Workflow Diagnosis

Debugging and troubleshooting a workflow is a challenging task. In this document, we will start from the debugging of the attributes and methods i.e. Business Object Builder, followed by a brief look into Events and subsequently Workflow as a whole.

One of the first things to do as a workflow developer is to look at the business object to see if it has what is needed in the way of attributes and methods. Once attributes and methods, (assuming you then build your workflow) are debugged, we need to know about linking it to an event (this is the most commonly used means of triggering a workflow) and then we will know why the workflow didn't get to the right person.

At the end of it all, there is the very powerful tool provided by SAP which allows you to perform a variety of tests from one transaction, SAP Workflow Diagnostic tool, transaction SWUD.

The Business Object Builder or Business Object Repository (BOR) and is accessed via transaction SWO1.

From this screen, enter an object key for the debugger. Just click 'Instance' and fill in the values with a document you have already created. A pop-up screen appears where you can enter the key values for the object.

This will display all the attributes for that object. This enables you to quickly view whether the values are as you expected. The circling arrows icon will indicate which attributes are virtual attributes. Scroll down on to see all the methods which are linked to that object.

To execute any method, just click on the execute button beside the required method.

NOTE: The application must be in debugging mode.

Now you can see the ABAP code behind the method.

From here, you can debug the program as like any other ABAP program. But there are some differences.

Points to keep in mind while debugging the workflow:

- Set a breakpoint in the code of attribute (or method) wherever required. You can then step through debugging the attribute or method as you would any other ABAP code.
- If you have calls to macros (swc_get_element, etc) you may need to set the values of the result elements inside the debugger (for example, when getting an element from the workflow container)

Debugging Events

Once you have debugged the attributes and methods, you would need to link it to an event. This is the most commonly used means of triggering a workflow. There are multiple ways to trigger and control the flow of a workflow, but the most commonly used is via events. They can be used to start and end workflows and workflow tasks. Event linkage in the workflow

- Event linkage (SWETYPV)
- Event creation (SWUE)
- Simulate event (SWU0)
- Check the event log (SWEL)

You need to know how to look at the event linkage to the workflow from both sides, that is, from the workflow to the event, and from the event to the workflow.

You also need to know how to check that the event is actually being triggered (just because an event is listed, does not necessarily mean it will be triggered. (It could be linked to some obscure customizing!)

You will want to be able to simulate the event, to check what happens if you know the event has been triggered but for some reason your workflow did not start, and to see if your workflow did start, but failed immediately.

Click on the Header button (Hat symbol at the top) > Version Independent (Task) > Start Events. Now check that all the triggering events for that Object Type are binded correctly and is in active state.
"A green light in the Active column means that your linkage is active. When the gray button is pressed (i.e.: it's not green yet) you will be given some binding messages. The binding to the event should always be checked."

If events are used in SAP Business Workflow (as triggering and terminating events, for wait steps), the entries required are made by the workflow system.

If you enter an event as a triggering event of a workflow, for example, the following entries are made automatically:
- Object type and event ID
- Workflow to be started as receiver type
- Function module SWW_WI_CREATE_VIA_EVENT as receiver function module

**Event Linkage with SWETYPV**

![image]

*The type linkage describes the assignment of a receiver function module and a receiver type to a particular combination of object type and event.

A type linkage must be created if the system is always to react to an event of a particular object type. The type linkages are evaluated at runtime by the event manager.*

The event receiver should define a type linkage using a function module provided.

![image]

If Linkage activated check box is checked, the Receiver Function module field contains SWW_WI_CREATE_VIA_EVENT function module and the Receiver type Function module is empty, then assume that the workflow is going to be triggered by the event.

**Event Triggering With SWUE**

![image]

You will probably need to test whether a workflow will be triggered by the event without always having to go through the business process of creating a new object instance every time. For these purposes, you can use the test tool SWUE to create the event.

After you fill in the object key, and click on the 'Trigger Receiver FM Synchr' flag, then click the 'Create Event' button.

This screen shows you the event and the receiver when a breakpoint has not been set. It can be a quick test to see that the workflow will be triggered.

Go to SE37 and set a breakpoint in the receiverFM, checkFM, or receiver type FM that is being called. Back out and go into SWUE and fill out the prerequisite key. Remember to check 'trigger receiver FM synchr'. When you push 'Create Event', you get into your function module at your breakpoint, and you will have the necessary parameters and object elements to proceed with debugging your FM.

**Event Simulation with SWU0:**

![image]

The Event simulator is a quick way to tell if your workflow is a candidate for starting. It can also tell you what other workflows may (or may not) be linked to the event. If you have found NO receiver type in the event log, then you should check transaction SWU0, Simulate Events.

![image]

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.

**Using the Event Log SWEL**

**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. When a new workflow goes into production, you can have the event log turned on. But you are advised not to leave it on for an extended period of time, as every event that occurs in the system, whether linked to a workflow or not, is logged in this file, and this can overburden the database system.
Here is an example of event log in SWEL. Please note: red indicates errors, green indicates success of a workflow. In case of errors, go back to se37 and set a breakpoint in the receiverFM and use SWU0 to create the event again.

**The Diagnostic Tool, SWUD**

At last we come to one of the most powerful testing tools available. SWUD is your single point of contact for performing diagnostics from the most basic (is the workflow customizing set up correctly) to more complex ('Task does not start', 'Workflow is hanging')

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. This list is also useful as a catalogue of the components of workflow for documentation or transport purposes.

The options from here are:

- Verify workflow customizing
- Problem: task does not start
- Problem: workflow is hanging
- Test environment

In order to know what else you can achieve with the help of this transaction, just go to this transaction and click on the Info button on the right side. This will give all the description regarding the test which you want to perform. 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.

1. **Verifying Automatic Customizing:**

   You can run transaction SWU3 directly, Transaction SWU3. In this transaction you can check the most important basic customizing for workflow. With the help of this transaction we can check the basic customization of the workflow such as the workflow administrator is maintained, RFC Destination is configured correctly.

2. **Task not started:**

   Task does not start takes you to another screen from which you can find all instances for a task, refresh the org environment, and run a consistency check on your tasks.

   !!

3. **Workflow is hanging:**

   Workflow is hanging checks for all instances of the task, checks the RFC logs, checks the ABAP dump analysis, and provides a consistency check for the workflow components.

4. **Test Environment:**

   The test environment is where I end up going the most...

   With the help of this option we can do multiple things such as enabling workflow trace, turning on the container monitors. When a task is executed, the contents of the container are shown in a dialog box. This is useful if you can't determine exactly what is being passed from one task to the next. You can turn the container monitor off by pressing the Debug off button.

   The workflow trace can be turned on from SWU8. With the help of this trace you can get a very detailed picture of what is going on inside your workflow, from workflow to task, and so forth.

   You can activate and deactivate by SWU8 transaction.
You can see the workflow log by clicking the workflow log button.

You can also see the graphic and know where the error is located.