SAP Ariba

SAP Ariba Integration with SAP Signavio Process Intelligence– Deployment Guide

Contents

1 Document History

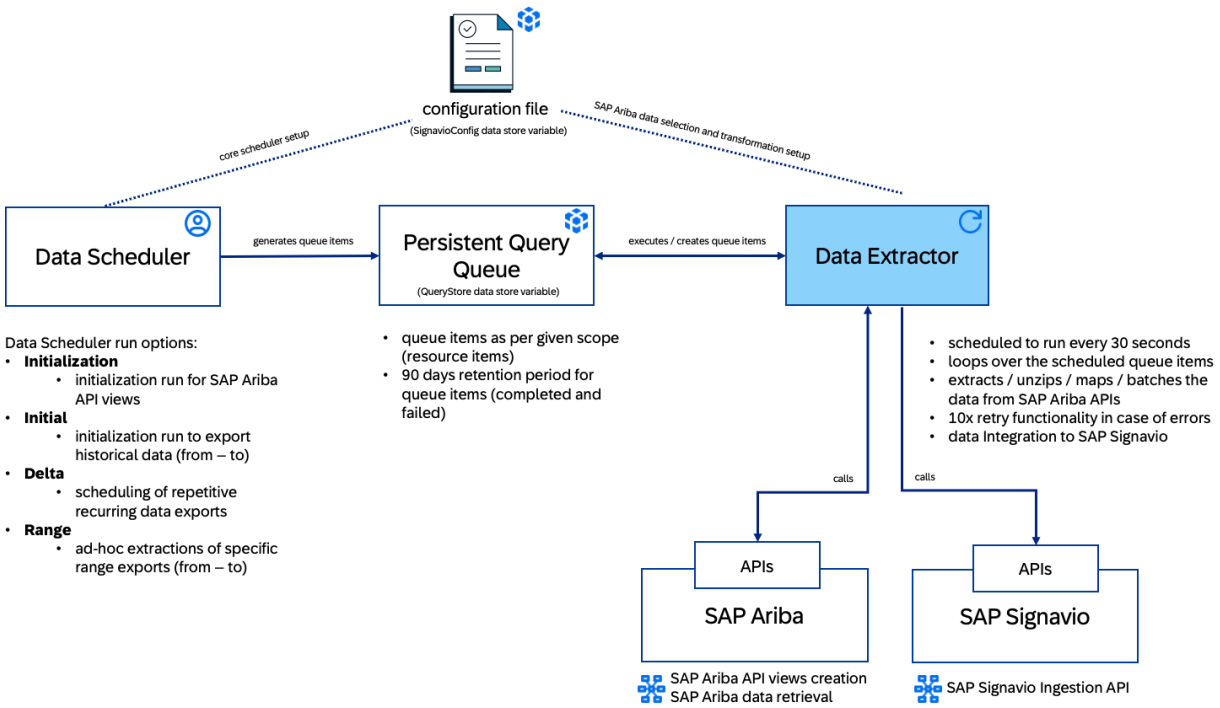
Document Version	Date	Author	Comment
1.0	2025/03/15	SAP	Initial release
1.0.1	2025/05/08	SAP	Screenshots updated
2.0.0	2025/07/11	SAP	Updated documentation for changes to configuration file
2.0.1	2025/10/14	SAP	Texts revised, graphics improved
2.0.2	2025/10/27	SAP	Updated documentation for new parent-child configuration
2.0.3	2025/12/02	SAP	Multi-vector, complex JSON flattening supported from Version 2.2.0
2.0.4	2026/02/06	SAP	Version 2.2.1 Enhancements: • Added support for extracting inactive records. • Fixed scheduled, time-based delta extraction. • Added delta Buffer parameter in Data Extractor and index functionality logic

2 Essential Steps for Configuring the SAP Ariba Connector

This section provides an overview of the setup and configuration of the SAP Ariba Connector to SAP Signavio Ingestion API. All steps are explained in more detail in the individual sections of the document.

1. Import the package “SAP Ariba Integration with SAP Signavio Process Intelligence” into your CI tenant. This package contains 7 iFlows (integration flows).
2. Prepare the configuration file and create or update the datastore for the config resource file.
3. Create security credentials for all APIs in scope, following the naming convention in the configuration file. This includes credentials for analytical, procurement, and sourcing reporting APIs, as well as the SAP Signavio Ingestion API. For more information on asynchronous operational and analytical reporting APIs, see the SAP Ariba Developer Portal.
4. Set up security credentials for error email notifications, connectivity checks, and import the SMTP certificate for the email adapter.
5. Run the Data Scheduler first to create the persistent queue before running the extractor.
6. To check the persistent queue or lock status, use HTTP POST endpoints via tools like Postman or Bruno. Make sure credentials are created in the subaccount tenant to access the iFlow endpoints. These endpoints are mainly for debugging or issue analysis and must not be used to alter the persistent queue during regular operations.

3 Setting Up and Running the Integration



The typical process for setting up and running the integration is as follows:

1. SAP provides a configuration template for updating the following fields:
   - Realm name in the dataSource key (SAPAribaRealm_{Realm})
   - TenantId
   - Credentials
   - TokenServiceUrl
   - BaseUrl
     - Make sure that tokenServiceURL (OAuth Server URL Prefix) and baseURL (base URL for API calls) in the dataSources section are set according to your SAP Ariba API data center. Here are some examples:
       - US baseURL: https://openapi.ariba.com, tokenServiceURL: https://api.ariba.com
       - EU baseURL: https://eu.openapi.ariba.com, tokenServiceURL: https://api-eu.ariba.com
       - AU baseURL: https://openapi.au.cloud.ariba.com, tokenServiceURL: https://api.au.cloud.ariba.com
       - IN baseURL: https://openapi.in.cloud.ariba.com, tokenServiceURL: https://api.in.cloud.ariba.com
       - CN baseURL: https://openapi.sapariba.cn, tokenServiceURL: https://api.sapariba.cn
       - JP baseURL: https://openapi.jp.cloud.ariba.com, tokenServiceURL: https://api.jp.cloud.ariba.com

        - UAE baseURL: https://mn1.openapi.ariba.com, tokenServiceURL: https://api.mn1.ariba.com
        - KSA baseURL: https://mn2.openapi.ariba.com, tokenServiceURL: https://api.mn2.ariba.com
    - Once you have updated the configuration, load it as described in section 4.6.

2. Run the Data Scheduler by configuring and deploying it once. See section 5.1 for more details on input screen parameters.
    - Run the Data Scheduler in "initialization" mode for the first time to create all SAP Ariba views for the asynchronous reporting API. Initialization is also used to patch the view with additional attributes.
    - After updating the parameters, choose Save & Deploy the iFlow named **Data Scheduler**.

3. Deploy the Data Extractor to run every 30 seconds and process Data Scheduler queue items. Make sure that the receiver email notification setup is complete, see section 6.1.2.

4. Repeat the Data Scheduler step 2 with "initial" mode after successful "initialization", specifying the **Extraction From Timestamp (mandatory)** and **Extraction To Timestamp** (optional) parameters for all views in scope.

5. Monitor the queue to ensure items are processed without errors.

6. Schedule "delta" loads for incremental updates after "initial" loads are successful, using the Data Scheduler with the desired delta period for each resource view item. This is optional and can be set as needed by the business.

4 Configuration File

The configuration file contains all security parameters for data sources (realm, reporting API base URLs, OAuth credentials), data targets (SAP Signavio Process Intelligence pipelines and credentials), configuration parameters, and resource items (fact tables from SAP Ariba reporting to be pushed to SAP Signavio Process Intelligence in CSV format).

The package includes three versions of the configuration file:

- single data source and single target
- FPC (Parent & Child setup) data sources and targets – ParentChildConfig_Template
- Multiple data sources (without FPC Ariba architecture) setup

Note: Do not change the structure or keys of the configuration file, as this can cause inconsistencies with the Data Scheduler or Data Extractor logic.

4.1 Data Sources

```
"dataSources": [
  {
    "SAPAribaRealm_{Realm1}": {
      "type": "SAP Ariba Realm",
      "tenantId": "{Realm1}",
      "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": "CsXcJ7GW7K4Ppr2MpISksM[redacted]",
          "credentials": "SAPAriba_{Realm1}_analytics-reporting"
        },
        "sourcing-reporting": {
          "createView": "/api/sourcing-reporting-view/v1/prod/viewTemplates",
          "jobSubmit": "/api/sourcing-reporting-job/v1/prod/jobs",
          "apikey": "9DzC6uFewQdfmXrNKwRx7j[redacted]",
          "credentials": "SAPAriba_{Realm1}_sourcing-reporting"
        },
        "procurement-reporting": {
          "createView": "/api/procurement-reporting-view/v2/prod/viewTemplates",
          "jobSubmit": "/api/procurement-reporting-job/v2/prod/jobs",
          "apikey": "OKRUaiIbONjMjJ1zxsXtRp[redacted]",
          "credentials": "SAPAriba_{Realm1}_procurement-reporting"
        }
      }
    }
  }
```

The data sources section contains the sender system details. The data source key SAPAribaRealm_{Realm1} corresponds to the 'Enter Data Source' field in Data Scheduler input screen (see section 5.1.2).

The configuration file accommodates the definition of multiple data sources (SAP Ariba realms). To facilitate this, a sample template named config_template_multidataSource.json is available in the content package. For configurations involving more than two data sources, the SAPAribaRealm_{Realm1} section needs to be replicated accordingly.

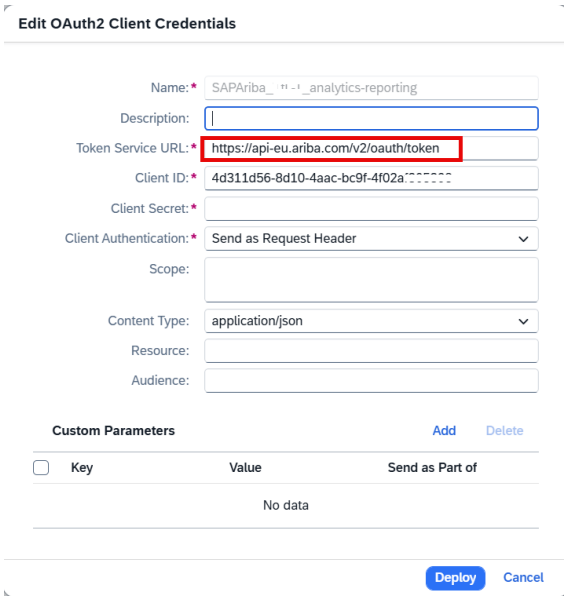
- tenantId: This is a placeholder for the customer realm id and {Realm} to be replaced with the actual realm name.
- tokenServiceURL: Populate this with the SAP Ariba reporting API token URL. See the SAP Ariba Developer Portal or section 3 for further details.
- baseURL: The SAP Ariba reporting API base URL, used in the Data Scheduler during the creation of queue items, varies depending on the SAP Ariba data center hosting the realm.
    - apis: This section contains a list of all reporting APIs that are applicable for that realm. In the context of SAP Signavio, we employ all analytical, sourcing, and procurement reporting APIs. This must align with the "api" key (path: resourceItemsConfig.api) in the resource items configuration (detailed below).
        - createView: This is an extension of the base URL used in the Data Scheduler for initialization mode. Do not modify unless the version is updated in the SAP Ariba development portal.

- jobSubmit: This is an extension of the base URL used for initial, delta, and range modes in the Data Scheduler. Similarly, do not alter this unless the version is updated in the SAP Ariba development portal.
- apiKey: The API key from the developer portal for the corresponding reporting API.
- credentials: This pertains to the security material name that is deployed in CI for each of the APIs. SAP recommends adopting the following naming convention for better readability:
    - SAPAribaRealm_{Realm}_analytics-reporting
    - SAPAribaRealm_{Realm1}_analytics-reporting
    - SAPAribaRealm_{Realm2}_analytics-reporting

Note: Within the Data Source section of the configuration file, all parameters must remain unaltered except for the following: realm name, baseURL (which is contingent upon the data center of the realm), apiKey, and credentials.

4.1.1 Data Source Credential Setup

To create security material for data sources, navigate to **Monitor > Manage Security** in CI. For SAP Ariba APIs, create credentials using OAuth2 client credentials, ensuring that the names match those in the configuration file, such as SAPAribaRealm_{Realm}_analytics-reporting. The token service URL must end with /oauth/token. See the SAP Ariba Developer Portal for the client ID and secret parameters.



4.2 Data Targets

```
"dataTargets": [
  {
    "SAPSignavio_editor": {
      "type": "SAP Signavio",
      "tenantId": "editor",
      "baseURL": "https://api.eu.signavio.cloud.sap/spi/ingestions/v1/data",
      "apis": {
        "ingestion": {
          "pipelines": [
            {
              "name": "sourcing_{Realm1}",
              "credentials": "SAPSignavio_Sourcing_{Realm1}"
            },
            {
              "name": "procurement_{Realm1}",
              "credentials": "SAPSignavio_Procurement_{Realm1}"
            },
            {
              "name": "sourcing_{Realm2}",
              "credentials": "SAPSignavio_Sourcing_{Realm2}"
            },
            {
              "name": "procurement_{Realm2}",
              "credentials": "SAPSignavio_Procurement_{Realm2}"
            }
          ]
        }
      }
    }
  }
],
```

The dataTargets section contains the receiver system details for SAP Signavio. The following parameters are required to configure data targets appropriately:

- type: This is set to 'SAP Signavio' and remains unchanged.
- tenantId: This is the SAP Signavio tenant identifier or name.
- baseURL: This is the SAP Signavio Ingestion API endpoint URL and must not be modified.
    - apis: This lists the SAP Signavio pipeline details
        - name: The name corresponds to the pipeline defined in the resourceItemsConfig section (specifically, resourceItems.{resourceitem}.extensions.pipeline) within the configuration file. This ensures the correct identification of respective pipelines for each fact table in the SAP Signavio tenant.
        - Credentials: This refers to the security material name deployed in the CI package for the SAP Signavio Ingestion API.

In cases where multiple data targets are used (for example, ingesting data into different SAP Signavio workspaces), the respective realm names (e.g., {Realm1}) need to be appended to both the name and credentials fields. For instance:

- "name”: procurement_{Realm1}
- "credentials”: SAPSignavio_Procurement_{Realm1}

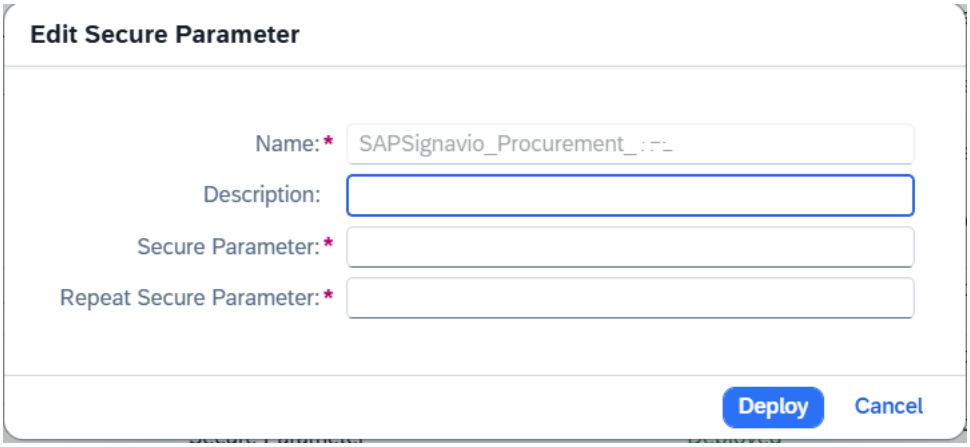
Ensuring these configurations are accurately set enables proper data targeting and secure communication with the SAP Signavio system.

For scenarios where you need separate data targets, allowing different SAP Signavio workspaces, this is fully supported. As data target is a list-type configuration, adding a new target is like adding a new data source. However, for a new target, replace the key '**SAPSignavio_editor**' with the desired target name. This target name must match the value specified in the path: resourceItems.{resourceitem}.extensions.dataTarget ensuring that the resource item is correctly linked to the intended data target and its respective pipeline.

If the value for this path is set to 'All', the default data target **SAPSignavio_editor** is used for the resource item that is being executed. If a specific value other than 'All' is provided, the system will look for the corresponding data target and the pipelines associated with it to fetch the necessary credentials.

4.2.1 Data Target Credential Setup

To create security material for data targets, navigate to **Monitor > Manage Security** in CI. For the SAP Signavio Ingestion API, create credentials as secure parameters, with names matching those in the configuration file, for example SAPSignavio_Procurement_{Realm}.



4.3 Pipeline Mappings

```
],
"pipelineMappings": [
  {
    "procurement": {
      "SAPAribaRealm_{Realm1}": "procurement_{Realm1}",
      "SAPAribaRealm_{Realm2}": "procurement_{Realm2}"
    }
  },
  {
    "sourcing": {
      "SAPAribaRealm_{Realm1}": "sourcing_{Realm1}",
      "SAPAribaRealm_{Realm2}": "sourcing_{Realm2}"
    }
  }
],
```

This section includes the mappings between Data Source Keys (see section 4.1) and Data Target (pipeline names as explained in section 4.2 under path: "dataTargets.apis.ingestion.pipelines.name"). This mapping identifies the relation between the SAP Ariba realm and the SAP Signavio pipeline in scope for each resource item.

Pipeline mappings contain two list items: procurement and sourcing. These keys must not be changed since these are linked to the pipeline under extensions section in resource items configuration as explained in section 4.5.

Each data source needs to be mapped to the respective procurement and sourcing pipelines if data is sent to separate pipelines as shown in the above screenshot.

If data from one or multiple data sources will be sent to only one pipeline, the same pipeline name must be mapped to both procurement and sourcing list items, for example:

```
],
"pipelineMappings": [
  {
    "procurement": {
      "SAPAribaRealm_{Realm1}": "pipeline_injestion"
    }
  },
  {
    "sourcing": {
      "SAPAribaRealm_{Realm1}": "pipeline_injestion"
    }
  }
],
```

4.4 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 the Data Extractor, applicable to all resource items in the queue. These can be adapted for specific resource items.

- dataSource: All (do not change)
- api: All (do not change)
- Timezone: The timezone for the date-time ranges based on the customer's preference. Allowed values are as below for common regions.

Region	Time Zone Identifier
UTC	UTC
New York (EST)	America/New_York
London (GMT/BST)	Europe/London
India (IST)	Asia/Kolkata
Sydney (AEST/AEDT)	Australia/Sydney
Dubai (GST)	Asia/Dubai
Singapore (SGT)	Asia/Singapore

- delimiter: delimiter while generating csv files (do not change)
- retentionPeriod: Since the persistent queue is stored in CI data store, a cleanup of this queue is required to release memory of the CI data store. Queue items that are older than a specified number of days in this parameter are removed.
- includeTenantId: This parameter configures if the realm column should be added for all tables in the configuration file

- tenantIdColumnName : CSV column name of the realm that is to be pushed to SAP Signavio.

Note: Within the config section, all parameters must remain unaltered except for the following if special scenarios occur: includeTenantId and tenantIdColumnName. These can be overwritten in each resource file. For example, if includeTenantId is set to 'true' at this level, includeTenantId at resource item level for specific fact tables can be set to 'false' to override the global configuration.

4.5 Resource Items Configuration

```
{
  "dataSource": "SAPAribaRealm_{Realm2}",
  "api": "procurement-reporting",
  "resourceItems": {
    "DES_Requisition": {
      "type": "view",
      "name": "Ariba_{Realm2}_requisition_<from>_<to>",
      "primaryKeys": "/*Placeholder for primarykey set in case it is different for dataSources*/",
      "vectors": [
        {
          "generateUUID": "Id",
          "vector": [
            "ApprovalRecords"
          ],
          "resourceItem": "requisition_ApprovalRecords",
          "name": "Ariba_{Realm2}_requisition_ApprovalRecords_<from>_<to>"
        }
      ],
      "columnMappings": "/*Placeholder for columnmappings set in case it is different for dataSources*/",
      "view": "/*Placeholder for view body in case view is different for dataSources*/",
      "extensions": [
        {
          "type": "Signavio",
          "dataTarget": "All",
          "api": "ingestion",
          "pipeline": "procurement",
          "schema": "/*Placeholder for schema body in case schema is different for dataSources*/"
        }
      ]
    },
```

The resourceItems configuration encompasses a comprehensive list of all resource items (for example, SAP Ariba view names) for each data source and API type (Sourcing, Analytical, Procurement). For each SAP Ariba reporting API, a list is provided of all resource items like DES_AuditEntry and DES_RequisitionLineItemFact, corresponding to the view names defined in SAP Ariba.

Under each resource item, the following parameters are specified:

- dataSource: This must be in the format "SAPAribaRealm_{Realm1}" and must match one of the dataSources described in section 4.1.
- type: This parameter can take two values: "view" or "vector." It is set to "view" for all the views that are defined in SAP Ariba. It is set to "vector" when additional fields need to be extracted from the main view as a separate CSV file (like a foreign key table). If a view contains vector fields that need to be extracted, they are defined under "vectors" (path: resourceItems.{{view}}.vectors) as explained below).
    - A vector file inherits the primary keys of the view to which it is linked. For example, the DES_Requisition view contains approval details (ApprovalRecords) in a vector

format that is required for SAP Signavio in a separate table/CSV. When defining the resource item for such vectors, the type must be set to "vector". (Path: resourceItems.{{vector}}.type)

- name: This parameter specifies the name of the CSV file in the SAP Signavio pipeline. It must be in the format "*Ariba_{Realm}<signaviotable>_<from>_<to>*" for better visualization of the logs in SAP Signavio. The <from> and <to> elements indicate the data ranges for which data is extracted.
- primaryKeys: This parameter is a placeholder as described in the template. It's used when the list of primary keys differs for various data sources. The value must be in the same format as the primary keys in the section about resource items common configuration. Leave it as-is with the placeholder if no changes are required.
- tenantIdColumnName: This parameter represents the column name for the realm in the CSV file. For example, "tenantIdColumnName": "Requisition_Realm" means that "Requisition_Realm" is the column name for the realm when extracting the CSV for DES_Requisition.
- vectors: This parameter contains a list of all possible vectors that need to be extracted from the resource item.
    - generateUUID: If a UUID needs to be generated and added as a separate column for the vector CSV, it must be defined. The value for this key is the column name for the UUID field, for example, "generateUUID": "id".
    - Vector: This specifies the key of the vector to be extracted from the resource item view. If the vector key is in a nested structure within the view, it must be accessed through dotted notation, for example AllOwners.AllOwners and Organization.Organization.
    - resourceItem: This is the name of the resource item of the vector as defined under resourceItems (path: resourceItems.{{vector}}), and the type for this vector resourceItem is always "vector".
    - name: This specifies the name of the vector CSV file in the SAP Signavio pipeline. It must be in the format "*Ariba_{Realm}_<signaviotable>_<from>_<to>*" for better visualization of the logs in SAP Signavio. The "<from>" and "<to>" elements indicate the data ranges for which data is extracted.

Note: Multi-vector extraction is supported in the CI package from version 2.2.0 and can be used when extending the standard accelerator with additional vector-based resource items. If a view contains multiple vectors, each vector can be configured to generate its own table in Signavio, as long as the vector has valid primary keys (either inherited from the parent view or explicitly defined in the vector resource item).

For example, a view such as DES_Requisition may include multiple vectors such as:

- DES_Requisition_ApprovalRecords
- DES_Requisition_ApprovalRequests

Each of these vectors can be defined under resourceItems with type: "vector" and will be extracted as separate tables in Signavio when primary keys and column mappings are correctly configured.

- columnMapping: This parameter is a placeholder as described in the template. It's used when column mappings differ for various data sources. The value must be in the same format as the column mappings in the section about resource items common configuration. Leave it as-is with the placeholder if no changes are required.

  Note: The extractor now supports automatic flattening of nested JSON structures within Ariba API responses. Nested fields that appear in dotted notation (for example, 'Description.Price', 'Supplier.PrimaryAddress.City') are flattened into top-level columns during CSV generation.
  - Dots ('.') in field names are replaced with underscores ('_') when written into the CSV header.
  - Each flattened field must exist in the target Signavio schema. Any additional fields that do not exist in the schema will be ignored.
  - Column mappings must reference the flattened field names using underscores.

  Example:
  ```
  "columnMappings":
  {
  ... ,
  "Supplier_PrimaryAddress_City" : "SupplierCity"
  }
  ```

- view: This parameter is a placeholder as described in the template. It's used when the view differs for various data sources. The value must be in the same format as the primary keys in the section about resource items common configuration. Leave it as-is with the placeholder if no changes are required.
- extensions: This parameter contains information about the target or receiver system, here SAP Signavio.
  - type: This must always be set to "Signavio" for the SAP Signavio use case and must not be changed.
  - api: This is set to "ingestion" and must not be changed.
  - pipeline: This defines the pipeline created in the SAP Signavio tenant.
    - For version 1.0.0 of the CI package, this value must match the pipeline credentials maintained in the dataTargets section, path: "dataTargets.apis.ingestion.pipelines.name".
    - In version 2.0.0 of this CI package, pipeline mappings on resource item level are introduced as explained in section 4.3 to simplify maintenance as attribute values are fixed with either "procurement" or "sourcing". This must not be changed.
  - schema: This parameter is a placeholder as described in the template. It's used when the schema differs for various data sources. The value must be in the same

format as the schema in the section about resource items common configuration. Leave it as-is with the placeholder if no changes are required.

Note: For multiple data sources, the JSON section for dataSource, api, and resourceItems must be repeated for all the data sources in scope. See the resourceItemsConfig section in the "Configuration File Template_Multiple Data Source" template to understand, which includes {Realm1} and {Realm2} placeholders.

Similarly, for FPC setup with multiple child realms, the JSON section for dataSource, api, and resourceItems must be repeated for all child realms to extract relevant procurement data from each child site.

4.6 Resource Items Common Configuration

```
"resourceItemsCommonConfig": [
    {
        "SAP Ariba Realm": {
            "DES_RequisitionLineItemFact": {
                "primaryKeys": [
                    "Realm",
                    "RequisitionId",
                    "RequisitionLineNumber",
                    "SourceSystem.SourceSystemId",
                    "SplitAccountingNumber"
                ],
                "columnMappings": {
                    "TimeUpdated": "ModifiedAt"
                },
                "view": {  ---> {{View body for Ariba reporting API}}
                "extensions": {
                    "schema": {  ---> {{Schema for Signavio Ingestion API}}
                }
            },
```

This section outlines the standard configuration parameters for all resource items, irrespective of the number of data sources or data targets and is applicable to all combinations of APIs and data sources defined in the resource item configuration (see section 4.5).

Note: This section must not be updated or modified under any circumstances, as these configurations are standard for data ingestion to SAP Signavio for all data sources. Changes can be made in the "resourceItemsCommonConfig" section to overwrite the changes globally for all data sources.

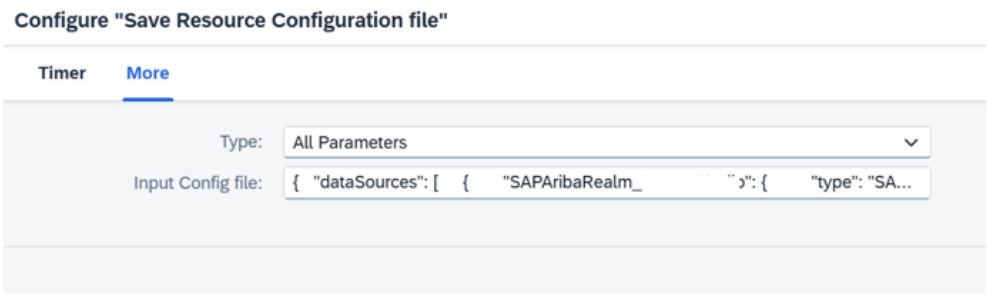
In the scenario of multiple data sources, if there is a requirement to have different configurations for a resource item under a specific data source and API combination, placeholders are provided for each of these parameters under the resourceItemsConfig section as explained in section 4.5.

- primaryKeys: The primary keys of the SAP Signavio tables are defined here. If the primary key is in a nested structure of the view, it must be accessed through dotted notation, for example SourceSystem.SourceSystemId.
- columnMapping: This parameter defines the translation of the SAP Ariba view structure field names to the desired SAP Signavio table fields as defined in the schema of the SAP Signavio Ingestion API. Note that column mappings for nested keys (complex object keys) must be updated with an underscore ('_') instead of a dot separator as shown below:
    - ```
      "columnMappings": {
          "Event_EventId": "EventId",
          "Event_VersionNumber": "EventVersion",
          "Event_ItemId": "ItemId",
          "TimeUpdated": "ModifiedAt"
      }
      ```
- view: This parameter defines the view structure for the SAP Ariba Reporting View Management API (see SAP Help Portal and SAP Ariba Developer). This is used in the data extractor for the "Initialization" query type of the persistent queue while creating or patching views.
- extensions: This parameter contains information about the target or receiver system, here SAP Signavio.
    - schema: This is the schema of the SAP Signavio Ingestion API for each resource item (view/vector). All fields described here will be considered while pushing data SAP Ariba extracted data to SAP Signavio. Since the SAP Signavio Ingestion API expects all fields in the CSV to be defined in the schema in the same order as in the schema, additional fields that are part of the SAP Ariba view but not part of the schema are excluded in the CSV.

4.7 Loading Configuration file to CI

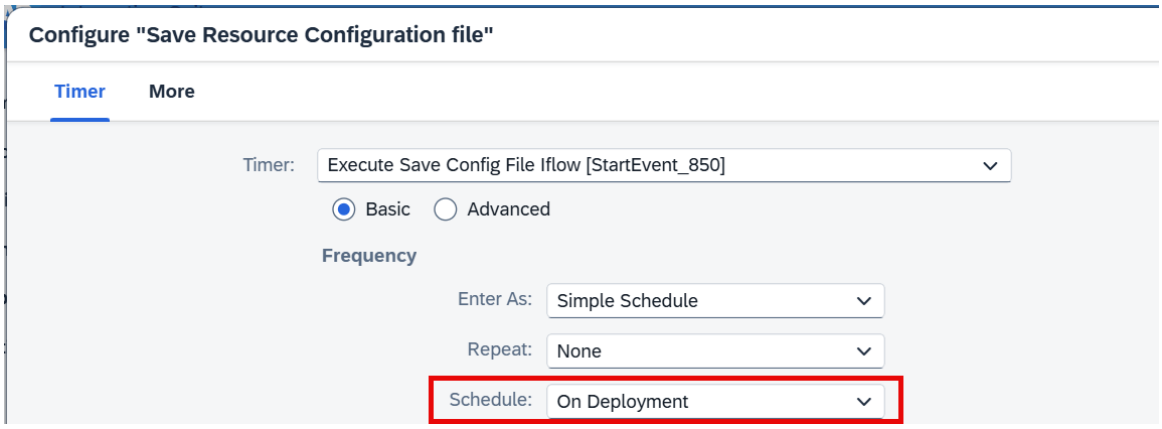
Use the input screen in the "Save Resource Configuration File" iFlow to upload the config JSON content. Save and deploy the iFlow.

Note: We recommend you to check and validate the format of the configuration JSON file using online JSON formatter tools (for example, JSON Formatter) before deploying the iFlow.

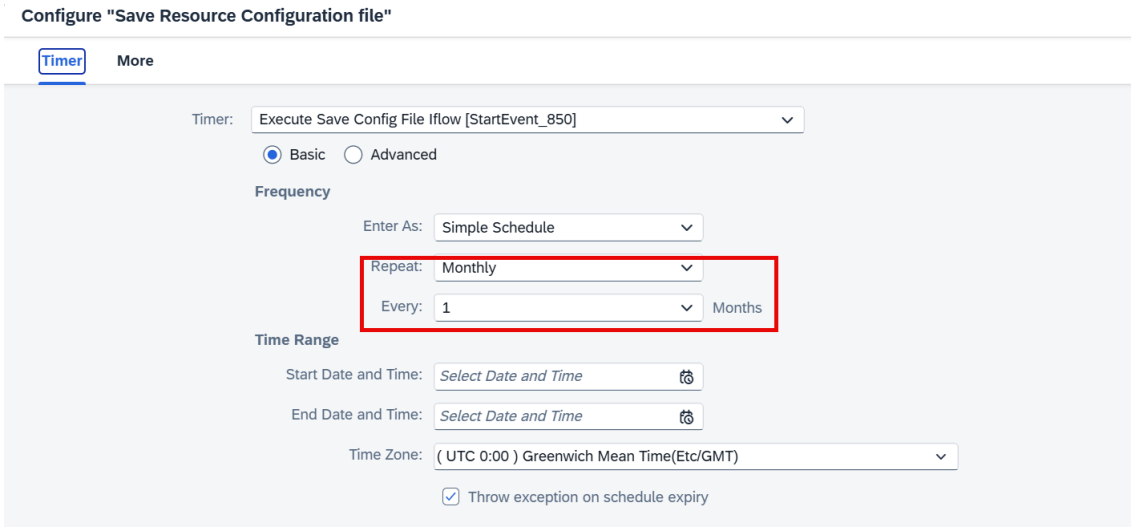


When initially deploying the configuration file, set the schedule to 'On Deployment'. Once deployed, change the scheduler to repeat monthly or as desired. This process must be followed
      ```

whenever updates are made to the configuration file. This ensures that the dataStore (SignavioConfig) does not expire and gets deleted after 180 days.



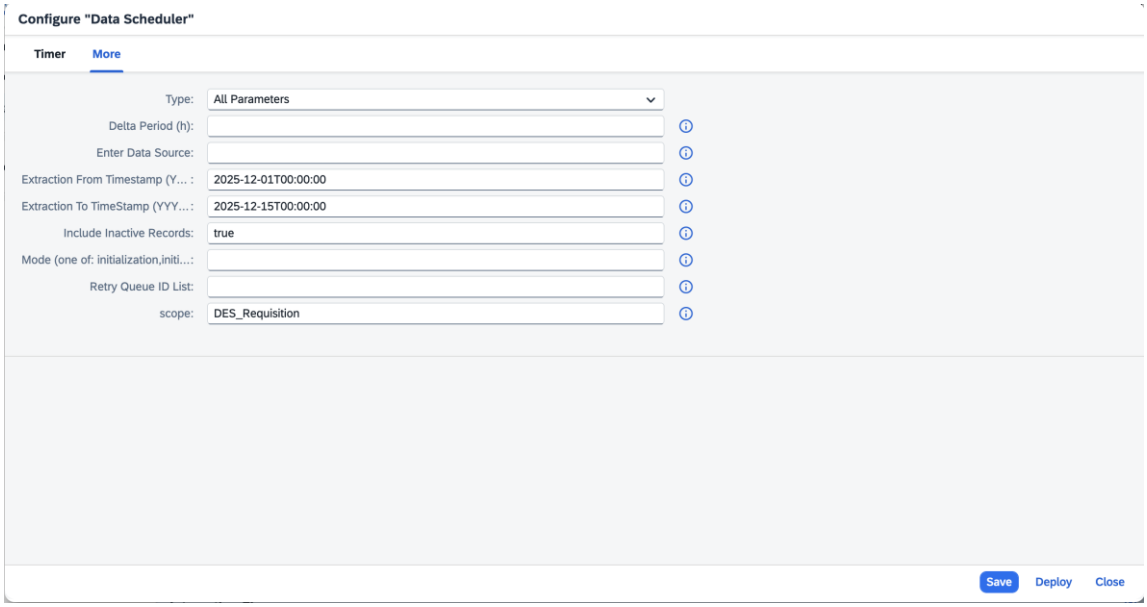
Repeat schedule setting:



5 Data Scheduler

The Data Scheduler is used to create or update the persistent query queue (QueryStore data store) with new queue items based on input from the administrator.

5.1 Input Screen



5.1.1 Delta Period

This parameter represents the time after which the next delta queue needs to be executed and is applicable only when the **Mode** parameter is set to 'delta'. If the value is set to 0, the delta execution is stopped/paused until further changes. Input must be provided in hours.

5.1.2 Enter Data Source

This is a mandatory field for all scheduler executions, and it takes one data source value per scheduler. This parameter supports multiple data sources in the configuration file, and scheduling needs to occur for all or specific resource items with respect to the data source. The value must match one of the data source keys described in the configuration file, for example SAPAribaRealm_{Realm1} or SAPAribaRealm_{Realm2}). See section 4.1 on data source configuration.

5.1.3 Extraction From Timestamp

This parameter specifies the starting timestamp for the extraction of the scope item(s) that are being executed. It must be formatted as YYYY-MM-DDThh:mm:ss, without timezone information, which will be determined from the configuration file. This field is mandatory when the "Extraction To Timestamp" parameter is specified for "initial" mode and for all cases when the mode is set to "range".

5.1.4 Extraction To Timestamp

This parameter specifies the ending timestamp for the extraction of the scope items that are being executed. It must be formatted as YYYY-MM-DDThh:mm:ss, without timezone information, which

will be determined from the configuration file. This field is mandatory when the mode is set to "range".

5.1.5 Include Inactive Records

This parameter specifies whether inactive records (i.e., previous versions of business objects that become inactive when a document is changed or versioned) should be included in the extraction.

- When set to true, both active and inactive records are extracted, ensuring complete version history and accurate process analysis.
- When set to false, only active records are extracted and historical versions are excluded.

For process mining use cases, setting this parameter to true is recommended to maintain full traceability of record changes and avoid data gaps in downstream analysis.

In case of delta loads, when the first delta is scheduled using the Data Scheduler, subsequent delta jobs are automatically generated by the Data Extractor and reuse the same extraction configuration. Therefore, the value of Include Inactive Records defined for the first delta remains effective for all following delta executions.

To modify this setting, the existing delta chain must be stopped, and a new delta must be scheduled with the updated configuration.

5.1.6 Mode

It is crucial to use one of the following options: "initialization", "initial", "delta", or "range".

- **Initialization:** This mode is to be employed when creating or patching a new SAP Ariba reporting view.
- **Initial:** Once the view is established in SAP Ariba using the initialization view, the initial mode is required to submit jobs with or without a date range. This facilitates SAP Ariba job submissions, fetches data asynchronously, and pushes it to SAP Signavio.
- **Range:** This mode functions similarly to the initial mode but differs in that it requires both 'from' and 'to' timestamps. This allows the execution of a specific date range, while the initial mode is not constrained by a date range.
- **Delta:** The delta mode must be used only after all past data has been extracted using the initial mode, and daily delta jobs need to be scheduled. The first delta job must be scheduled with a specific delta period. Subsequent deltas are automatically generated once the previous delta is completed for the specified resource item (scope) in the data extractor.
  The execution time of the next delta is calculated based on the delta job trigger time, the configured Delta Period, and an optional Delta Buffer (in minutes) defined in the Data Extractor. This prevents scheduling drift caused by variations in file extraction duration.

  To pause the delta schedule of jobs, set the "Delta Period" to '0'. This ensures that new deltas are not generated in the data scheduler and data extractor.

5.1.7 Scope

This parameter takes values of resourceItems (SAP Ariba view) from the configuration file, as detailed in the "Resource Items Configuration" section. If all resource items should be included, the input for this field can be specified as "All". Alternatively, a list of specific resource items, separated by commas, must be provided.

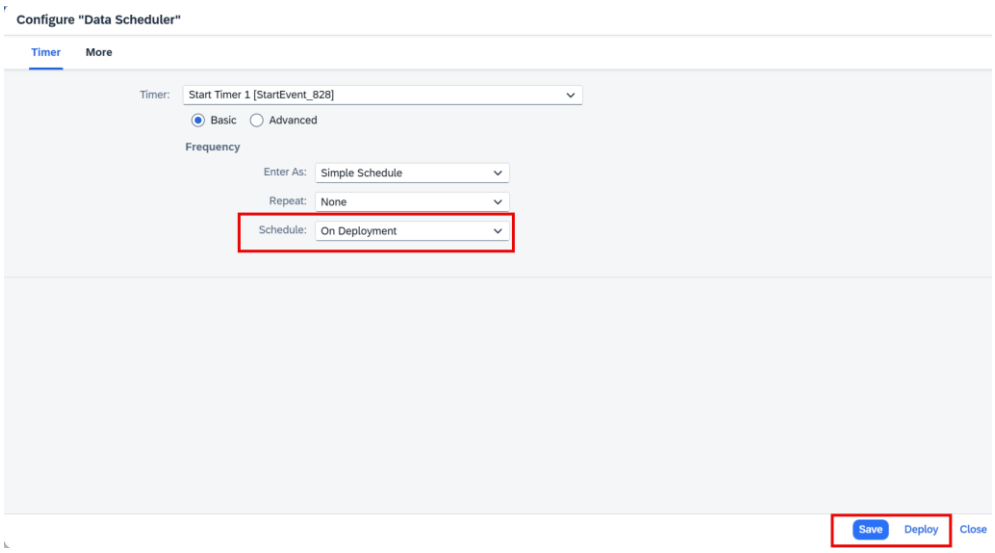
5.1.8 Retry Queue Id List

This input is used to retry failed queue items in the persistent queue. If specific failed queue items need to be re-executed, clear the **Mode** field and input the comma-separated queue items from the persistent queue.

Persistent queue details can be retrieved from the error notification email or by using the endpoint as explained in section 5.4.1.3. This action will reset the queue item status from "Failed" to "Scheduled", and the items will be processed by the extractor again.

5.2 How to Run

Once the above parameters are defined, save and deploy the iflow named "**Data Scheduler**".



This iFlow must not be scheduled frequently using a timer. It must only be scheduled with the "on deployment" setting, as this is an activity to be performed by the administrator.

5.3 When to Run

The Data Scheduler needs to be run manually in the following scenarios:

- **Initialization:** When SAP Ariba views are to be created or patched for specific or for all resource items.
- **Initial Mode:** To schedule jobs using the initial mode. Subsequent job queue items will be managed by the extractor.

- **First Delta Run:** To schedule the first delta run. Subsequent delta job items will be automatically generated by the Data Extractor.
- **Stop Delta:** To stop the delta schedule as per business decisions, as explained in the different Modes section above.

6 Data Extractor

The Data Extractor processes the queue created by the Data Scheduler and updates the persistent queue with status information. It also generates subsequent queue items required for reporting API processing.

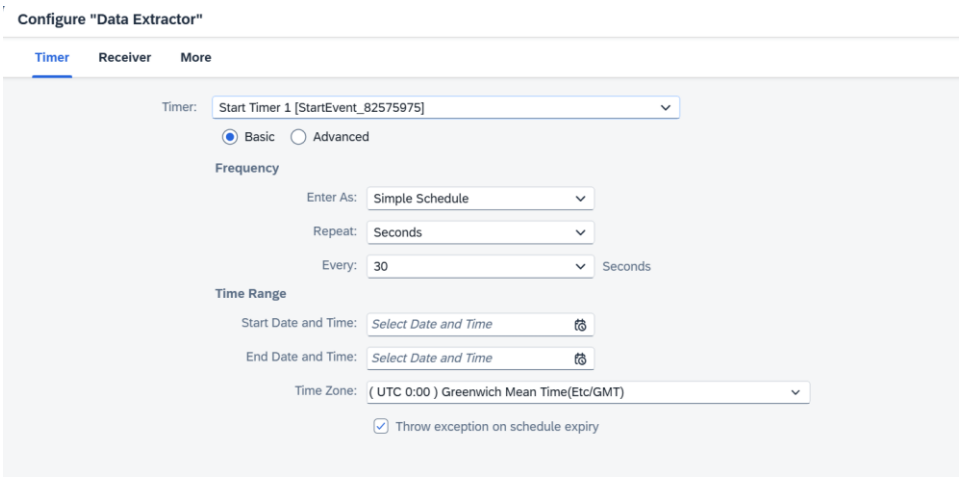
In case of any failure in the iFlow or if the queue status is updated to "Failed", an email notification is triggered to the recipients.

6.1 Input Screen

The Data Extractor comes with a predefined configuration. It is recommended to retain these default settings, except for the Receiver parameters, which should be updated to enable email notifications.

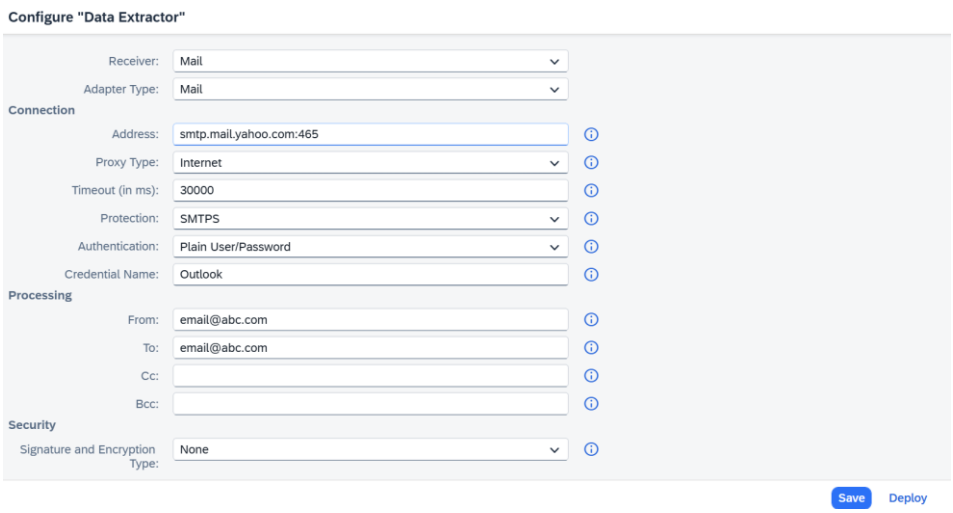
6.1.1 Timer

By default, the Data Extractor is set to run every 30 seconds. Do not change this setting as increasing the frequency might exhaust API rate limits/minute for Ariba. Frequency of the extractor can be reduced if it's acceptable to have delays in extraction, for example, delta loads.



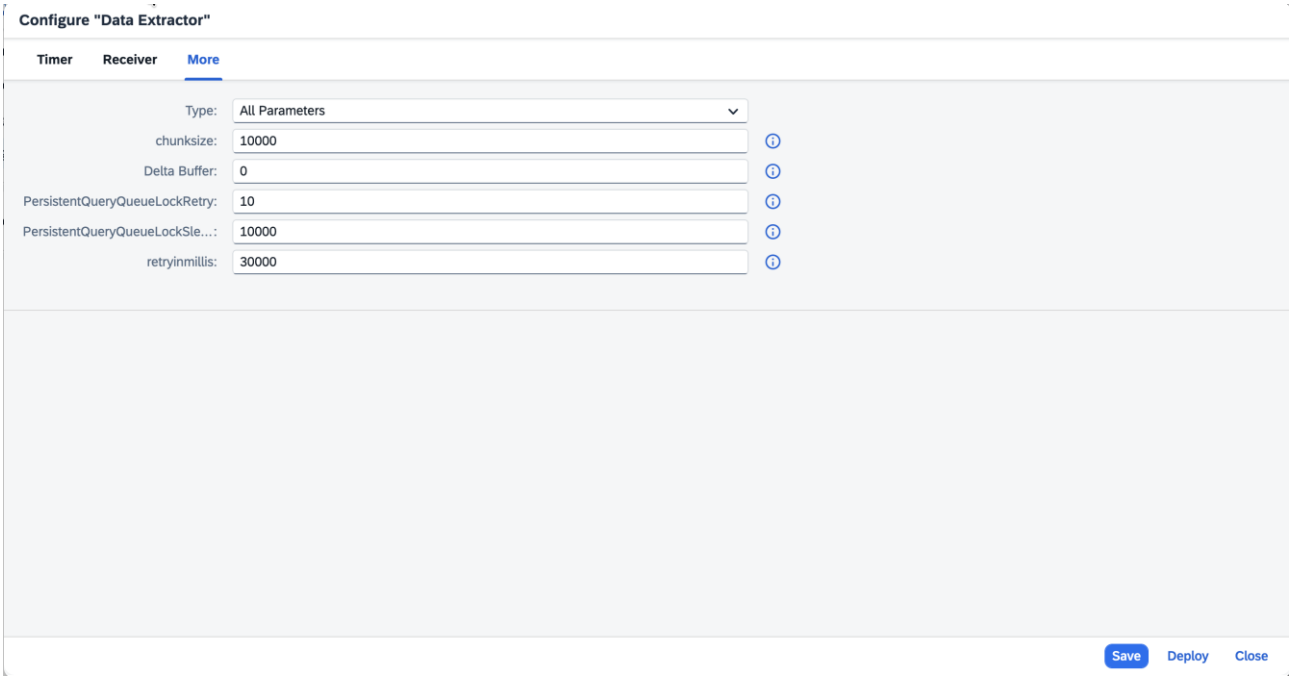
6.1.2 Receiver

These settings are intended for receiving failure email notifications within the iFlow. The desired configuration in the CI environment for setting up the email adapter must be completed as a prerequisite for the Data Extractor.



6.1.3 More

Under the **More** tab, the parameters PersistentQueryQueueLockRetry and PersistentQueryQueueLockSleep are preset like those on the data scheduler input screen. They do not need to be changed.



Retryinmillis: This parameter represents the retry time after which the persistent queue needs to be executed again. This occurs when a persistent queue item goes into the "Scheduled" status with a retry counter due to a timeout from an API call.

Chunk Size: This parameter represents the number of entries in a CSV file before pushing to SAP Signavio. If the number of entries exceeds 1000, the CSV will be split, and the SAP Signavio Ingestion API will be called in sequence. For better API performance, leave this value between 1000 and 1500.

Delta Buffer: This parameter specifies the buffer (in minutes) applied when scheduling the next delta run from the Data Extractor. It is used to avoid overlaps between the execution timestamp and the data extraction window.

In earlier versions, the next delta was scheduled based on the timestamp of the Ariba file extraction completion, which could be several minutes later than the initial delta job trigger time and could cause gradual shifts in execution. Logic has been rectified in Data extractor to schedule the next Delta with actual delta job execution which avoids the drift of delta execution times.

With the Delta Buffer parameter, the next delta execution can be controlled to add offset, if required. If an offset is not required and if it impacts ETL for Signavio, set the parameter to '0' or leave it blank.

- 1 → Adds a 1-minute offset (default, safe behavior)
- 0 → No offset; delta scheduling strictly follows the trigger time

This allows users to control whether a minimal safety offset is applied or whether delta execution should remain tightly aligned to the trigger time.

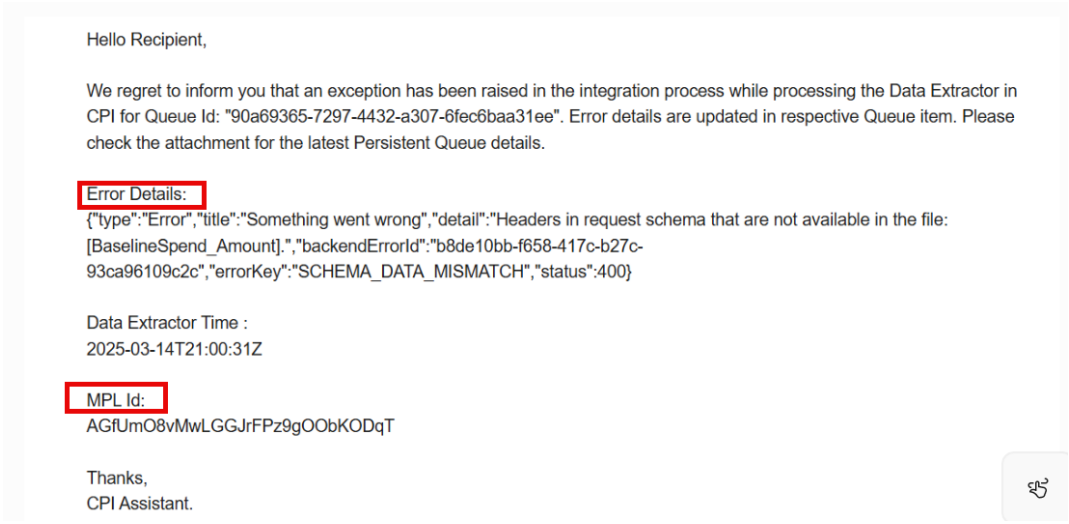
6.2 How to Run

Once the parameters described in section 6.1 are configured, deploy the iFlow.

6.3 Email body

Emails contain the following information:

- **Queue ID:** Unique identifier of the persistent queue item in the persistent queue data store (Query store).
- **Error Details:**
  - Error Code**:** This includes the error code from the iFlow. It can be a technical message or the response from the API, depending on the failure message.
- **MPL ID:**
  - iFlow ID: This can be used to search for the iFlow in the CI monitor messaging.
- **Data Extractor Time:**
  - The time at which the Data Extractor ran, based on the time zone set in the configuration file.
- **Attachment:**
  - A file is also attached for failure notification, containing the latest Persistent Queue data (Query Store content) at the time of the run. Users can download and check the queue ID and the failed entries.



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

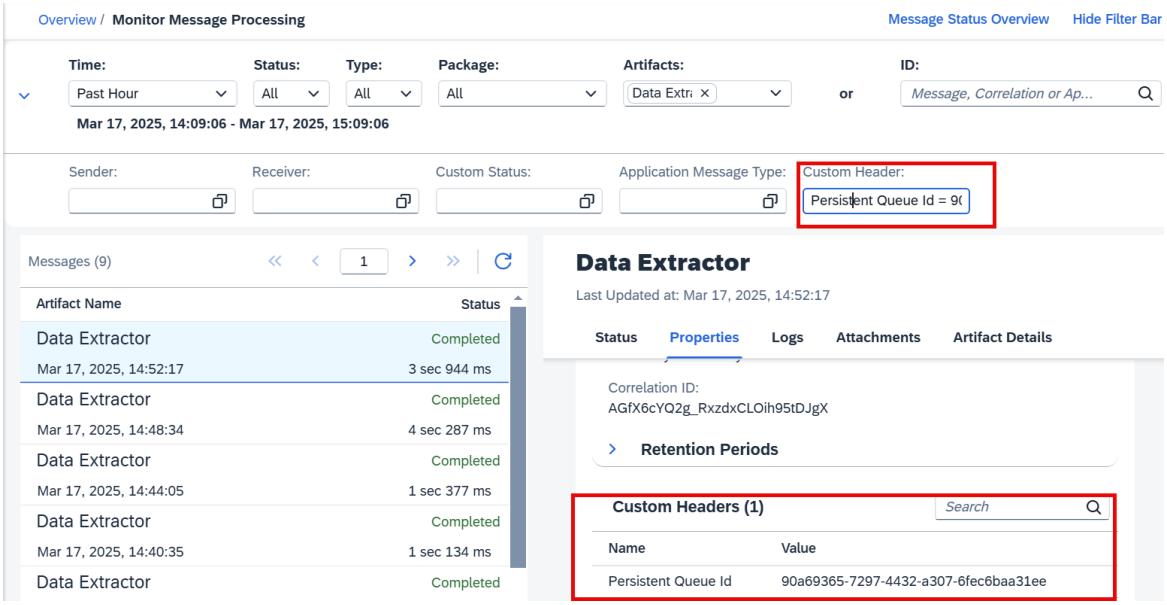
MPL Id:
AGfUmO8vMwLGGJrFPz9gOObKODqT

Thanks,
CPI Assistant.

7 Monitoring

If any iFlow fails, the recipients will receive an error email notification, as explained in section 6.3. A custom header property for the queue ID is created, allowing users to search using the queue ID or MPL ID to filter the iFlow that needs further analysis.

Additionally, standard HTTP response attachments are included in the iFlow. These attachments consist of responses from the SAP Ariba Reporting API, labeled as "Ariba log", and the SAP Signavio Ingestion API response, labeled as "Signavio log". These logs can be used to check the status of the API if required.



7.1.1 Additional CI Content for Monitoring

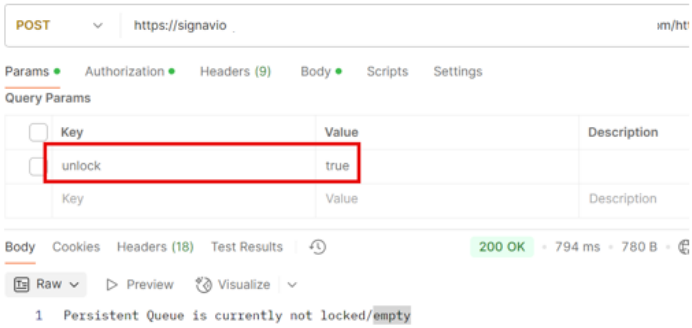
There are additional iFlows created specifically for debugging and monitoring purposes by the customer support team for issue analysis. These iFlows are accessible via HTTPS endpoints. OAuth credentials need to be set up in the subaccount tenant to access these endpoints. This setup ensures secure and authorized access for troubleshooting and monitoring.

7.1.1.1 Get PersistentQueryQueueLock

This iFlow endpoint is used to read the persistent queue lock ID and to unlock the PersistentQueryQueueLock global variable. To unlock the lock ID, the unlock parameter needs to be passed (true/false).

Note: Unlocking the queue should not be done as regular practice. This should be considered a last resort for the technical support team to unlock the queue when it is stuck in a locked state due to a technical failure of the "Data Extractor" that was not handled within the Data Extractor.

Proper caution should be exercised to avoid disrupting normal operations and to ensure the stability of the integration processes.



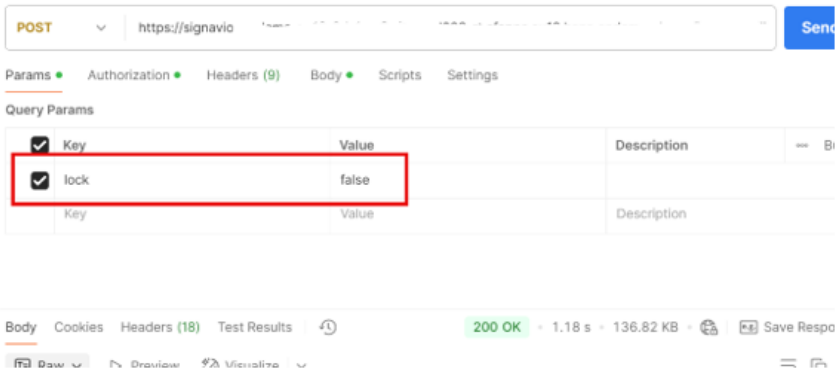
7.1.1.2 Read Resource Configuration file

This endpoint is used to read the content of the loaded configuration as described in section 4.6.

7.1.1.3 Read Persistent Queue Data Store

This endpoint needs to be used to read the content of the persistent queue data store (QueryStore). It can be used to monitor the processing of queues by the operations team.

When posting to this HTTPS endpoint, ensure that the lock parameter is not defined or is set to false in a productive scenario. If the lock parameter is set, it will lock the queue with the prefix EXT_{{iFlow id of this post call}}. If the queue is locked with an EXT_prefix, use the process described in section 6.1 to unlock the queue.



7.1.1.4 Persistent Queue Data Store Update

This endpoint is provided to update the status of any persistent queue IDs. It is intended for use by the operations team in the following scenarios, particularly when they monitor queues:

1. When any queue item is stuck in the 'inExecution' status due to exceptions or timeouts that were not caught.
2. To stop any "initial" or "range" queue items that are in the "scheduled" status due to valid business reasons.

Note: This endpoint must not be used to set up new queue items or for retrying failed queue items, as the Data Scheduler is responsible for those tasks.

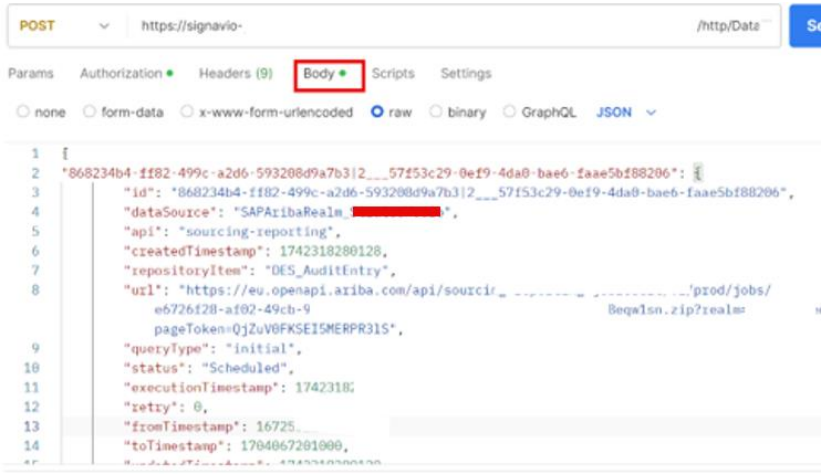
Process to Update:

1. Read the queue items as described in section 6.4.1.3.
2. Identify the queue items for which the status needs to be updated.
3. Prepare the body in the following format by enclosing the queue items in a 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*****f02-49cb-9a9d-
1e51e2d3c28b1742317560591/files/Fm8eqw1sn.zip?realm={RealmName}&pageToken=QjZuV0FKSEI5MERPR3lS",
      "queryType": "initial",
      "status": "Failed",  ---- cancelled or failed as per the business decision
      "executionTimestamp": 1742318280128,
      "retry": 0,
      "fromTimestamp": 1672531201000,
      "toTimestamp": 1704067201000,
      "updatedTimestamp": 1742318280128
     "failedReason" : "Cancelling 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",
```

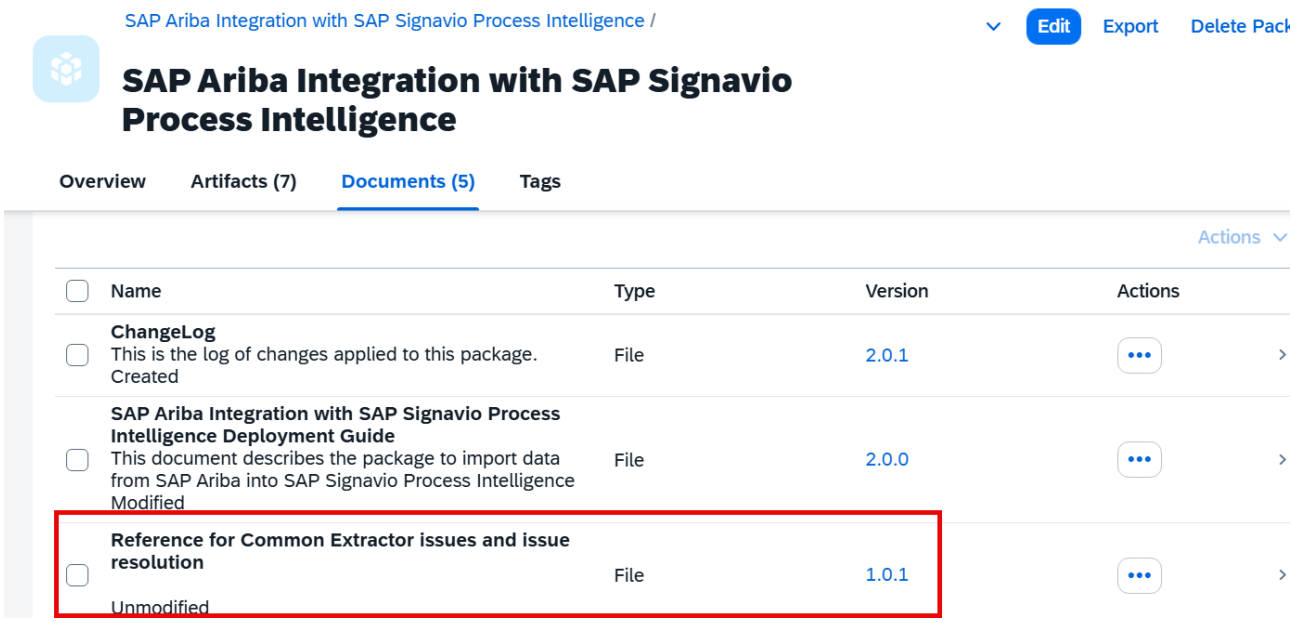
```
      "dataSource": "SAPAribaRealm_{RealmName}",
      "api": "sourcing-reporting",
      "createdTimestamp": 1742318280128,
      "repositoryItem": "DES_AuditEntry",
      "url": "https://eu.openapi.ariba.com/api/sourcing-reporting-
job/v1/prod/jobs?realm={RealmName}&pageToken=QjZuV0FKSEJ4TXdYbnlR",
      "queryType": "initial",
      "status": "Scheduled", ---- cancelled or failed as per the business decision
      "executionTimestamp": 1742318280128,
      "retry": 0,
      "fromTimestamp": 1672531201000,
      "toTimestamp": 1704067201000,
      "updatedTimestamp": 1742318280128
     “failedReason” : “Cancelling the job due to XXXXXXXX” ---- Optional
   }
   }
```


4. Execute the POST HTTPS endpoint using the prepared body content.



7.1.2 Troubleshooting

In case of errors during extraction, refer to the “Reference for Common Extractor Issues and Issue Resolution” guide provided in the package.



8 Steps for Package Upgrade from SAP Business Accelerator Hub

When a new package version is released in SAP Business Accelerator Hub to address bug fixes or enhancements, follow these steps to ensure the smooth operation of Data Extractor or Data Scheduler:

1. Review and adopt the latest configuration template:
    a. If there is an updated version of the configuration template, review the changes as described in section 4.
    b. Adopt the latest template if necessary.
2. Update data sources and data targets:
    a. In the new configuration template, update the Data Sources and Data Targets with the respective credentials.

    Note: Previous configuration files will remain backward compatible. However, it is recommended to use the latest template from SAP Business Accelerator Hub.

3. Undeploy existing iFlow (if applicable):
    a. If the older version of the Data Extractor iFlow is scheduled to run every 30 seconds, undeploy the iFlow.
4. Check and reset the PersistentQueryQueueLock:
    a. Verify if the “PersistentQueryQueueLock” is empty or locked with the iFlow ID.
    b. If it is not empty, delete the global variable to reset the lock via **Monitor** -> **Variables** in the CI tenant, or follow the steps in section 7.1.1.1 to clear the lock using monitoring content.
5. Import updated package content:
    a. Copy or adopt the updated package content into the design space of the CI tenant.

6. Deploy the updated configuration file:
    a. Redeploy the updated configuration file as described in section 4.6.
7. Schedule new queue items (optional):
    a. Schedule new queue items from the Data Scheduler of the latest package if required.
8. Deploy the latest version of data extractor:
    a. Deploy the latest version of the Data Extractor as described in section 6.2.