

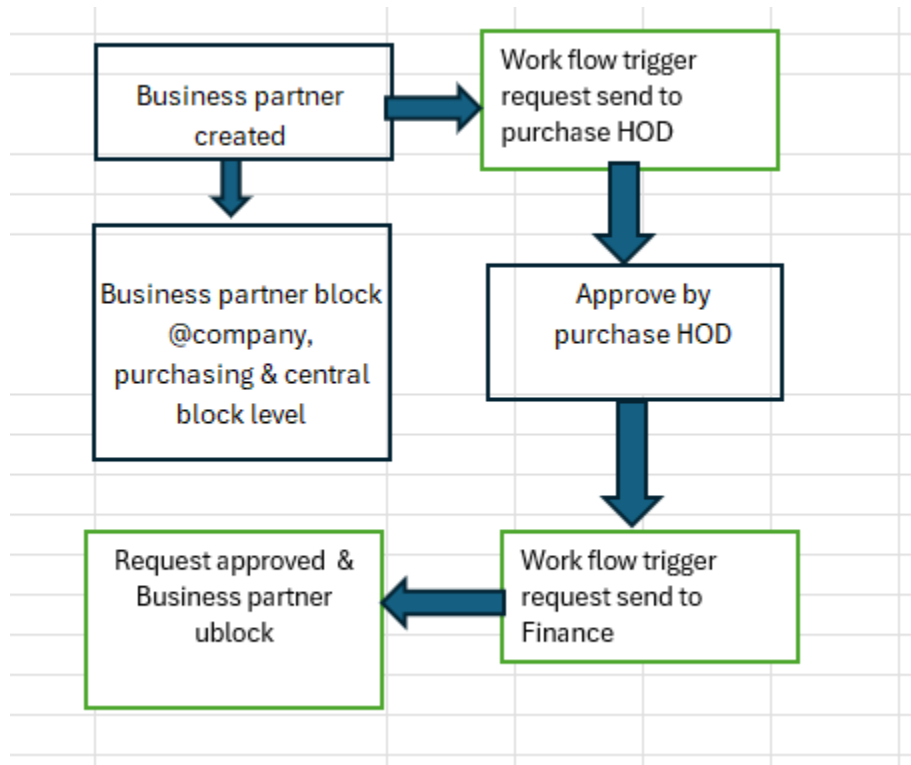
## WORKFLOW EXAMPLE

### 1. Requirement

Business requirement is after create business partner in system through system approval request should trigger, approval can view all the business partner details on same approval screen. Approval should be 3 level approval as by HOD , Finance , customer.

Once a business partner is created in the system it should block automatically. And after final approval business partner should unblock.

### 2. Process flow



## WORKFLOW EXAMPLE

### 3. Functional Logic :

- When user create business partner in transaction code BP.
- Firstly we need to check Business partner must be BLOCK. As BUT000 – XBLCK = X .
- We need to check partner group field as . **KTOKK** or as Business Partner Grouping.
- We need to set logic approval based on this field as .

**1) FOR VENDOR- KTOKK = ZDOM, ZBRK, ZCAP, ZIMP, ZTRP.**

**2) FOR FARMER- KTOKK = ZFRM.**

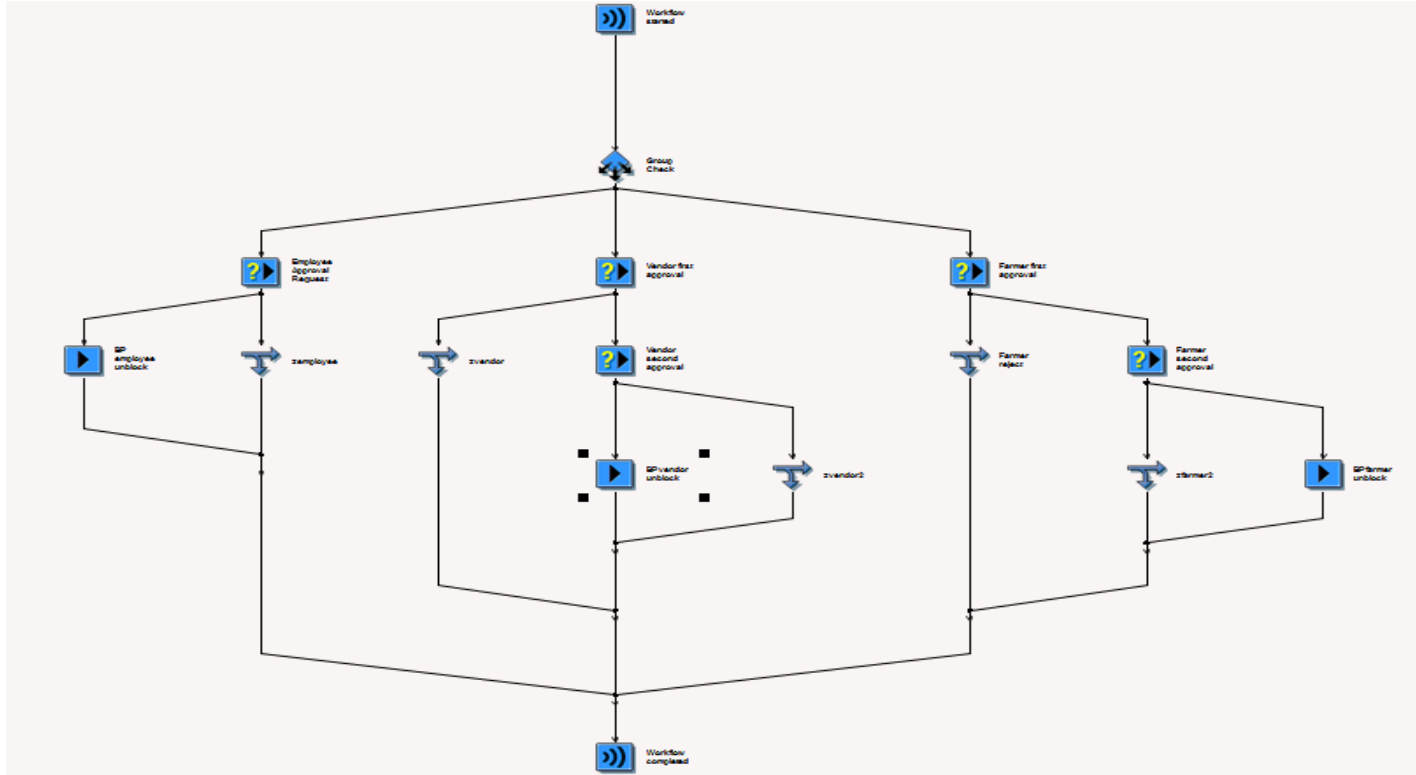
**3) FOR EMPLOYEE – KTOKK = ZEMP.**

- And passing the lifnr value or as Business partner number.
- We are passing these 2 values in workflow. Based on these 2 values our workflow will be flow.
- **After Workflow will end . then we need to unblock a field as**
- BUT000 – XBLCK = X - central block (tab- general status)
- To unBlock field we need to create a enhancement where we can set the value of this XBLCK field as blank for **UNBLOCKED**.
- If it is X then it is blocked.
- So basically we have to find enhancement when BP is SAVE .
- And call workflow there by passing group and business partner.
- After 2 level of approval then we have to unblock it.

**Mail format is given in FS for HOD and Finance.**

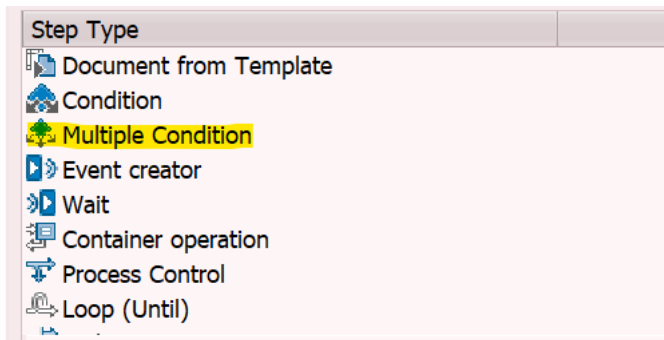
## WORKFLOW EXAMPLE

### 4. Workflow Design.

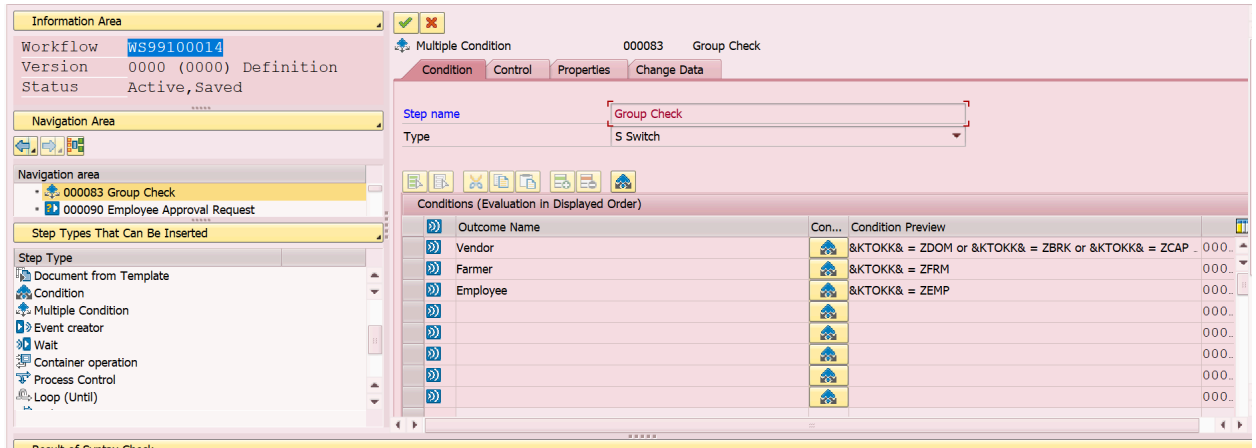


### 5. STEPS To design this workflow.

1. Go to SWDD TCODE and Create a Workflow. Our generated workflow no is WS99100014.
2. So here firstly we are creating a Multiple condition as

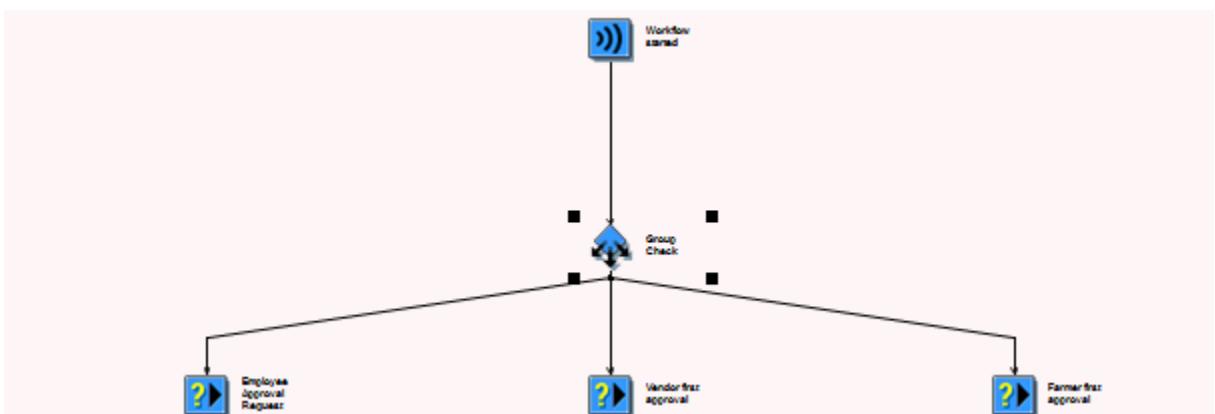


## WORKFLOW EXAMPLE



So here we are giving multiple conditions as vendor , Farmer and Employee.  
Based on this condition workflow will be flowed.

### 3. After this step we are creating a **User Decisions Steps.**



This Approval should be as follow based on the condition it will be trigger.  
This as

### 1) Employee.

- For Employee first we need to create a Task .
- To create a Task go to TCODE- **PFTC\_INS**
- Click on Create and give object and method as below.

## WORKFLOW EXAMPLE

**Standard Task: Display**

Standard task: 99100061 ZBPEMP  
 Name: Employee block in BP  
 Package: ZMM      Applic. Component: \_\_\_\_\_

Basic data | Description | Container | **Triggering events** | Terminating events | Default rules | SAPphone

Name: ZBPEMP  
 Abbr.: Employee block in BP  
 Release status: Not defined

Work Item Text: \_\_\_\_\_

Object method: BO BOR Object Type: DECISION    Object: Decision  
 Method: PROCESS    Perform check  
 Synchronous object method  
 Object method with dialog

- Go to Container Tab and add field as

**Standard Task: Display**

Standard task: 99100061 ZBPEMP  
 Name: Employee block in BP  
 Package: ZMM      Applic. Component: \_\_\_\_\_

Basic data | Description | **Container** | Triggering events | Terminating events | D

Expression      M.    Description      Initial value

▶ Ad Hoc Objects		Ad Hoc Objects of Workflow Instance	< Not Set >
▶ Attachments		Attachments of Workflow Instance	< Not Set >
▪ Agent		Actual Agent of Workflow Activity	< Not Set >
▶ Grouping Character		Grouping Characteristic for Workflow Instances	< No Instance >
▶ Step		Step Instance	< No Instance >
▶ Agents		Result of Agent Determination	< No Instance >
▶ Second.Method Object		Secondary Method Objects of Workflow Activity	< Not Set >
▪ Ext. user decision		Extended user decision	< Not Set >
▪ Decision_Note		Decision_Note	< No Instance >
▶ Template for notes		SOFM template for notes	< No Instance >
▶ _Wi_Object_ID		_Wi_Object_ID	< No Instance >
▪ _WI_Result		_WI_Result	< Not Set >
▪ lifnr		lifnr	< Not Set >
▪ ktokk		ktokk	< Not Set >

## WORKFLOW EXAMPLE

The screenshot shows the 'Display Container Element' window for element 'lifnr'. The 'Texts' section contains 'Name' and 'Short Descript.' both set to 'lifnr'. The 'D. Type' tab is active, showing 'SelectionOfPredefinedTypes' with radio buttons for 'Object Type', 'ABAP Dict. Reference', and 'ABAP Dict. Data Type'. The 'ABAP Dict. Data Type' option is selected, with 'Type Name' set to 'LIFNR' and the description 'Account Number of Supplier'.

Note Give parameter setting as

The screenshot shows the 'Parameter Settings' section for the 'Initial Valu' tab. It includes checkboxes for 'Import' (checked), 'Export' (checked), and 'Mandatory' (unchecked).

The screenshot shows the 'Display Container Element' window for element 'ktokk'. The 'Texts' section contains 'Name' and 'Short Descript.' both set to 'ktokk'. The 'D. Type' tab is active, showing 'SelectionOfPredefinedTypes' with radio buttons for 'Object Type', 'ABAP Dict. Reference', and 'ABAP Dict. Data Type'. The 'ABAP Dict. Data Type' option is selected, with 'Type Name' set to 'KTOKK' and the description 'Vendor account group'.

## WORKFLOW EXAMPLE

- For mail we are adding Mail content in this Description TAB as by editing it.

**Standard Task: Display**

Standard task: 99100061      ZBPEMP

Name: Employee block in BP

Package: ZMM      Applic. Comp

Basic data    Description    Container    Triggering events    Terminating

Text type: **D120 Task description**      Language: E English

After Task has been created go to workflow and give task number in control tab as below.

**Workflow Builder - Display 'ZWF\_BP'**

Information Area: Workflow WS99100014, Version 0000 (0000) Definition, Status Active, Saved

Navigation Area: 000090 Employee Approval Request, 000094 Vendor first approval

Step Types That Can Be Inserted: Document from Template, Condition, Multiple Condition, Event creator, Wait

User Decision: 000090 Employee Approval Request

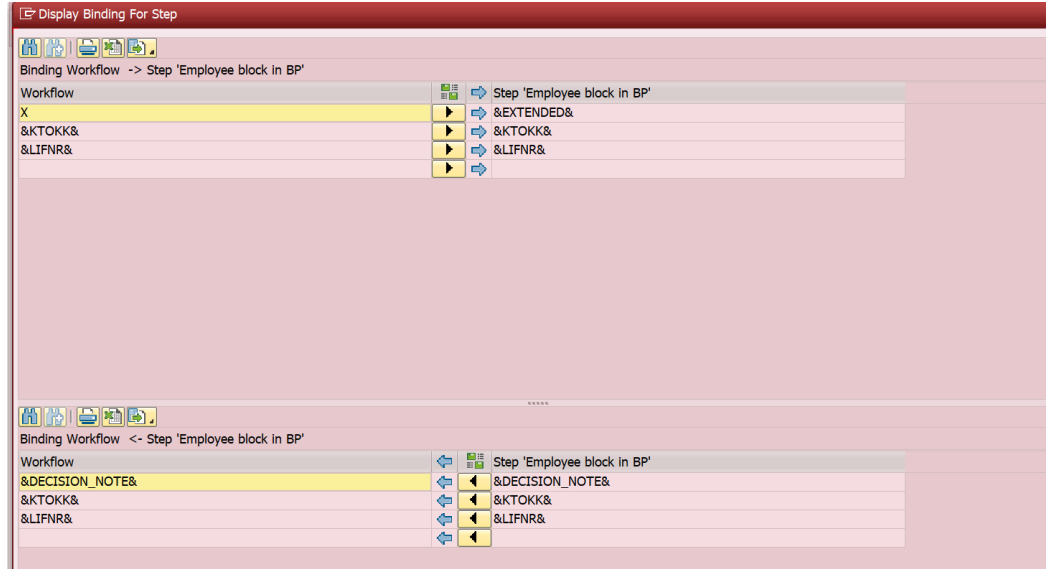
Control Tab: Task TS99100061 Employee block in BP, Step Name Employee Approval Request, Binding (Exists)

Task Properties: Agent Assignment, Task Complete, Background Processing, Confirm End of Processing

Template for Attachm. [Task description - click here to go to task...](#)

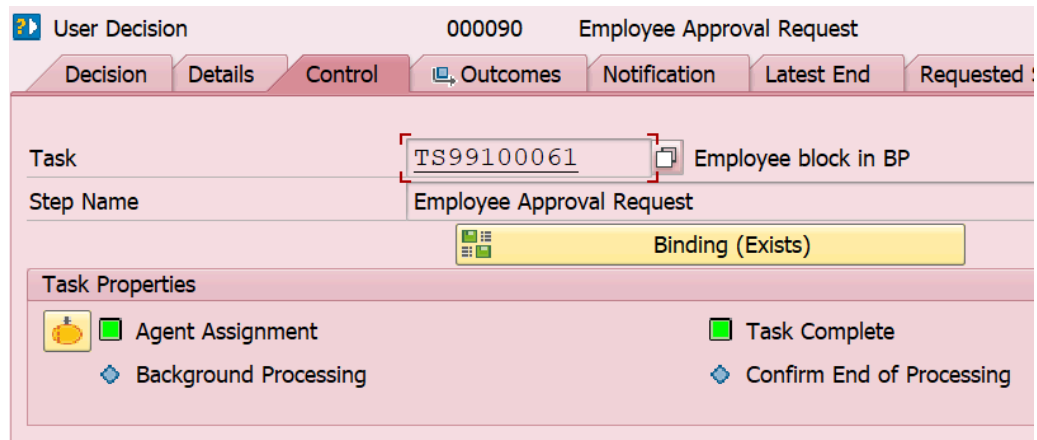
Click on Binding(Exits) button and give binding as follow.

## WORKFLOW EXAMPLE



It will be auto generated to.

Click on Agent Assignment and give Agent.



Check here Task properties should be green as

Agent assignment and task complete.

As this ...

## WORKFLOW EXAMPLE

**Standard task: Maintain Agent Assignment**

Attributes... Org. assignment

Name	ID	General or Background Task
Employee block in BP	TS 99100061	General Task
Monali Bargal	US B.MONALI	

So here we have to add 2 buttons for approve and reject as .

✓ ✗

User Decision 000090 Employee Approval Request

Decision Details Control Outcomes Notification Latest End Requested Start Latest Start Requester

Title Employee first approval

Parameter 1 Parameter 2

Parameter 3 Parameter 4

Agents

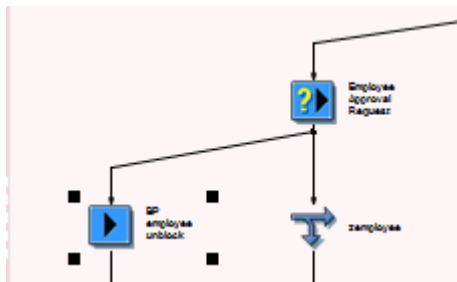
US User B . MONALI Monali Bargal

Excluded

Decision Options

Decision Texts	Outcome Name	Decision Note
Approve	Approve	3 Unsupported
Reject	Reject	3 Unsupported

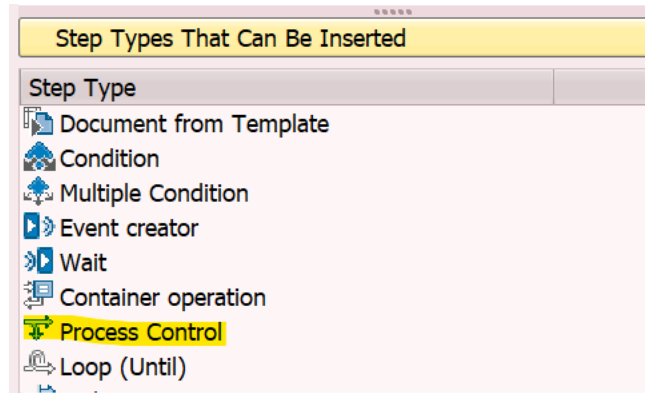
Save and click on Execute.



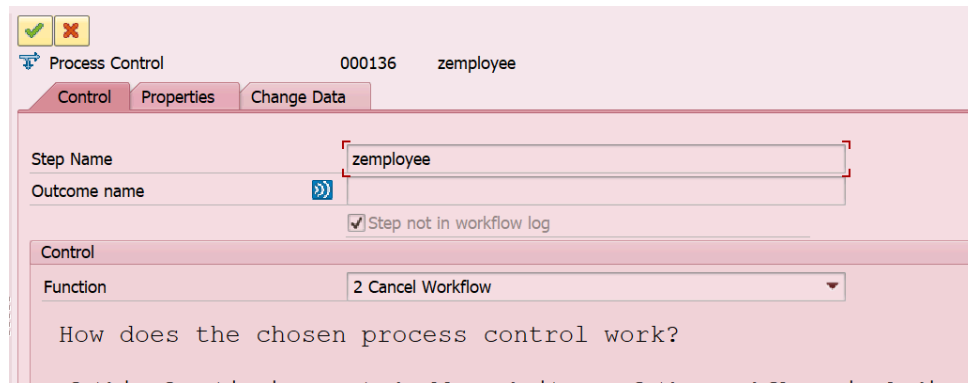
So here 2 Nodes are generated as

- 1) For Approve
  - 2) For Reject
- To stop code in Reject we are adding
  - Process Control Step as follow

## WORKFLOW EXAMPLE

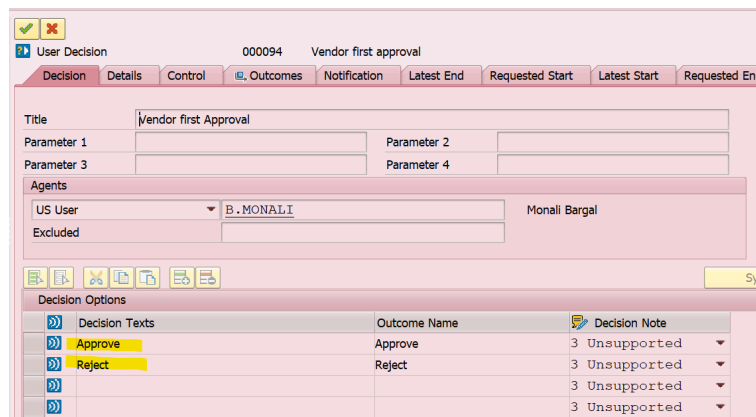


- To stop a Workflow we need to set Process Control as follows



## 2) Employee.

- First step is to give decisions as follows



- To create a Task go to TCODE- **PFTC\_INS**
- Click on Create and give object and method as below.

## WORKFLOW EXAMPLE

Basic data | Description | Container | Triggering events | Terminating events

Name

Abbr. ZBPVEN

Name Vendor block in BP

Release status Not defined

Work Item Text

Work item text

Object method

Object Category BO BOR Object Type

Object Type DECISION Object: Decision

Method PROCESS Perform check

Synchronous object method

Object method with dialog

- Then in description add a Mail Body as

Basic data | Description | Container | Triggering events | Terminating events | Default rules

Text type: 0120 Task description Language: E English

Dear Sir,

Business partner &lifnr& has been created its approval request is Pending with you, for processing the request kindly login into SAP and follow below steps.

GO to T. code SWBP -> click on inbox -> click on request -> Approve / Reject.

Regards,  
SAP team

- And in Container maintain fields as lifnr and Ktokk as below.

Expression	M. Description	Initial value
Ad Hoc Objects	Ad Hoc Objects of Workflow Instance	< Not Set >
Attachments	Attachments of Workflow Instance	< Not Set >
Agent	Actual Agent of Workflow Activity	< Not Set >
Grouping Character	Grouping Characteristic for Workflow Instances	< No Instance >
Step	Step Instance	< No Instance >
Agents	Result of Agent Determination	< No Instance >
Second.Method Object	Secondary Method Objects of Workflow Activity	< Not Set >
Ext. user decision	Extended user decision	< Not Set >
Decision_Note	Decision_Note	< No Instance >
Template for notes	SOFM template for notes	< No Instance >
_WI_Object_ID	_WI_Object_ID	< No Instance >
lifnr	lifnr	< Not Set >
ktokk	ktokk	< Not Set >
_WI_Result	_WI_Result	< Not Set >

- After that save and go to back and save as give agent and do Binding init

## WORKFLOW EXAMPLE

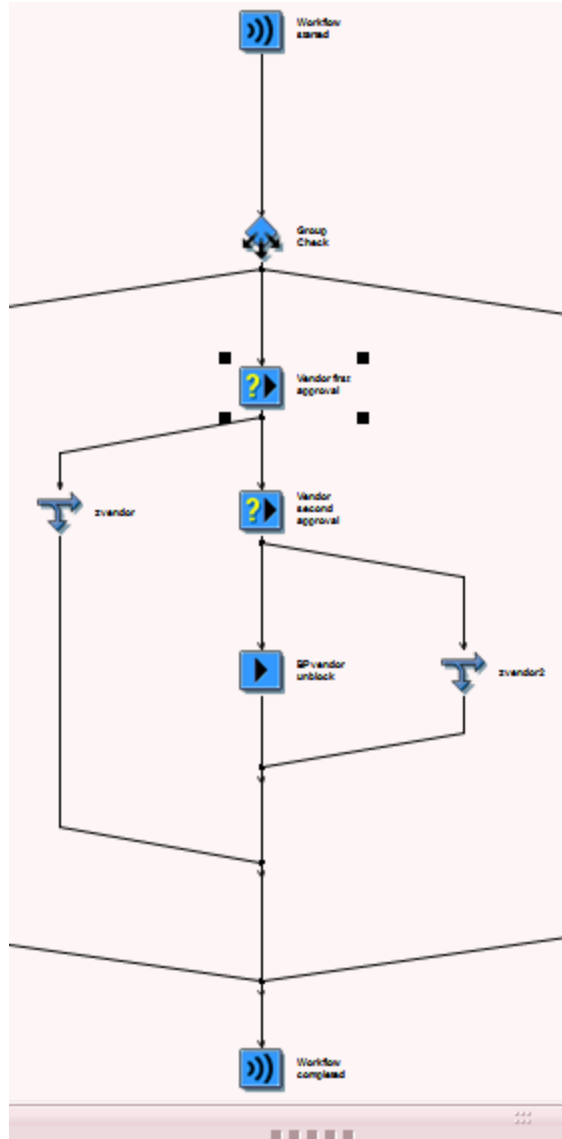
The screenshot shows a workflow configuration window for 'Vendor first approval' with ID 000094. The 'Task' field contains 'TS99100062' and the 'Step Name' is 'Vendor first approval'. A yellow box labeled 'Binding (Exists)' is visible. Under 'Task Properties', there are four options: 'Agent Assignment', 'Background Processing', 'Task Complete', and 'Confirm End of Processing'. A 'Template for Attachm.' field is at the bottom.

- Binding should be done by auto generated. And execute it.

The 'Display Binding for Step' dialog shows two panels. The top panel, 'Binding Workflow -> Step 'Vendor block in BP'', lists a workflow with steps: AEXTENDEDIA, AKTORIKIA, AKTORIKIA, ALIPNRIA, ALIPNRIA, AKTORIKIA, and AKNOTE\_REFERENCEIA. The bottom panel, 'Binding Workflow -< Step 'Vendor block in BP'', lists steps: ADECESSION\_NOTEA, ADECESSION\_NOTEA, AKTORIKIA, AKTORIKIA, ALIPNRIA, and ALIPNRIA.

- So after that here we have 2 Steps as approve ,reject.
  - 1) For Approve
- We have to use 2nd Approval for to finance as shown

## WORKFLOW EXAMPLE



So here.

- We are creating a Task

Standard Task: Display	
Standard task	99100066 ZVENDACC
Name	Vendor second approval
Package	ZHW
Applic. Component	
Basic data   Description   Container   Triggering events   Terminating events   Default rules   SAPPhone	
Name	ZVENDACC
Abkr.	Vendor second approval
Release status	Not defined
Work Item Text	Vendor second approval
Object method	
Object Category	BO BOR Object Type
Object Type	DECISION Object: Decision
Method	Perform check
<input checked="" type="checkbox"/>	Synchronous object method
<input checked="" type="checkbox"/>	Object method with dialog

- For mail .

## WORKFLOW EXAMPLE

**Standard Task: Display**

Standard task: 99100066 ZVENDACC  
 Name: Vendor second approval  
 Package: ZMM Applic. Component

Basic data | Description | Container | Triggering events | Terminating events | Default rules | SAPphone

Text type: 0120 Task description Language: E English

Dear Sir,

Business partner &lifnr& has been approved by purchase HOD its approval request is pending with you, for processing the request kindly login into SAP and follow below steps.

GO to T. code SWBP -> click on inbox -> click on request -> Approve / Reject

Regards,  
 SAP team  
 This is system generated email please do not reply, for any further assistance kindly contact to SAP IT Team.

### ■ For container

Standard task: 99100066 ZVENDACC  
 Name: Vendor second approval  
 Package: ZMM Applic. Component

Basic data | Description | Container | Triggering events | Terminating events

Expression	M. Description	Initial value
▶▶▶ Ad Hoc Objects	Ad Hoc Objects of Workflow Instance	< Not Set >
▶▶▶ Attachments	Attachments of Workflow Instance	< Not Set >
▶▶▶ Agent	Actual Agent of Workflow Activity	< Not Set >
▶▶▶ Grouping Character	Grouping Characteristic for Workflow Instances	< No Instance >
▶▶▶ Step	Step Instance	< No Instance >
▶▶▶ Agents	Result of Agent Determination	< No Instance >
▶▶▶ Second.Method Object	Secondary Method Objects of Workflow Activity	< Not Set >
▶▶▶ _WI_Result	_WI_Result	< Not Set >
▶▶▶ Ext. user decision	Extended user decision	< Not Set >
▶▶▶ Decision_Note	Decision_Note	< No Instance >
▶▶▶ Template for notes	SOFM template for notes	< No Instance >
▶▶▶ _WI_Object_ID	_WI_Object_ID	< No Instance >
▶▶▶ ktokk	ktokk	< Not Set >
▶▶▶ lifnr	lifnr	< Not Set >

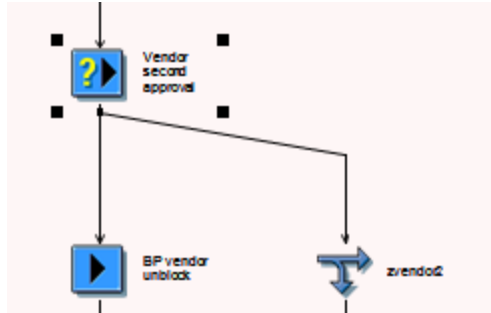
### ■ Assign agent

Task Properties

Agent Assignment  Task Complete

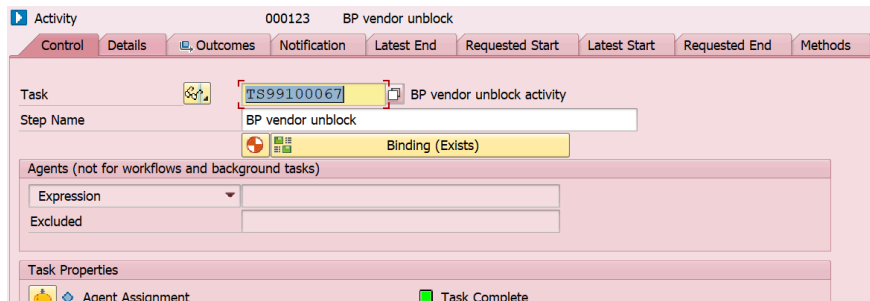
- Binding Should be done or autogenerate binding.
- Save , Check and activate.
- After 2nd approval we need to create a activity.

## WORKFLOW EXAMPLE

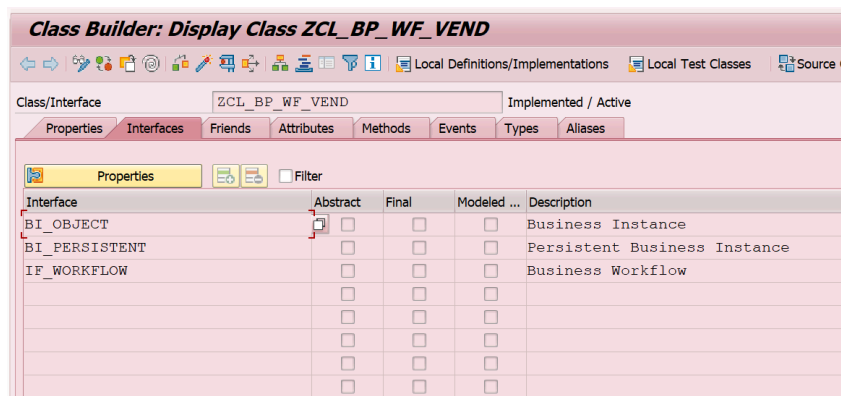


As BP vendor unblock.

- SO in this activity we are unblocking. Through class for this purpose we are creating a class . and define method and in this method we are writing a logic to unblock the BP through calling a change BAPI and adding a logic as BUT000 – XBLCK = ‘ ’. this field should be Blank to unblock.
- In this we are creating a Task as follow.



- Now we are creating a Class and Method as follows  
Go to se24->create a class->zcl\_bp\_wf\_vend  
And in this class we are adding interface as



And in methods we are defining a methods as below and at last we are declaring the method name as vend\_blk.

## WORKFLOW EXAMPLE

**Class Builder: Display Class ZCL\_BP\_WF\_VEND**

Class/Interface: ZCL\_BP\_WF\_VEND Implemented / Active

Properties Interfaces Friends Attributes Methods Events Types Aliases

Parameters Exceptions Sourcecode

Method	Level	Visibility	Me...	Description
BI_PERSISTENT~FIND_BY_LPOR	Static Method	Public		Find Using Local Persistent Object Reference
BI_PERSISTENT~LPOR	Instance Method	Public		Local Persistent Object Reference
BI_PERSISTENT~REFRESH	Instance Method	Public		Flag to Reload from Database
BI_OBJECT~DEFAULT_ATTRIBUTE_VALUE	Instance Method	Public		Value of Default "Attribute" (as Data Reference)
BI_OBJECT~EXECUTE_DEFAULT_METHOD	Instance Method	Public		Execute Default Methods
BI_OBJECT~RELEASE	Instance Method	Public		Release for Garbage Collector to Delete
VEND_BLK	Static Method	Public		Vendor block

So in this method we are writing a logic and calling it in activity task as this.

Basic data Description Container Triggering events Terminating events Default rules

Name

Abbr. ZWF\_VEND

Name BP vendor unblock activity

Release status Not defined

Work Item Text

Work item text BP vendor unblock activity

Object method

Object Category CL ABAP Class

Object Type ZCL\_BP\_WF\_VEND BP vendor block appr

Method VEND\_BLK

Synchronous object method

Execution

2) For Reject.

--- The workflow is set to be stopped. With process control



4.

**3) Employee.**

## WORKFLOW EXAMPLE

In the same manner we are adding a logic and workflow flow.

### 6. How to call workflow in Program.

- To use this workflow we need to call it int Enhancement
- For this workflow we are creating a program
- And submitting that program into the enhancement using submit and return.
- So here we are creating a program as ZBP\_WF\_APRV

```
*-----*
* Report ZBP_WF_APRV
*-----*
*
*-----*
REPORT zbp_wf_aprv.

DATA :
*   LV_TASK TYPE SWR_STRUCT-TASK ,
lv_user      TYPE sofoleutil-sender_id,
lv_container TYPE TABLE OF swr_cont,
lv_attachment TYPE TABLE OF soattlstil,
lv_wrkitem   TYPE swr_struct-workitemid,
it_list      TYPE TABLE OF soattlstil,
lt_cont      TYPE STANDARD TABLE OF swr_cont,
lv_return    TYPE sy-subrc,
lv_name      TYPE sy-uname,
*   lt_cont  TYPE STANDARD TABLE OF swr_cont,
ls_cont      TYPE swr_cont.

PARAMETERS : p_KTOKK TYPE bus_joel_main-creation_group.
*           p_LIFNR TYPE

START-OF-SELECTION.

lv_name = sy-uname.

ls_cont-element = 'KTOKK'.
ls_cont-value = p_ktokk.
APPEND ls_cont TO lt_cont.

* ls_cont-element = 'LIFNR'.
* ls_cont-value = p_LIFNR.
* APPEND ls_cont TO lt_cont.

CALL FUNCTION 'SWU_START_WORKFLOW'
EXPORTING
  task           = 'WS99100014'
  language       = sy-langu
  user           = lv_name
*   SENDER_ID    =
IMPORTING
  return_code    = lv_return
  workitem_id    = lv_wrkitem
TABLES
  container      = lt_cont
*   MESSAGE_LINES =
  attachment_list = it_list.

IF sy-subrc = 0.

ENDIF.
```

## WORKFLOW EXAMPLE

- And calling this program into that class with `vend_bck` method by using `submit` and `return`.

how to debug workflows

ChatGPT said:

Debugging **SAP Workflows** can be tricky, but if you understand the right tools and techniques, it becomes much easier. Let's go step-by-step 🙌

---

### ◆ 1. Basic Understanding

SAP Workflow = combination of

- **Workflow Definition (template)** → created in **transaction SWDD**
- **Runtime Workflow Instance (work item)** → triggered when workflow runs

When debugging, we mainly look at:

- The **workflow container values**
  - The **steps executed**
  - The **agent determination**
  - The **task / method execution (ABAP code)**
- 

### ◆ 2. Check Workflow Log

Transaction: **SWIA** or **SWI1**

Steps:

## WORKFLOW EXAMPLE

1. Open **SWIA**
2. Enter the workflow ID or task ID
3. Select your workflow instance → **Workflow Log** (button or menu)
4. In the log, you can see:
  - Each step's status (executed, ready, error)
  - Agents determined
  - Container values (via right-click → *Display container*)

● If you see “Error” in log → double-click → system shows message text and possible cause.

---

### ◆ 3. Activate Workflow Debugging (Global)

If you want to debug **workflow execution logic**, use:

**Transaction:** **SWUS** (to test workflows manually)

or

**Transaction:** **SWU9** (for debugging)

**Steps in SWU9:**

1. Enter workflow template ID (WS...)
2. Click **Execute with debugging**
3. The workflow will run in debug mode, pausing at:
  - Task execution
  - Rule resolution
  - Binding steps

## ◆ 4. Debug Task's ABAP Code

Most workflow steps call **methods** from **Business Object Repository (BOR)** or **ABAP Classes**.

### Case 1 – BOR Method:

- Use transaction **SWO1**
- Find the method (under object)
- Note the **Function Module name** (often **SWW\_\***)
- Set breakpoint in SE37 for that FM

### Case 2 – Class Method:

- Find method name in task (via **PFTC** or SWDD)
- Go to **SE24**
- Set breakpoint in method implementation

When workflow runs and reaches that step, the debugger will stop there.

---

## ◆ 5. Debug Rule Resolution

If agent determination (rule) is failing:

**Transaction:** **PFAC**

1. Open the rule
2. Run test with container values
3. If rule calls a **Function Module**, set breakpoint in that FM

## WORKFLOW EXAMPLE

You can also debug directly in workflow runtime log when it calls rule container binding.

---

### ◆ 6. Debug Work Item Execution

When a user executes a work item and you want to debug:

1. Go to **SWI1** → select work item
  2. Note the **task number (TS...)**
  3. Go to **PFTC** → **Task** → get method/class details
  4. Set breakpoint and then execute work item (via **SBWP** or **SWI1** → “Execute Work Item”)
- 

### ◆ 7. Workflow Container Debugging

In the workflow log:

- Right-click → “Display Container”
- Check all variable bindings
- Verify data flows correctly between workflow → task → method

If wrong, fix **Binding** between containers (SWDD → double-click arrow between steps).

---

### ◆ 8. Technical Workflow Tools

Tr an sa	Purpose
----------------	---------

## WORKFLOW EXAMPLE

cti  
on

S  
W  
I1      Display workflow  
instances

S  
W  
IA      Admin view of  
workflows

S  
W  
U  
D      Workflow diagnosis

S  
W  
U  
9      Test workflow with  
debugging

S  
W  
EL  
S  
/  
S  
W  
EL      Event trace (to check  
event trigger)

S  
W  
O  
1      Business Object  
Repository

PF  
TC      Task / Template  
definition

## WORKFLOW EXAMPLE

S User authorization  
U for WF agent  
01

S Delete workflows  
W (cleanup)  
W  
L

In **SAP ABAP Workflow**, the concept you're referring to — where a work item (approval task) is **automatically forwarded to a manager if the intended agent is not available** — is known as **"Substitution"** or **"Agent Determination / Rule Resolution"** depending on how you implement it.

Let's break it down clearly 🙌

---

### 1. Substitution (Primary Concept)

This is the **main SAP workflow concept** for handling unavailability.

#### ◆ What It Means

If an agent (approver) is **not available** (e.g. on leave), SAP Workflow can automatically **delegate their work items to a substitute**, often the manager or a designated backup.

#### ◆ How It Works

- Users maintain **substitution rules** in their SAP Inbox (Business Workplace) using transaction code **SBWP** or via Fiori My Inbox app.
- Substitution can be:
  - **Active:** Work items are automatically routed to the substitute.
  - **Passive:** Substitute can view and act on behalf of the original user manually.

## WORKFLOW EXAMPLE

- Workflow runtime checks these rules and assigns or forwards the work item accordingly.

### ◆ Example

Agent **A123** is out of office → A substitution rule exists that routes their work to their manager **M456**. So when the workflow runs, the **substitute (manager)** receives the work item instead.

---

## 2. Agent Determination (Rule-Based)

If you want to handle this dynamically in the workflow itself (not via user-maintained substitution):

### ◆ You can define a Rule (via transaction **PFAC**)

A rule with a **function module** determines the responsible agent at runtime.

For example:

- The function module checks:
  1. If the default agent is available.
  2. If not, fetches their manager from Org Management (HRP1001 relationship A003 – “manages”).
  3. Returns that manager as the responsible agent.

### ◆ This is called:

“Rule-Based Agent Determination” or “Responsibility Rule”

### ◆ Example Workflow Step Configuration:

- Task → "Maintain Agents" → Choose “Rule” → Enter rule number (e.g. **90000012**).
- Rule uses a function module (e.g. **Z\_DETERMINE\_APPROVER**) that handles availability and escalation.

### 3. Organizational Management Integration

You can also use **organizational objects** (Position → Holder → Manager relationship).

SAP Workflow can automatically **escalate to the manager** using the organizational hierarchy (Org Units and Positions).

---

#### Summary

Concept	Purpose	How it helps
<b>Substitution</b>	Handles temporary user unavailability	Automatically forwards work items to a substitute (manager)
<b>Rule-Based Agent Determination</b>	Dynamically finds an approver	Custom logic can assign to manager if approver not found/available
<b>Org Management Relationships</b>	Uses org structure to find manager	Escalates to manager automatically using position hierarchy

## WORKFLOW EXAMPLE