SAP BUSINESS TECHNOLOGY PLATFORM | EXTERNAL

Developers Guide
SAP Ariba Integration with SAP Build Process Automation

THE BEST RUN SAP

Table of Contents

Overview

This document provides information about Integration part of Live Process Package **Ariba Procurement Operations Visibility** available in **SAP API Business Hub**. The main audience of this document are SAP Integration Suite developers.

This document does not include the Configuration part of Integration models. Please refer to configuration guide on how to configure integration model **Extract and Transform Purchase Requisitions data from SAP Ariba** before the start.

Description of integration process

The integration model **Extract and Transform Purchase Requisitions data from SAP Ariba** is designed to transfer data changes in purchase requisitions data from SAP Ariba to process visibility.

The integration model allows for periodic (scheduled) loading of data changes in purchase requisitions data in SAP Ariba.

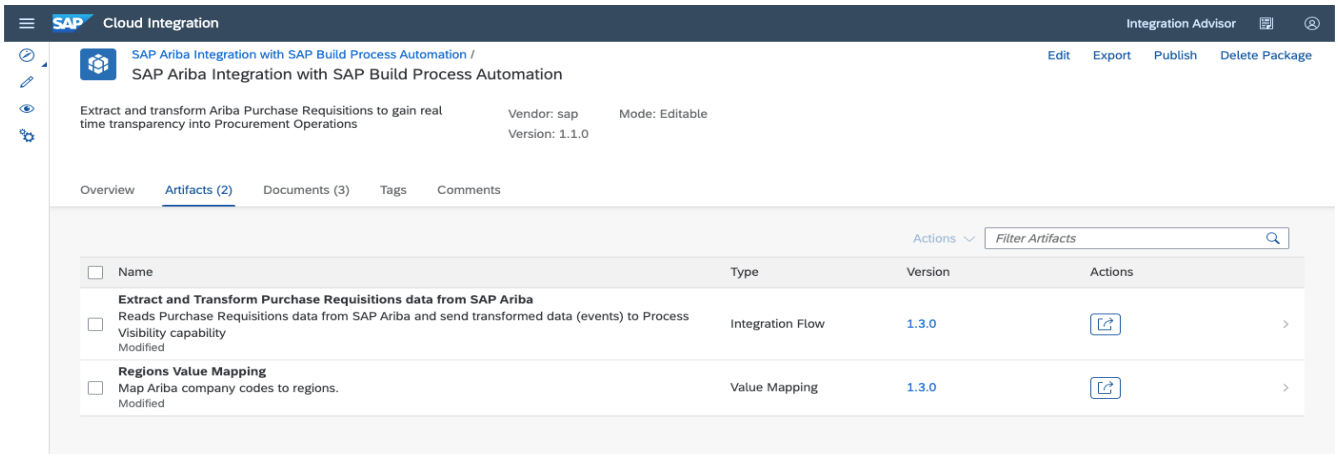
Work of the integration model can be described in the following steps:

1. **Loading purchase requisition data from SAP Ariba** for the past period since the last run of the integration model.
2. **Process Visibility Events data preparing for requisitions data updates** in SAP Ariba using Decisions. The list of Business is used to check the received data from SAP Ariba for certain changes in purchase requisitions. As a result of this step, a list of events will be generated. Decisions API is used for the rule call.
3. **Sending events to SAP Process Visibility.** List of events of SAP Ariba Purchase Requisition data updates to Process Visibility. Process Visibility API is used to send events.
4. **Visibility Events data preparing for all line items** of the requisition with “Ordered” Status. Decision is called for each line item of all records to check for the status "Ordered". If some line items status equals "Ordered", then “Process Completed” event is generated for this line item. Decisions API is used for the rule call.
5. **Sending events to SAP Process Visibility** about all Ordered line items of requisitions. Process Visibility API is used to send events.

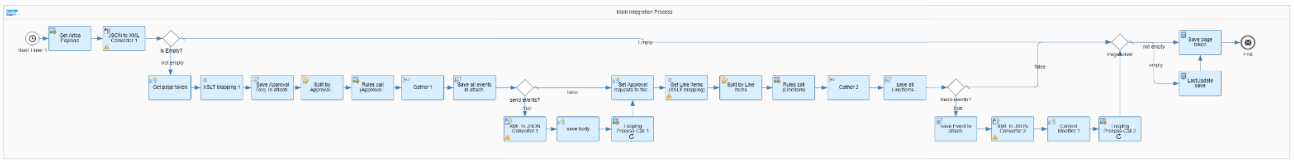
Main Process Extract and Transform Purchase Requisitions data from SAP Ariba.

Each of the listed steps is presented in the diagram of the integration module as a sub-process (Local Integration Process). The detailed information about all steps of sub-processes described below.

Integration model **Extract and Transform Purchase Requisitions data from SAP Ariba** should be imported from the standard package **SAP Ariba Integration with SAP Build Process Automation**.



The main process of the integration model is launched on a schedule (for example, once every 5 minutes). Setting up the launch schedule for the integration model is available in the configuration (see Configuration guide).



Main process steps description

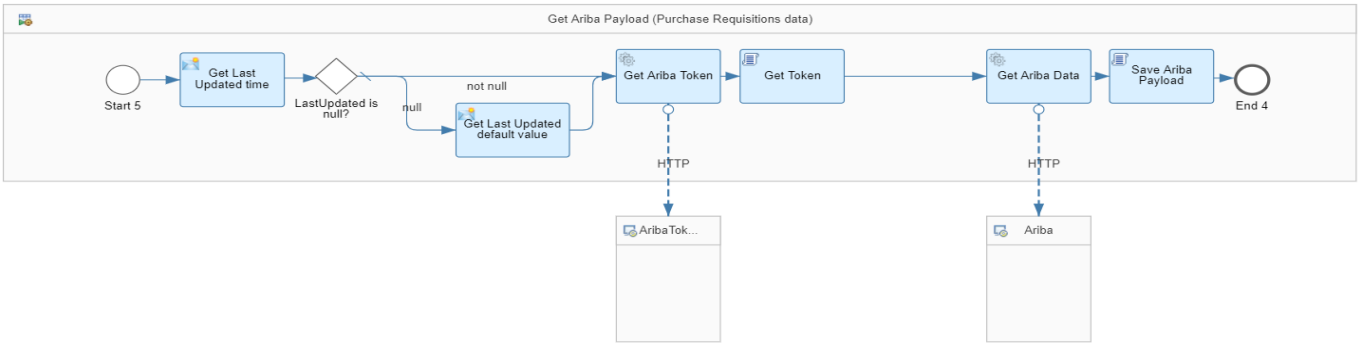
No	Step Name	Step type	Description
1	Get Ariba Payload	Sub-process	Sub-process “Get Purchase Requisitions Data from SAP Ariba” called to retrieve updates of purchase requisitions data from SAP Ariba. (Get Purchase Requisitions Data from SAP Ariba)
2	JSON to XML Converter 1	JSON to XML Converter	Converting the received data from Ariba (JSON) to XML format for further processing in step 5.
3	Routing step “Is Empty Response”	Router	Check for any updates of SAP Ariba side at specified time period. If data is received, then data processing started, integration flow goes to step 4, else the integration flow goes to step 22.

4	Get page token	Groovy script	Saving the PageToken property. The parameter is used for page-by-page data loading in case the limit of data loaded for one SAP Ariba API call is exceeded. Maximum number of records that can be received from SAP Ariba in one API call is 40. If the limit on the number of records is exceeded, then the pagetoken value is stored in a INTEGRATION SUITE Global variable and will be used at the next call of the integration interface. The names of INTEGRATION SUITE Global Variables can be configured before start of integration model (see Configuration guide).
5	Get Approval requests (XSLT mapp.)	XSLT Mapping	The XSLT mapping step transforms the received data into an XML structure of the following format: //ApporvalPayload/ApprovalRequests/LineItems Transformation is used to be able to use the Iterating split step in integration model (steps 7, 15).
6	Save Approval Req. in attach.	Groovy script	Saving SAP Ariba Purchase requisition data in an attachment.
7	Split by Approval Request	Iterating split	Processing ApprovalRequests data from SAP Ariba. A sub-process of Rule calls is started for each ApprovalRequest element (step 8).
8	Rules call (Approval Requests)	Sub-process	Sub-process called to get a list of SAP Visibility Events based on the data in the Approval request element using the Decisions.
9	Gather 1	Gather	Concatenating of the received event lists for each Approval Request element from Ariba into one common list with Visibility events data.
10	Save all events in attach.	Groovy script	Saving a general list of SAP Visibility events in a message attachment.
11	Routing step “Event List is Empty?”	Router	Checking for Events in the list. If the Event list is not empty then integration flow goes to the step №12, else integration flow goes to the Routing step “PageToken” (step 22).
12	Sending Events	Sub-process	Sub-process “Get Visibility Events for all Records” called for preparing and sending events to Process Visibility (Get Visibility Events for all Records).
13	Set Approval requests to the body	Content Modifier	Save value of property ApprovalRequests to the message body
14	Get Line Items (XSLT mapp.)	XSLT Mapping	The XSLT mapping step transforms the received data into an XML structure of the following format: // LineItems Result structure contains all LineItems for all Ariba Records.
15	Split by Line Items	Iterating split	Processing Line Items data from SAP Ariba (for all records). A sub-process of Rule calls is started for each LineItem element (step 16).
16	Rules call (LineItem)	Sub-process	Sub-process “Get Visibility Events for all Line Items” called to get a list of Events based on the data in the Line-Item element (Get Visibility Events for all Line Items).
17	Gather 2	Gather	Concatenating of the received event lists for each Line Item element from Ariba into one common list.
18	save all LineItems Events	Groovy script	Save all Events for Line Items elements to message attach.
19	Routing step “Send events?”	Router	Checking the events list size. If the Event list size more than 0 then integration flow goes to the step №20, else integration flow goes to the

			Routing step “PageToken” (22).
20	Save Event to attach.	Groovy script	Save event list to attach.
21	Sending Events	Sub-process	Sub process call for sending LineItems events to Process Visibility.
22	Routing step “PageToken”	Router	Checking value of PageToken message property (saved at step 4). The parameter is used for page-by-page data loading in case the limit of data loaded for one SAP Ariba API call is exceeded. If message property “PageToken” is empty (there are all data for current time period was received from Ariba), then iflow goes to the “Last Update timestamp save” step (23). Else iflow goes to the “Save Ariba page token” step (24).
23	Last Update TimeStamp save	Content Modifier	Save timestamp of Last run of integration model to the INTEGRATION SUITE Global Variable. This step called only for case when the PageToken is empty. The names of INTEGRATION SUITE Global Variables can be configured before start of integration model (see Configuration guide).
24	Save Ariba page token	Content Modifier	Save PageToken value to the INTEGRATION SUITE Global Variable. The names of INTEGRATION SUITE Global Variables can be configured before start of integration model (see Configuration guide).

Sub-process Get Purchase Requisitions Data from SAP Ariba

Sub-process **Get Purchase Requisitions Data from SAP Ariba** used for loading purchase requisition data from SAP Ariba for the past period since the last run of the integration model. SAP Ariba Open API used for service calls.



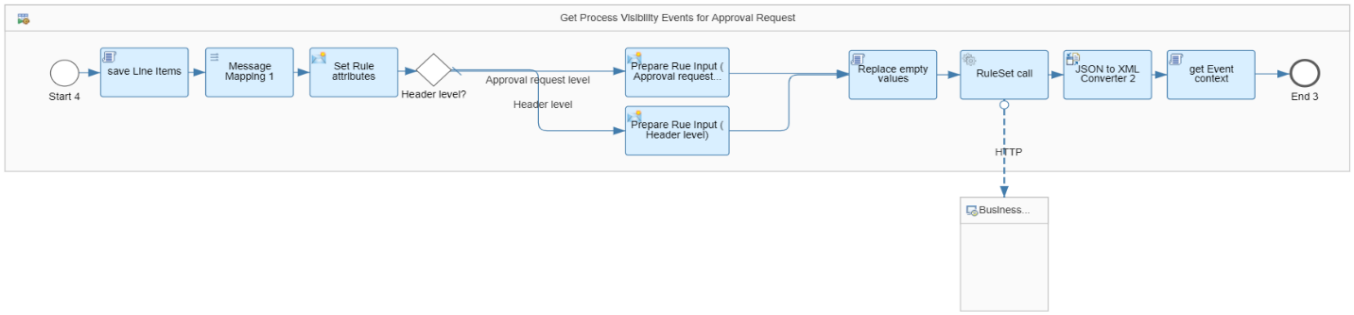
Sub-process steps description.

No	Step Name	Step type	Description

1	Get DateTime properties	Content Modifier	Creating of the following properties: - page_token (Global Variable) - LastUpdated (Global Variable) - CurrentDateTime (expression) LastUpdated and CurrentDateTime properties are used as a filter values in query of service for get the Ariba data (step 6).
2	Routing step “LastUpdated is null?”	Router	If global variable “AribaLastSucessfullyUpdate_<RealmName>” not exist or empty, then integration flow goes to the step 3, else integration flow goes to the step 4.
3	Get Last Updated default value	Content Modifier	Default value for LastUpdated property can be configured before the start of integration model (see Configuration guide).
4	Get Ariba Token	Request Reply	External call step for getting SAP Ariba Auth. Token using the service: https://<Ariba_runtime_url>/v2/oauth/token Configuration parameters “Address” and “Credential name” should be configured for this step before (see Configuration guide). Access_token element expected in response body (JSON format).
5	Get Token	Groovy script	Script for save the Ariba token from response body to the Message Header in the following format: - Name: “Authorization” - Value: “Bearer <Ariba token>”.
6	Get Ariba Data	Request Reply	External call step for getting SAP Ariba Data using the service: https://<Ariba_runtime_url>/<resource> Configuration parameters “Address” and “Credential name” should be configured for this step before (see Configuration guide).
7	Save Payload to attach.	Groovy script	Saving SAP Ariba payload to Message Attachments. The response body is saved in an optimized version (not original Ariba response format). Response body contains only the elements of Ariba data used in the integration model.

Sub-process Get Process Visibility Events for Approval Request

Sub-process **Get Process Visibility Events for Approval Request** used to prepare of visibility events data using the Decisions. The event list is generated by processing SAP Ariba data (Approval requests) using the Decisions Services (see Configuration guide). SAP Decisions API used for Decisions service calls.

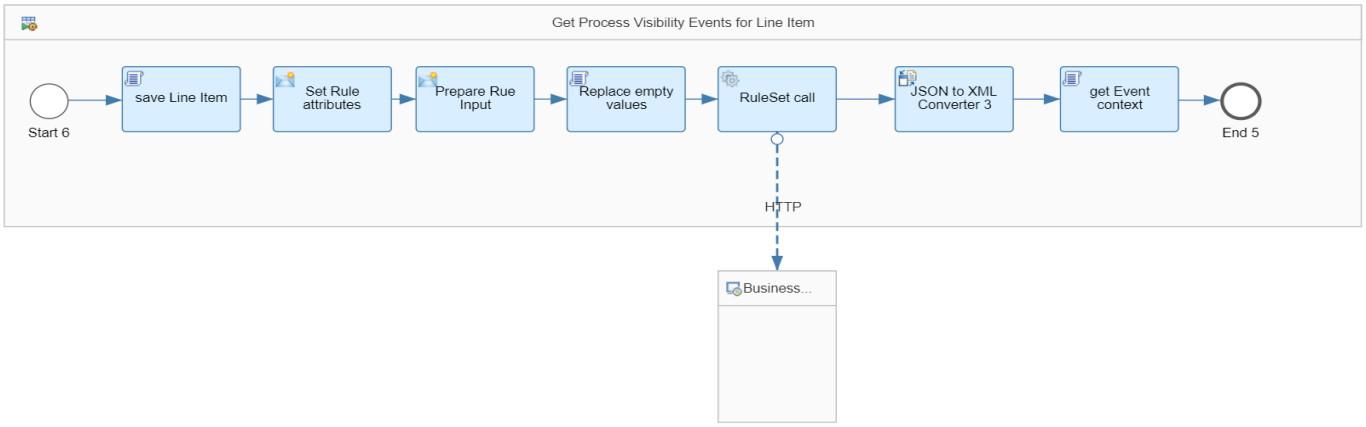


Sub-process steps description.

No	Step Name	Step type	Description
1	save Line Items	Content Modifier	Saving message body to the message property “lineItems”. The property will be used at the “get Event context” (step 10)
2	Message Mapping 1	Message mapping	Transformation message for reducing the size of message body
3	Set Rule attributes	Content Modifier	Creating of message properties from message body (Approval request data): - Approver - CreatedDate - ApprovalRequestFlag - PRApprovalDate - ApprovedDate - AssignedDate - RuleName - ReceivedDate - AreOrdersStuck - StatusString - OrderedDate - ActivationDate - ApprovalRequired - ApprovedBy - EscalationDate - AssignedTo - SubmitDate - LastModified - UniqueName - TimeUpdated The properties will be used for preparation of request body for the rule calls at the steps 5/6.
4	Router step “Header level?”	Router	If ApprovalRequestFlag equals “RequestHeader” then integration flow set to step 5, else - step 6.
5	Prepare Rue Input (Header level)	Content Modifier	Preparation of message body (Header level) for rule call “AribaPRHeaderEventTransformation” (step 8).
6	Prepare Rue Input (Approval request level)	Content Modifier	Preparation of message body (Approval request level) for rule call “AribaEventTransformation” (step 8).
7	Replace empty values	Groovy script	Preparation of a message body to the correct format for the Rule Service call (empty values should be set as null).
8	RuleSet call	Request Reply	Call Decisions using the Decisions API service: https://< rule_hostname>/rule-path/rest/v2/rule-services Request body was prepared on steps 5/6. Response body contains generated event list for Approval request. Configuration parameters “Rules API Address” and “Rules Credentials” should be configured for this step before (see Configuration guide).
9	JSON to XML Converter 2	JSON to XML Converter	Converting the received data from Decisions Management (JSON) to XML format for further processing in step 10.
10	get Event context	Groovy script	Preparation of Event list for sending to the SAP Process Visibility (sub-flow Sending events to Process Visibility). Received events data of processing Approval request (step 8) using for each line item of the Approval Request: Each of event from events list data for Approval request (step 8) set to the each of line item of the current Approval request in the new events list for Process Visibility.

Sub-process Get Process Visibility Events for Line Item

Sub-process **Get Process Visibility Events for Line Item** used to prepare of process visibility events data using the Decisions. The event list is generated by processing SAP Ariba data (Line Items) using the Decisions Services (see Configuration guide).
Decisions are called for each line item of all records to check for the status "Ordered". If Line-Item Status equals "Ordered", then “Process Completed” event is generated for this line item.
SAP Decisions API used for Decisions service calls.



Sub-process steps description.

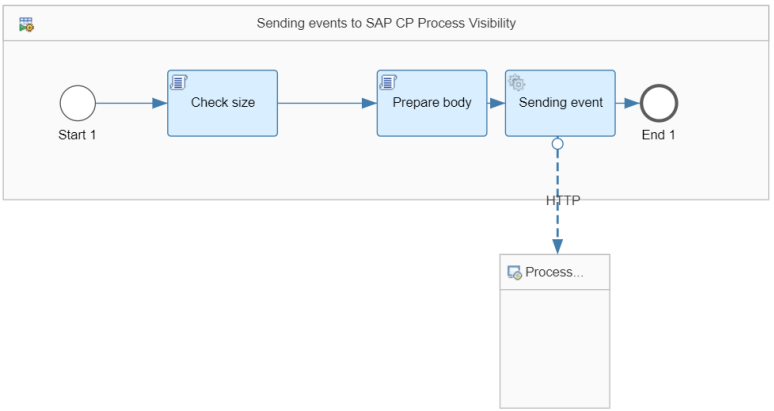
No	Step Name	Step type	Description
1	save Line Item	Content Modifier	Saving message body to the message property “lineItemCurrent”. The property will be used at the “get Event context” (step 7)
2	Set Rule attributes	Content Modifier	Creating of message properties from message body (Approval request data): - LastModified - SupplierId - RequisitionId - LineItemId - OrderedDate - StatusString - LastUpdated The properties will be used for preparation of request body for the rule calls at the step 3.
3	Prepare Rue Input	Content Modifier	Preparation of message body for rule service call “AribaPRLineItemEventTransformation” (step 5).
4	Replace empty values	Groovy script	Preparation of a message body to the correct format for the Rule Service call (empty values should be set as null).
5	RuleSet call	Request Reply	Call Decisionsservice using the Decisions API: https://<ruleAPIhostname>/rule-path/rest/v2/rule-services Request body was prepared on steps 3/4. Response body contains generated event list for Line Item. Configuration parameters “Rules API Address” and “Rules Credentials” should be configured for this step before (see Configuration guide).

6	JSON to XML Converter 3	JSON to XML Converter	Converting the received data from Decisions Management (JSON) to XML format for further processing in step 7.
7	get Event context	Groovy script	Preparation of Event list for sending to the SAP Process Visibility (sub-flow Sending events to Process Visibility). Received events data of the processing Line Item (step 6) set into the new Events list for Process Visibility in the required format.

Sub-process Sending events to Process Visibility

Sub-process **Sending events to Process Visibility** used for sending events data (each of event define Ariba Purchase requisition new updates) to the Process Visibility using the process visibility API.

Sub-process used at the 'Sending Events' Loop step of Main process. The stop condition for a loop step is the value of the message property - 'generalEventListEmpty'. Value of message property calculated every iteration of loop step.



Sub-process steps description.

No	Step Name	Step type	Description
1	Check size	Groovy script	Checking the size of message body with all not sent Process Visibility Events. If the message body size less or equals payload size limit then all Events are saved to message body for Process Visibility Service call (step 3), else part of events are selected that satisfy the condition (events size less or equals size limit) and set to the message body for Process Visibility Service call (step 3). If message body contains all generated events then "generalEventListEmpty" message property equals '-' (stop condition of loop processing) else "generalEventListEmpty" equals 'X'.
2	Prepare body	Groovy script	Prepare a message body to the correct format for Process Visibility service call (empty values should be set as null).

3	Sending event	Request Reply	Call Process Visibility service using the Process Visibility API: https://<process_visibility_hostname>/visibility-path/rest/v1/data-acquisition/data Request body was prepared on steps 1-2. Configuration parameters “Process Visibility API Address” and “Visibility Credentials” should be configured for this step before (see Configuration guide).