS Entry name earlier than specified timestamp. Example of value: “2025-01-01T12:34:56.789Z”.
 - HTTP request headers: No specific headers.
 - HTTP request body: No body.
- **Move selected Entries from the source Data Store to the target Data Store.** More generic operation than “Move to NoRetry DS” that allows to move from any source DS to any target DS (in case target DS does not exist, it will be created during the Write operation). There are 2 options to execute this request: either to provide list of entries to be moved or to provide selection criteria to move matching entries.
 - **Move DS entries provided in the request body from the source Data Store to the target Data Store**
 - HTTP method: PUT
 - Query:
 - **dsname** (Mandatory): name of the source Data Store whose entries should be moved to the target Data Store. Example of value: “MATMAS_NDLV_NoRetry”.
 - **trg-dsname** (Mandatory): name of the target Data Store where entries should be moved from the source Data Store. Example of value: “MATMAS_NDLV”.
 - HTTP request headers: No specific headers.
 - HTTP request body: JSON array with DS Entry IDs.
 - Example of body:


```
[
                "zkv04~(OB)~IS_CI_Receiver~AGf-bpTD4JXQxwgjRKoiAmz1JcxS~2025-04-15T14:35:00Z~5",
                "zkv04~(IB)~AGf-vkUnqoRREci7eqxlhdoiV_QX~2025-04-15T20:15:01Z~1"
              ]
```

- **Move DS entries matching selection criteria from the source Data Store to the target Data Store**
 - HTTP method: PUT
 - Query:
 - **dsname** (Mandatory): name of the source Data Store whose entries should be moved to the target Data Store. Example of value: “MATMAS_NDLV_NoRetry”.
 - **trg-dsname** (Mandatory): name of the target Data Store where entries should be moved from the source Data Store. Example of value: “MATMAS_NDLV”.
 - **interface-pid** (Optional): Partner ID to select all the entries from the Data Store that belong to the specified interface. Example of value: “MATMAS_S4H”.
 - **receiver** (Optional): Receiver name to select all the entries from the Data Store that belong to the specified receiver (for the stages when receiver is unknown (e.g. Inbound), the value “null” can be sent and then all entries without receiver will be selected). Example of value: “S4HCLNT100”.
 - **time-from** (Optional): Interval start timestamp in UTC format to select all the entries from the Data Store that have timestamp in DS Entry name later than specified timestamp. Example of value: “2025-01-01T12:34:56.789Z”.
 - **time-to** (Optional): Interval end timestamp in UTC format to select all the entries from the Data Store that have timestamp in DS Entry name earlier than specified timestamp. Example of value: “2025-01-01T12:34:56.789Z”.
 - HTTP request headers: No specific headers.
 - HTTP request body: No body.

Pipeline API - Data Store Manage Entries

- **Read 1 Data Store Entry** (DS entry format – refer to [Documentation](#) → *Data Store Entry Body Format*)
 - HTTP method: GET
 - Query:
 - **dsname** (Mandatory): name of the Data Store. Example of value: “MATMAS_NDLV”.
 - **dsentry** (Mandatory): name of the Data Store Entry. Example of value: “zkv04~(OB)~IS_CI_TECH~AGf-bsnV1oy3AnuezjgpDUy9G3je~2025-04-15T14:35:53Z~6”.
 - HTTP request headers: No specific headers.
 - HTTP request body: No body.

- **Read 1 Data Store Entry** (message payload format – original decoded message body)
 - HTTP method: GET
 - Query:
 - **dsname** (Mandatory): name of the Data Store. Example of value: “MATMAS_NDLV”.
 - **dsentry** (Mandatory): name of the Data Store Entry. Example of value: “zkv04~(OB)~IS_CI_TECH~AGf-bsnV1oy3AnuezjgpDUy9G3je~2025-04-15T14:35:53Z~6”.
 - **entry-format** (Mandatory): with value “message”. Otherwise, DS entry format will be returned in the response, i.e. previously described GET call. Remark: in this request original message payload is base64-decoded from the Data Store and returned in the original format (without Data Store specific fields).
 - HTTP request headers: No specific headers.
 - HTTP request body: No body.

- **Put 1 Data Store Entry to the Data Store.** This operation is error prone (as there is a risk that some mandatory Process Integration Pipeline message headers won't be provided in this PUT request and further pipeline steps won't be possible in such case). If modification of messages is not required or not allowed in the Cloud Integration, it can be disabled on the Integration Flow level.
 - o HTTP method: PUT
 - o Query:
 - **dsname** (Mandatory): name of the Data Store where the DS Entry to be placed.
Example of value: "MATMAS_NDLV".
 - o HTTP request headers: all pipeline headers must be sent. Examples of headers are provided below (can be other headers as well):
 - **partnerid** (typically Mandatory)
 - **sap_sender** (might be Mandatory)
 - **sap_senderinterface** (might be Mandatory)
 - **sap_messagetype** (recommended)
 - **sap_applicationid** (recommended)
 - **sap_receiver** (Mandatory after Receiver Determination)
 - **sap_receiveralias** (Mandatory after Receiver Determination)
 - **sap_outboundprocessingendpoint** (Mandatory for Outbound processing)
 - ...
 - o HTTP request body: XML body of the DS Entry ID in the format according to [Documentation](#)
→ *Data Store Entry Body Format*.
 - Example of body:


```
<?xml version="1.0" encoding="UTF-8"?>
<Message>
  <MsgHeader>
    <RestartQueue>PIPQ04</RestartQueue>

    <EntryID>ZMATMAS~(OB)~Receiver1~AGfvg9UQyZlIMbpHbq7EXsnP6YMk
~2025-04-04T07:01:41Z~0</EntryID>
    <RetryDataStore> ZMATMAS_Retry</RetryDataStore>
    <MaxRetryCount>3</MaxRetryCount>
    <Error>Exception.Base64encoded </Error>
  </MsgHeader>
  <Payload>Payload.Base64encoded</Payload>
</Message>
```

- **Delete selected Entries from the Data Store.** There are 2 options to execute this request: either to provide list of entries to be deleted or to provide selection criteria to delete matching entries. **Note: the action cannot be undone, the messages will be permanently deleted from the tenant.**

- **Delete DS entries provided in the request body from the Data Store**

- HTTP method: DELETE
- Query:
 - **dsname** (Mandatory): name of the Data Store whose entries should be deleted. Example of value: "MATMAS_NDLV".
- HTTP request headers: No specific headers.
- HTTP request body: JSON array with DS Entry IDs.
 - Example of body:

```
[
  "zkv04~(OB)~IS_CI_Receiver~AGf-bpTD4JXQxwgjRKoiAmz1JcxS~2025-04-15T14:35:00Z~5",
  "zkv04~(IB)~AGf-vkUnqoRREci7eqxIhdoiV_QX~2025-04-15T20:15:01Z~1"
]
```

- **Delete DS entries matching selection criteria from the Data Store**

It is recommended to execute a call "Read all Entries from the Data Store" first with the selection criteria that you would like to apply and if DS entries list is correct, then execute this deletion request. Alternatively, you can copy the DS entries list from the response of a GET call (Read all Entries from the Data Store) and Delete DS entries by providing DS Entry names in the message body (request above).

- HTTP method: DELETE
- Query:
 - **dsname** (Mandatory): name of the Data Store whose entries should be deleted. Example of value: "MATMAS_NDLV".
 - **interface-pid** (Optional): Partner ID to select all the entries from the Data Store that belong to the specified interface. Example of value: "MATMAS_S4H".
 - **receiver** (Optional): Receiver name to select all the entries from the Data Store that belong to the specified receiver (for the stages when receiver is unknown (e.g. Inbound), the value "null" can be sent and then all entries without receiver will be selected). Example of value: "S4HCLNT100".
 - **time-from** (Optional): Interval start timestamp in UTC format to select all the entries from the Data Store that have timestamp in DS Entry name later than specified timestamp. Example of value: "2025-01-01T12:34:56.789Z".
 - **time-to** (Optional): Interval end timestamp in UTC format to select all the entries from the Data Store that have timestamp in DS Entry name earlier than specified timestamp. Example of value: "2025-01-01T12:34:56.789Z".
- HTTP request headers: No specific headers.
- HTTP request body: No body.