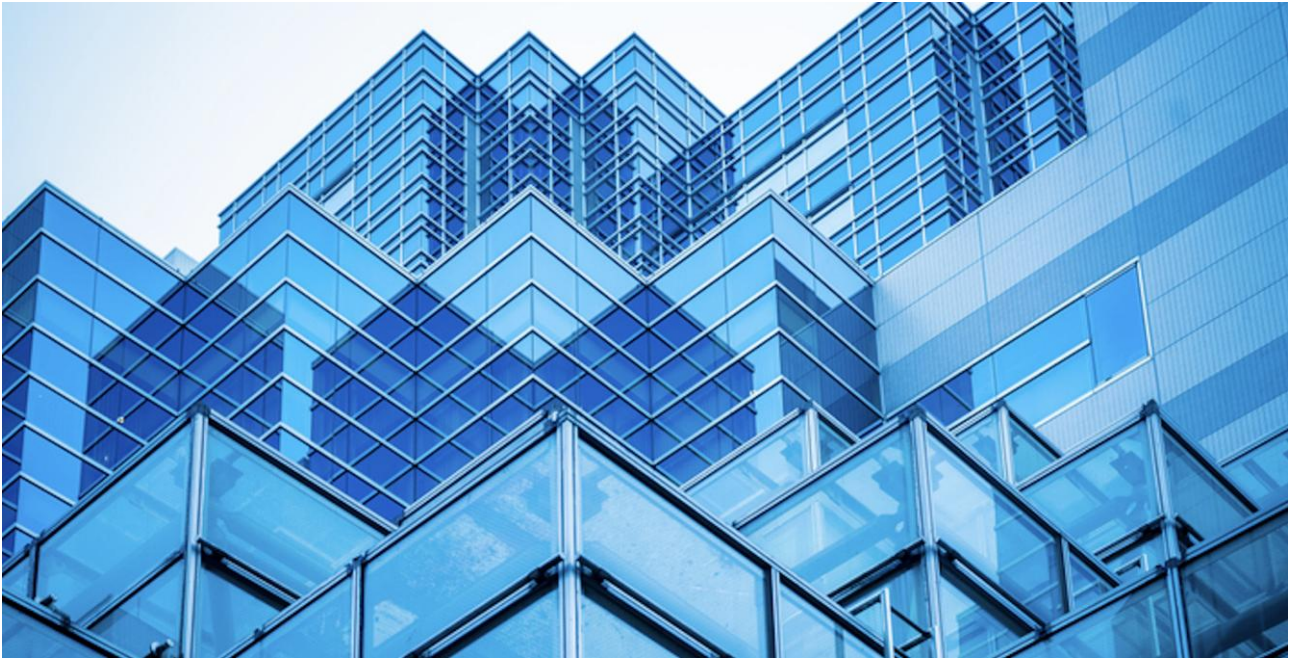




PUBLIC

SAP Build Process Automation

Intelligent Processing of Invoices without Purchase Orders Configuration and User Guide

Contents

Change Log	3
Overview	4
Prerequisites	6
Configuration Guide	6
Setup Business Technology Platform Cockpit	6
Configure SAP Build Process Automation Destination	6
Configure SAP S/4HANA Destination	7
Configure SAP Integration Suite Destination	7
Configure SAP Build Process Automation Email Destination	7
User Roles Configuration	7
Cloud Connector Configuration	8
Setup Content Package	9
Import SAP Build Process Automation content	9
Import and deploy TASK UI application	9
Configuration & Modification	9
Configure Decisions	9
Configure Email	12
Create Process	13
Create Forms	13
Create Decision	13
Create Process Visibility Scenario	13
Release and Deploy the Process	13
Configure and Deploy Task UI	14
Configure and Deploy Start UI	15
Configure SAP Work Zone	17
User Guides	17
Capabilities	17
Actions	19
SAP UI5 SCREEN	20
Appendix	22
Support	26

Change Log

Version	Date	Description
1.0.0	May 21, 2025	Document update: Appendix updated

This document contains 3 sections. Each section covers different information about this template content package.

1. **Overview:** In this section, you will get quick overview of the use case, what are high-level components used and how the template works in nutshell. Prerequisite section provides information of different services of SAP Business Technology Platform that are required to use this template.
2. **Configuration Guide** It contain sections which will guide you to setup your (a) SAP Business Technology Platform Account with destination, cloud connector, user roles etc., (b) Integration Content via SAP Integration Suite, (c) SAPUI5 application via SAP Business Application Studio and (d) Import and configure the template using SAP Build Process Automation design studio and (e) SAP Work Zone to access the My Inbox or Task Center and other applications.
3. **User Guide:** This section provides details about different artefacts that are used in this template like process definition, decision diagram, action project details, forms, email notifications etc. to better understand how different capabilities are used in this template.

Note:

- Learning contents are published only for demo and reference purposes. We do not provide any support to learning / sample content.
- This documentation is not a detailed guide to setup SAP Business Technology Platform services. It assumes that IT admin who is setting the content is skilled with SAP Business Technology Platform environment. **Configuration section must be followed.**
- This template content is to accelerate your solution development. You must modify this template according to your requirements to achieve the desired business goal. To use this template content, you need to have basic knowledge and understanding of SAP Build Process Automation, SAP Integration Suite, and its capabilities.

Overview

This document provides technical information about what needs to be configured to make the package – Intelligent Processing of Invoices without Purchase Orders to run. The main audience of this document are technical personas like IT Administrators or Developers.

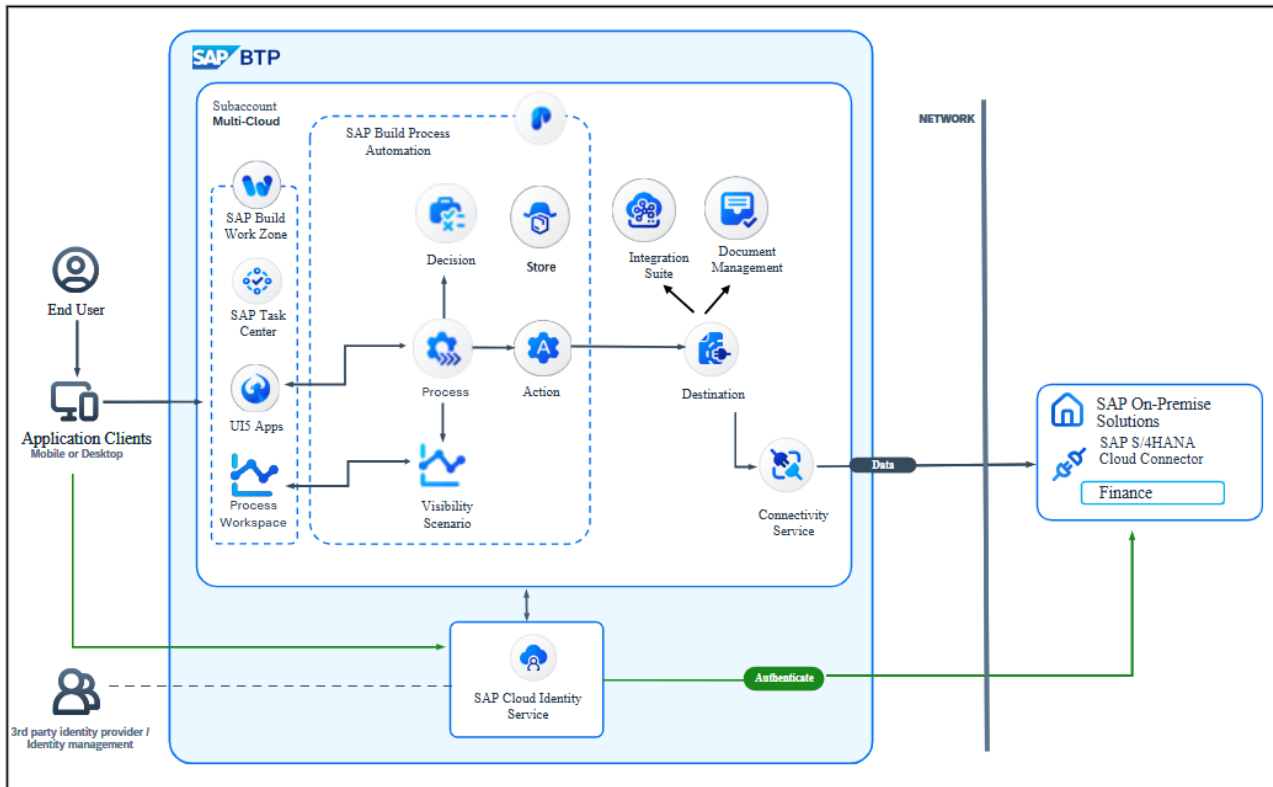
This content package enables SAP S/4HANA customers to automate approvals for invoices without purchase orders using SAP Build Process Automation without additional development in SAP S/4HANA. Intelligent Processing of Invoices without Purchase Orders application helps customer to post Invoice data into SAP S/4HANA, and check and display an existing invoice. For creation and change of the invoice, customers can enable approval process before updating in SAP S/4HANA.

This content package can be used with both SAP S/4HANA Cloud Private Edition (2023 onwards) and SAP S/4HANA On-Premise.

Features:

- Validation of Non-PO Vendor Invoice attributes through decisions.
- Automatic email notification to parties involved.
- Duplicate invoice detection to avoid redundant processing.
- Out-of-the-box visibility into key process performance indicator

Process Flow Diagram



Solution flow :

1. The Accounts Payable Executive initiates a Non-PO Vendor Invoice request by submitting details, including Company Code, Vendor Number, Invoice Date, Amount, and attachments or the initiator can upload a pdf invoice copy from which invoice details we will be auto populated in the Start UI fields through the SAP UI5 application.
2. The system validates the invoice request using decision to check for mandatory fields, such as Vendor Number, Reference, and Amount, as well as compliance with policies for attributes like Cost Center, GL Account, and Currency. If validation fails, the requester is prompted to correct the errors.
3. Upon successful validation, the request is routed to the approver, and an email notification is sent with a task assigned in SAP My Inbox.
4. Approvers evaluate the request and take one of the following actions:
 - a. Approve: Advances the request to the next approver or triggers final posting in SAP S/4HANA
 - b. Decline: Sends the request back to the requester for updates, including approver comments.
5. If the request is sent for rework, the requester updates the necessary details, such as line items or tax codes, and resubmits the request. The approval process restarts, following the configured process steps.
6. After all approval levels are completed, the system posts the Non-PO Vendor Invoice in SAP S/4HANA, and the status is updated to "Approved."
7. The requester and relevant stakeholders receive notifications about the final status and details of the posted invoice.

Note:

- The process can be triggered via a Start UI.

Required SAP Business Technology Platform Services

The following SAP Business Technology Platform services are required to consume the **Intelligent Processing of Invoices without Purchase Orders**:

- SAP Build Process Automation
- SAP Connectivity service
- SAP Business Application Studio
- SAP Application Runtime Service
- SAP Document Management Service
- SAP Integration Suite
- SAP Work Zone, standard or advanced
- SAP Cloud Identity Services - Identity Authentication (optional)
- SAP Business Technology Platform, Cloud Foundry runtime.

Prerequisites

For this template to run successfully the following prerequisites are needed

- Get access to BTP account.
- Need valid SAP S/4HANA user.
- This template requires Integration package 'SAP Build Process Automation Integration with SAP ERP for Invoice' to run. You can download this package separately from Integration Suite tenant. To understand more, refer [help document](#). The guide to set-up and configure integration package is mentioned along with the integration package itself.

Configuration Guide

Intelligent Processing of Invoices without Purchase Orders template requires SAP Build Process Automation subscription and Destination Setup for SAP Build Process Automation. Follow the [setup and configuration section](#).

Setup Business Technology Platform Cockpit

Configure SAP Build Process Automation Destination

To access the SAP Build process Automation, you can configure the destination. Please create destination accordingly

Destination Property	Value
Name	sap_process_automation_service
Type	HTTP
URL	<"endpoints"."api">
Client ID	<"uaa"."clientid">
Client Secret	<"uaa"."clientsecret">
Token Service URL	<"uaa"."url">/oauth/token
Proxy Type Internet	Internet
Authentication	OAuth2ClientCredentials

Add the property **sap.processautomation.enabled** as **true** to view the destination in the SAP Build Process Automation tenant. Please refer [help document](#) for more details.

Configure SAP S/4HANA Destination

This content package can be used with S/4HANA On-Premise. Destination can be created accordingly.

Destination Property	Value
Name	<Name>
Type	HTTP
URL	<SAP S/4HANA endpoint> Example: https://<s4system-name>-api.xxxxxxxx
Proxy Type Internet	Internet
Authentication	Basic Authentication
Username	<Username> of the SAP S/4HANA Communication User
Password	<Password> of the SAP S/4HANA Communication User

Configure SAP Integration Suite Destination

Configure Destination to connect to Integration Suite destination cloud system.

Destination Property	Value
Name	IntegrationSuite
Type	HTTP
URL	<Integration Suite Destination endpoint> https://<Integration Suite>.hana.ondemand.com
Proxy Type Internet	Internet
Authentication	Basic Authentication
Username	<Username> of the SAP S/4HANA Communication User
Password	<Password> of the SAP S/4HANA Communication User

Configure SAP Build Process Automation Email Destination

Mail notifications are used in this template to send notifications to requestor, approver, and admin during different stages of the process. Follow the instructions to setup mail destination in your SAP Business Technology Platform Account.

Configuring destination, [Configure Mail Destination](#)

- For more information refer to Create [HTTP Destinations](#) and [OAuth User Token Exchange Authentication](#).

User Roles Configuration

To access this template, the business user will need these roles:

Role Collection Name	Purpose
TaskCenterAdmin	Permission to execute calls to the connector status API and monitor configured destinations and running background jobs.
Process Automation Developer	Standard Business User, can model and publish processes
ProcessAutomationParticipant	To execute (like accessing MyInbox, running the automations, etc.) this content package, you need to have this role assigned to the respective user.
ProcessAutomationAdmin	To perform process admin activities like accessing monitoring applications etc., you need to have this role assigned to the respective user.
Business_Application_Studio_Administrator	Allows administrators to manage (export and delete) user data.
Business_Application_Studio_Developer	Allows developers to load and develop applications using SAP Business Application Studio.

For more information about SAP Build Process Automation roles, see [Authorizations](#).

Cloud Connector Configuration

For SAP S/4HANA on-premise landscape, configure cloud connector to enable secure tunnel to SAP Business Technology Platform tenant. Please refer the help documentation to [configure Cloud Connector](#).

Services/Resources that need to be exposed from SAP S/4HANA on-premise using Cloud Connector.

Resources	Protocol	Backend-Type
BAPI_GL_ACC_GETLIST	BAPI/RFC	SAP S/4HANA
BAPI_COMPANYCODE_GETLIST	BAPI/RFC	SAP S/4HANA
MMPUR_UI_GET_SUPPLIERS	BAPI/RFC	SAP S/4HANA
BAPI_BUSINESSAREA_GETLIST	BAPI/RFC	SAP S/4HANA
BAPI_PROFITCENTER_GETLIST	BAPI/RFC	SAP S/4HANA
BAPI_COSTCENTER_GETLIST	BAPI/RFC	SAP S/4HANA
/SAPNEA/SMAPI_PLANT_GETLIST	BAPI/RFC	SAP S/4HANA
RFC_READ_TABLE	BAPI/RFC	SAP S/4HANA
BPAR_C_FI_TERM_OF_PAYMENT_KEY	BAPI/RFC	SAP S/4HANA
BAPI_CONTROLLINGAREA_FIND	BAPI/RFC	SAP S/4HANA
FI_TAX_SERVICES_CALCULATE	BAPI/RFC	SAP S/4HANA
BAPI_ACC_DOCUMENT_POST	BAPI/RFC	SAP S/4HANA
BAPI_ACC_DOCUMENT_CHECK	BAPI/RFC	SAP S/4HANA
BAPI_VENDOR_GETDETAIL	BAPI/RFC	SAP S/4HANA
BAPI_COMPANYCODE_GETDETAIL	BAPI/RFC	SAP S/4HANA
BAPI_USER_GETLIST	BAPI/RFC	SAP S/4HANA
BAPI_EXCHRATE_GETCURRENTRATES	BAPI/RFC	SAP S/4HANA
/sap/opu/odata/sap/ZAPI_ACC_DOC_POST_SRV/Acc_DocSet	OData	SAP S/4HANA

Setup Content Package

Import SAP Build Process Automation content

Standard automation packages released by SAP will be available at the store. To know more about the store visit [SAP Build Process Automation Store](#).

This template can be downloaded from the Store:

1. Navigate to the Store in your SAP Build Process Automation account.
2. Select the Project Type filter as "Business Process."
3. In the Search bar type "**Intelligent Processing of Invoices without Purchase Orders**"
4. Select the "Create from Template" button to add the template into your lobby.
5. After successfully adding the template, navigate back to the lobby to find the "**Intelligent Processing of Invoices without Purchase Orders**" template available for use.

Caution: Be sure to select the **Project Type** as "**Business Process**" in the filters.

Import and deploy TASK UI application

The content package requires the SAP UI5 user interface which is available in Intelligent Processing of Invoices without Purchase Orders Process content package. Below are details of Task UI application

Task UI Application: NonPOVendorInvoice-ui_withAI

Following steps can be followed to utilize the SAP UI5 application provided with this process package:

- Download the SAPUI5 application Task UI from the imported store project.
- Open Business Application Studio and import the previously downloaded SAPUI5 project.
- Build SAPUI5 project using mta.yaml file and the generated project (generated mtar file will be available in the mta_archives folder) file can be deployed.

Configuration & Modification

Once you have added the content to your lobby, you will find the project in your project list. Adapt the content template based on your requirements.

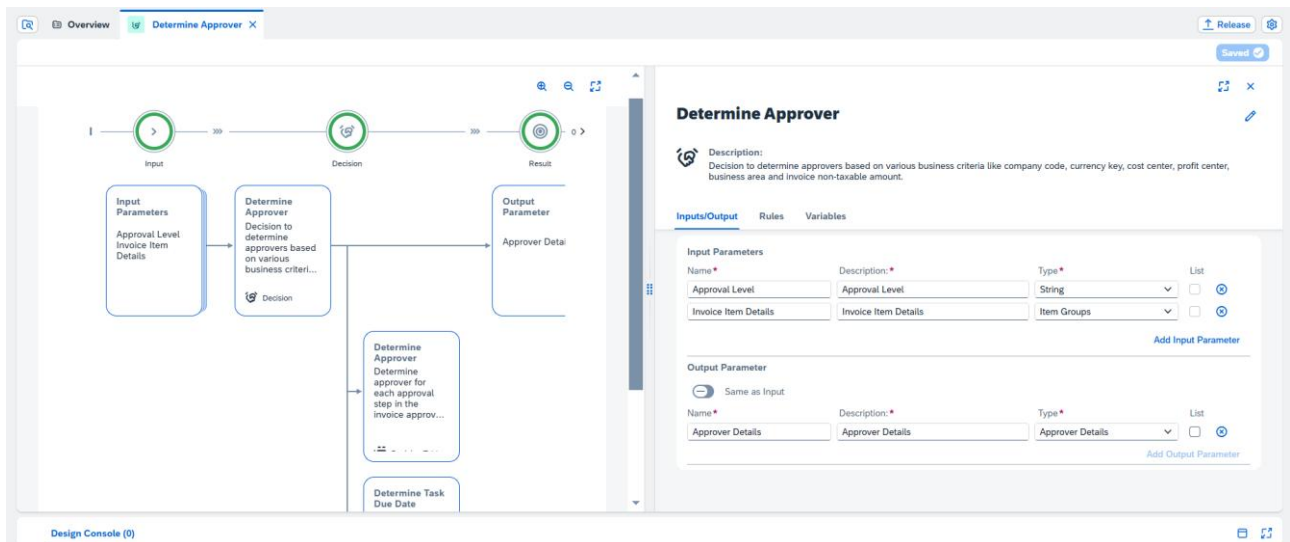
Configure Decisions

There are 5 decisions in this template. We can configure as required.

1. Determine Approver

Decision to determine approvers based on various business criteria like company code, currency key, cost center, profit center, business area and invoice non-taxable amount. To modify the decision:

- In process editor, double click to open the decision.
- Select the rule from the decision diagram.



- Edit the rule to add more rows or update existing value.

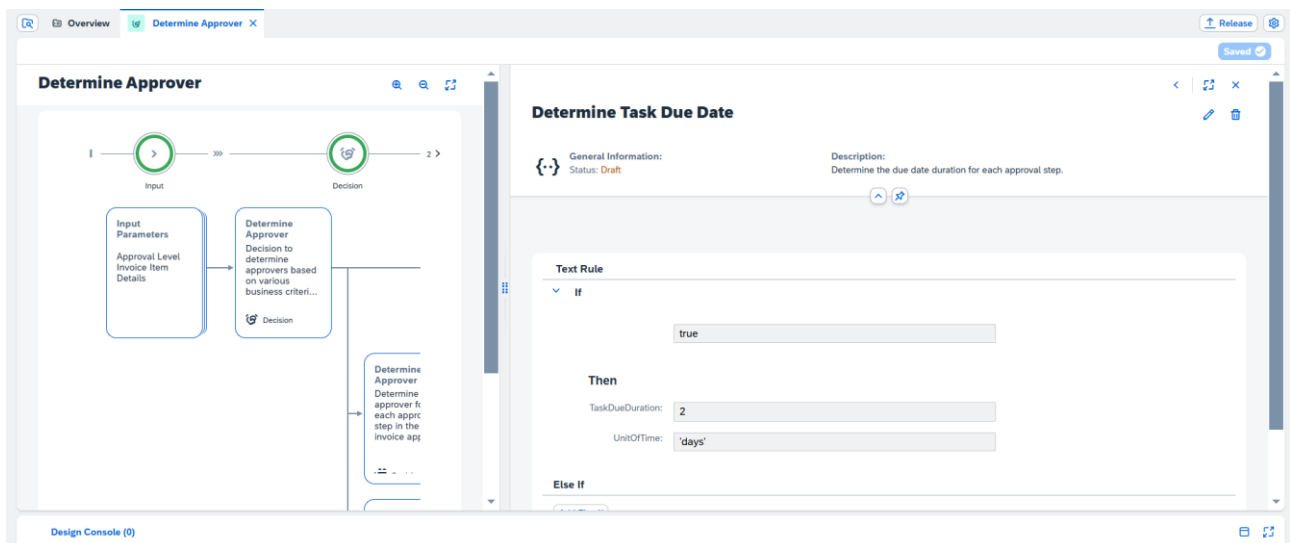
Determine Approver

General Information:
Status: Draft
Hit Policy: **FIRST MATCH**

Description: Determine approver for each approval step in the invoice approval process based on the conditions like company code, currency key, g/ account, plant, cost center, business area, profit center, release group and threshold invoice amount.

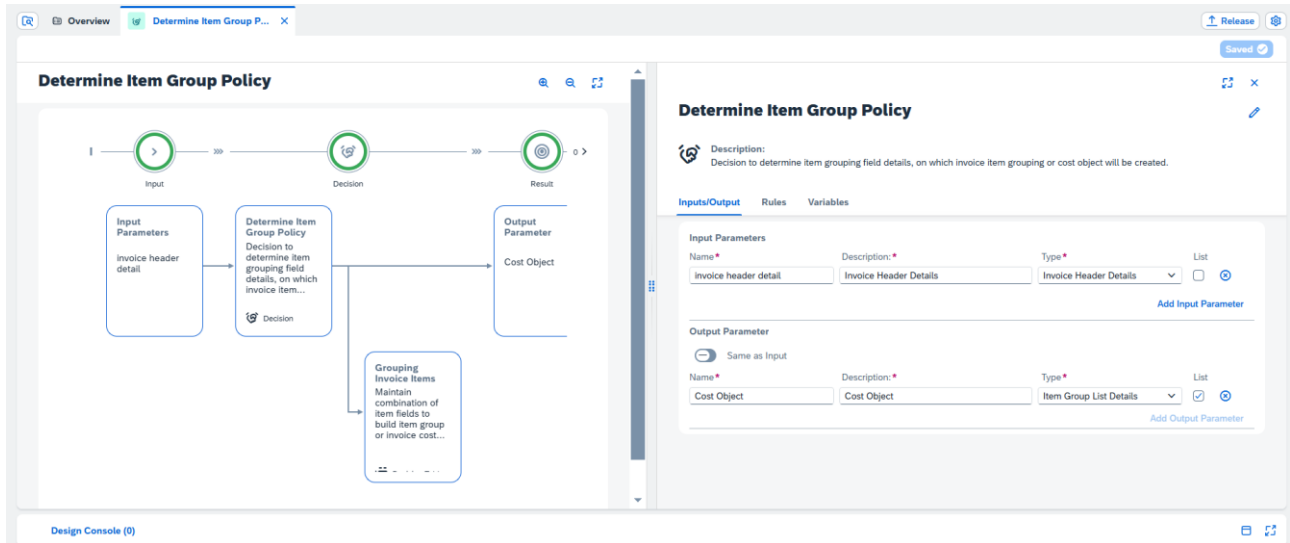
Decision Table

If									Then			
g/Account	plant	companyCode	currencyKey	costCenter	businessArea	profitCenter	amount	Approval Level =	Email Id	User Id	User Group	isApprovalReq...
☐								'Local Mana...				true
☐								'Area Manager'				true
☐								'Local Mana...				true



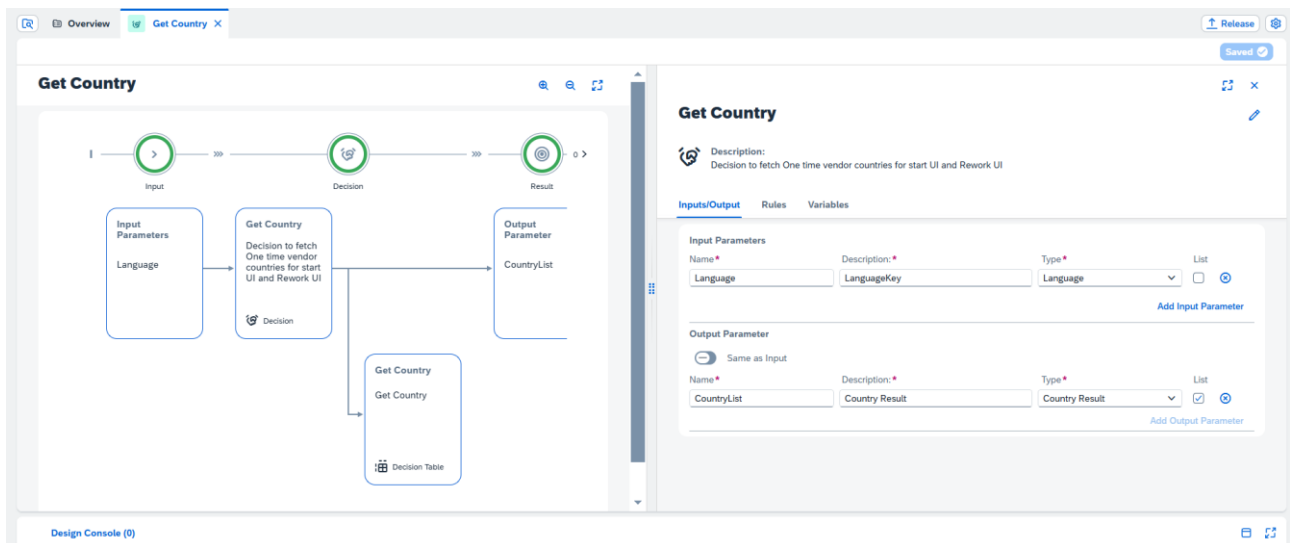
2. Determine Item Group Policy

Decision to determine item grouping field details, on which invoice item grouping or cost object will be created.



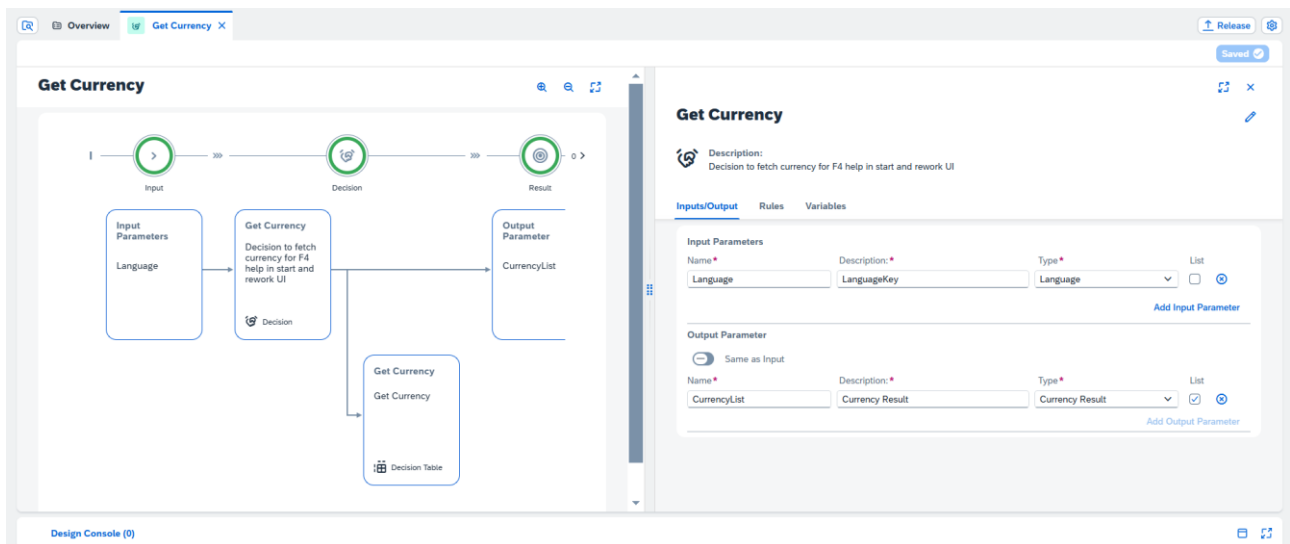
3. Get Country

Decision to fetch one time vendor countries for start and rework UI



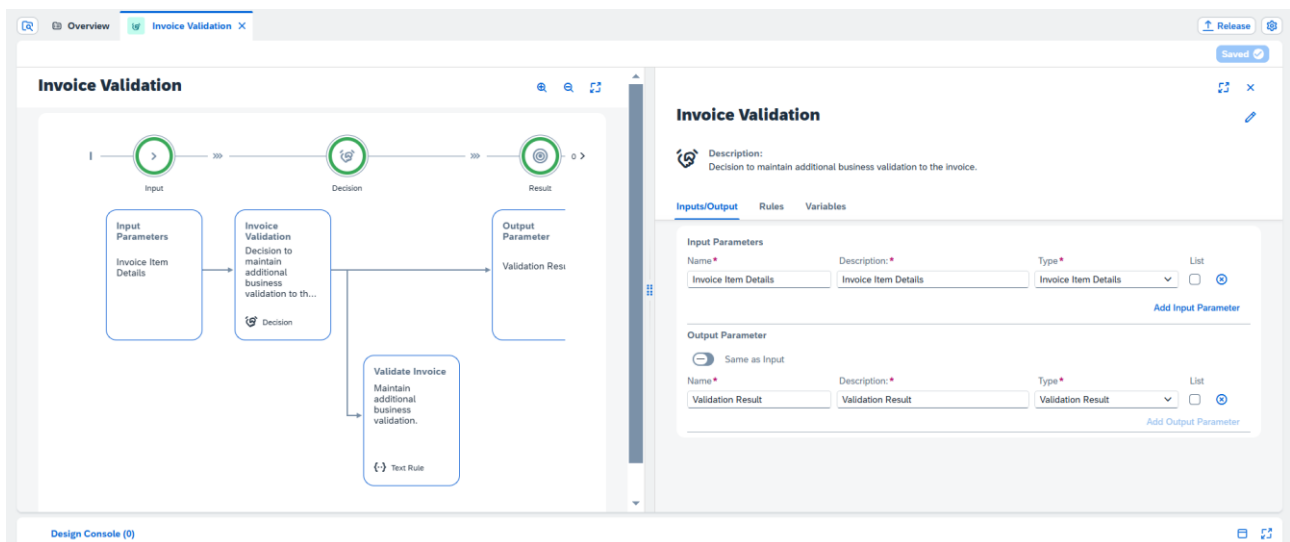
4. Get Currency

Decision to fetch currency for F4 help in start and rework UI



5. Invoice Validation

Decision to maintain additional business validation to the invoice.



Configure Email

Email notifications are sent to requestor and approver at different places during process and task execution. All the email artefacts in the process have default template body which can be modified.

To update the email body content, do the following:

- Select the mail artefact in the process.
- From the **Settings** in the right-side panel, check the **Mail Header** to confirm the **To** field (or email recipients) and the **Subject**. You can adjust the subject and recipient, if needed.
- You may also add someone in **CC**.
- Click to Open '**Mail Body Editor**'.

Dear Approver,

Please find below the Vendor Invoice pending for your approval. Kindly respond in **My Inbox** application.

Invoice Details:

Company Code: <> companyCode

Vendor Number: <> vendorNumber

One Time Vendor: <> oneTimeVendor

One Time Vendor Name: <> oneTimeVendorName

One Time Vendor City: <> oneTimeVendorCity

One Time Vendor Country: <> oneTimeVendorCountry

Invoice Date: <> invoiceDate

Amount: <> invoiceAmount

Payment Method: <> paymentMethod

Payment Terms: <> paymentTerm

Exchange Rate: <> exchangeRate

Reference: <> reference

Text: <> text

Requester: <> requester

Requester Name: <> requesterName

This is an automatically generated message. Please do not respond to it.

Close

- The mail editor allows you to write and adjust text according to your needs. You can also reuse information that has been collected during the process. If you want to use additional process context data, you can simply drag the fields from the left menu to the editor on the right.

For more information on email configuration and text, refer to this [link](#)

In case any further extension needs to be made then follow the reference links below.

Create Process

[Create a Business Process | SAP Help Portal](#)

Create Forms

[Create a Form | SAP Help Portal](#)

Create Decision

[Create a Decision | SAP Help Portal](#)

Create Process Visibility Scenario

[Create a Visibility Scenario | SAP Help Portal](#)

Release and Deploy the Process

After configuring your process, it's time to publish, test and run it. To run a process, you must first release and deploy it.

Release - This locks the version of your process, meaning that no further edits can be made to that version.

At this stage, the process can't be run and isn't available to participants yet.

To release a version of your process:

© 2023 SAP SE or an SAP affiliate company. All rights reserved. See Legal Notice on www.sap.com/legal-notice for use terms, disclaimers, disclosures, or restrictions related to SAP Materials for general audiences.



- First ensure that the status is listed as 'Editable'.
- Then click 'Release'.

For more information please visit: [SAP Build Process Automation](#)

Note: If this is the first time you've released the process, a version labelled 1.0.0 is released. For subsequent releases, you're prompted to select a version type before the process is released.

Import Destination

Actual destinations that are created in SAP BTP cockpit will be required when you deploy the process. This destination will be mapped to the destination environment variables during deployment.

To configure destination,

- Click on **Control Tower**.
- Select **Destinations** from the right panel.
- Click **Add**

Select the destinations created in the setup section to be added.

For more information and screen shots, please refer to the below link: [Manage Destinations](#)

Deploy – This takes a released version of a process and enables it to be actively run and monitored. A deployed version of this approval process unlocks the request form, allowing participants to submit requests and the approvers to view items in their inbox.

To deploy a released version of your process,

- First ensure that the status is listed as 'Released'.
- Then click '**Deploy**'.

For more information and screen shots, please refer to the below link: [Deploy Project](#)

To deploy your project, you will be prompted to provide a destination as an environment variable. Select the destination from the drop-down and fill other environment variables as well.

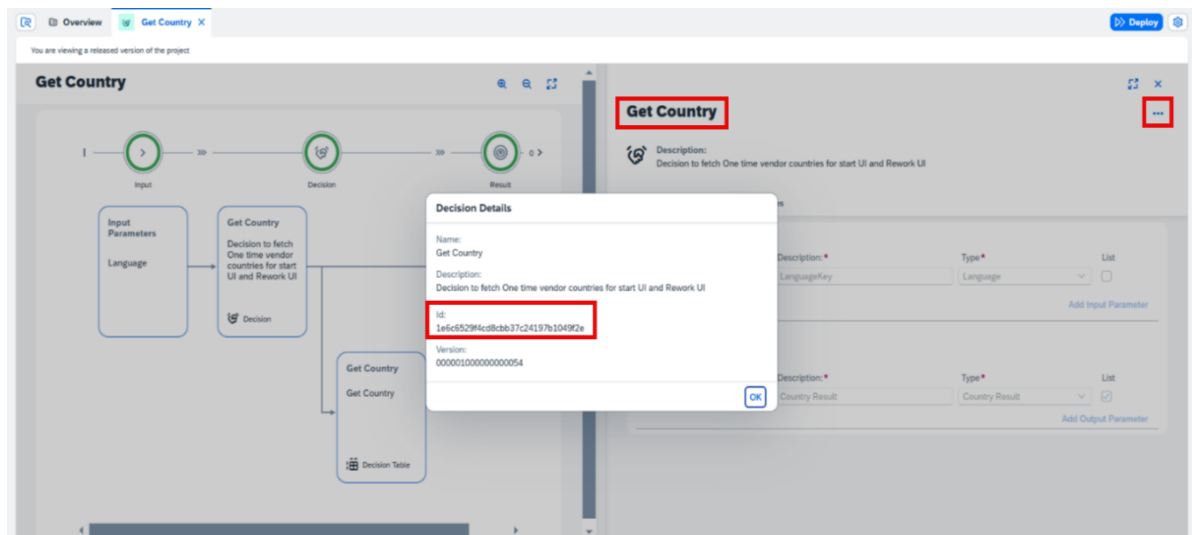
- **SAP S/4HANA:** Business Technology Platform destination to connect with SAP S/4HANA system. As shown in SAP S/4HANA destination set-up-section.
- **IntegrationSuite:** Business Technology Platform destination to connect with destination.

Note: If you do not map your destination environment variable with actual SAP Business Technology Platform destination during deployment, then the process will enter erroneous state when it tries to connect to the external system.

Configure and Deploy Task UI

After the process has been deployed, the decision Id is changed. The updated decision Id must be changed in the corresponding Task UI application:

1. Once the application and the process has been deployed, go to the deployed process and click on the decision: **Get Country**. Here, open the decision and see in the deployed version, you will have 3 dots to the right side. Click on that and you will get a link to 'view details'. Click on view details and you will find the id for the decision. This is the rule id in this case. Note it down.



- Go to Business Application Studio and open the corresponding task UI code; go to the file, **/ReworkUI/webapp/Util/Util.js**. For the function, **getValueHelpDialog**, change the rule id that you have noted above.

```
// Value helps dialog for Header and Item Value helps
getValueHelpDialog: function (oView, sName, bValidate, sInputValue, sCall, sParameter) {
    var sUrl, sTitle, oData, sCode, bTrigger, sValue = undefined;

    this.sName = sName;
    this.sInputValue = sInputValue;
    var slang = sap.ui.getCore().getConfiguration().getLanguage(),
        slanguageRule = slang.toUpperCase();
    var slanguage = slang && slang.split("-") && slang.split("-")[0] && slang.split("-")[0].toUpperCase() && slang.split("-")[0].toUp

    this.oView = oView;
    this.bValidate = bValidate;

    if (sName === "One-time Vendor Country" || sName === "Currency") {
        var sRequestId;
        if (sName === "One-time Vendor Country") {
            sTitle = "Select Vendor Country";
            this.sModelName = "oneTimeVendorCountryModel";
            sRequestId = "1e6c6529f4cd8cbb37c24197b1049f2e";
        } else if (sName === "Currency") {
            sTitle = "Select Currency";
            this.sModelName = "currencyListModel";
            sRequestId = "b14a5e2d458ce973a383de743dbb21fe";
        }
    }
}
```

- Repeat the steps 1 and 2 to replace other decision id's present in the application. Refer below the table for more details:

S No.	Decision Name	UI5 Application File Path	Function Name	Decision Id to be replaced
1	Get Currency	/ReworkUI/webapp/Util/Util.js	getValueHelpDialog	b14a5e2d458ce973a383de743dbb21fe

Configure and Deploy Start UI

The content package requires the SAP UI5 user interface which is available in **Intelligent Processing of Invoices without Purchase Orders** content package. Below are details of Start UI application

Start UI Application: NonPOVendorInvoice-ui_withAI

Following steps can be followed to utilize the SAP UI5 application provided with this process package:

1. Download the SAPUI5 application Start UI from the imported store project.
2. Open Business Application Studio and import the previously downloaded SAPUI5 project.
3. The definition id can be found in the Monitoring section, inside Process and Workflow Definitions. You need to enter the project name and go to the main process. The ID mentioned in the process would be the definition id. Refer [determine the workflow definition ID](#) for more details
4. This definition id must be maintained in in Create.controller.js file. Please refer below screenshot

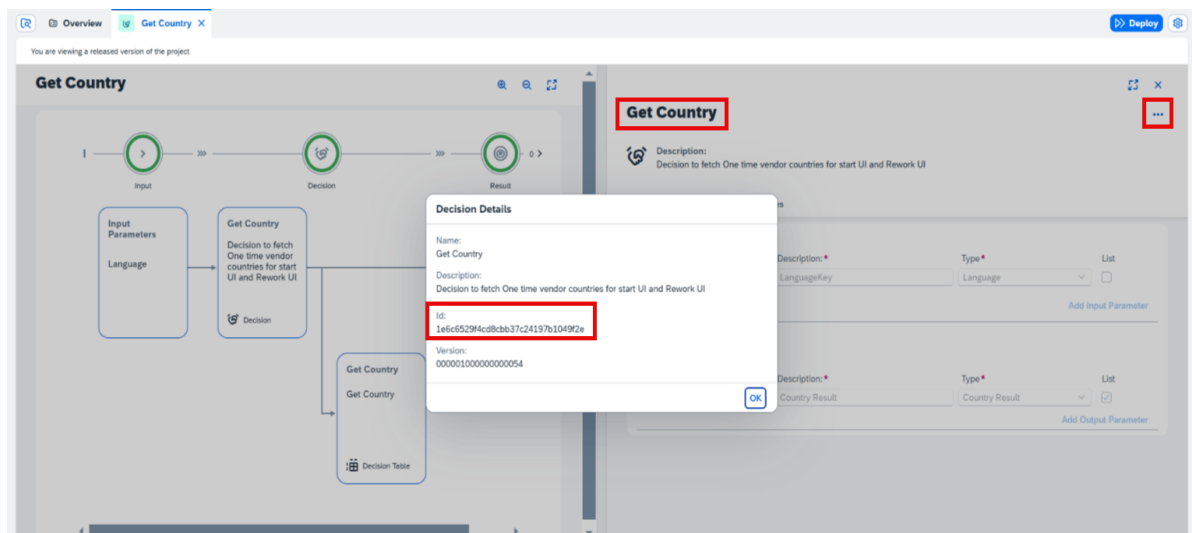
```
// Triggering Work Flow Instance
triggerBackendCallToSaveChanges: function (sToken, oDialog) {
    // set parameters
    this.setParameters(oDialog);

    var oBusyIndicator = Util.getBusyLoader();
    oBusyIndicator.open();
    var aData = {
        definitionId: 1ef6e5294cd8bb37c24197b10492e,
        context: Util.oParamObject
    };

    var iData = JSON.stringify(aData);

    var oController = this;
    $.ajax({
        url: this._getWorkflowRuntimeBaseUrl() + "/workflow-instances",
        method: "POST",
        async: false,
        contentType: "application/json".
    });
}
```

5. Once the process has been deployed, go to the deployed process and click on the decision: **Get Country**. Here, open the decision and see in the deployed version, you will have 3 dots to the right side. Click on that and you will get a link to 'view details'. Click on view details and you will find the **id for the decision**. This is the rule id in this case.



6. This is different for each of us when we deploy the package. Hence this must be changed in the Business Application Studio. To do this, Go to **/StartUI/webapp/Util/Util.js** file and click on **ctrl+F** and search for the text, 'rule'. You will find the rule id used in the file as shown below. Change this to the ID that you have noted in the above step.

```

// Value helps dialog for Header and Item Value helps
getValueHelpDialog: function (oView, sName, bValidate, sInputValue, sCall, sParameter) {
    var sUrl, sTitle, oData, sCode, bTrigger, sValue = undefined;

    this.sName = sName;
    this.sInputValue = sInputValue;
    var slang = sap.ui.getCore().getConfiguration().getLanguage(),
        slanguageRule = slang.toUpperCase();
    var slanguage = slang && slang.split("-") && slang.split("-")[0] && slang.split("-")[0].toUpperCase() && slang.split("-")[0].toUp

    this.oView = oView;
    this.bValidate = bValidate;

    if (sName === "One-time Vendor Country" || sName === "Currency") {
        var sRequestId;
        if (sName === "One-time Vendor Country") {
            sTitle = "Select Vendor Country";
            this.sModelName = "oneTimeVendorCountryModel";
            sRequestId = "1e6c6529f4cd8cbb37c24197b1049f2e";
        } else if (sName === "Currency") {
            sTitle = "Select Currency";
            this.sModelName = "currencyListModel";
            sRequestId = "b14a5e2d458ce973a383de743dbb21fe";
        }
    }
}

```

- Repeat the steps 5 and 6 to replace other decision id's present in the application. Refer below the table for more details:

S No.	Decision Name	UI5 Application File Path	Function Name	Decision Id to be replaced
1	Get Currency	/StartUI/webapp /Util/Util.js	getValueHelp Dialog	b14a5e2d458ce973a383de743d bb21fe

- Once changed, build SAPUI5 project using mta.yaml file the generated project (generated mtar file will be available in the mta_archives folder) file can be deployed. Please also remember to refresh the work zone tile again. Refer [help documentation](#).

Configure SAP Work Zone

Configure SAP Work Zone to access the start UI for this automation.

Once SAP Work zone is configured, perform the following steps to create the tile for the start UI:

- Open content manager and click on the HTML5 Apps.
- Select the Import Manage Vendor Invoices Start UI from the list of apps and choose Add.
- Add the relevant roles to the user. Intelligent Processing of Invoices without Purchase Orders Start UI will be available to the user as a tile in the SAP Work Zone.

If you plan to use SAP Work Zone service, then configure [SAP Work Zone with SAP Build Process Automation Applications](#) using help documentation.

User Guides

Capabilities

The following are the capabilities of this template:

For detailed information about different artifacts, please refer [help documentation](#)

Type	Name	Description
Decision	Determine Approver	Decision to determine approvers based on various business criteria like company code, currency key, cost center, profit center, business area and invoice non-taxable amount.
Decision	Determine Item Group Policy	Decision to determine item grouping field details, on which invoice item grouping or cost object will be created.
Decision	Get Country	Decision to fetch one time vendor countries for start and rework UI
Decision	Get Currency	Decision to fetch currency for F4 help in start and rework UI
Decision	Invoice Validation	Decision to maintain additional business validation to the invoice.
Process	Approval	Process to approve or decline details of the invoice.
Process	Finalization	Process to finalize the approval result of invoice
Process	Initialization	Process to prepare payload for approval
Process	Invoice Header Process	Process to group invoice items and send it for approval
Process	Item Level Approval	Process to approve the invoice item group
Form	Approver Task Ui	Form to approve invoice details
Form	Rework Task UI	Form to submit or cancel invoice details after changing the values according to the approver comments.
Data type	Approval DT	Data type to store group item level invoice details to send as input for approval process
Data type	Approver Details	Data type to store the approver details
Data type	Country Result	Data type to store list of one-time vendor countries
Data type	Currency Result	Data type to store currency results
Data type	History Details	Data type to store history details of approval and rework form
Data type	internal	Data type to store values that are used internally within the process
Data type	Internal Details	Data type to store internal details of the process
Data type	Invoice details	Data type for maintaining all the data about invoice details, item details and header details.

Data type	Invoice Header Details	Data type to store header details of the invoice
Data type	Invoice Item Details	Data type to store item details of the invoice
Data type	Item Group List Details	Data type to store cost object value for invoice item group
Data type	Item Groups	Data type to store item group details for approval/rework
Data type	itemTreeTable	Data type to store processed item details
Data type	Language	Data type to store language key used in a decision for start UI
Data type	Post Invoice Payload	Data type to store payload for post invoice action
Data type	reworkReason	Data type to store the rework reasons
Data type	Validation Result	Data type to store the validation result of invoice

Actions

Intelligent Invoice - Validate Invoice Action

Action to validate invoice details from SAP S/4HANA

Method	API	API Path	Purpose
POST	/http/CheckInvoice	/http/CheckInvoice	validate invoice details from SAP S/4HANA

Intelligent Invoice Processing - Post Invoice Action

Action to post invoice details to SAP S/4HANA after approval

Method	API	API Path	Purpose
POST	/sap/opu/odata/sap/ZAPI_ACC_DOC_POST_SRV/Acc_DocSet	/sap/opu/odata/sap/ZAPI_ACC_DOC_POST_SRV/Acc_DocSet	post invoice details to SAP S/4HANA after approval

Intelligent Invoice Processing - Calculate Tax Action

Action to calculate tax for the invoice amount based on provided tax code

Method	API	API Path	Purpose
POST	/http/fetchTaxcalculation	/http/fetchTaxcalculation	calculate tax for the invoice amount based on provided tax code

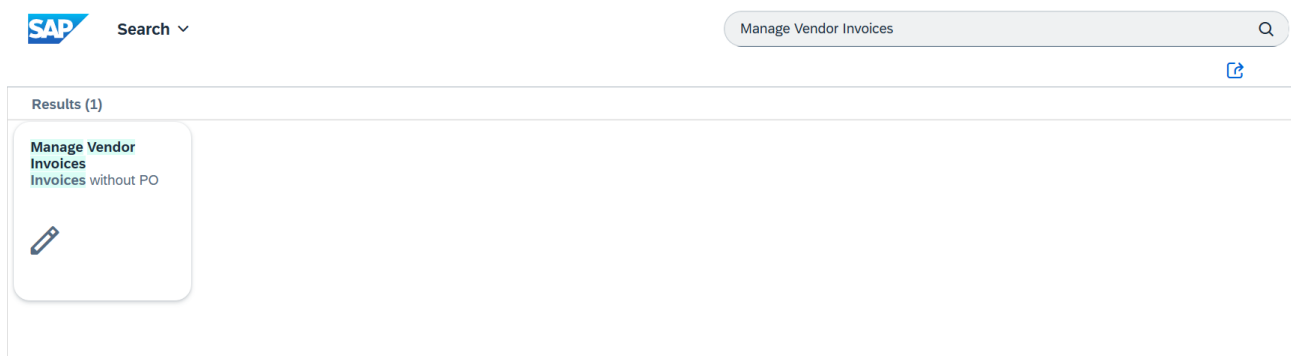
SAP UI5 SCREEN

This is the Start UI application to submit the request for Intelligent Processing of Invoices without Purchase Orders must be entered here to trigger the process.

The tile name that must be chosen in the Work Zone is: **'Manage Vendor Invoices'**.

Once navigated to the tile, User can start the process.

1. Access app Manage Vendor Invoices.



2. The user can initiate the creation of a new Vendor Invoice without Purchase Order Request by manually entering the values.

The screenshot shows the 'Create' screen for 'Manage Vendor Invoices'. The screen has a top navigation bar with the SAP logo and the title 'Manage Vendor Invoices'. Below the navigation bar, there is a 'Create' section with tabs: 'Upload Invoice', 'Header Details', 'Attachments(0)', 'Item Details', and 'Duplicate Invoices'. The 'Upload Invoice' tab is selected. In this tab, there is a text input field for 'Upload Invoice:' with the placeholder 'Choose a file for Upload...' and a 'Browse...' button. Below this is a blue 'Upload Invoice' button. The 'Header Details' tab is also visible, showing a form with fields for 'DocX Job ID:', 'Company Code:', 'Vendor Number:', 'One Time Vendor:', 'Invoice Date:', 'Amount:', 'Exchange rate:', and 'Tax Amount:'. The 'Company Code' field is filled with 'LP Company Code US- Ops' and the 'Vendor Number' field is filled with 'LP Local Vendor 13'. The 'Amount' field is filled with 'USD'. At the bottom right of the screen, there are 'Save' and 'Cancel' buttons.

3. The user must give all the header details manually like Company Code, Vendor Number, Amount etc and Item details like G/L Account, Amount in Document Currency etc using F4 help.
4. On clicking 'Check Duplicate Invoices' button, invoices that are already posted to SAP S/4HANA will be displayed with the status 'Completed'. And invoices that are processing will be displayed with the status 'In Approval'

In 'Header Details', we have 19 fields:

DocX job Id: Unique identifier for the DocX job processing.
Company code (Mandatory): Unique code representing the company entity.
Vendor number (Mandatory): Identifier for the vendor in the system.
One-time vendor: Temporary vendor without a permanent record.
Invoice date (Mandatory): Date when the invoice was issued.
Amount (Mandatory): Total invoice value in the relevant currency.
Currency key: Code representing the currency used.
Exchange rate: Conversion rate between currencies.
Tax amount: Tax applied on the invoice amount.
Total amount: Sum of the invoice and tax amounts.
Payment method: Method selected for payment processing.
Payment terms: Agreed conditions for invoice payment.
Days: Number of days in the payment term.
Due date: Date payment is due based on terms.
Reference (Mandatory): Invoice reference for tracking or clarification.
Text: Description or notes related to the invoice.
Requester: Person initiating the invoice process.
Requester name: Name of the requester.

In 'Attachments' tab, we have 1 field:

Attachments (Mandatory): Attachments: Supporting documents related to the invoice.

In 'Item Details', we have 10 fields:

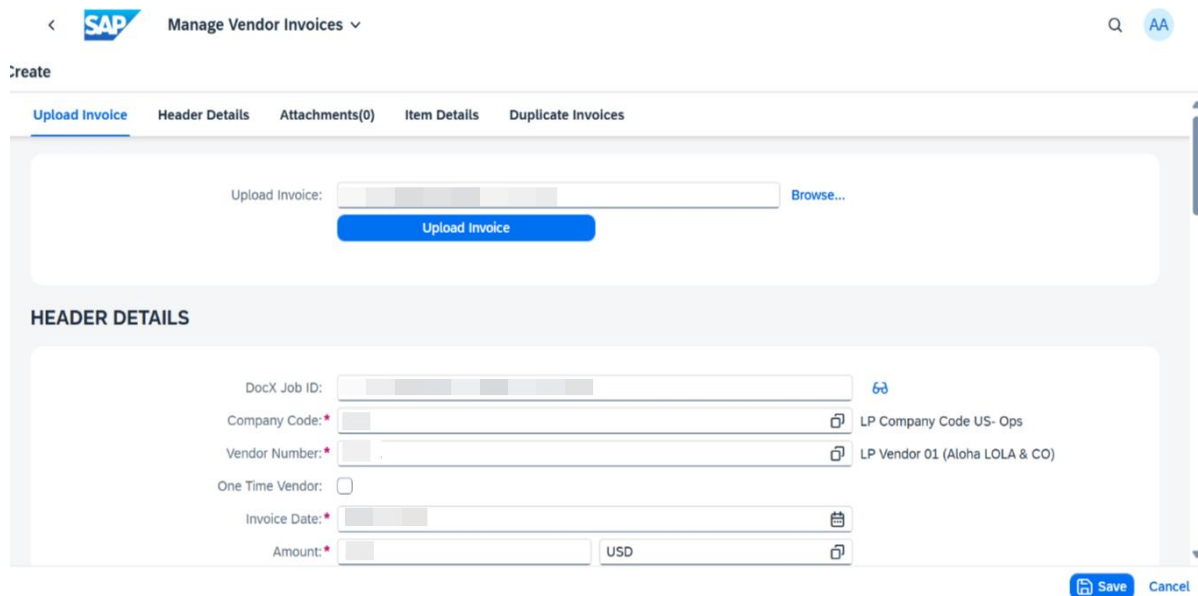
G/L account (Mandatory): General ledger account for the invoice.
Cost center: Department or unit bearing the cost.
Amount in document currency (Mandatory): Value expressed in the original transaction currency
Currency key: Represents the specific code identifying that currency
Tax code: Code for tax type applied.
Business area: Specific area of business related to the invoice.
Profit center: Unit responsible for profit within the organization.
Plant: Operational location or facility.
Item text: Description of individual invoice items.
Internal order: Specific internal tracking number.

In 'Duplicate Invoices', we have 7 fields:

Accounting document number: System-generated document identifier.
Reference: A unique identifier or code used to track a specific transaction or document.
Vendor Number: A unique code assigned to a supplier for identification in the system.
Invoice Date: The date when an invoice is issued for a transaction or purchase
Status: Current progress or state of the invoice.
Requested by: Individual who made the invoice request.
Requested on: Date the invoice request was initiated.

Upload Invoice Feature

The user will also be able to create a new Vendor Invoice without Purchase Order Request using the 'Upload Invoice' button. When a Purchase Order Invoice is uploaded the DocX Job ID will get populated automatically. Then click the (specs)  icon which is present along with the DocX Job ID to populate the header details and line item details.



In the 'Header Details' tab, there are 5 mandatory fields and 1 non-mandatory field that will be automatically filled after clicking the 'specs' icon. The attachments and G/L account must be entered manually.

Once all the mandatory/required details are filled, click on the Save button in SAP UI5 screen to start the process

Appendix

Setup service: `/sap/opu/odata/sap/ZAPI_ACC_DOC_POST_SRV/Acc_DocSet`

1. Go to the **SAP GUI** and enter **SEGW** in the command field to open the **Gateway Service Builder**.
2. In the **Service Builder** screen, click on the **"Create Project"** button.
3. Enter a name 'ZAPI_ACC_DOC_POST' for your project in the **Project Name** field. This name should be unique and descriptive.
4. Provide a relevant short description of your project.
5. In the left-hand tree structure, right-click on **Data Model** and choose **Import -> RFC/BOR interface**. Enter the entity name and type and name in Data Source Attributes section then click Next.

Wizard Step 1 of 3: Create Entity Type from Data Source

Import Entity Type From Data Source

Entity Type Name

Target System

☒ Local
☐ Remote

RFC Destination

Data Source Attributes

Type

Name

☒ Create Default Entity Set

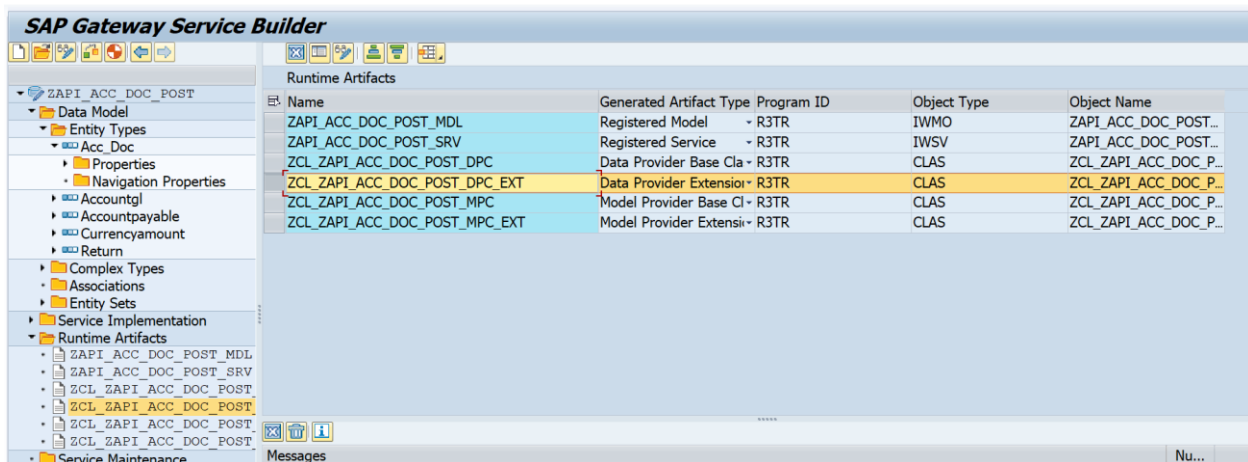
6. Select the fields and structures to create the entity.

Wizard Step 2 of 3: Create Entity Type from Data Source

Select Parameter(s)

Data Source Parameter	Assign Structure	Mandatory	Default Value	Description
☐ CUSTOMERCPD	☐	☐		One-time customer
☒ DOCUMENTHEADER	☐	☒		Header
☐ OBJ_KEY	☐	☐		Reference Key
☐ OBJ_SYS	☐	☐		Logical system
☐ OBJ_TYPE	☐	☐		Ref. procedure
☒ ACCOUNTGL[]	☐	☐		G/L account item
☒ ACCOUNTPAYABLE[]	☐	☐		Vendor Item
☐ ACCOUNTRECEIVABLE[]	☐	☐		Customer Item
☐ ACCOUNTTAX[]	☐	☐		Tax item
☐ ACCOUNTWT[]	☐	☐		Withholding tax information for FI Interface
☐ CONTRACTITEM[]	☐	☐		Additional Contract Accounts Receivable and Payable Document Line Item
☐ CRITERIA[]	☐	☐		CO-PA Account Assignment Characteristics
☒ CURRENCYAMOUNT[]	☐	☒		Currency Items
☐ EXTENSION1[]	☐	☐		Container for 'Customer Exit' Parameter
☐ EXTENSION2[]	☐	☐		Reference Structure for BAPI Parameters EXTENSIONIN/EXTENSIONOUT
☐ PAYMENTCARD[]	☐	☐		Payment Card Information
☐ REALESTATE[]	☐	☐		Real Estate Account Assignment Data
☒ RETURN[]	☒	☒		Return parameter
☐ VALUEFIELD[]	☐	☐		CO-PA Account Assignment Value Fields

Back Next Cancel



7. Give the entity type a name as 'Acc_doc'
8. After creating the entity type, you can add **properties** (fields/attributes) that correspond to the attributes of your data model. After defining entity types and services, generate runtime objects using **Service Builder**, this will generate the necessary classes in the backend.
9. Redefine the method "/iwbep/if_mgw_appl_srv_runtime~create_deep_entity" and add the code within the Method & Endmethod given as below:

METHOD /iwbep/if_mgw_appl_srv_runtime~create_deep_entity.

DATA: BEGIN OF ls_payload.

INCLUDE TYPE zcl_zapi_acc_doc_post_mpc=>ts_acc_doc.

DATA: to_accgl TYPE TABLE OF zcl_zapi_acc_doc_post_mpc=>ts_accountgl,
to_accpay TYPE TABLE OF zcl_zapi_acc_doc_post_mpc=>ts_accountpayable,
to_currmnt TYPE TABLE OF zcl_zapi_acc_doc_post_mpc=>ts_currencyamount,
to_return TYPE TABLE OF zcl_zapi_acc_doc_post_mpc=>ts_return,
END OF ls_payload.

DATA: ls_doc_header TYPE bapiache09,
lv_obj_type TYPE bapiache09-obj_type,
lv_obj_key TYPE bapiache09-obj_key,
lv_obj_sys TYPE bapiache09-obj_sys,
lt_gl_acc TYPE TABLE OF bapiacgl09,
lt_acc_pay TYPE TABLE OF bapiacap09,
lt_curr_amnt TYPE TABLE OF bapiaccr09,
lt_return TYPE TABLE OF bapiaret2,
ls_return TYPE bapiaret2,
lv_entity_set_name TYPE /iwbep/mgw_tech_name.

DATA(exception) = NEW /iwbep/cx_mgw_busi_exception().

DATA(msg_container) = exception->get_msg_container().

* Get Entity Set Name

lv_entity_set_name = io_tech_request_context->get_entity_set_name().

CASE lv_entity_set_name.

WHEN 'Acc_DocSet'.

io_data_provider->read_entry_data(
IMPORTING
es_data = ls_payload).

ls_doc_header-username = sy-uname.
ls_doc_header-header_txt = ls_payload-header_txt.
ls_doc_header-comp_code = ls_payload-comp_code.

```

ls_doc_header-doc_date = sy-datum.
ls_doc_header-pstng_date = sy-datum.
ls_doc_header-doc_type = ls_payload-doc_type.
ls_doc_header-ref_doc_no = ls_payload-ref_doc_no.

```

```

ls_payload-username = sy-uname.
ls_payload-doc_date = sy-datum.
ls_payload-pstng_date = sy-datum.

```

```

LOOP AT ls_payload-to_accgl ASSIGNING FIELD-SYMBOL(<fs_accgl>).
  APPEND VALUE #( itemno_acc = <fs_accgl>-itemno_acc
    gl_account = <fs_accgl>-gl_account
    item_text = <fs_accgl>-item_text
    comp_code = <fs_accgl>-comp_code
    tax_code = <fs_accgl>-tax_code
    costcenter = <fs_accgl>-costcenter
  ) TO lt_gl_acc.
ENDLOOP.

```

```

LOOP AT ls_payload-to_accpay ASSIGNING FIELD-SYMBOL(<fs_accpay>).
  APPEND VALUE #( itemno_acc = <fs_accpay>-itemno_acc
    vendor_no = <fs_accpay>-vendor_no
    comp_code = <fs_accpay>-comp_code
    pmnttrms = <fs_accpay>-pmnttrms
  ) TO lt_acc_pay.
ENDLOOP.

```

```

LOOP AT ls_payload-to_curramnt ASSIGNING FIELD-SYMBOL(<fs_curramnt>).
  APPEND VALUE #( itemno_acc = <fs_curramnt>-itemno_acc
    currency = <fs_curramnt>-currency
    amt_doccur = <fs_curramnt>-amt_doccur
  ) TO lt_curr_amnt.
ENDLOOP.

```

```

CALL FUNCTION 'BAPI_ACC_DOCUMENT_POST'
  EXPORTING
    documentheader = ls_doc_header
  IMPORTING
    obj_type = lv_obj_type
    obj_key = lv_obj_key
    obj_sys = lv_obj_sys
  TABLES
    accountgl = lt_gl_acc
    accountpayable = lt_acc_pay
    currencyamount = lt_curr_amnt
    return = lt_return.

```

```

ls_payload-obj_type = lv_obj_type.
ls_payload-obj_key = lv_obj_key.
ls_payload-obj_sys = lv_obj_sys.
ls_payload-ac_doc_no = lv_obj_key.

```

```

CLEAR ls_payload-to_return.
IF line_exists( lt_return[ type = 'E' ] ).
  CALL FUNCTION 'BAPI_TRANSACTION_ROLLBACK'.

```

```

LOOP AT lt_return ASSIGNING FIELD-SYMBOL(<fs_return>).
  APPEND VALUE #( type = <fs_return>-type
    id = <fs_return>-id

```

```

        number    = <fs_return>-number
        message    = <fs_return>-message
    ) TO ls_payload-to_return.
ENDLOOP.

ELSE.
CALL FUNCTION 'BAPI_TRANSACTION_COMMIT'
EXPORTING
    wait = abap_true.

LOOP AT lt_return ASSIGNING FIELD-SYMBOL(<fs_ret>).
    APPEND VALUE #( type    = <fs_ret>-type
                    id      = <fs_ret>-id
                    number   = <fs_ret>-number
                    message  = <fs_ret>-message
    ) TO ls_payload-to_return.
ENDLOOP.

ENDIF.
copy_data_to_ref(
    EXPORTING
        is_data = ls_payload
    CHANGING
        cr_data = er_deep_entity
    ).
ENDCASE.

ENDMETHOD.

```

- After implementing the logic, register and activate the service using the **/IWFND/MAINT_SERVICE** transaction and test the service in the SAP Gateway Client (/IWFND/GW_CLIENT).

Support

There is no support available for template content. If you experience general issues with SAP Build Process Automation, please follow the links below:

- [Troubleshoot Support Guided Answer](#)
- [Test an Automation Help portal Article](#)
- [Test an Application Help portal Article](#)
- [Get traces Support Guided Answer](#)

If the issue persists, please raise an incident via SAP Support Portal on **BPI-PA-BPC** component.

This template can be modified using SAP Process Automation. For example: add decisions to determine the approver based on a certain business logic or add an action to post collected information to backend SAP system. Refer [help portal](#) for more details.

www.sap.com