

# SAP Ariba Integration with SAP Signavio Process Intelligence– Deployment Guide



**Run Simple**

SAP Ariba Integration with SAP Signavio Process Intelligence

# Contents

1	Document History.....	2
2	Pre-requisites.....	2
3	Process Flow.....	2
4	Configuration File:.....	3
4.1	Data Sources.....	3
4.2	Data Targets.....	4
4.3	Config.....	5
4.4	Resource Items Configuration.....	6
4.5	Loading Configuration file to CI.....	9
5	Data Scheduler.....	10
5.1	Input Screen.....	11
5.1.1	Delta Period.....	11
5.1.2	Extraction From Timestamp.....	11
5.1.3	Extraction To Timestamp.....	11
5.1.4	Mode.....	11
5.1.5	Scope.....	12
5.1.6	Retry Queue Id List.....	12
5.2	How to Run.....	12
5.3	When to Run.....	13
6	Data Extractor.....	13
6.1	Input Screen.....	13
6.1.1	Timer.....	14
6.1.2	Receiver.....	14
6.1.3	More.....	14
6.2	How to Run.....	15
6.3	Email body.....	15
6.4	Monitoring.....	16
6.4.1	Additional CI content for Monitoring.....	17

## 1 Document History

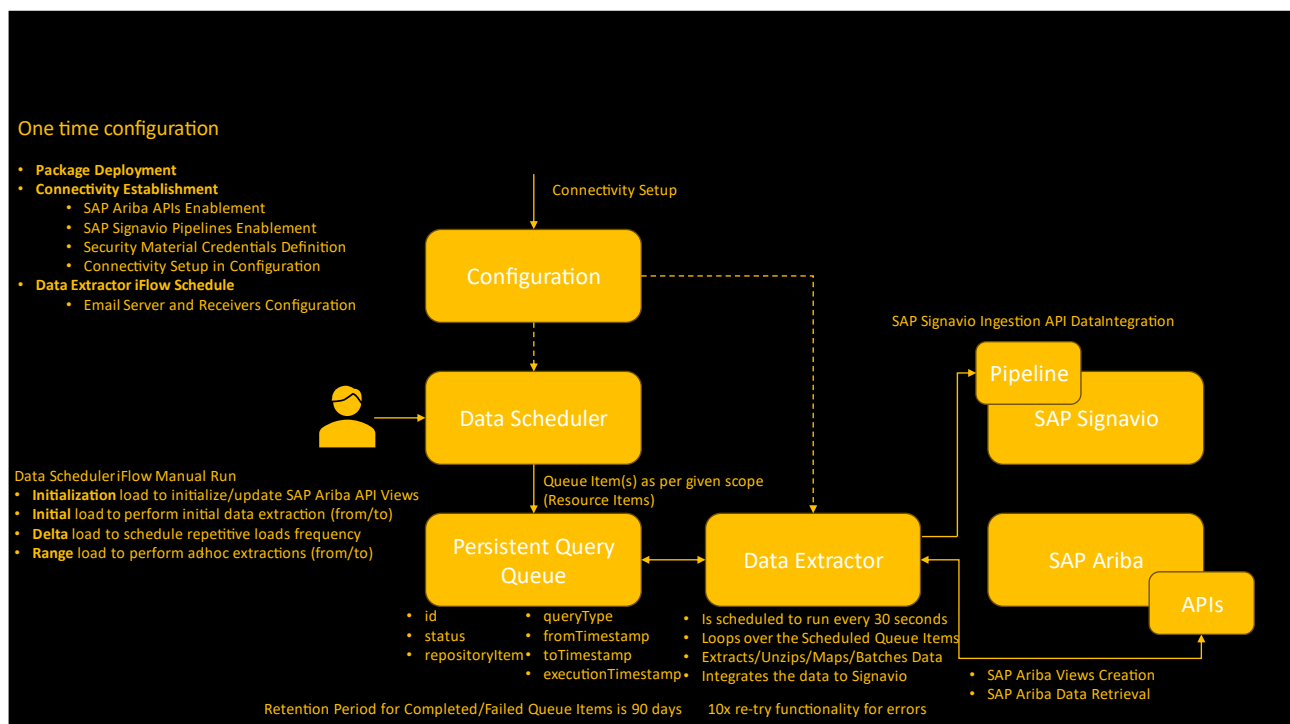
Document Version	Date	Author	Comment
1.0	2025/03/15	Charan Samanchi	Initial release
1.0.1	2025/05/08	Saniya Mapari	Screenshots updated

## 2 Pre-requisites

Following are checks before Deploying Data Scheduler or before scheduling Data Extractor.

- Import package content “SAP Ariba Integration with SAP Signavio Process Intelligence” in CI tenant which consists of 7 iflows.
- Prepare config file and create/update the datastore for the config resource file. Details explained in Section 4.5.
- Make sure to create security credentials for all APIs in scope as per naming convention in configuration file (analytical/procurement/sourcing reporting API & SAP Signavio Ingestion API).
- Security credentials for Error email notification, connectivity checks and certificate import for setting up email adapter needs to be done.
- Data scheduler needs to run first to create persistent queue before running extractor. More details on this are explained later below
- To fetch the persistent Queue/ lock status, https post endpoints need to be executed through postman/bruno. Make sure that credentials are created in subaccount tenant to access endpoints of iflow. These are mainly used for debugging or issue analysis purpose but must not be used to alter persistent queue during regular operations.

## 3 Process Flow



1. Config template is shared by SAP team. Only update realm name in dataSource key SAPAribaRealm\_{**Realm**}, tenantId, tokenServiceUrl, baseUrl and credentials. Replace {Realm} in config template with actual realm.
2. Once updated, load it as mentioned in section 4.5.
3. Run data Scheduler by configuring and deploying only once. Please refer section 5.1 for more details on input screen parameters.
4. Deploy Data Extractor to run periodically every 30 seconds to process Data Scheduler queue items. Ensure Receiver email notification setup is done as mentioned in section 6.1.2

## 4 Configuration File:

[Configuration File](#) contains all security parameters for data Sources ( realm, reporting API base URLs , Oauth credential), data Targets ( SAP Signavio Process Intelligence Pipelines and credential) information, configuration parameters and resource Items ( which contains Fact tables from SAP Ariba reporting to be pushed to SAP Signavio Process Intelligence in csv).

**Note:** Make sure that config file structure and keys are unchanged. This will cause inconsistency with logic in data Scheduler/ extractor while fetching relevant information. Only expected values are to be updated.

### 4.1 Data Sources

#### 4.1 Data Sources

```

"dataSources": [
  {
    "SAPAribaRealm_StratusMedio": {
      "type": "SAP Ariba Realm",
      "tenantId": "Sti_...",
      "tokenServiceURL": "https://api-eu.ariba.com",
      "baseUrl": "https://eu.openapi.ariba.com",
      "apis": {
        "analytics-reporting": {
          "createView": "/api/analytics-reporting-view/v1/prod/viewTemplates",
          "jobSubmit": "/api/analytics-reporting-job/v1/prod/jobs",
          "apikey": "Cs...xqKd",
          "credentials": "SAPAriba_...o_analytics-reporting"
        },
        "sourcing-reporting": {
          "createView": "/api/sourcing-reporting-view/v1/prod/viewTemplates",
          "jobSubmit": "/api/sourcing-reporting-job/v1/prod/jobs",
          "apikey": "9DzC6i...4NZ",
          "credentials": "SAPAriba_...o_sourcing-reporting"
        },
        "procurement-reporting": {
          "createView": "/api/procurement-reporting-view/v2/prod/viewTemplates",
          "jobSubmit": "/api/procurement-reporting-job/v2/prod/jobs",
          "apikey": "OKRI...oLmMZ",
          "credentials": "SAPAriba_...o_procurement-reporting"
        }
      }
    }
  }
]

```

As illustrated in above example, data Sources are sender system details. data source Key is in the format ( SAPAribaRealm\_{**RealmName**}) which matches with “data Source of persistent queue items“ in persistent queue and “data Source of resource items“ in config file.

- i) tenantId : {RealmName}
- ii) baseUrl : SAP Ariba reporting API base URL which will be used in data Scheduler while creating queue items. This will vary according to SAP Ariba data center where realm is hosted
- iii) apis : It contains list of all reporting APIs that are in scope for that realm. In this case for SAP Signavio, we are using all analytical, Sourcing and Procurement reporting APIs. It must match with the “api“ key (path: **resourceItemsConfig.api** ) in resource items config ( explained below).
  - (a) createView : it is extension to base URL which will be used for **initialization** mode in data Scheduler
  - (b) jobSubmit: it is extension to base URL which will be used for **initial/delta/range** mode in data scheduler.
  - (c) apiKey: API key from developer portal for the respective reporting API
  - (d) credentials : Credentials are Security Material Name that is deployed in CI for each of the API. SAP recommend to follow naming convention for better readability (Eg: SAPAribaRealm\_{**RealmName**}\_analytics-reporting)

Note that All the parameters in this Data Source section remains same except realm name, base URL ( depending on the data center of the realm), apikey and credentials.

## 4.2 Data Targets

```
"dataTargets": [
  {
    "SAPSignavio_editor": {
      "type": "SAP Signavio",
      "tenantId": "editor",
      "baseUrl": "https://api. .... .com/api/ingestions/v1/data",
      "apis": {
        "ingestion": {
          "pipelines": [
            {
              "name": "sourcing",
              "credentials": "SAPSignavio_Sourcing_ .... .io"
            },
            {
              "name": "procurement",
              "credentials": "SAPSignavio_Procurement_ .... .io"
            }
          ]
        }
      }
    }
  }
],
```

As shown in above example, data targets are the receiver system details.

- i) type : for Signavio, it is always “SAP Signavio“ ( not to be changed).

- ii) tenantId : SAP Signavio Process Intelligence tenant identifier/ name.
- iii) baseUrl: it is the ingestion API endpoint URL ( not to be changed)
- iv) apis : Since data extractor used ingestion API to ingest csv to SAP Signavio, all the pipeline details that are created are listed in under apis.ingestion.pipelines
  - (a) name : this must match with the pipeline defined in *resourceItemsConfig.resourceItems.{resourceitem}.extensions.pipeline* path in config file. This is used to identify respective pipeline for each fact table in SAP Signavio Process Intelligence tenant
  - (b) credentials : Credentials are Security Material Name that is deployed in CI for ingestion API.

### 4.3 Config

```

"config": [
  {
    "dataSource": "All",
    "api": "All",
    "timezone": "UTC",
    "delimiter": ",",
    "retentionPeriod": 60,
    "complexObjectItemsSeparator": "_",
    "includeTenantId": true,
    "tenantIdColumnName": "Realm",
    "formatters": {
      "date-time": "Timestamp"
    }
  }
],

```

This section includes global configuration parameters for data Extractor which is applicable for all resource items in queue and can be overwritten for each *resourceitem* under *resourceItemsConfig* section (if required).

- i) dataSource : All ( Do not change)
- ii) api : All (Do not change)
- iii) timezone : it represents the timezone to be set for the datetime ranges based on the customer location preference. Allowed values are as below for more common regions

Region	Time Zone Identifier
UTC	UTC
New York (EST)	America/New_York
London (GMT/BST)	Europe/London
India (IST)	Asia/Kolkata

Region	Time Zone Identifier
<b>Sydney (AEST/AEDT)</b>	Australia/Sydney
<b>Dubai (GST)</b>	Asia/Dubai
<b>Singapore (SGT)</b>	Asia/Singapore

- iv) delimiter: delimiter while generating csv files. ( Do not Change).
- v) retentionPeriod: Since the persistent queue are stored in CI data store, clean up of the queue is required to remove completed/Failed items to release memory of CI data store. Queue items are older than specified number of days in this parameter except "lastSuccess" query type queue item are removed. ( change only if required).
- vi) includeTenantId : true/false – This represent if the realm Column needs to be added for all tables in configuration file. This can be overwritten at each resource file. For example, if includeTenantId is set to 'true' at this level, "includeTenantId" at resource item level for specific fact can be set to 'false' which overrides the global config.
- vii) tenantIdColumnName : Csv column name of the realm id that is to be pushed to SAP Signavio Process Intelligence. Similar to that (vi), this value can be overridden if "tenantIdColumnName" is defined at any resource item

#### 4.4 Resource Items Configuration

```
{
  "dataSource": "( SAPAribaRealm_{RealmName}) ",
  "api": "procurement-reporting",
  "resourceItems": {
    "DES_Requisition ": {
      "type": "view",
      "tenantIdColumnName": "Requisition_Realm",
      "name": "Ariba_{RealmName}_auditentry_<from>_<to>",
      "primaryKeys": [
        "Id",
        "Realm",
        "SourceSystem.SourceSystemId"
      ],
    },
  },
  "vectors": [
    {
      "generateUUID": "Id",
      "vector": [
```

```

    "ApprovalRecords"
  ],
  "resourceItem": "requisition_ApprovalRecords",
  "name": "Ariba_{RealmName}_requisition_ApprovalRecords_<from>_<to>"
}
]
"columnMappings": {
  "TimeUpdated": "ModifiedAt"
},
"view": {{View body for Ariba reporting API}}
"extensions": [
  {
    "type": "Signavio",
    "dataTarget": "All",
    "api": "ingestion",
    "pipeline": "sourcing",
    "schema": {{Schema for Signavio Ingestion API}}
  }
]

```

Resource Items configuration contains list of all Resource items ( i.e., SAP Ariba View Names) for each api type ( Sourcing/analytical/procurement). For each SAP Ariba reporting API, list of all resource items ( eg: DES\_AuditEntry/ DES\_RequisitionLineItemFact etc., ) which is same as the View names to be defined in SAP Ariba are listed out. Under each resource items, following parameters are defined.

dataSource must be in the format of “SAPArribaRealm\_{**RealmName**}” (eg: SAPArribaRealm\_AribaRealm) and it should match with dataSources as mentioned in section 3.1

**Note** that this section shouldn’t be changed until unless there are changes to primary keys or if any additional attributes are required for SAP Signavio Process Intelligence. Leave this as-is until unless SAP Signavio Process Intelligence team expects any changes.

- i. **Type:** it takes two values “view/vector”. It is set to View for all the views that are defined in SAP Ariba. It is set to vector when set of additional fields are to be extracted from the main view as separate csv (similar to foreign key table). If a view has vector fields that are to be extracted, it is defined under “vectors” ( path: resourceItems.{{view}}.vectors ) explained in (v.).

Vector file also inherits the primary keys of the view that it is linked to.

Eg: DES\_Requisition view contains approval details (*ApprovalRecords*) of the requisition in vector format that is required for SAP Signavio in separate table/csv. While defining the resource item for such vectors, type should be of vector. ( path : resourceItems.{{vector}}.type)

- ii. **Name :** Name of the csv file in SAP Signavio Process Intelligence pipeline. It is in the format of “Ariba\_{**RealmName**}<signaviotable>\_<from>\_<to>” for



better visualization of the logs in SAP Signavio Process Intelligence. <from> and <to> are the data ranges for which data is extracted.

- iii. **primaryKeys** : Primary keys of the SAP Signavio Process Intelligence table are defined here. If the primary key is in the nested structure of the view, it has to be accessed through dotted notation ( Eg: SourceSystem.SourceSystemId)
- iv. **TenantIdColumnName**: This represents the column name of Realm in csv file. For example, "tenantIdColumnName": "Requisition\_Realm" means that Requisition\_Realm is the column name for Realm while extracting csv for DES\_Requisition
- v. **Vectors**: it contains list of all possible vectors that are to be extracted from resource item.
  1. **generateUUID**: if UUID needs to be generated and added as separate column for vector csv, it needs to be defined and value for this key is the column name for the uuid field ( "generateUUID" : "id" ).
  2. **Vector**: Key of the vector that are to be extracted from resource item view. . If the vector key is in the nested structure of the view, it has to be accessed through dotted notation ( Eg: AllOwners.AllOwners, Organization.Organization etc)
  3. **resourceItem** : name of the resource item of the vector that is defined under resourceItems ( path : *resourceItems.{{vector}}* ) and type for this vector resourceItem is always of type "vector" as explained in i).
  4. **name**: Name of the vector csv file in SAP Signavio Process Intelligence pipeline. It is in the format of "Ariba\_{RealmName}\_<signaviotable>\_<from>\_<to>" for better visualization of the logs in SAP Signavio Process Intelligence. <from> and <to> are the data ranges for which data is extracted.
- vi. **ColumnMapping**: it defines the translation of SAP Ariba View structure field name to desired SAP Signavio Process Intelligence table field as defined in Schema for Ingestion API.

**Note** that column mappings for nested keys ( complex objects keys) are to be updated with underscore '\_' instead of dot separator as mentioned below.

```
Eg: "columnMappings": {
    "Event_EventId": "EventId",
    "Event_VersionNumber": "EventVersion",
    "Event_ItemId": "ItemId",
    "TimeUpdated": "ModifiedAt"
}
```

- vii. **View:** it defines the view structure for SAP Ariba reporting view Management API ( refer <https://help.sap.com/docs/ariba-apis/analytical-reporting-api-for-strategic-procurement-and-operational-procurement/sample-request-to-create-custom-view-template?q=view+Management+API> ). This is used in data extractor for “*initialization*” querytype of the persistent queue while creating/patching views.
- viii. **Extensions:** Extensions contain the information of the target or receiver system, SAP Signavio Process Intelligence in this case.
  1. **Type:** It is always “Signavio” ( for Signavio use case) and shouldn’t be changed.
  2. **Api:** “ingestion” ( not to be changed)
  3. **Pipeline :** it defines the pipeline that is created in SAP Signavio Process Intelligence tenant. This value should match with pipeline credentials maintained in dataTargets section ( Path: “*dataTargets.apis.ingestion.pipelines.name*” ).
  4. **Schema :** this is the schema of the Ingestion API for each resource item ( view/vector). All the fields that are mentioned here will be considered while pushing SAP Ariba extracted data to SAP Signavio Process Intelligence. Since Ingestion API expects all the fields in csv to be defined in schema in the same order as in schema, additional fields that are part of SAP Ariba view but not part of schema are excluded in csv.

#### 4.5 Loading Configuration file to CI

A simple input screen to copy the content of config json content and paste in “Input Config File” input box is provided in the iflow “**Save Resource Configuration file**”.

Save and deploy the iflow.

Configure "Save Resource Configuration file"

Timer **More**

Type: All Parameters

Input Config file: { "dataSources": [ { "SAParibaRealm\_": { "type": "SA..."

when initially deploying the configuration file, use Schedule as “On Deployment” and once deployed, change the scheduler to repeat monthly once ( or as desired). Same process needs to be followed when there is any updates to configuration file.

This is to ensure dataStore (SignavioConfig) is not expired and deleted after 180 days.

Configure "Save Resource Configuration file"

**Timer** More

Timer:

Basic  Advanced

Frequency

Enter As:

Repeat:

**Schedule:**

### Repeat schedule setting:

Configure "Save Resource Configuration file"

**Timer** More

Timer:

Basic  Advanced

Frequency

Enter As:

**Repeat:**

**Every:  Months**

Time Range

Start Date and Time:

End Date and Time:

Time Zone:

Throw exception on schedule expiry

## 5 Data Scheduler

Data Scheduler is used to create/update Persistent Query Queue (QueryStore data store) with new queue items as per input from Administrator.

Below are the key parameters for input

## 5.1 Input Screen

Configure "Data Scheduler"

Timer More

Type: All Parameters

Delta Period (h): 1

Extraction From Timestamp ... : 2023-01-01T00:00:01

Extraction To TimeStamp (Y... : 2024-01-01T00:00:01

Mode (one of: initialization,... : initial

Retry Queue ID List:

scope: DES\_ReceiptFact

Save Deploy Close

### 5.1.1 Delta Period

This represents the Time after which next delta queue needs to be executed and it is applicable only when Mode is set to '*delta*'. If the value is set to 0, it means delta execution is stopped/halted until further changes. Input should be in **hours**

### 5.1.2 Extraction From Timestamp

- It is extraction from timestamp of the scope item(s) that are being executed.
- It must be in the format of YYYY-MM-DDThh:mm:ss (without **zone** – it will be determined from config file)
- It is mandatory when "Extraction To Timestamp" is specified for "initial" mode, and for all cases when mode is "range"

### 5.1.3 Extraction To Timestamp

- It is extraction to timestamp of the scope item(s) that are being executed
- It must be in the format of YYYY-MM-DDThh:mm:ss (without zone – it will be determined from config file)
- It is mandatory when mode is "range"

### 5.1.4 Mode

- It must always be any of "initialization", "initial", "delta" or "range"
  - Initialization: this needs to be used when a new SAP Ariba reporting view needs to be created/patched.
  - Initial: Once the view is created in SAP Ariba using initialization view, initial mode needs to be used to submit jobs with/without date range so that, SAP Ariba jobs are submitted and fetches asynchronously and pushed to SAP Signavio Process Intelligence.

- Range: it behaves same as initial and only difference is that range expects from and to timestamps so that only specific date range is executed, whereas initial is not restricted over date range.
- Delta: Delta needs to be used only when all the past data is extracted using initial/range, and daily delta jobs are needed to be scheduled. Only initial delta job needs to be scheduled with specific delta period.
  - Subsequence deltas are automatically generated once previous delta is completed for specific resource item (scope) in data Extractor.
  - To stop delta schedule of jobs, Delta Period needs to be set to '0'. This ensures new deltas are not generated in data Scheduler and data extractor.

#### 5.1.5 Scope

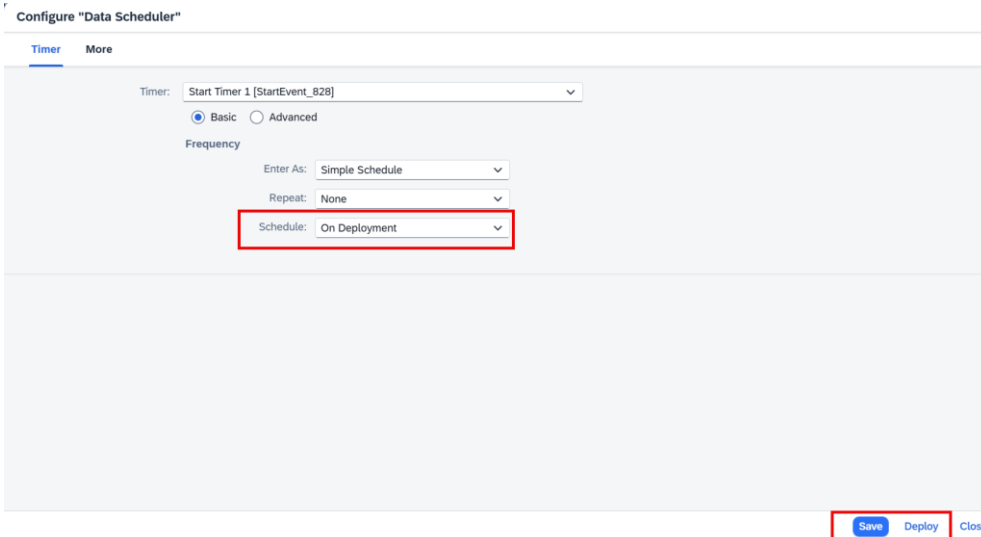
- This takes values of resourceItems (SAP Ariba view) from the config file as explained in “Resource Items Configuration” section.
- If all resource items are to be included, input for this field can be “All” or else, list of resource items separated by comma are to be listed.

#### 5.1.6 Retry Queue Id List

- This input shall be used to retry Failed queue items in persistent queue.
- If a Specific queue item which failed needs to be reexecuted, clear the “**Mode**” and input comma separated queue items from persistent queue.
- Persistent Queue details can be fetched from error notification email or using endpoint as explained in 5.4.1.3.
  - This will reset the queue item status from “Failed” to “Scheduled” and will be picked up by extractor again

## 5.2 How to Run

Once above parameters are defined, Save and deploy the iflow “Data Scheduler”.



This iflow is not to be scheduled frequently using Timer. It is to be Scheduled only with “on deployment” as this is activity to be performed by Administrator.

### 5.3 When to Run

- Data Scheduler needs to be run manually when SAP Ariba views are to be created/patched for specific or for all resource items – **Initialization**
- It needs to be run to schedule the job using **initial**. Subsequent job queue items are taken care by extractor.
- It needs to be run to schedule first **delta** run. Subsequent delta job items are automatically generated from Data Extractor.
- It needs to be run to stop delta (as per business decision) as explained above in different Modes.

## 6 Data Extractor

Data Extractor processes the queue created from data Scheduler and in turn, updates the persistent queue with status and generated subsequent queue items required for reporting API processing.

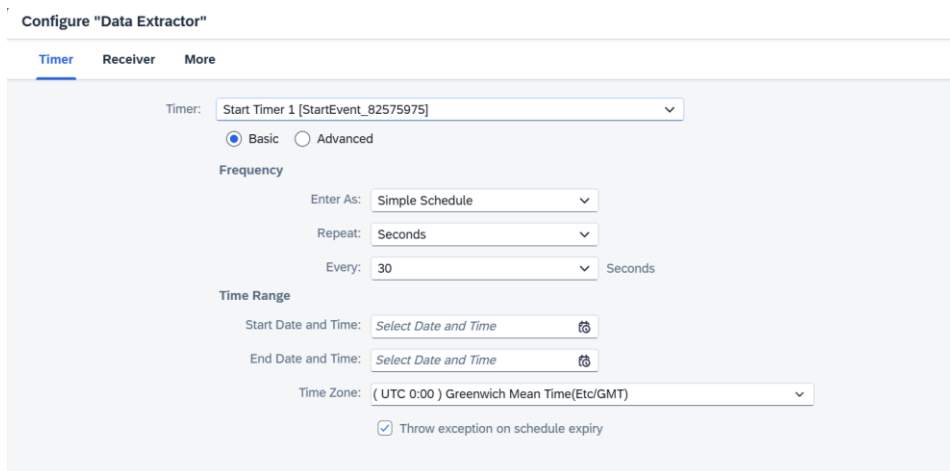
It triggers email notification to recipients when there is any failure in iflow or if status of queue is updated to **Failed**

### 6.1 Input Screen

When configuring the data Extractor, there are preconfigured set of values like Data Scheduler which can be updated. It is recommended not to change any value and deploy as-is, except the **Receiver** settings.

### 6.1.1 Timer

By default, Data Extractor is set to run for every 30 seconds. Do not change unless there is need to increase/reduce the frequency



Configure "Data Extractor"

Timer: Start Timer 1 [StartEvent\_82575975]

Basic  Advanced

Frequency

Enter As: Simple Schedule

Repeat: Seconds

Every: 30 Seconds

Time Range

Start Date and Time: Select Date and Time

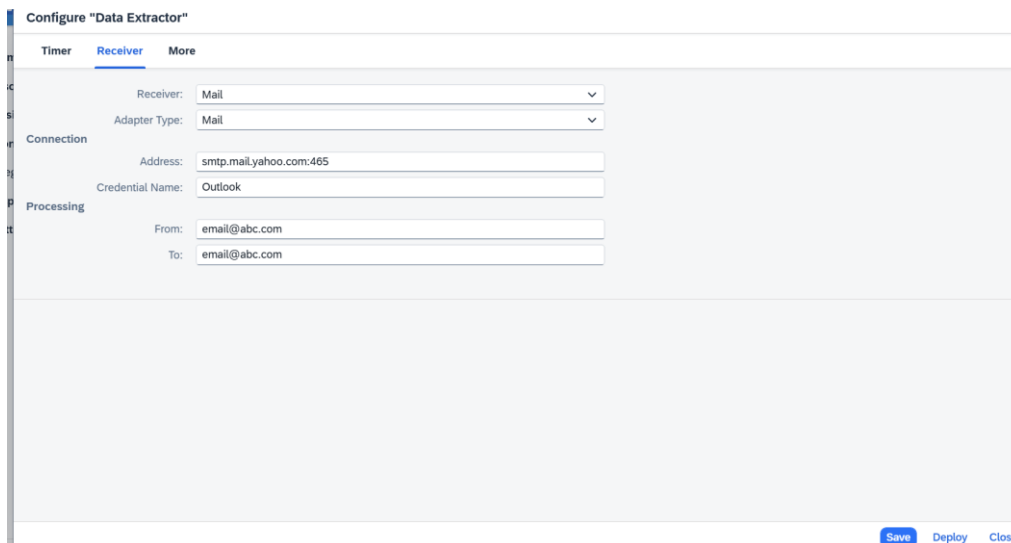
End Date and Time: Select Date and Time

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

Throw exception on schedule expiry

### 6.1.2 Receiver

These settings are for receiver failure email notification of iflow. Desired configuration in CI for setting up email adapter need to be done as prerequisite for data extractor.



Configure "Data Extractor"

Receiver: Mail

Adapter Type: Mail

Connection

Address: smtp.mail.yahoo.com:465

Credential Name: Outlook

Processing

From: email@abc.com

To: email@abc.com

Save Deploy Close

### 6.1.3 More

Under tab more, we have additional parameters as below. PersistentQueryQueueLockRetry and PersistentQueryQueueLockSleep parameters doesn't need to be changed and are defaulted Similar to Data Scheduler input screen.

Configure "Data Extractor"

Timer Receiver **More**

Type:	All Parameters
chunksize:	1000
PersistentQueryQueueLock... :	10
PersistentQueryQueueLock... :	10000
retryinmillis:	30000

Save Deploy Close

**Retryinmillis:** It represents retry time after which persistent queue needs to be executed again. It happens when the persistent queue item goes into "Scheduled" status with retry counter due to time out from API call.

**Chunk size:** this represents the number of entries in csv file before pushing to SAP Signavio Process Intelligence. If entries are >1000, it will split csv and calls ingestion API in sequence. Leave it with 1000-1500 for better performance of API.

## 6.2 How to Run

Once the parameters as mentioned in 6.1 are done, deploy the iflow.

## 6.3 Email body

Email contains

**Queue id** : Unique id of the persistent Queue item in Persistent Queue Data Store ( Query Store).

### **Error Details:**

Error Code from iflow. It can be technical message/ response of the API depending on the failure message

### **MPL ID:**

Iflow ID to search the iflow in CI Monitor messaging

### **Data Extractor Time:**

Data Extractor time as per the time zone set in config file

**Attachment** is also added for failure notification with latest Persistent Queue ( queryStore data store content) at that run. Users can download and check the Queue ID and failed Reason within attachment for further analysis.



Hello Recipient,

We regret to inform you that an exception has been raised in the integration process while processing the Data Extractor in CPI for Queue Id: "90a69365-7297-4432-a307-6fec6baa31ee". Error details are updated in respective Queue item. Please check the attachment for the latest Persistent Queue details.

**Error Details:**

```
{"type":"Error","title":"Something went wrong","detail":"Headers in request schema that are not available in the file: [BaselineSpend_Amount].","backendErrorId":"b8de10bb-f658-417c-b27c-93ca96109c2c","errorKey":"SCHEMA_DATA_MISMATCH","status":400}
```

Data Extractor Time :  
2025-03-14T21:00:31Z

**MPL Id:**

AGfUmO8vMwLGGJrFPz9gOOObKODqT

Thanks,  
CPI Assistant.



Most common error details that would be encountered are as below



Error Causes and Fix.xlsx

## 6.4 Monitoring

If any iflow fails, recipient(s) will receive error email notification as explained in 6.3. Custom header property for queue id is created to search with queue id or MPL ID can be used to filter the iflow that needs further analysis.

Standard HTTP response attachments are added in iflow along with responses of SAP Ariba reporting API as "Ariba log" and SAP Signavio Process Intelligence ingestion response as "signavio log" as attachments to check the status of API if required.

Overview / Monitor Message Processing Message Status Overview Hide Filter Bar

Time: Past Hour Status: All Type: All Package: All Artifacts: Data Extr: X or ID: Message, Correlation or Ap... Q

Mar 17, 2025, 14:09:06 - Mar 17, 2025, 15:09:06

Sender: Receiver: Custom Status: Application Message Type: Custom Header: Persistent Queue Id = 90a69365-7297-4432-a307-6fec6baa31ee

Messages (9) << < 1 > >> ↻

Artifact Name	Status
Data Extractor	Completed
Mar 17, 2025, 14:52:17	3 sec 944 ms
Data Extractor	Completed
Mar 17, 2025, 14:48:34	4 sec 287 ms
Data Extractor	Completed
Mar 17, 2025, 14:44:05	1 sec 377 ms
Data Extractor	Completed
Mar 17, 2025, 14:40:35	1 sec 134 ms
Data Extractor	Completed

**Data Extractor**  
Last Updated at: Mar 17, 2025, 14:52:17

Status Properties Logs Attachments Artifact Details

Correlation ID: AGfX6cYQ2g\_RxzdxCLOih95tDjgX

> Retention Periods

**Custom Headers (1)** Search Q

Name	Value
Persistent Queue Id	90a69365-7297-4432-a307-6fec6baa31ee

### 6.4.1 Additional CI content for Monitoring

There are additional iflows created for debugging and monitoring purposes by customer support team for any issue analysis. These are https endpoint and oauth credentials needs to be setup in subaccount tenant to access these endpoints.

#### 6.4.1.1 Get PersistentQueryQueueLock

This iflow endpoint is used to read the persistent queue lock id and to unlock “PersistentQueryQueueLock” global variable.

unlock parameter needs to be passed (true/false) to unlock the lock id.

**Note** that unlocking should not be done as regular practice. This must be the last resort for the technical support team to unlock the queue when queue is stuck in locked with iflow id because of some technical failure of “Data Extractor” that is not handled in data Extractor.

POST https://signavio... xm/ht

Params Authorization Headers (9) Body Scripts Settings

Query Params

Key	Value	Description
unlock	true	
Key	Value	Description

Body Cookies Headers (18) Test Results 200 OK - 794 ms - 780 B

Raw Preview Visualize

1 Persistent Queue is currently not locked/empty

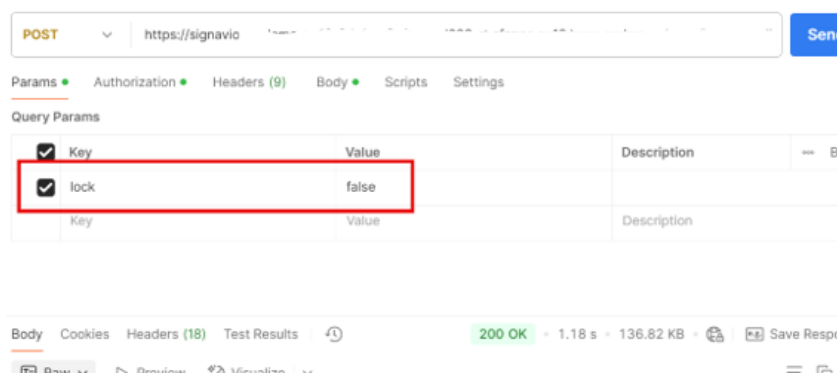
### 6.4.1.2 Read Resource Configuration file

This endpoint needs to be used to read the content of Configuration loaded as per section 4.5.

### 6.4.1.3 Read Persistent Queue Data Store

This endpoint needs to be used to read the content of Persistent Queue data Store (QueryStore). This can be used regularly if required and when required to monitor the processing of queues by operations team.

When posting this https endpoint, make sure that lock parameter is not defined or set to 'false' in productive scenario as this locks the queue with prefix **EXT\_**{{iflow id of this post call}}. In case it is locked by EXT\_ prefix, use 6.1 process to unlock the queue immediately.



### 6.4.1.4 Persistent Queue Data Store Update

This iflow https endpoint is provided to update the status of any persistent queue Ids. This is to be used for following scenarios by operations team who monitors queue regularly.

1. When any queue item is stuck in 'inExecution' status because of any exceptions or timeouts that were not caught.
2. To Stop any "initial" or "range" queue items which are in "scheduled" status because of any valid business reasons.

Note that it should not be used to setup new queue items or for retrying failed queue items, as Data Scheduler is used for it.

Process to update:

1. Read the queue items as mentioned in 6.4.1.3.
2. Identify the queuitems for which status need to be updated
3. Prepare the body as in below format by enclosing the queuitems in Map structure

```
{
  "868234b4-ff82-499c-a2d6-593208d9a7b3|2__57f53c29-0ef9-4da0-bae6-faae5bf88206":
  {
```

```

    "id": "868234b4-ff82-499c-a2d6-593208d9a7b3|2___57f53c29-0ef9-4da0-bae6-faae5bf88206",
    "dataSource": "SAPAribaRealm_{RealmName}",
    "api": "sourcing-reporting",
    "createdTimestamp": 1742318280128,
    "repositoryItem": "DES_AuditEntry",
    "url": "https://eu.openapi.ariba.com/api/sourcing-reporting-jobresult/v1/prod/jobs/e6726****f28-af02-49cb-9a9d-1e51e2d3c28b1742317560591/files/Fm8eqw1sn.zip?realm={RealmName}&pageToken=QjZuV0FKSEISMERPR3IS",
    "queryType": "initial",
    "status": "Failed", ---- canceled or Failed as per the business decision
    "executionTimestamp": 1742318280128,
    "retry": 0,
    "fromTimestamp": 1672531201000,
    "toTimestamp": 1704067201000,
    "updatedTimestamp": 1742318280128
    "failedReason" : " Canceling the job due to XXXXXXXX" ---- Optional
  },
  "868234b4-ff82-499c-a2d6-593208d9a7b3|2___90a0b14c-84ae-4afa-9fb5-5893faf8079a": {
    "id": "868234b4-ff82-499c-a2d6-593208d9a7b3|2___90a0b14c-84ae-4afa-9fb5-5893faf8079a",
    "dataSource": "SAPAribaRealm_{RealmName}",
    "api": "sourcing-reporting",
    "createdTimestamp": 1742318280128,
    "repositoryItem": "DES_AuditEntry",
    "url": "https://eu.openapi.ariba.com/api/sourcing-reporting-job/v1/prod/jobs?realm={RealmName}&pageToken=QjZuV0FKSEJ4TXdybnIR",
    "queryType": "initial",
    "status": "Scheduled", ---- canceled or Failed as per the business decision
    "executionTimestamp": 1742318280128,
    "retry": 0,
    "fromTimestamp": 1672531201000,
    "toTimestamp": 1704067201000,
    "updatedTimestamp": 1742318280128
    "failedReason" : " Canceling the job due to XXXXXXXX" ---- Optional
  }
}

```

- Execute the post https end point to update the queue by passing the body content created in iii.

