**What is SAP workflow and Why we need it ?**

* SAP Workflow ensures " the right work is brought in the right sequence at the right time to the right people".
* It is a tool designed to facilitate and automate business processes that require tasks to be performed by people.
* Each step of a business transaction can be easily monitored, and processes are completed from the beginning to the end.
* Workflow allows process owners to keep an eye on deadlines, provides statistics on the length of time to complete work processes, determine the workload with regard to individual employees and save processing time.

**Classifications,**

1. Triggering Events
2. Bindings & Containers
3. Agents & Agent Determinations
4. Task Creations
5. Deadline Monitoring
6. Step Types
7. Debugging

Transaction Codes :

Workflow Customising : SWU3

Workflow Builder : SWDD

Workflow Tasks : PFTC

Workflow Rule : PFAC

Business workplace : SBWP

Business Objects : SWO1

Workflow Logs : SWIA

Event Trace : SWELS

Display Event trace : SWEL

Event type Linkages : SWE2

Workflow restart after error : SWPR

Triggering Events,

* By calling the function module in the programs
* Events could be triggered when change documents are written
* Using Message Control

## **Using Check Function Module in SAP Workflow**

Following is a custom function module created for our demo purpose, which will check the conditions before triggering the workflow.

FUNCTION ZGEE\_SALES\_ORG.

\*"----------------------------------------------------------------

\*"\*"Local Interface:

\*" IMPORTING

\*" VALUE(OBJTYPE) TYPE SWETYPECOU-OBJTYPEx

\*" VALUE(OBJKEY) TYPE SWEINSTCOU-OBJKEY

\*" VALUE(EVENT) TYPE SWETYPECOU-EVENT

\*" VALUE(RECTYPE) TYPE SWETYPECOU-RECTYPE

\*" EXPORTING

\*" REFERENCE(RESULT) TYPE C

\*" TABLES

\*" EVENT\_CONTAINER STRUCTURE SWCONT

\*" EXCEPTIONS

\*" NOT\_TRIGGERED

\*"----------------------------------------------------------------

TABLES: VBAK.

DATA VKORG LIKE VBAK-VKORG.

SELECT SINGLE VKORG INTO VKORG

FROM VBAK WHERE VBELN = OBJKEY.

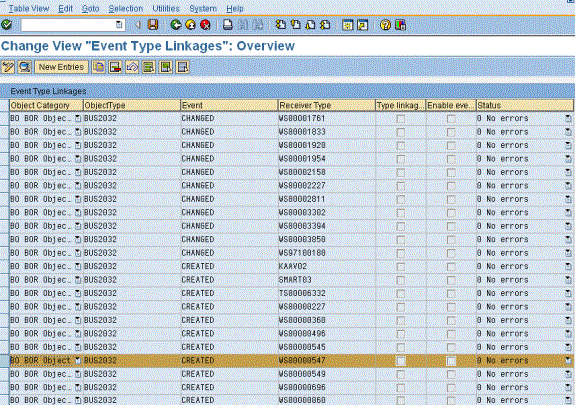
IF VKORG = '3000'.

RAISE NOT\_TRIGGERED.

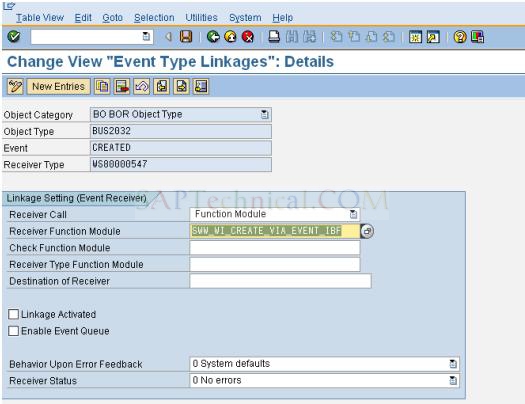
ENDIF.

ENDFUNCTION.

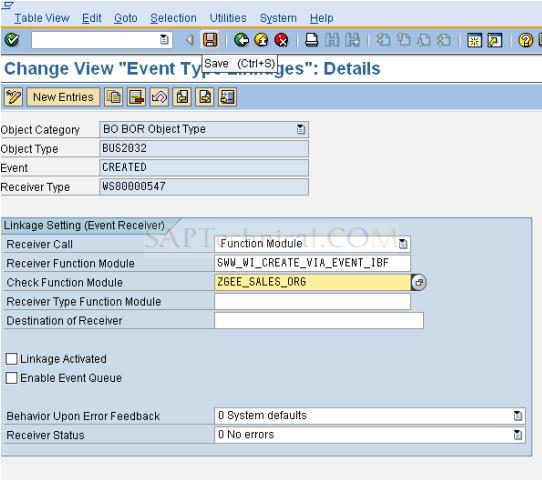
Now go to transaction SWE2 and insert your object type, event, and the receiver type.



  Double click on your entry, following screen appears:



Provide the name of the customized function module name created earlier in the check function module field.



Now workflow is triggered only when the sales order is created with the sales organization 3000.

Binding & Containers

1. Workflow container
2. Task container
3. Method container
4. Rule container
5. Event container

**Possible Binding Matrix:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Container Type** | **Workflow** | **Task** | **Rule** | **Event** | **Method** |
| **Workflow** | No | Yes | Yes | Yes | No |
| **Task** | Yes | No | Yes | No | Yes |
| **Rule** | **Yes** | No | No | No | No |
| **Event** | Yes | Yes | No | No | No |
| **Method** | No | Yes | No | No | No |

Agents & Agent Determinations

There are 3 types of Agents used in workflows, all are used in conjunction to find the selected agents for a work item.

[](https://blogs.sap.com/wp-content/uploads/2015/10/agents_821833.png)

**Possible Agents**are assigned to the task; this is the pool of agents that will be considered later to become an agent of the work item.

**Responsible Agents**are the agents that are returned from workflow agent definition or the default role of the task. A Responsible agent can only process a work item if they are a possible agent and if they are not an excluded agent.

**Excluded agents**cannot be an agent of the task even if they are listed as a responsible and/or selected agent. Excluded agents are defined at the workflow level.

**A user may find themselves in multiple groups:**

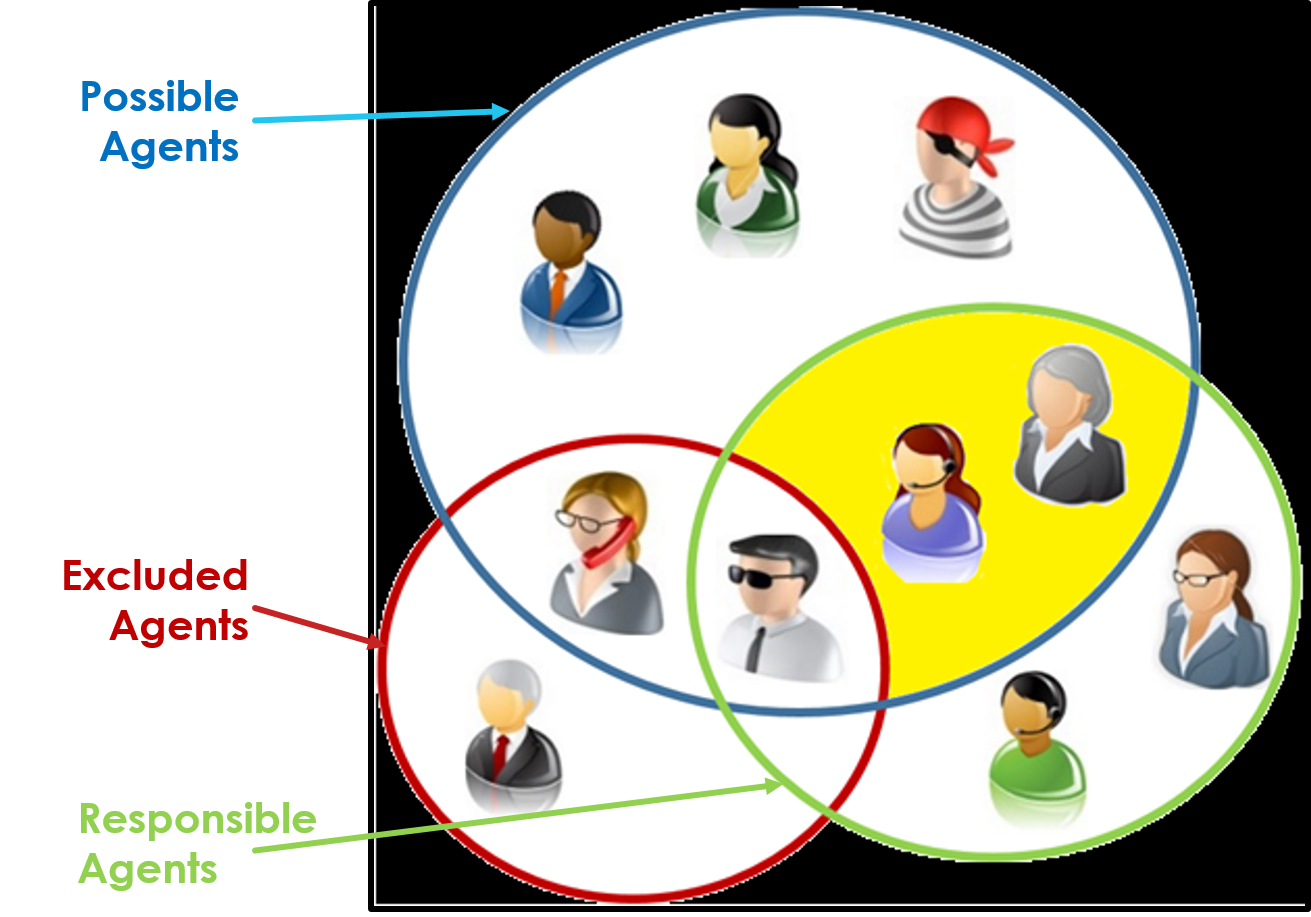
[](https://blogs.sap.com/wp-content/uploads/2015/10/woman_821868.jpg)

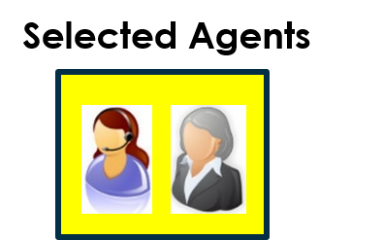
This woman is in the pool of Possible Agents but she is not one of the Responsible Agents so she will not be able to process the work item.

[](https://blogs.sap.com/wp-content/uploads/2015/10/man_821869.jpg)

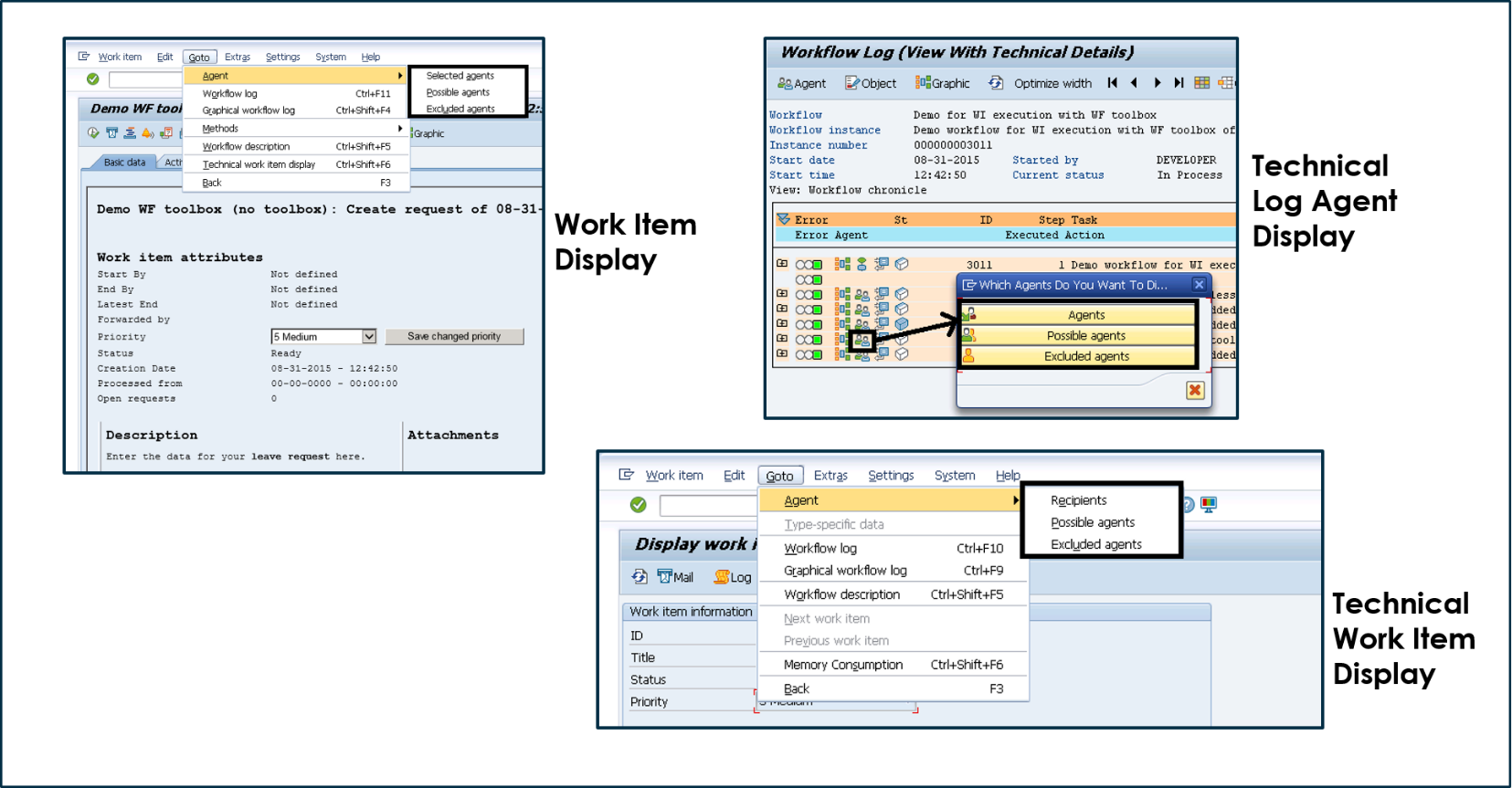
This man is in the pool of Possible Agents and he is in the pool of Responsible Agents; however, he is also on the list of excluded agents so he will not be able to process the work item.

The **Selected Agents** of the work item will be the ones that are at the intersection of the Possible Agents and Responsible Agents and are NOT included in the intersection of Excluded Agents

[](https://blogs.sap.com/wp-content/uploads/2015/10/venn_821870.png)

[](https://blogs.sap.com/wp-content/uploads/2015/10/selected_agents_821872.png)

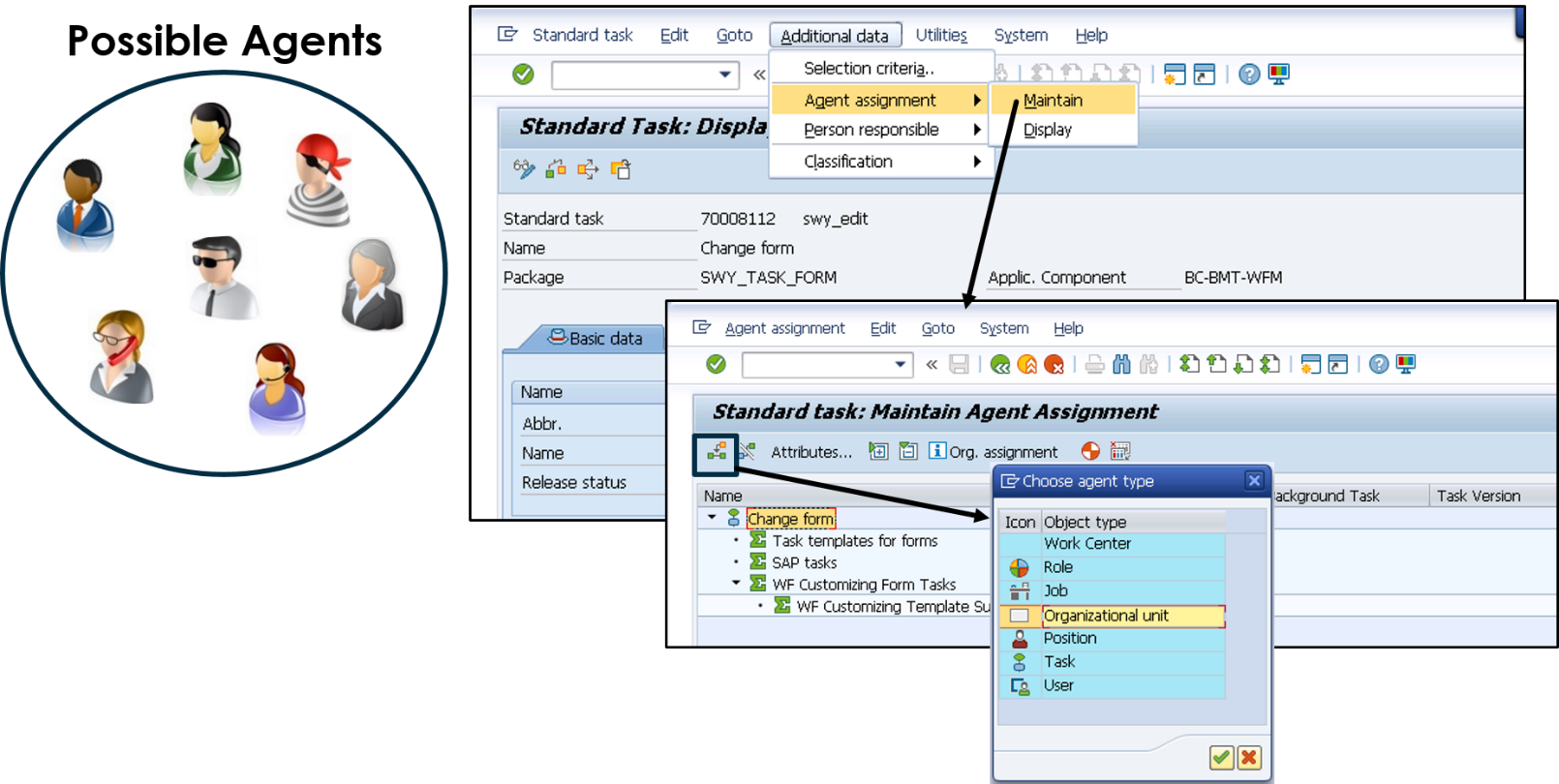
**Workflow Log Display of Agents**

[](https://blogs.sap.com/wp-content/uploads/2015/10/showing_agents_821979.png)

The workflow environment is not very consistent with how it names the Selected Agents; ‘Selected Agents’ or just ‘Agents’ or ‘Recipients’    These are all the same.   Note, you are not offered the opportunity to view the Responsible Agents.

**Definition of Possible Agents**

The possible agent is defined under ‘Additional data’ from the task definition.   You may define the possible agents as one of the following or of a combination of multiple agent types.

**[](https://blogs.sap.com/wp-content/uploads/2015/10/define_possible_agents_821990.png)**

The task agent assignment may be any one of the following:

1) Role

2) Workflow Task

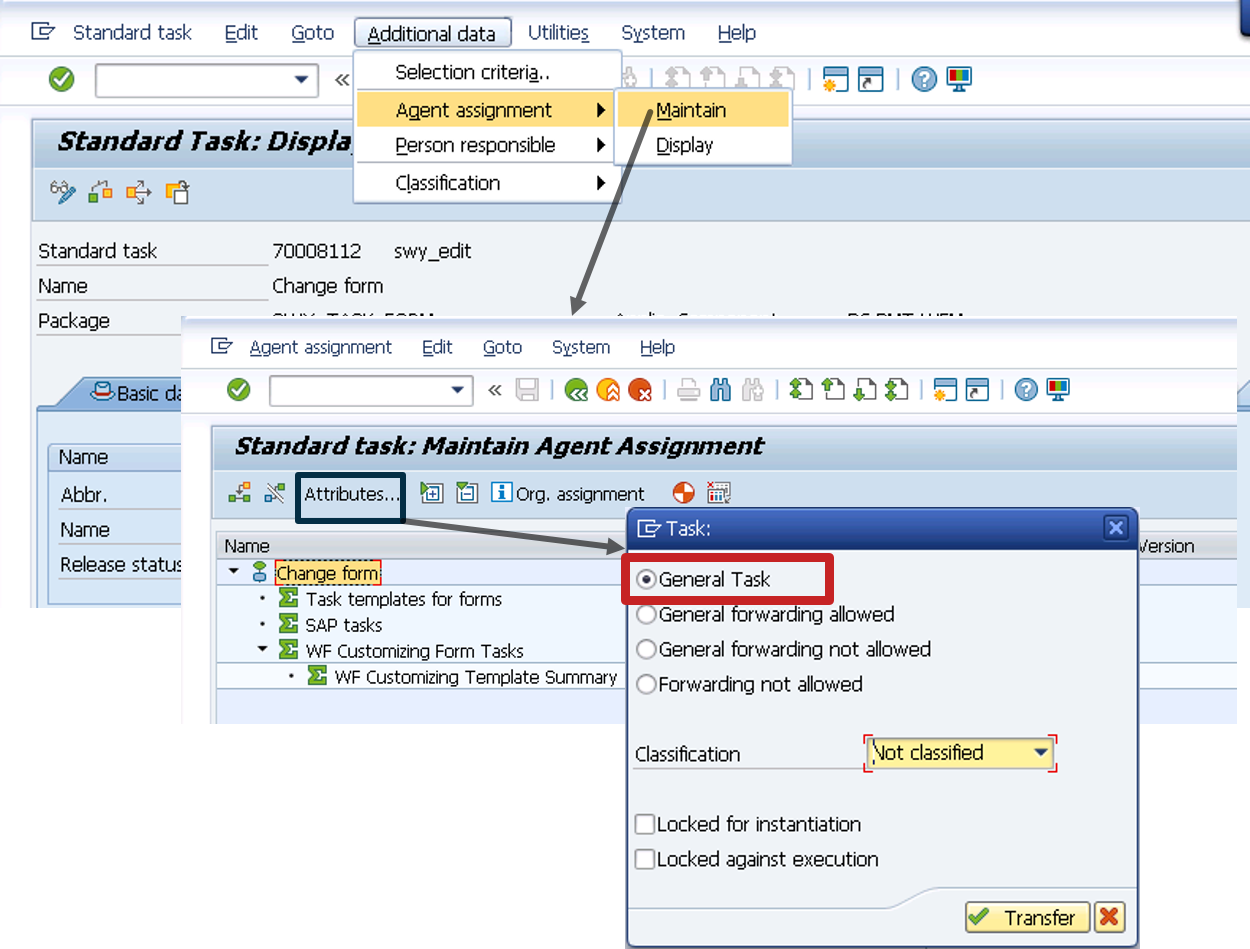
3) User

4) Organizational Unit

* + Organizational Object
  + Job
  + Work Center
  + Position

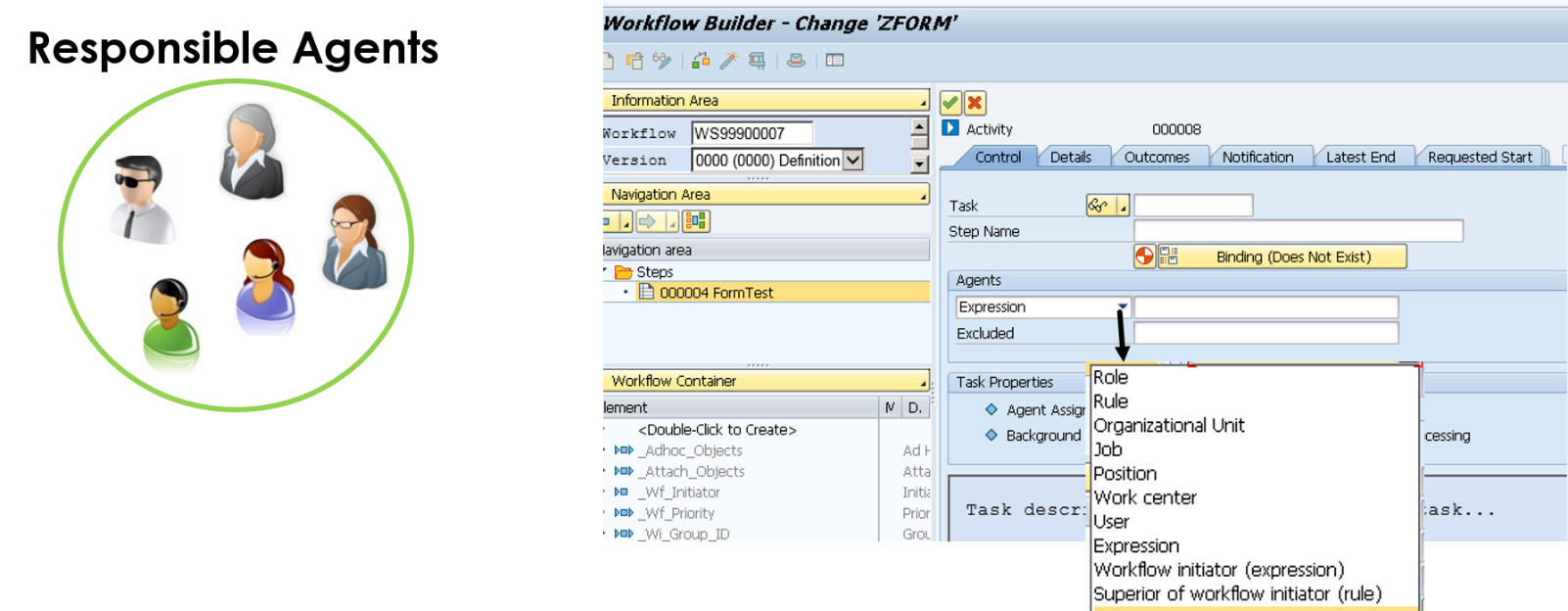
Alternatively, the entire SAP population can be a possible agent if the “General Task” option is selected.

* This can be**DANGEROUS** if no selected agent is found; **EVERYONE** will receive a work item if the rule is not set to error out if no one is found, or responsible agent is not a rule and found empty.
* This is Wasteful of resources in SAP Business Workplace.

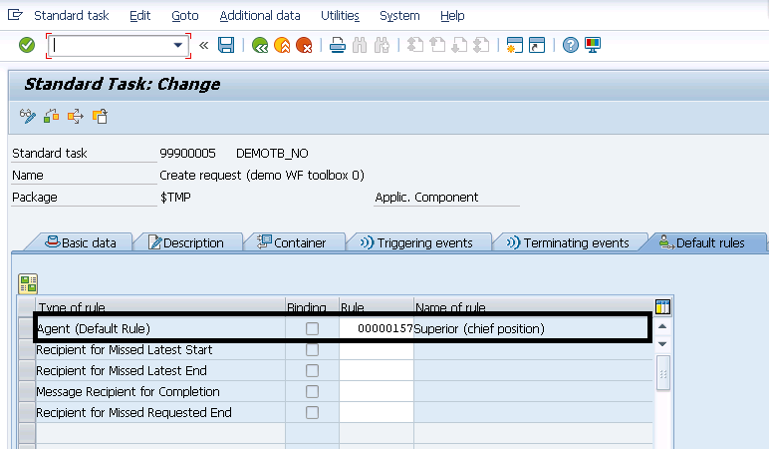
[](https://blogs.sap.com/wp-content/uploads/2015/10/general_task_821992.png)

**Definition of Responsible Agents**

The Responsible Agents  can be defined in the workflow definition on the step that points to the workflow task. There are more definition options than what was found for the Possible Agents on the Task.   In addition to these, when defining the selected agents you can use all of the options available for a **Rule**, an **Expression** (workflow container element), the**Workflow Initiator**, or the **Superior of the Workflow Initiator.**

**[](https://blogs.sap.com/wp-content/uploads/2015/10/define_responsible_agent_821993.png)**

* Another place you can define the Responsible Agents  is by defining a default rule in the task definition on the ‘Default Rule’ tab. This default rule will be used if a Role is defined for finding the Responsible Agents is in the workflow definition AND no agents are found AND the rule is not set to error if no agents are found.
* The default rule is necessary for single step tasks (because there is no workflow definition).
* The default rule is seldom utilized.
* Only have the ability to add a rule, no organizational objects or user assignment.

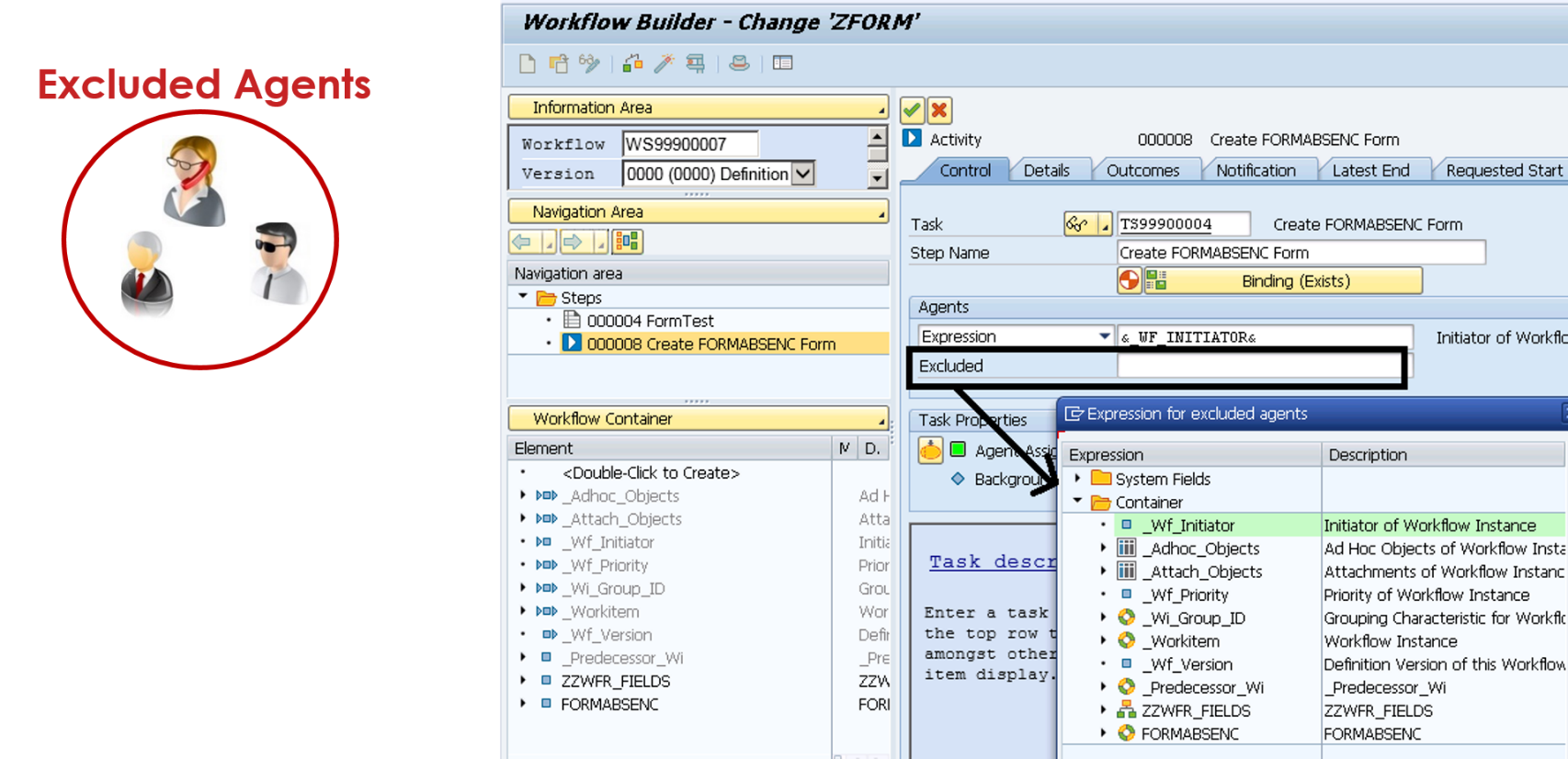
[](https://blogs.sap.com/wp-content/uploads/2015/10/default_rule_821880.png)

**Definition of Excluded Agents**

Excluded Agents are not allowed to process a work item even if they are listed as a possible agent and a responsible agent.

**Examples of Excluded Agents:**

* 2 approvals are required from a pool of managers; once a manager supplies their approval, they will be listed as an excluded agent so they cannot also supply the second approval.
* A user in a shared services center has the ability to create and approve requests for payment; but can never create and approve the same request for payment.   Once the user creates the request for payment, they are now listed as an Excluded Agent so they will not be able to approve their own request.

[](https://blogs.sap.com/wp-content/uploads/2015/10/define_excluded_agent_821980.png)

Excluded Agents are defined in the workflow definition on the task activity step.  Excluded Agents can only be an expression; meaning they have to be defined as a workflow container element.

# **Maintenance of Agent Assignment**

## Use

This function assigns possible agents to a task.

**Note**

No possible agents are assigned initially to an SAP task or an SAP workflow. You can perform this assignment in Customizing. You can change and extend these assignments at a later date as well.

## Integration

**Establishing recipients at runtime**

The system establishes the recipients of a work item at runtime. The recipients are derived from the intersection of possible agents and

responsible agents barring

excluded agents.

**Note**

If this intersection is empty, no users receive the work item. In [Workflow Runtime Administration](https://help.sap.com/viewer/4400bdc8dd4648a5a2e5c1c8e05198d7/7.4.19/en-US/4f257c69e85a18c3e10000000a421937.html) a function is available to determine such work items..

You specify the responsible agents for a step in the step definition. You can specify not only a system user, but also a role or an object of Organizational Management such as a position, a job, an organizational unit, or a work center. Alternatively, you can define a container element that contains the object of Organizational Management at runtime or a rule for dynamic agent determination.

You can define agents excluded from processing within the step definition.

The user who actually executes the work item at runtime is the actual agent.

The possible agents for a task must be maintained so that the recipients can be established at runtime.

**Using a default rule**

You can specify a default rule when defining a task so that the agents are established dynamically when this task is executed.

The default rule is evaluated if the task is executed alone. The default rule takes on the function of the responsible agents in this case.

The default rule is also evaluated if the task is used in a workflow definition and no responsible agents are specified in the step definition in question.

## Features

**General**

Task

The possible agents of a task are all of the users who are organizationally responsible and authorized to execute the task at runtime. Only they can receive a work item for this task in their Business Workplaces. Starting the task in dialog outside a workflow is included in execution.

**Note**

You must assign possible agents to a task that requires a dialog for the task to be executed at runtime.

You can create several assignments for a task and also use various types of agent doing so.

**Example**

The task "Check invoice" is linked with the following possible agents.

Organizational unit Purchasing

Organizational unit Internal auditing

Job Accountant

You do not have to assign possible agents to tasks designated as background tasks. You must assign possible agents to those background tasks that are also to be started as single steps in dialog.

Workflow (multistep task)

The possible agents of a workflow are those users who are authorized to start the workflow manually. You can perform the assignment of possible agents in the basic data of a workflow definition or in the relevant multistep task.

You only have to assign possible agents to a workflow if the workflow is to be started directly by users. If the workflow is only to be started by events, do not assign any possible agents to the workflow.

Task groups

If you assign possible agents to a task group, this assignment in passed on to all tasks, workflows and task groups contained in the task group.

Check whether the assignment defined here really is to apply to **all** task groups, tasks, and workflows contained.

**Basic settings**

Task

Choose the basic property from the following alternatives:

* **General Task**

If you define a task as a general task, all users can execute the task. This is useful if the task is used in a workflow and you only want to define the recipients in the step definition. A recipient can forward associated work items to all users.

**Note**

Work items whose tasks are defined as general tasks and for which no responsible agents or default rules are defined are offered to all users of the SAP System for execution in their Business Workplaces.

* **General forwarding allowed**

A work item that represents a task with this property can be forwarded by one of its recipients to all users, even if they are not possible agents of the task.

* **General Forwarding not Allowed**

A work item that represents a task with this property can be forwarded by one of its recipients only to the possible agents of the task.

* **Forwarding not allowed**

A work item that represents a task with this property cannot be forwarded by one of its recipients.

Workflow

With a workflow (multistep task) you can only decide whether it is a general task or not.

## Activities

To process the basic properties of a task, choose **Additional Data**  **Classification**  **Change** in task maintenance.

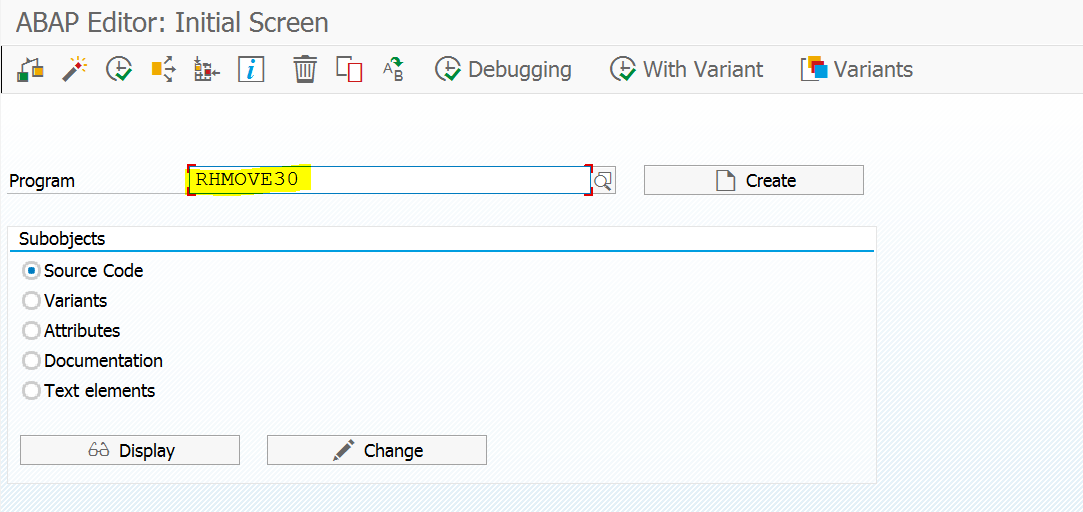
To assign a task you have defined to its possible agents or process its basic properties, choose **Additional Data**  **Agent Assignment**  **Maintain** and select the task.

Select **Attributes...** to define the basic properties of the task.

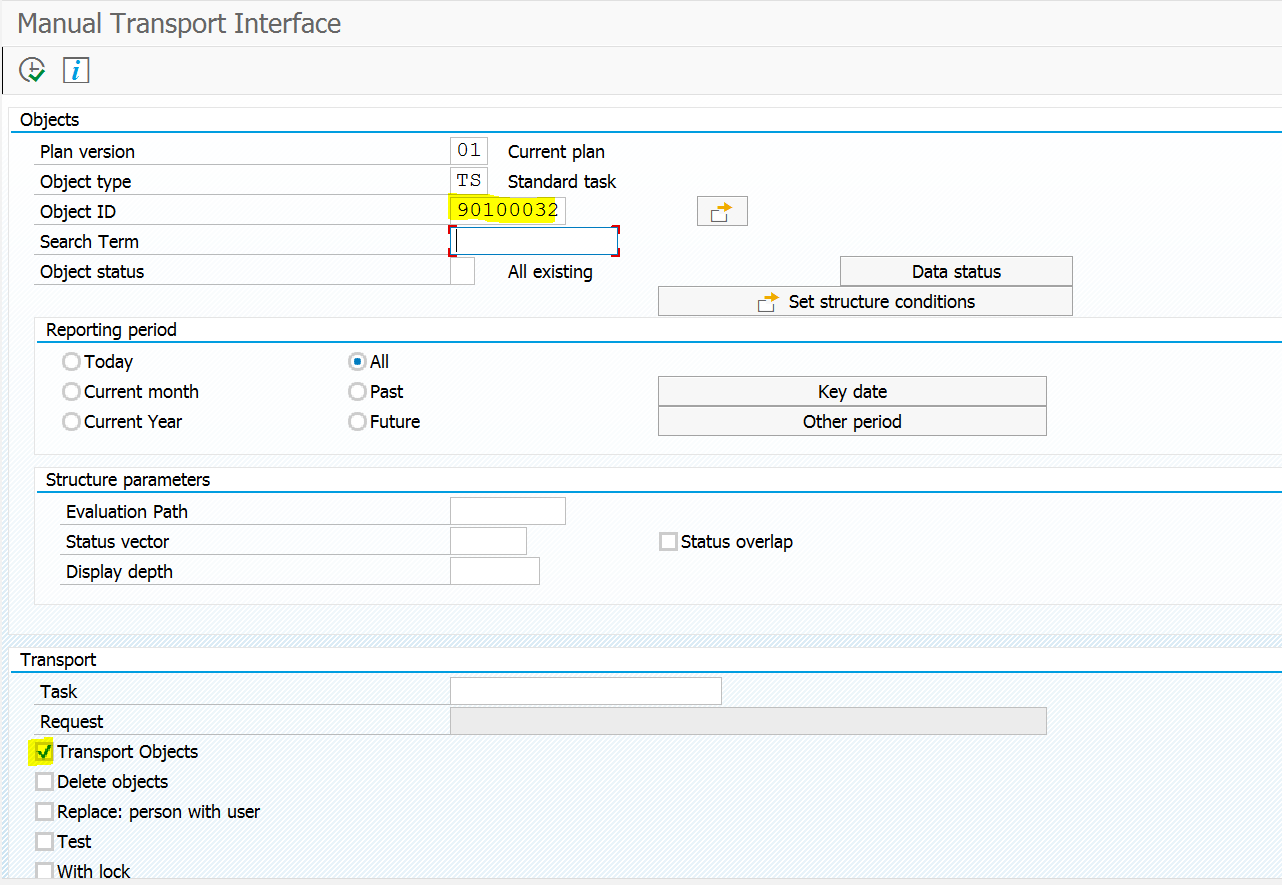
Choose **Agent Assignment**  **Create...** to assign the possible agents.

I have developed workflows and faced general task issues for agent assignment in quality and production systems so i want to write blog that how to add general tasks into custom transport in detailed.

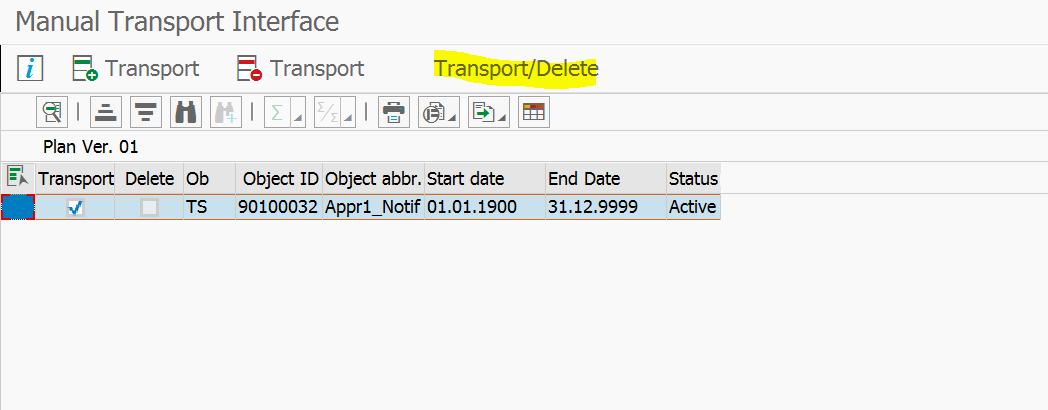
* Execute the report **RHMOVE30** in SE38.



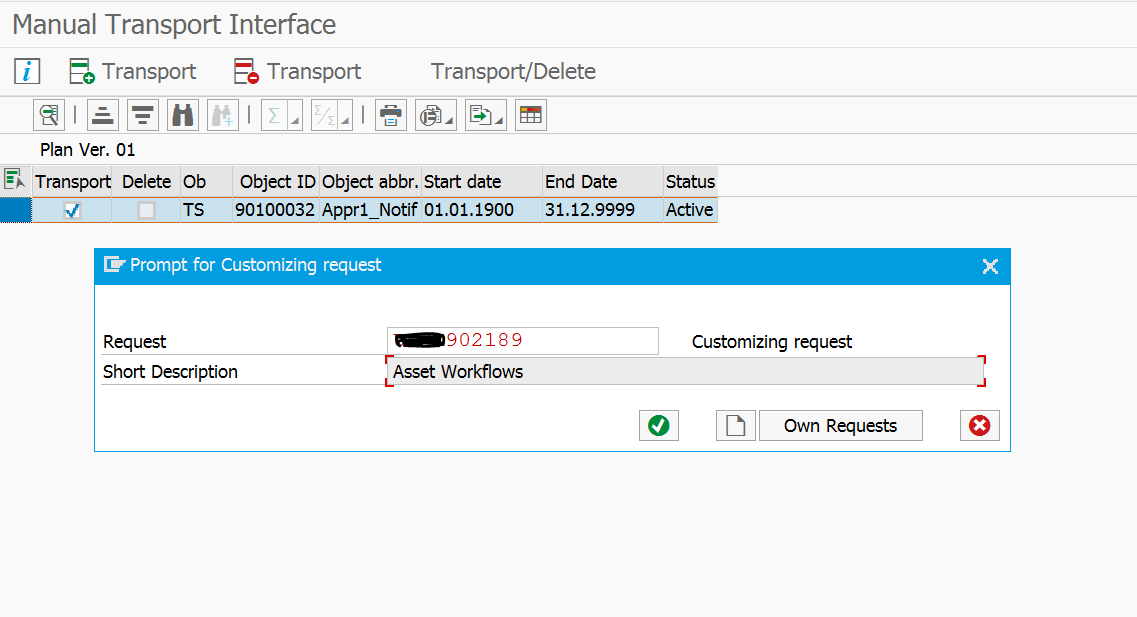
Enter task number which you want to transport agent details, check transport objects check box and click on execute button.



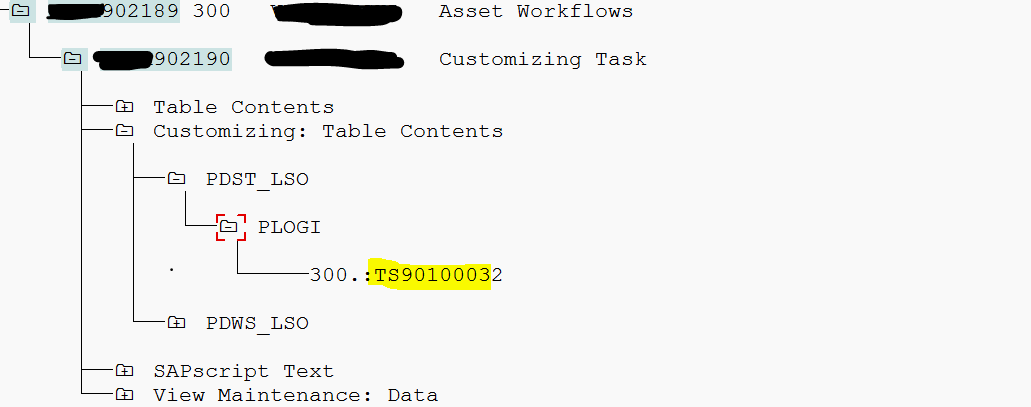
* In the next screen select the entries and click on Transport/Delete button to add it into transport.



Next screen it display custom transport pop up screen and add it in respective transport.



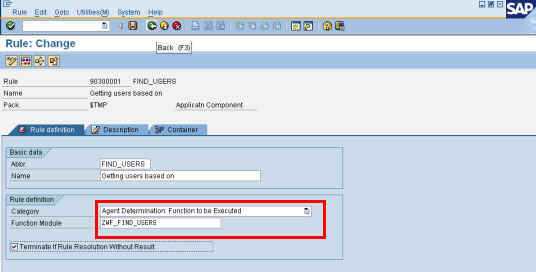
Successfully added in custom transport and release the transport and import into quality and production.



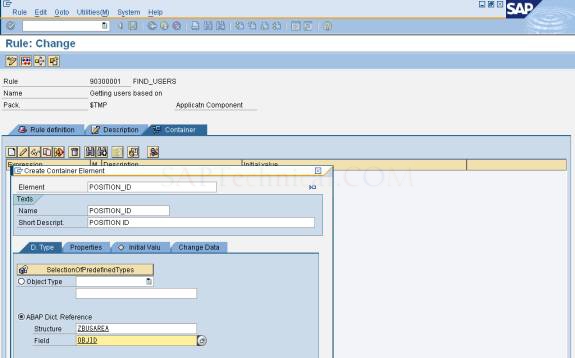
Some common macros defined in include <CNTN01> for working with Containers

|  |  |
| --- | --- |
| **Functionality** | **Macro** |
| To write a single line variable in Container | SWC\_SET\_ELEMENT |
| To read a single line variable from Container | SWC\_GET\_ELEMENT |
| To write a multiline variable or internal table in Container | SWC\_SET\_TABLE |
| To read a multiline variable or internal table from Container | SWC\_GET\_TABLE |
| To clear the container | SWC\_CLEAR\_CONTAINER |

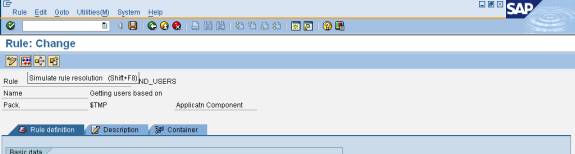
**Creating a rule using the PFAC Transaction**  
  
Assign the function module ZWF\_FIND\_USERS in the "Rule Definition Tab"



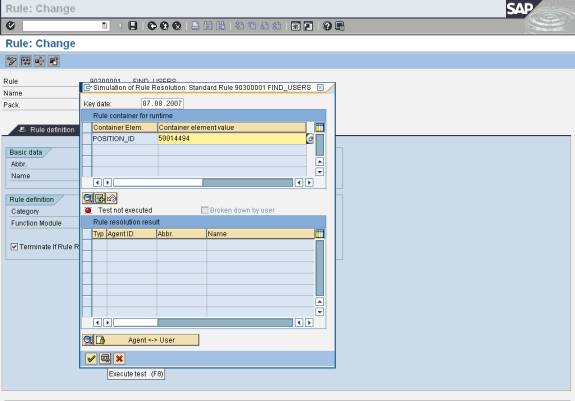
Go to the "Container Tab" and create a container element for the “Position id” which will be passed to the function module



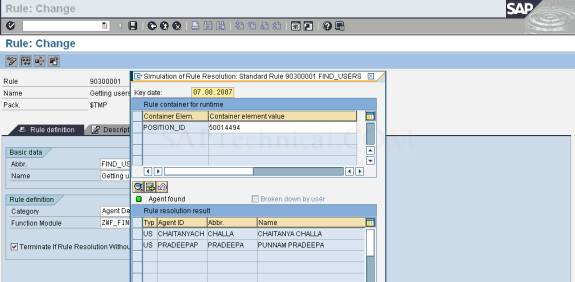
 Our rule is created now. You can test the rule within the PFAC transaction by clicking the "Simulation" button on the application toolbar. Now this rule is ready to be used in any workflow according to your requirement via the rule container of the workflow.



 Enter the Position Id, which is created in (PPOCW)



 It displays Asset Analysts (users) in the “Rule resolution result”



 Now you can use this rule in any of your workflow definitions.

# What is Deadline Monitoring in SAP Workflow?

SAP Workflow, as the name suggests, is responsible to ensure that Work in the SAP System Flows in a consistent, systematic and timely manner.

Every business process consists of doable actions (Work) that has to be completed. For example: Approving Employee Time-sheet, Checking if a new hire form is filled completely, Checking/ Approving Purchase Requisition etc. All these are doable actions.

Some of this work can be done in the background without any manual intervention where as other kind of work needs active participation by an actual person/user ( also known as agent ) .

For example : If we take leave approval for instance, then the actual approving of the leave cannot be done automatically. As per the business process a manager needs to approve an employee’s leave request. However some activities like updating the available leave balance of the employee when the leave is approved does not need manual intervention and can be done in the background.

Background activities are usually done instantaneously by a system user id WF\_BATCH. However activities which need to be done by a user like a manager etc usually have to be done in the foreground.

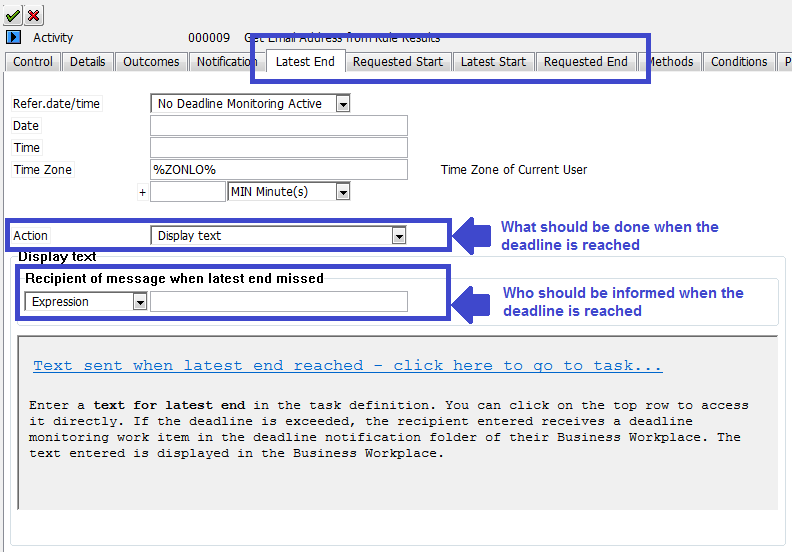
Some of these activities or actions are non critical activities and hence usually need not be completed on an urgent basis. For example approval of employee performance appraisals.

However other activities like a sales order approval, timesheet approval etc are time critical activities that have to be completed within a stipulated time period.

For such cases SAP gives us a functionality to apply Deadlines and Escalations to our workflow tasks.

So, if you are designing a workflow and the requirement is that a particular Task, for example approval of employee’s leave request should be completed by the manager within a given time frame then we can set a deadline for the task. Once the deadline is reached, we can then apply further logic using escalation functionality in order to escalate the approval task(work) to their superior manager or any other agent/user as required by the process.

* Requested Start
* Latest Start
* Requested End
* Latest End



**Requested Start :** This is Earliest point in time at which the work item can be executed.

Example : Even if the employee submits their travel request to the manager, the company policy requires that a 2 days waiting period is required before the manager can actually approve the request.

A dialog work item (approver task) appears in the Business Workplace of the manager when the requested start is reached if the work item is already available for processing. If the dialog work item is not created until after the requested start, it appears immediately in the Business Workplaces of its recipients.

With background work items, execution starts when the requested start is reached at the earliest.

**Latest Start :** Deadline by which one of the recipients of a work item must have started to process it.

Example : If the employee submits their performance appraisal then it is required that the manager should START reviewing the appraisal document within a certain period of time.

**Requested End** : Deadline by which the processing of a work item should be terminated.

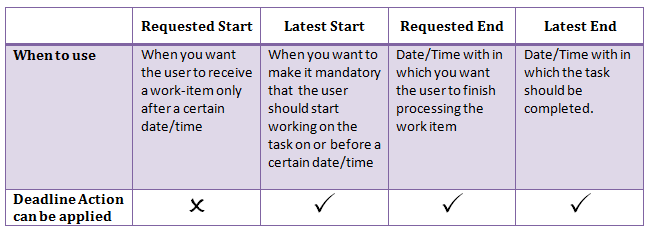
Example : In the above case of performance appraisal, the appraisal approval by manager should complete within a given time period.

**Latest End :**Latest point in time for the processing of a work item to end.

As I have noticed – in all practical cases Requested End and Latest End behave the same way, just that if a user opens a work item ( Reserves the work item ) then the Requested End date deadline will not kick in. Only the Latest End date will be valid. Hence usually its is advisable to set Latest End greater than Requested End.

Example : In the above case of performance appraisal, the appraisal approval by manager should completed within a given time period greater that requested end otherwise the appraisal will escalate to higher manager.

*The processing of a work item ends when it assumes either the status logically deleted or the status completed.*

[](http://www.beginners-sap.com/wp-content/uploads/2015/11/Deadline-Matrix1.png)

In the next part of the blog we will see an implementation of deadline monitoring in a workflow.

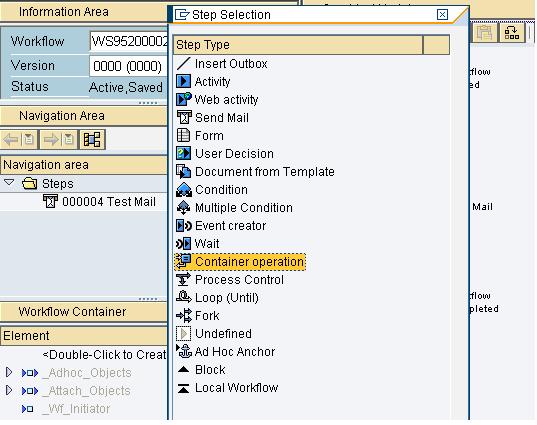
# **Brief Overview of Workflow Step-Types**

This WIKI will help the readers to know more about the Step-Types that are used in Workflows and a wide Implementation of these Step-Types. Apart from sending mails, Workflows can handle a lot many affairs. This will clear the air that Workflows are generally used for sending mails.

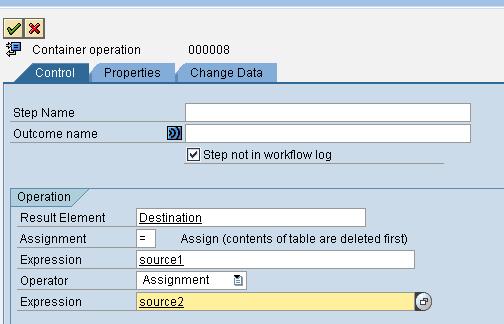
Workflows can work like Algorithms in the background as well as Foreground to make the task appear in a Flow, Basically like the Flow of Work. Various Functionalities can be Implemented like, Flow based on calculations, Flow based on Conditions, Flow based on Conditional Events, Uploading Documents, Filling up Forms etc.

## **Container Operation**

                Container Operation is a Step-Type that is used for calculations with the help of Workflow Containers.



This Step-Type is used to derive various Arithmetic calculations from the Workflow Elements. To implement this Step-Type, we should give an Arithmetic Condition, based on whose result the WF moves further.



The various Arithmetic functions available are:

       Multiplication

       Addition

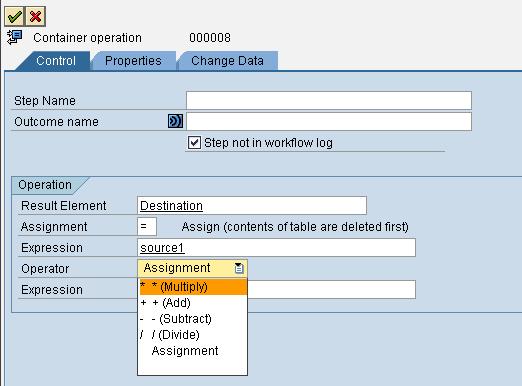
       Subtraction

       Division

One should create 3 WF Containers, for example:-

**C = A + B (where all these 3 should be WF Containers)**

**Result Element = Source1 [(Operator)] Source2**



Note:

       Here the Source and the Target Elements should be the Workflow Containers.

       The Outcome of this Step-Type would be Numeric/Decimal always with 1 value.

       This Arithmetic Operations take place between the values of the Workflow Containers.

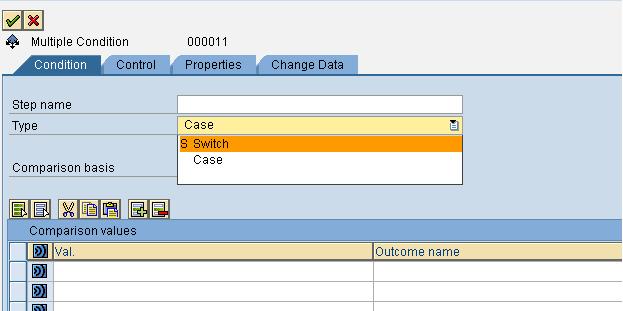
    These WF Containers can have Dynamic values as well as Hard-Coded values from the WF Container.

## **Multiple Conditions**

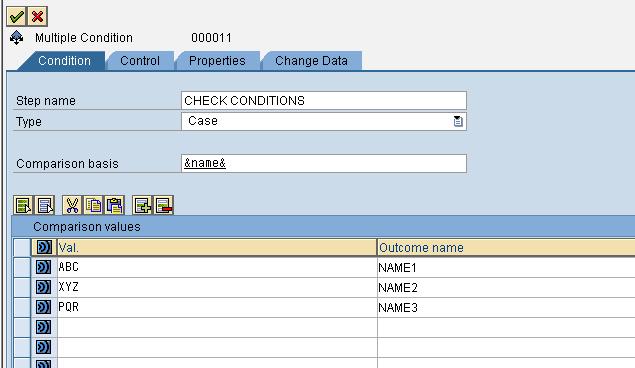
    Multiple Conditions is a case that is implemented where we have to check so many conditions at 1 time.  
Multiple Conditions can be implemented in 2 ways:-

       Case

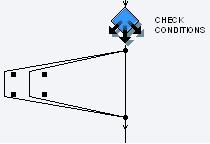
       Switch



If the condition is like this:-  
                    
**If name = 'ABC'**  
**Do this.**  
\*                       \* **Else if name = 'XYZ'**  
**Do this.**  
\*                       \* **Else if name = 'PQR'**  
**Do this.**  
\*                       \* **Else**  
**Do this**  
In this example ()name should always be defined as the Workflow Container Element. The values that we pass to this WF Container Element using Binding are the Actual Values that we look for Comparison.



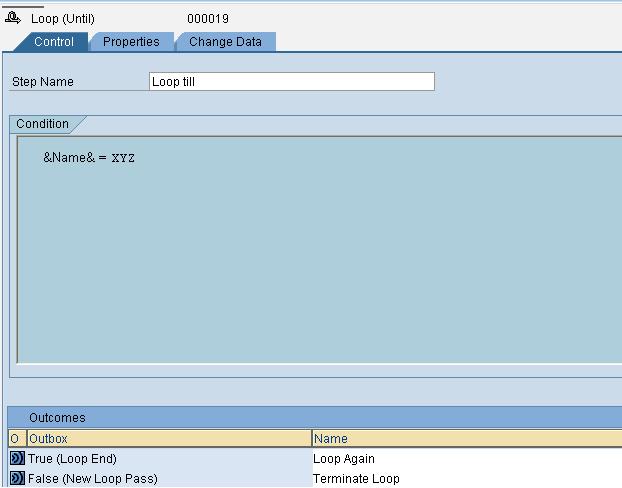
 Number of Values for Comparison = Number of Branches from the Multiple Condition Step-Type



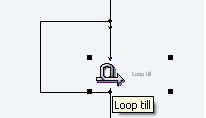
There are 3 conditions to check the name, so now we have 3 branches that will separately process each of the Condition being implemented.

## **Loop**

Loop is a condition that is used when one has to check for a particular Condition being FALSE or TRUE. Here we can put any Condition involving 1 or more than 1 Container Elements.

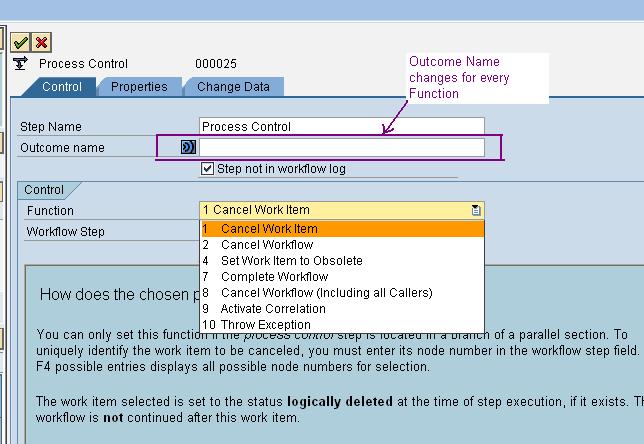


If the Condition gets satisfied, the Loop Continues or else if the Condition goes False, The Loop Terminates and Moves on to the Next Element in the WF's.

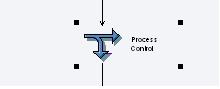


## **Process Control**

Process Control is a very important Step-Type, since it performs a lot of Functions.

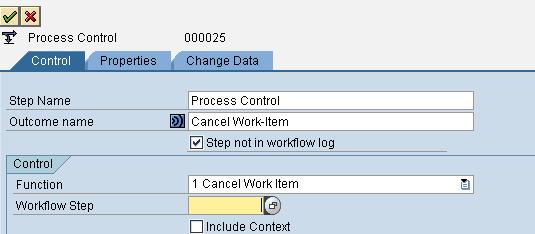


Now let us look at each of the Functions individually.



### ****Cancel Work item****

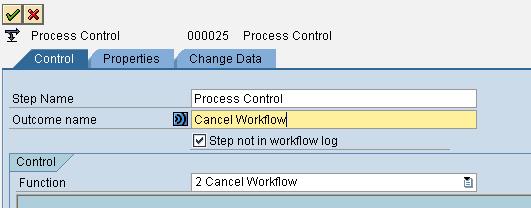
The process control step should be located in the branch of a parallel section. To uniquely identify the work item to be canceled, we must enter its node number in the 'workflow step' field. Press F4 to find all possible entries for the node numbers.



The work item selected is set to the status **logically deleted** at the time of  step execution. The workflow is **not** continued after this work-item. It terminates.

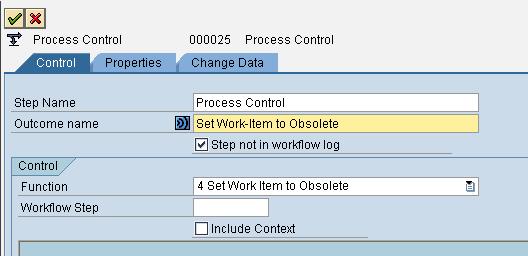
### ****Cancel Work-Flow****

Using this function, all work items of the workflow are set to the status **logically deleted**. If the step Process control is located in a workflow that is used as a sub workflow, the branch of the calling workflow that contains the sub workflow is stopped.



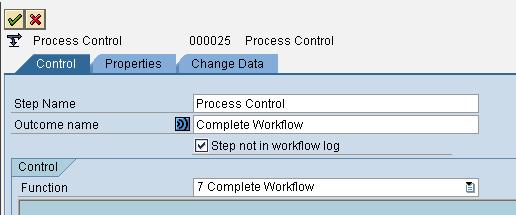
### ****Set Work-Item to Obsolete****

Use this Function, when process control is located in the branch of a modeled deadline monitoring. We can then set the work item that has exceeded its deadline, to the status **completed** and continue processing in the branch processing obsolete. In order to uniquely identify the work item to be cancelled, we must enter its node number in the field **workflow step**. Press F4 to find all possible entries for the node numbers.

   
The work item selected is set to the statuscompletedat the time of step execution.

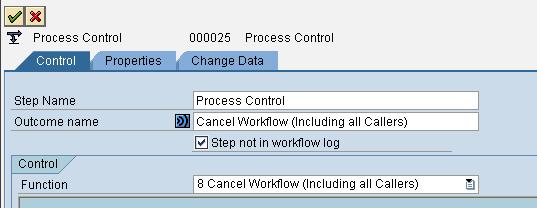
### ****Complete Workflow****

Using this function, workflow work items are set to the status **completed.** If the step process control is located in a workflow that is used as a sub workflow, the calling workflow is continued. For Details of Sub-Workflow or Local Workflow.



### ****Cancel Workflow (Including all Callers)****

Using this function, all work items of the workflow are set to the statuslogically deleted. If the step Process control is in a workflow that is to be used as a sub workflow, the calling workflow as well as all the work items in it is set to the statuslogically deleted.Thewholeworkflow is stopped using this function.



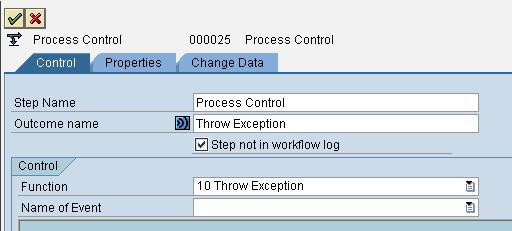
### ****Activate Correlation****

A correlation enables us to identify objects that belong together. If we have defined a correlation by using the correlation editor (Environment->Development->Correlation Editor), we can activate it here. It can then be used in a wait step.Choose the event that the correlation is to be activated for, and the correlation to be activated. Specify how the value of the correlation is to be determined:

* **BasisElement:** If we have defined the binding in the definition of the correlation, we can specify a container element that references the corresponding basis element from the correlation definition.
* **Binding:** Choose this entry if we want to define a binding between the Workflow container and the correlation container.  
     

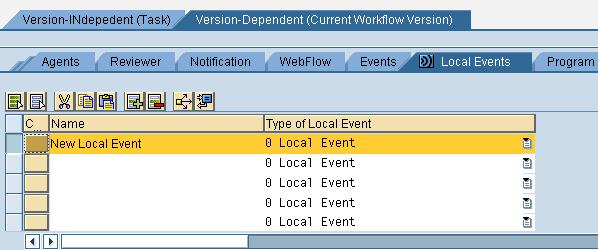
In the fieldCorrelation Instance, choose the container element for the correlation object.  
The correlation object shows the value of the correlation. The containerelement for the correlation object must have the predefined data type CL\_SWF\_MOD\_CORRELATION.

### ****Throw Exception****

Here we can trigger local events of typeException.  
   

We can define a local event or an exception in the version-specific basic data of the Workflow. We define the exception handler in the block.  
We define local events in the version-dependent basic data of the Workflow.

### ****Local Event****



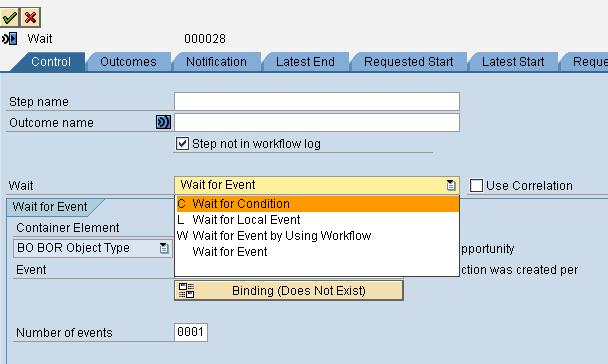
Give a name to the Local Event created; here the name is'New LocalEvent'.  
A local event can be the following:

* An exception that is triggered by a step of typeProcess Control. You define the exception handler in the block.
* A local event that is triggered by an event controller or an event. You can use a local event to start a local Workflow. Furthermore, a step of typeWait for Eventcan wait for a local event.

## **Wait**

Wait Step-Type is used when we need to wait for either an Event to happen or a Condition to get satisfied, so that we can proceed further with the Workflow.

 There are 4 conditions on which WAIT works:-



**C: Wait for Condition**

**L: Wait for Local Event**

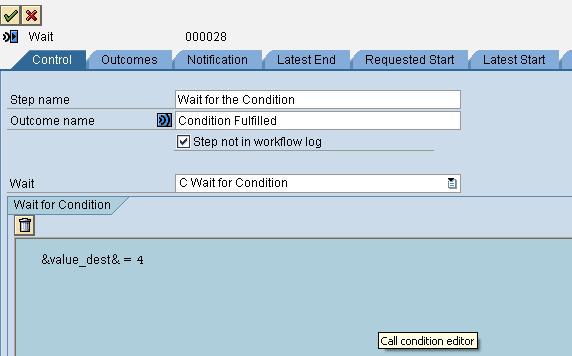
**W: Wait for Event by Using Workflow**

**Wait for Event.**

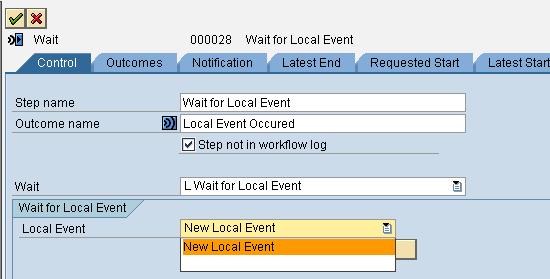
 Let us see these individually.

### ****C: Wait for Condition****

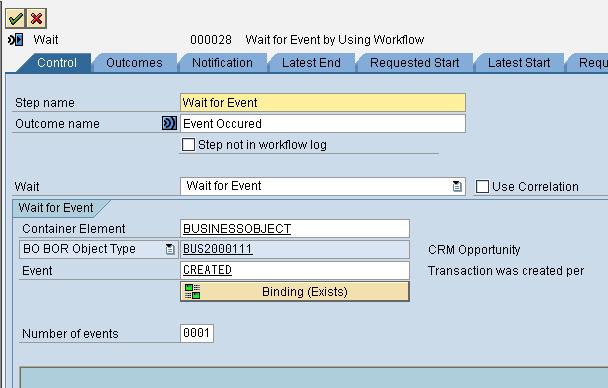
The wait step is complete once the specified condition has been fulfilled.



### ****L: Wait for Local Event****

This step enables us to stop the execution of this branch of our Workflow until a local event has arrived.  
   

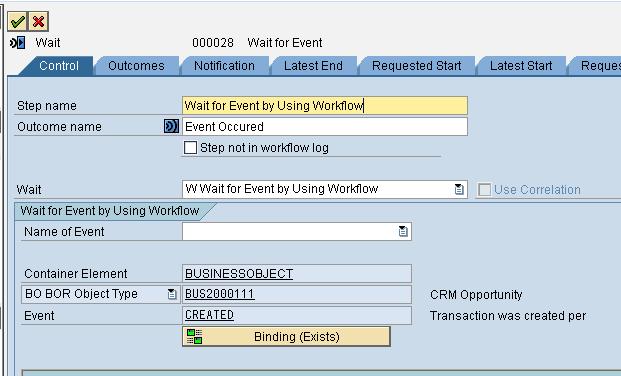
### Wait for Event

* With this step, we stop the execution of this branch of our workflow until a defined event for a particular object occurs.The object for which an event must occur must be in a container element of the workflow container. Enter the desired container element and select an associated event. We only have to define a binding if we want to use the data of the event in the workflow.  
     

We should not use this step type in a fork to wait for an event that is triggered in another branch of the same fork.If the event arrives, it completes **all** wait steps that are waiting for this event.

### ****W: Wait for Event by Using Workflow****

* When waiting for an event using Workflow, the event is initially received by the Workflow and transferred to the next step, as soon as it is active.

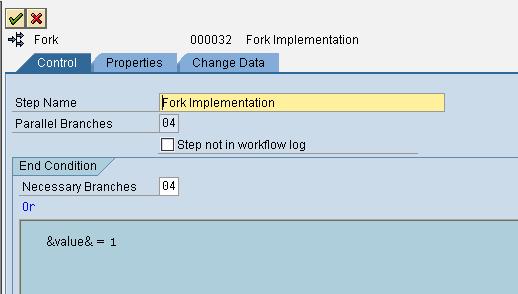


When waiting using Workflow, an event can complete a maximum of **one** wait step. If more than one wait step is active, the event completes the oldest wait step.

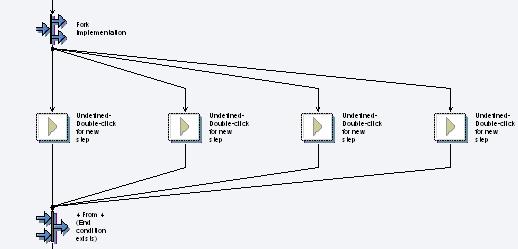
  

## **Fork**

This Step-Type is generally used when we need parallel processing which may or may not be based on any condition. Specifying a Condition is not mandatory for a FORK to work.



The number of Parallel Branches specifies that these many are the ways for parallel processing.

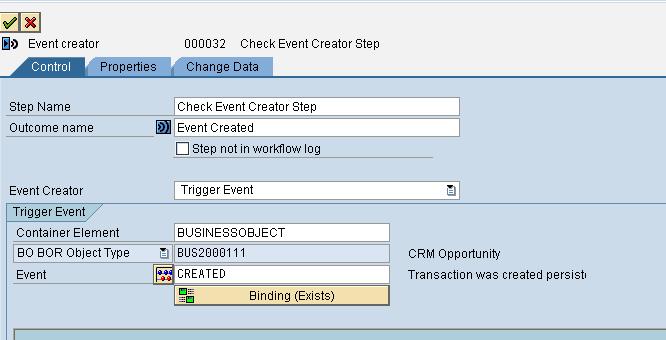


## **Event Creator**

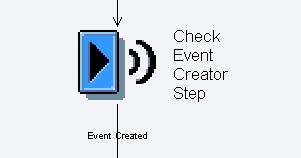
This Event Creator step type is used when we have to implement a BO Event or a local Event after the WF has started.

For this event Creator Step, the event must be defined for the BOR object type or class and a container element must exist in the workflow container, which refers to this BOR object type or class. Enter the required container element and choose an event. Do the necessary Binding.

We can use this event to, start or continue another workflow.



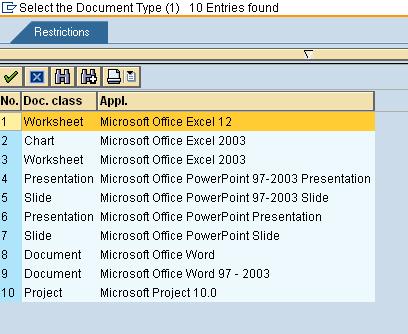
After this Event Creator, you can implement a Local Workflow to Start based on this Event Triggering Condition.



After Implementing This Step-Type, the next step would continue only when this Event is created in the Application. Until and unless, this Event is satisfied, WF won't continue from here.

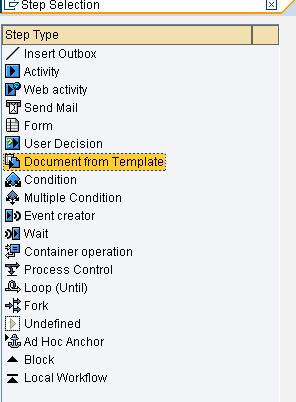
## **Document from Template**

The step type 'Document from Template' enables us to create documents in Workflow with different PC applications. These documents are based on a template that we create with the PC application when we define the Workflow. This template can reference elements from the Workflow container. We can display and edit templates created in this way throughout the Workflow.  
The Different PC Applications are:-



One must have saved the Workflow to be able to create a template.The receiver of the Workflow must have installed the PC application locally on their PC to be able to execute the work item. When the Workflow is executed, the template is opened in the relevant PC application and can be completed there. The document is then saved in a container element in the Workflow container.  
How to create a Document Template?

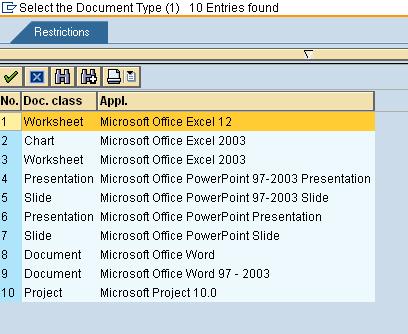
Choose this Step-Type.



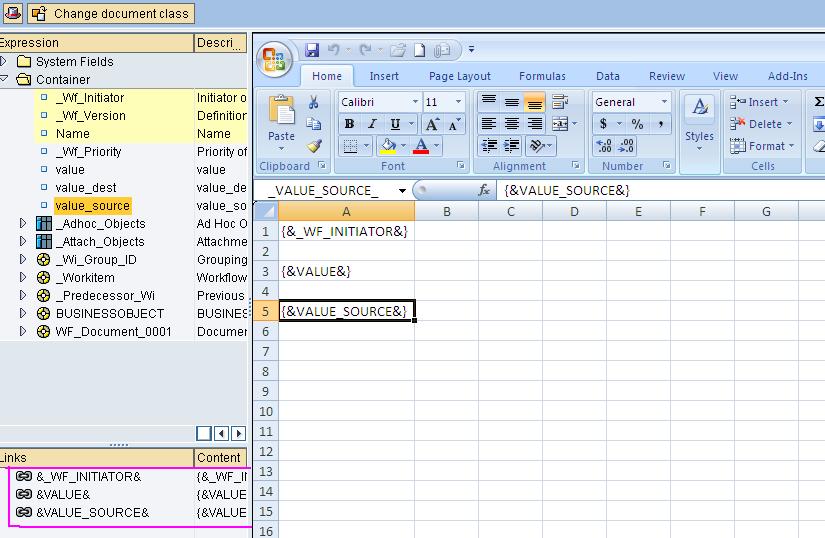
A workflow Container named WF\_DOCUMENT\_001 automatically gets created. Click on 'CREATE'.



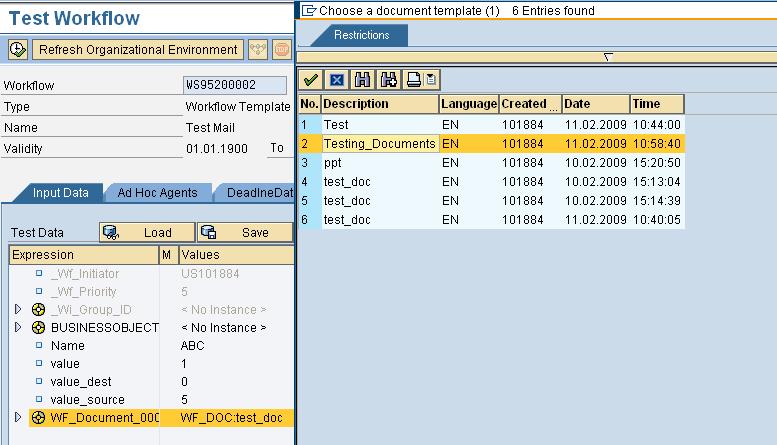
Select one of the Options.



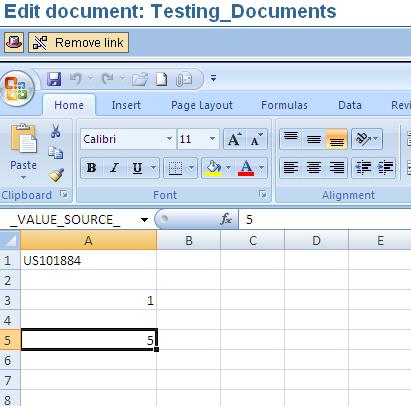
After Selecting a PC Application, The system fields and the container elements of the workflow container are offered for selection in the navigation area. They can be transferred into the template by double-clicking.



Select to assign a name for the document template and SAVE the Document template. Now Execute the Workflow and it gives us the Option to select from the various Templates Created. The Container Fields that we have inserted in our Template gets replaced by the Run-time Values.



Select the Document that you created and find the runtime values.



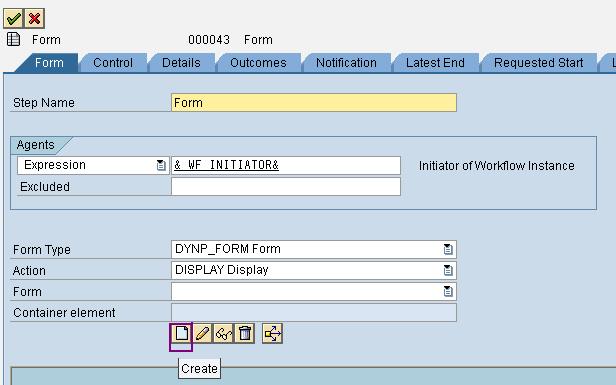
## **Form**

 We can use a form to display or edit data of a container element of the workflow container that refers to a structure. We can use all the fields in the structure in the form. We can use a wizard to generate the form we need and we can edit the generated form and make our individual settings.

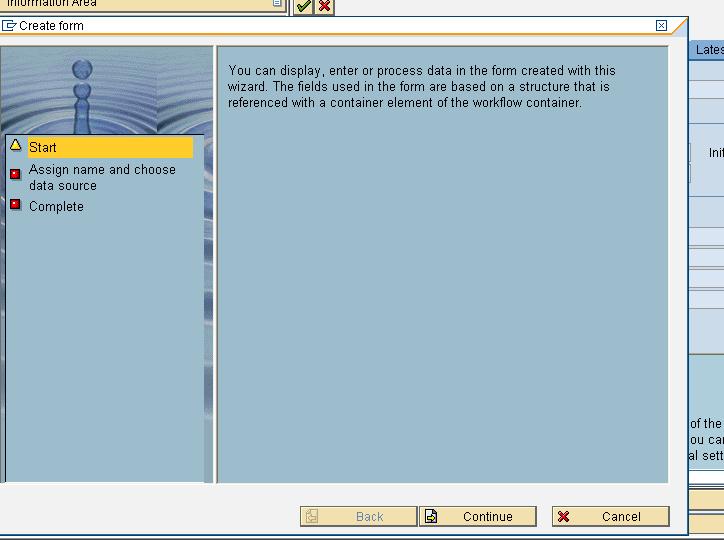
Depending on the action chosen, the form is displayed in either display or edit mode when executed. Additional pushbuttons are generated for the approval actions. As well as the form, the workflow toolbox is automatically available in execution.

How to create and use a FORM?

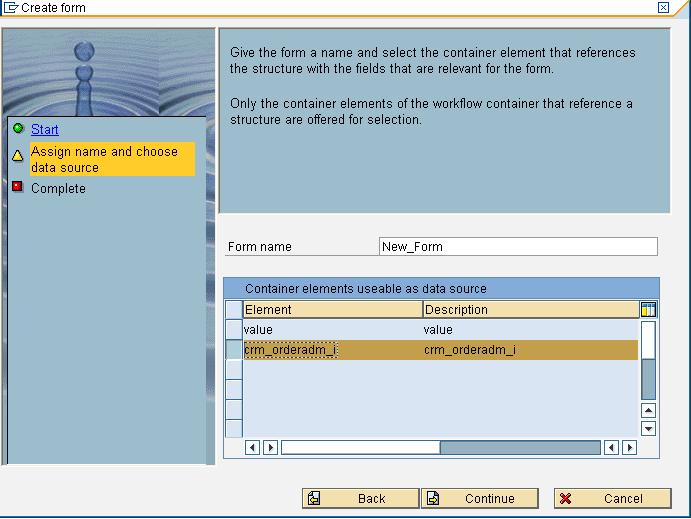
Create a WF Container – the Type of which you need the FORM to be. Select the FORM Step-Type there.



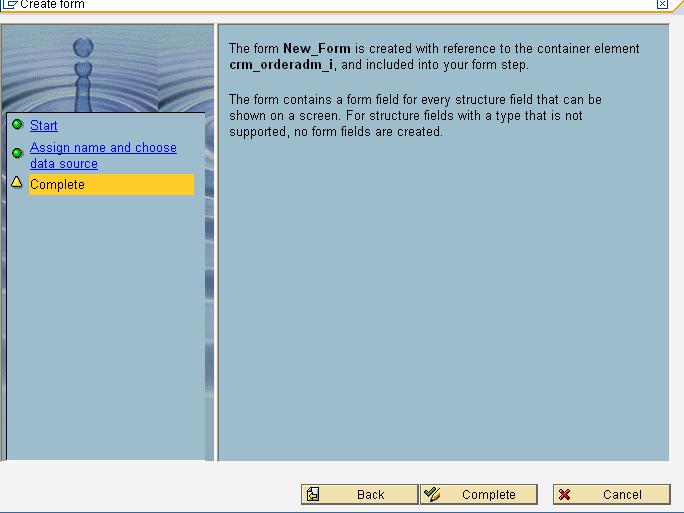
Click on CREATE.



Click on CONTINUE.

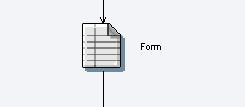


Click on CONTINUE.

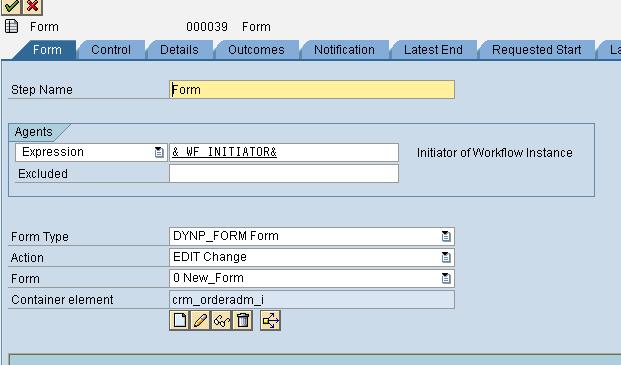


Click on COMPLETE.

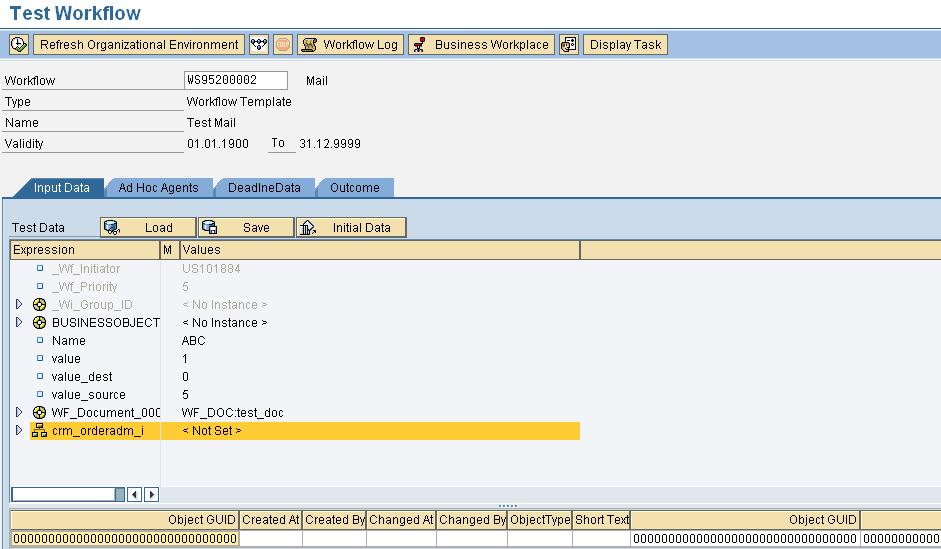
Form Created Successfully.



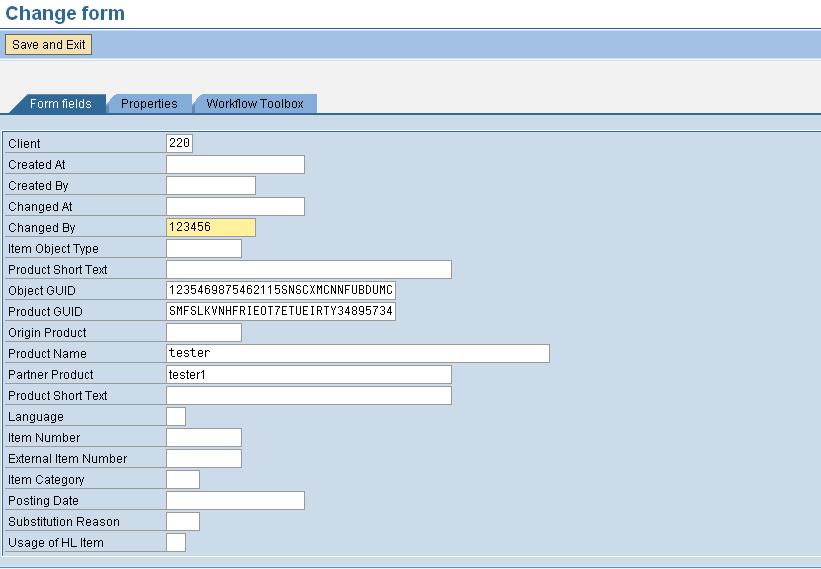
The FORM Format.



Now Execute the WF.



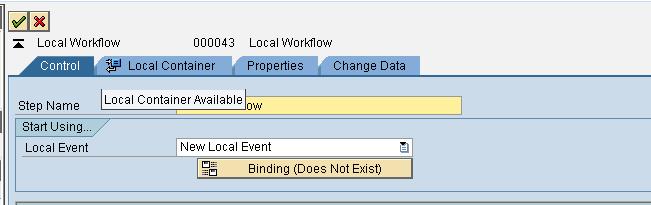
 The FORM opens up for you to fill. Just Fill the FORM and Press SAVE AND EXIT Button.



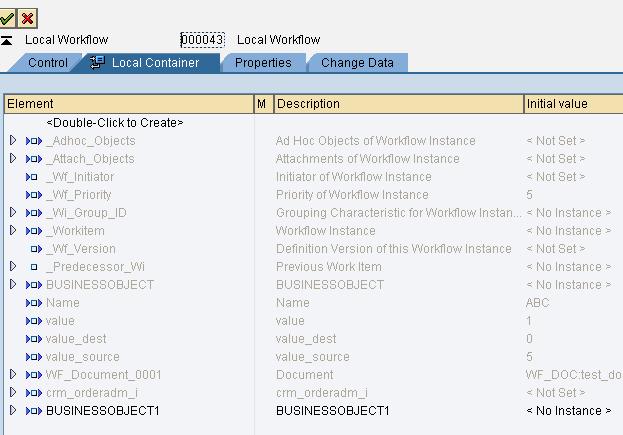
After you press the SAVE AND EXIT Button, the data gets saved in the required table from which the WF Container was made: - CRM\_ORDERADM\_I in this case.

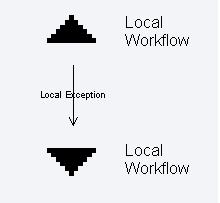
## **Local Workflow**

        In a local Workflow we can model a part of the Workflow that is started by a local event.



In this Local Workflow we should create our own Workflow Container Elements, called as Local Containers. These additional container elements are then available to all steps that belong to this local Workflow and can be filled from the Workflow container by either an initial value or a binding.



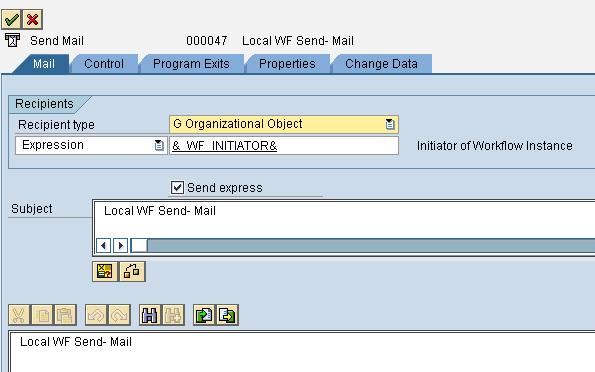


We only need to define a binding if we want to use the event data in a Workflow. We must set the import indicator for local container elements that are filled by a binding from the Workflow container to ensure that the binding can be traced.

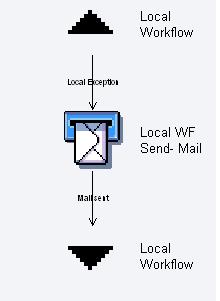
Within this Local Workflow you can still create some more WF Steps, so that this should act as a Sub-Workflow for the original WF in which this Local Workflow was created.

Create a Send-mail Step in this Local Workflow:-

Right Click on the Down Arrow for Local Exception, and Click Create. Select the Send-Mail and enter the values.



The Local Workflow Looks like this:-



Implement this Local Workflow after the Event Create Step-Type and see the Effect. This Local Workflow will trigger for the Specified Event happening after the start of the original Workflow. In the Event Creator Step-Type, Use the Local Workflow as the Event Creator: - TRIGGER LOCAL EVENT.

WORKFLOW DEBUGGING

**Debugging a background workflow process**

*In your method write the following code:*

data exit.

Do.

if exit = 'X'.

exit.

endif.

enddo.

Run the workflow, causing an infinite loop on that step, and then go to SM50.

Here you can debug the process.

**Tansaction SWUD**

SWUD is the diagnosis transaction. This will guide you through the fault finding process. The check-list order is determined dynamically according to:

a) your workflow definition and

b) the statistics of the most common mistakes made when creating/activating a workflow.

The testbed, provides a launchpad for tests to ensure that your workflow definition is rugged enough for the production environment. It also allows you to access the tasks, subflows, object types, delegated object types and agent assignments directly, without going through the workflow builder first. (Foreground tasks with no agent assignment are marked in red from release 4.5 onwards). This list is also useful as a catalogue of the components of workflow for documentation or transport purposes.

Click on the "info" icon to read a description of the test being performed.

Any workflow accessed through this transaction is added to the last-used list. The workflow can be recalled at a later date by clicking on the reuse icon in the left-hand column of the last-used list.

The next section describes the transactions called from SWUD. Experts may find it easier calling them directly.

**Secondary Diagnosis Transactions**

 *Transaction SWU3:* Verify Workflow Customizing.

In this transaction the most important basic customizing for workflow is checked. If you find any errors here, read the documentation for the error. If your system is BBP or CRM, see note 60801.

 *Transaction SWEL:* Event Log.

Note: Before you can use the event log, you must make sure it is activated. You check this in transaction SWELS. If you are not sure whether or not it is active, just deactivate and activate the event log once.

WARNING: An activated event log can lead to bad performance, and must not be activated constantly in a productive system.

 *Transaction SWE2*: Event type linkage

A workflow is usually started or triggered by an EVENT so it is important to make sure that an event occurrs. In the event log you can see the event and its business object type. If an event has been successfully created, it is important to check if there is a receiver type for this event. The receiver type is the workflow that has been triggered by the event. The coupling between an event and the receiver can be maintained in trans. SWE2.

 *Transaction SWU0:* Event simulation

If you have found NO receiver type in the event log, then you should check transaction SWU0, Simulate Events. In this transaction you can simulate an event. The system will check all workflows that could be triggered by the event and tells you whether or not they were triggered. Information is displayed for the workflows that were not triggered successfully.

 *Transaction SWI1*: Work item list

If a workflow was been correctly triggered and started at least one WORK ITEM should be created. You can view the work items in transaction SWI1, Work Item Selection. The easiest way for the selection is just to go into this transaction after you have checked the event log. By default all the work items of the last hour will be displayed. You will find in this transaction the work item number, the status of the work item and the workflow task that has created the work item. Furthermore, (in the technical workitem display) you can have a look at the container by selecting a work item and using the menu GOTO -> CONTAINER. To display the work item details double click on the item. From the details you can see the selected/ possible/excluded agents for this work item by using the menu GOTO -> AGENT -> SELECTED/POSSIBLE/EXCLUDED AGENTS.

When the status of the work item shows an error you should have a look at the workflow log using the menu EDIT -> DISPLAY WORKFLOW LOG. Here you can see errors that have occurred.

*Transaction PFTC/SWDD:* Task editor/Workflow Builder

If you want to have a look at the definition of the workflow you can use transaction PFTC, Maintain Tasks. The button WORKFLOW BUILDER brings you to a graphical display of the workflow structure which is fairly easy to read and understand. On the tabstrip TRIGGERING EVENTS you can check if the triggering is activated or not and deactivate/activate it by clicking on the icon in front of the event.

*Transaction OOCU:* Customizing overview

To check if the relevant tasks are activated and users are assigned, you can use transaction OOCU; Task Customizing Overview. In this transaction all the tasks are sorted by the application components and are thus easy to find. Note that if you have almost no entries in this transaction, you should execute the report RS\_APPL\_REFRESH in transaction SE38.

*Transaction SWUE/SWUS*: Event/workflow trigger

If you have activated a user exit, e.g. for the role determination, and you have problems with that, you most probably have to debug the workflow. This can be done using the transaction SWUS or SWUE.

With SWUE you can create an event. After you entered object type and event you have to fill the event parameters and the key for a specific object. If you want to debug the workflow you have to set the flag TRIGGER RECIEVER FM SYNCHR. If you have set a break point in your user exit or in any other function module of the workflow the system will stop at this break point and you can start analyzing.

The second possibility to debug a workflow is to use the transaction SWUS, the workflow test environment. In this transaction you have to enter the workflow task and fill the input data. Please use the F4-help to fill the input data. Here again you have to set a break point in your user exit or in any other function module of the workflow and then the system will stop there.

Summary of the manual alternative to SWUD

Now that you have a better understanding of how to use these transactions, you can follow the steps below to quickly analyze your workflow problems:

1.Go to transaction SWU3 and check if everything is okay. If it is not, please fix it before you proceed.

2. Go to transaction SWEL and check if an event is created. If not, check in transaction SWELS if the event log is activated.

 If you see an event but no workflow is triggered, proceed with step 3.

 If you see an event and a workflow is triggered, but the status of the triggering shows an error, proceed with step 3.

 If you see an event and a workflow is triggered and there is no error, proceed with step 4.

 If the event log is activated and you do not get any event, open up a message using the component of the application to which the business object type belongs. From release 20C for creation of shopping carts the architecture of workflows changed and no events are visible

3. Go to transaction SWU0 and simulate the event. Is the workflow that you expect to be triggered visible in the simulation?

 If it is not, then check the workflow definition in PFTC. The event must be defined as triggering event.

 If it is, but the event linkage is not activated, go to transaction OOCU and activated it.

 If it is visible, but it is not activated, go to transaction PFTC and activate it.

4. Go to transaction SWI1. Search for the work item that belongs to the workflow. Check the status of the work item.

 If it shows an error, look at the workflow log. Do you see any errors? If yes, you can probably find a solution in the SAPNet Frontend. (Tips and tricks on how to search for information and solutions in the SAPNet Frontend can be found in note 192194.) If you do not find any help in the SAPNet Frontend, please open up a message with the component BC-BMT-WFM and tell us the number of the error message and the name of the workflow.

# [**Restarting SAP Workflows**](https://wiki.scn.sap.com/wiki/display/ERPFI/Restarting+SAP+Workflows)

[Skip to end of metadata](https://wiki.scn.sap.com/wiki/display/ERPFI/Restarting+SAP+Workflows#page-metadata-end)

* Created by Guest, last modified by Anonymous on [Jul 18, 2011](https://wiki.scn.sap.com/wiki/pages/diffpagesbyversion.action?pageId=244810140&selectedPageVersions=5&selectedPageVersions=6)

[Go to start of metadata](https://wiki.scn.sap.com/wiki/display/ERPFI/Restarting+SAP+Workflows#page-metadata-start)

Certain times we might need to restart workflows for various reasons. In SAP, we can restart the workflows as below:-

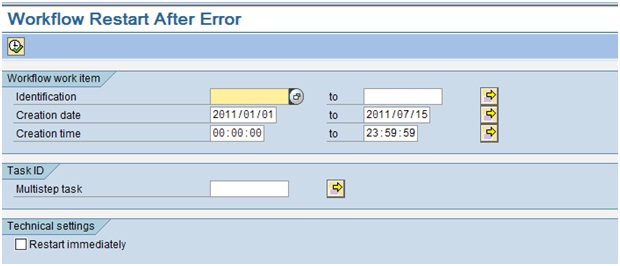
(1)    Restarting the workflow which is in error

(2)    Restarting the workflow which is not in error

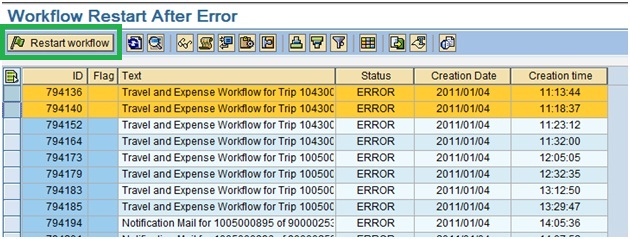
This link will explains how to restart a workflow in above scenarios:-

**(1)** **Restarting the workflow which is  in error:-**

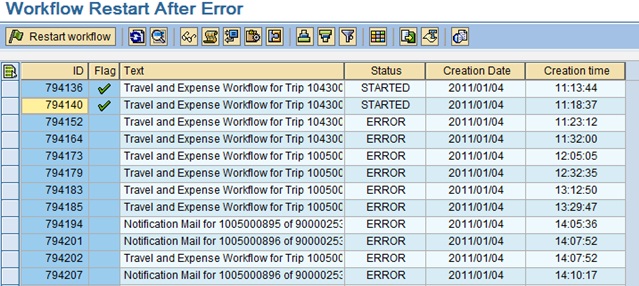
A workflow which is in error, can be restarted via the standard transaction SWPR. In this method, the same workflow gets restarted and no new workflow is generated. The workflow gets restarted from the point where it was staucked.

Go to transaction SWPR and the following screen will come:-  


Here enter your workflow work item ID, if you want to restart a particular workflow, otherwise you can also select based on time interval. If option “Restart Immediately” is selected, then all the workflows matching your selection screen will get restarted as soon as you execute this screen.

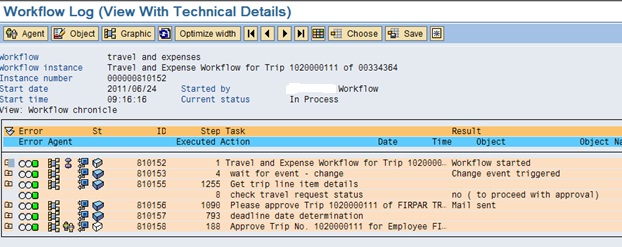
After executing, below kind of screen will appear showing the workflows corresponding to your selection:-  
    
Here you need to select the lines containing your workflows, and then click on button “Restart Workflow”.

Following message will come at the bottom of the screen about restart status:-  
https://wiki.scn.sap.com/wiki/download/attachments/244810140/Image0003.jpg?version=1&modificationDate=1310706403000&api=v2

And the flag goes green for the workflows restarted:-  


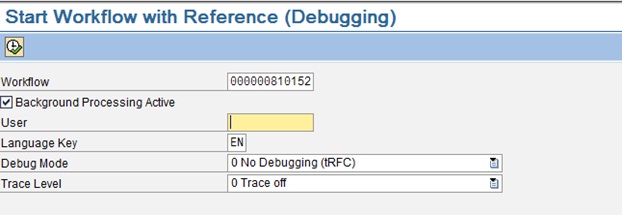
**(2)** **Restarting the workflow which is NOT in error:-**

A workflow which is not in error, can not be restarted via the standard transaction SWPR. For this case, a new workflow needed to be generated with reference to the old workflow and old workflow needed to be deleted logically.

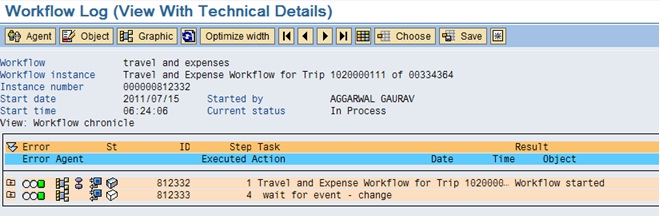
For example if you want to restart the below workflow which is not in error:-  


For this you need to do below two steps:-

**Step 1:-**

Go to transaction SWUS\_WITH\_REFERENCE and give the workflow instance (ID) and execute:- (You can enter the user name also which you want to display for workflow restart)  


After execution below message will come on the screen:-  
https://wiki.scn.sap.com/wiki/download/attachments/244810140/Image0007.jpg?version=1&modificationDate=1310706435000&api=v2

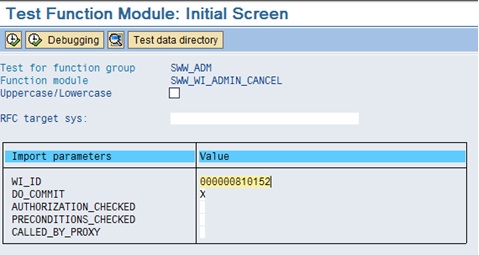
You can check this new workflow created:-  


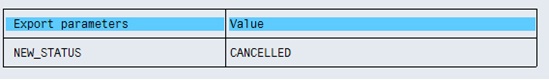
(Please note that this new workflow also might need certain action to proceed, e.g. in a travel and expense workflow, the new workflow needed to be reapproved in transaction PRAP by AP clerk.)

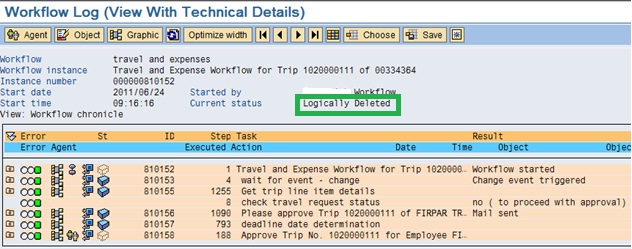
**Step 2:-**

Even the new workflow has been created, but the old workflow is still in the same status, so you need to logically delete the old workflow. You can do it via two ways:-

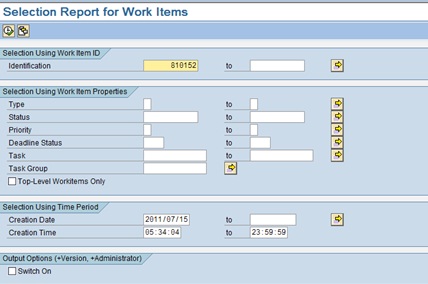
**Method 1:-**

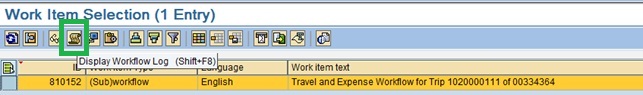
Execute function module SWW\_WI\_ADMIN\_CANCEL in transaction SE37:-  


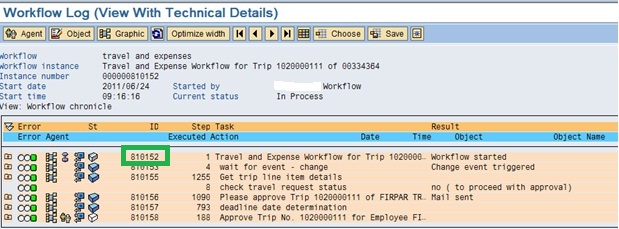
Here give the workflow instance (ID) of old workflow and execute. And the new status will come as status in the execution output:-  


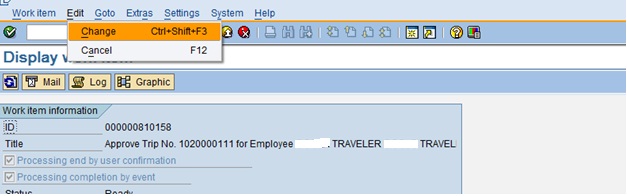
Check the workflow logs of the old workflow, which will now show the status “Logically Deleted”:-  


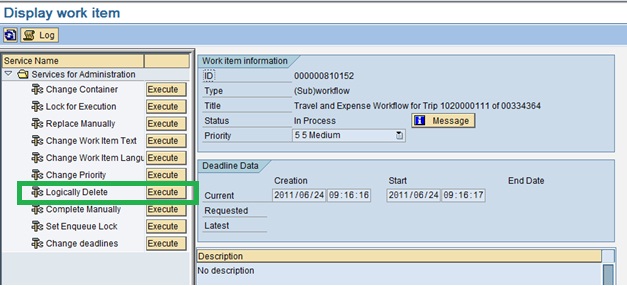
**Method 2:-**

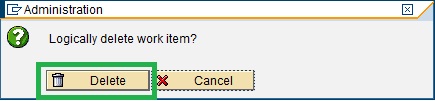
Go to transaction SWI1, enter workflow instance (ID) of old workflow and execute:-  


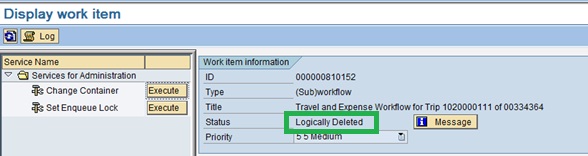
In the output, select your work item line and click on log button:-  


In the logs, double click on the first line in workflow logs:-  


Then on the next screen, click on Change option in Edit menu:-  


On the next screen, click on “Execute” button for “Logically Delete” option:-  


Below pop-up will appear, confirm it:-  


The work flow gets deleted:-  


Check the old work flow logs, which will now show the status “Logically Deleted”:-  
