Workflow: Cross System and Cross Client

Applies to:

SAP ECC 6.0.

For more information, visit the Business Process Modeling homepage.

Summary

The article describes the concept and approach for cross system and cross client workflows.

Author: Bijal Parmar

Company: Larsen & Toubro Infotech Limited

Created on: 15 July 2009

Author Bio

Bijal Parmar is a Software Consultant in Larsen & Toubro Infotech Limited. She has three years of SAP experience. She has worked extensively in ABAP, Workflow and Visual Composer.

Table of Contents

Concepts

Cross System

In some scenarios, the business process workflow is distributed in more than one system. It means the workflow is triggered in one system and executes some tasks and then triggers a second workflow in another system. Processing is done in the second workflow and again comes back to first workflow. This means there is a switch between workflows across systems. The major task is to pass data from one workflow to other.

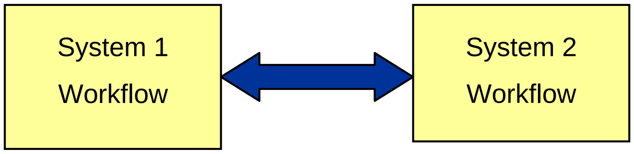
Cross Client

Like Cross systems, we can have workflows talking across clients of same system. Example, Workflow is triggered when a Purchase Order is changed in system SYS 1 (Client 100). It determines the changes. Based on it, a work item is generated for approval of changes done. The agents (Suppliers) are in same SYS 1 but different client (Client 200). So, the Dialog task for Approval will be in client 200.

Cross System Workflows

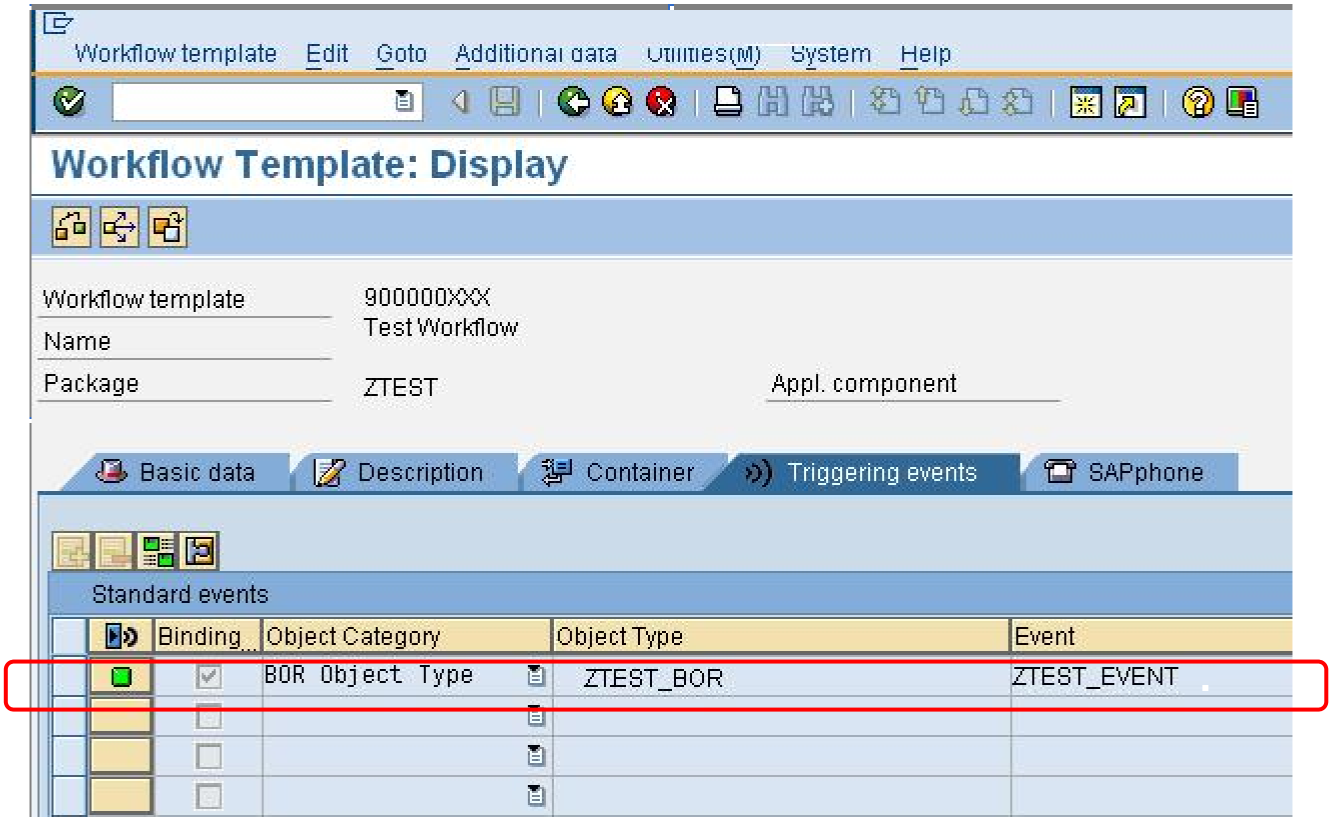
Design

Assume we have two systems and two workflows.



Start events are created for both workflows in their Business object and it is activated in the workflow in corresponding systems using transaction PFTC / SWDD

Go to Triggering events, add the start event and activate the event, this will create a customizing transport request.



Workflow Template: Display

Workflow template: 900000XXX

Name: Test Workflow

Package: ZTEST

Appl. component:

Standard events

	Binding	Object Category	Object Type	Event
🟩	☑	BOR Object Type	ZTEST_BOR	ZTEST_EVENT
	☐			
	☐			
	☐			

- When the first workflow is triggered, the data is passed to the first workflow event container. And, from event container data is passed to the workflow container.
- We will trigger the second workflow from first workflow.
  For Example, the first workflow is triggered is System 1, when a PO changed. The change is processed in first workflow tasks, and then the work item for approval is to be sent to agent in System 2 with information of what data is changed in workflow. In such cases we have to pass data from one system workflow to other system workflow.
- First workflow will trigger the start event of second workflow in another system and pass data to it. The data is passed in the event container of second workflow.

Approach

The approach is, in task of the first workflow you call the below Function module with destination of Second system and pass data in event container. This will trigger start event of second workflow.

This is an example of a code sample:

```
CALL FUNCTION 'SWE_EVENT_CREATE’ DESTINATION ‘SYSTEM2’
EXPORTING
objtype                                     = ‘Z_SYS2’
objkey                                      = ‘450000111’
event                                       = ‘ZSTART_EVENT_SYS2’
take_workitem_requester              = ‘X’
start_with_delay                     = ‘X’
start_recfb_synchron                 = ‘X’
IMPORTING
event_id                                            = w_event_id
TABLES
event_container                                = event_container
EXCEPTIONS
objtype_not_found                             = 1
OTHERS                                            = 2.
```

Note: Check the availability of the destination system before making a RFC call. Use function module **‘RFC_PING’** for it.

We will need to pass variables, structures and internal tables to the second workflow. But using the above code does not pass the internal tables to second workflow in our workflow. It will pass only structures and variables.

This issue has been posted on SDN before, but solution was not available. Hence, this article will address the method to solve the same as stated below.

To resolve this, create a custom function module in System 2, and call the 'SWE_EVENT_CREATE’ function module within it.

Call this custom function module with destination ‘System2’ in task of first workflow and pass all data to it. This will inheritably call the function module to trigger start event of second workflow using 'SWE_EVENT_CREATE’ FM.

It will populate all data in event container of second workflow.

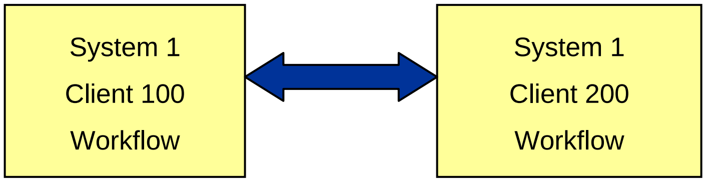
Summary

The function module 'SWE_EVENT_CREATE’ is to be called in the system in which the workflow resides.

Cross Client Workflows

Design

Assume we have one system and we will create 2 workflows, but in different clients.



When we create a workflow, we will see it using any client. In the above scenario, the triggering of the workflow must happen in proper client. Hence, also if you create two workflows using same system and same client. Important is to start/trigger the workflow, in proper client.

In above diagram, we have client 100 and 200, we will create workflows in client 100. (This is when you have development rights to your userid of the system only in one client)

In such cases the business object for both workflows can also be common. You create 2 start events, one for each workflow in the BOR.

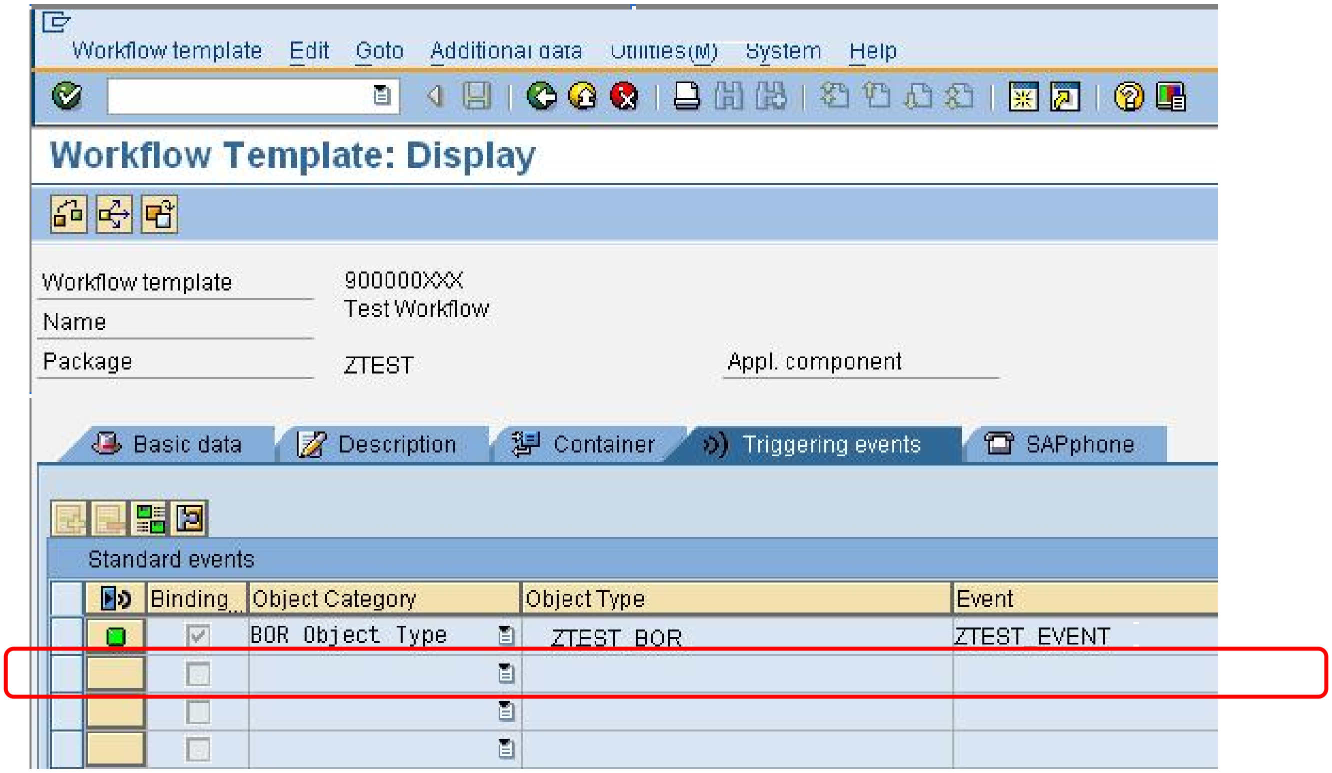
You can use the CALL FUNCTION 'SWE_EVENT_CREATE' to trigger the workflows.

Workflow to Event Linkage is Client specific

We need to activate the start event for each workflow in its particular client.

In above example though we created both workflows in client 100, when we need to create the customizing request and activate the start event for second workflow, we need to login to client 200 and do it.

In PFTC/SWDD, create start event and activate the workflow in its corresponding clients.



Workflow Template: Display

Workflow template 900000XXX

Name Test Workflow

Package ZTEST

Appl. component

Standard events

	Binding	Object Category	Object Type	Event
🟩	☑	BOR Object Type	ZTEST_BOR	ZTEST_EVENT
	☐			
	☐			
	☐			

Summary

Workflow to event linkage is client specific.

Related Content

https://wiki.sdn.sap.com/wiki/display/ABAP/SAP+Workflow+EVENTS

https://forums.sdn.sap.com/thread.jspa?threadID=113766

https://forums.sdn.sap.com/thread.jspa?threadID=168757

For more information, visit the Business Process Modeling homepage.

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.