SAP Workflow and Events - Control the flow!

**Day and Time:** Thursday, May 16, 2013 - 12:30 PM - 1:30 PM Session 0811

**Content Area:** Business Integration, Technology and Infrastructure (BITI)

Susan Keohan, MIT LL
Martin Maguth, Norikkon
LEARNING POINTS

- How to use an event to control the SAP Business Workflow
- How to trigger an event
  - Standard SAP Methods
- Know about the advantages of using events
  - Traceable
  - Deactivatable/activateable
  - Forward-looking – feeds into Process Observer/SAP Operation Process Intelligence
Radio example for events

SAP Programs
(One Radio Station \(\Rightarrow\) one Object Type)

Transmitter

Events

Antenna

Receiver “On”

Start Workflow Instance

Terminate Workitem

Transactions
to configure events

Simulate
event SWUO

APAP Program

Manual (Re) Trigger
SWUE

Events of different
Business Objects
or ABAP Objects

Trace events
SWEL

Switch event trace on or off
SWELS

RFC check
SM58

Event Linkage
SWETYPV

Event queue
SWEQADM

Instance linkage
SWEINST

Follow @ASUG365 and #ASUG2013 on Twitter
Overview: Events to Control Workflow

- Trigger a new instance of a Workflow
  - Select the Workflow to be started
  - Check, if the workflow should be started at all
  - Start Workflow instance immediately
  - Use the event queue to start a Workflow instance

- Terminate a dialog workitem
  - Set the workitem to complete, so that the workflow instance can continue

- Wait for an event
  - Complete a section of a Workflow instance
  - Complete a whole Workflow instance

- Start a ABAP function module
  - Consume the event without any impact on workflow
Different ways to trigger an event

- SAP Configuration of Events
  - HR Events
  - Status Management
  - Change Documents
  - Message Control
  - Trigger a BOR event via the ABAP function module
    SAP_WAPI_CREATE_EVENT (Commit Work)
  - Transaction SWETYPV
Different ways to trigger an event

- Triggered by program
  - SAP provided program triggers the event
    - Read SAP Help for Workflow Scenarios, to find the required IMG configuration steps
    - Use the event trace to find events
  - Custom developed program will trigger the event
    - Trigger a BOR event via the ABAP function module `SAP_WAPI_CREATE_EVENT`
    - Note: The event only occurs after the ABAP statement “COMMIT WORK.”
Information delivered by events

- Object Type
- Object Key Value
- Name of the events
  - The event name indicates the status change
- User who triggered the event
  - This user will become the workflow initiator for a workflow start
  - Or the actual agent who completed a workitem
- Additional parameters
  - For example: the required release code of a purchase requisition

What data is delivered?

How is the data delivered?
- The Binding definition in the Workflow Builder stores the data in the
  - Workflow Container
  - Task Container
Event of a Business Object Type

Display Object Type BUS2009

Object type BUS2009 - Purchase requisition item

- Interfaces
- Key fields
- Attributes
- Methods
- Events

- PurchaseReqItem.released
- PurchaseReqItem.reset
- PurchaseReqItem.rejected
- PurchaseReqItem.releaseStepCreated
- PurchaseReqItem.rejection_start
- PurchaseReqItem.rejection_stop
- PurchaseReqItem.significantlyChanged

- Requisition Item Released
- Not Used
- Release of Purchase Requisition Item Refused
- Purchase Requisition Item Release Step Created
- Requisition Rejected
- Cancel Rejection
- Purchase Requisition Item Significantly Changed

Object Type BUS2009: Display Parameters for Event RELEASESTEPCREATED

- Overview

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Obj. Type</th>
<th>First Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReleaseCode</td>
<td>BUS2009</td>
<td>30A</td>
</tr>
</tbody>
</table>

Follow @ASUG365 and #ASUG2013 on Twitter
### SAP Configuration of Events

- It’s all right here (Tools> Business Workflow> Development> Events)

<table>
<thead>
<tr>
<th>Events</th>
<th>Event creation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Set Up with Wizard</td>
</tr>
<tr>
<td></td>
<td>Change to HR Master Data</td>
</tr>
<tr>
<td></td>
<td>SWEHR1 - Linkage: Object Type to HR Infotype</td>
</tr>
<tr>
<td></td>
<td>SWEHR2 - Event - Infotype Operation (SAP)</td>
</tr>
<tr>
<td></td>
<td>SWEHR3 - Event - Infotype Operation (Customer)</td>
</tr>
<tr>
<td>Change documents</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCDO - Overview</td>
</tr>
<tr>
<td></td>
<td>SWED - Define workflow properties</td>
</tr>
<tr>
<td></td>
<td>SWEC - Linkage</td>
</tr>
<tr>
<td></td>
<td>BSVW - Status management</td>
</tr>
<tr>
<td></td>
<td>NACE - Message Control</td>
</tr>
<tr>
<td></td>
<td>AWUW - Logistics Information System (LIS)</td>
</tr>
<tr>
<td></td>
<td>Event Linkages</td>
</tr>
</tbody>
</table>
For HR Events, look at transactions SWEHR1 and SWEHR2 – these are SAP-defined events

Then, if you need to, configure your own events using SWEHR3
SWEHR3 uses the same Object Type, Infotypes, etc as SWEHR2

You define the Rules for when your event is raised in custom functions – make sure you follow the templates!
Change Documents are useful to trigger events when you know a Change Document is written if Master Data changes.

Quick Check - go to TX SE12 and check for the ‘Change Document’ flag on the data element.

If the Change Document box is not ticked, you may have to use a different method.
If you can use Change Docs, there is a nice condition editor to help you complete your mission.

After you select the Field(s) which you need to trigger the event from, you have the ability to format complex conditions prior to raising the event.
Many objects in SAP already use Status Management – Quality Inspection, Projects, Engineering Change Requests...
For customer statuses, you define the status (or use a system (ie: I) status, and it can be in conjunction with others.
**Check Function Modules** - define specific criteria that must be met in order to raise the event.

**Receiver Type Function Modules** will dynamically determine which workflow you need to start.

---

### Change View "Event Type Linkages": Overview

<table>
<thead>
<tr>
<th>Object Category</th>
<th>ObjectType</th>
<th>Event</th>
<th>Receiver Type</th>
<th>Type link...</th>
<th>Enable ev...</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOR Object Type</td>
<td>LIKP</td>
<td>CHANGED</td>
<td>WS96600200</td>
<td></td>
<td></td>
<td>No errors</td>
</tr>
<tr>
<td>BOR Object Type</td>
<td>LIKP</td>
<td>CREATED</td>
<td>WS96600200</td>
<td></td>
<td>✓</td>
<td>No errors</td>
</tr>
<tr>
<td>BOR Object Type</td>
<td>LIKP</td>
<td>CREATED</td>
<td>WS98800048</td>
<td></td>
<td></td>
<td>No errors</td>
</tr>
</tbody>
</table>

---

### Change View "Event Type Linkages": Details

Object Category: BOR Object Type

Object Type: LIKP

Event: CHANGED

Receiver Type: WS96600200

**Linkage Setting (Event Receiver)**

- Receiver Call: Function Module
- Receiver Function Module: SWU_UI_CREATE_VIA_EVENT
- Check Function Module: ZWF_ECC_SHIP_ORDER_CHECK_FM
- Receiver Type Function Module: 
- Destination of Receiver: 

---

Follow @ASUG365 and #ASUG2013 on Twitter
Have the Event send you mail

UserID as Receiver

You get mail in your SAPOffice inbox, and can see the Event Parameters too.

Follow @ASUG365 and #ASUG2013 on Twitter
Output documents are created using SAP’s output determination procedure

Access output management for applications via SAP Reference IMG (TAC SPRO)

Generally, output is created based on condition table for communication purposes between business partners, e.g. print, fax, EDI, or between systems, e.g. ALE

Instead of creating an output document, SAP raises an event in the system
Create output type with transmission medium ‘9’

Maintain condition record and use communication to define event
“Publish & Subscribe” interface available for FI-GL, FI-AP, and FI-AR

Business transaction event linked to business object event, i.e. 00000130 Post Document -> BKPF.CREATED

Template function modules for event processing with parameter interface available for supported relationships

Transaction to access business transaction events is SWU_EWBTE for BTE wizard; you can use special TAC for custom development, e.g. FIBF
BUSINESS TRANSACTION EVENTS

This wizard helps you to define the creation of an event. To do this, you link an event of a business object type with a Business Transaction Event. The occurrence of this Business Transaction Event results in the creation of a new Business Object. If you want to link event with a business object type, enter the required Business Transaction Event and a product that you identify your extension with. The product cannot exist yet; it is created automatically.

Specify the system created event name of the business object type that will be attached to the Business Transaction Event. All entries required for the creation of an event have been made. Choose Complete to make the required changes in the database.

- The function module is created in the function group 2001.
- Event linkage is entered for:
  - Product ZASUG with event 00001030
  - Business object type BKPF, event CREATED

You may have to specify the package or a change request when creating the individual components.
- SAP application creates business transaction event
- Function module processes data to be passed on to business object event
- Business object event is raised
- Event manager determines available active link(s) and starts corresponding workflows
Event of a ABAP Object Type

The Event Parameter can be an object of another type.
### Change View "Event Type Linkages": Overview

<table>
<thead>
<tr>
<th>Event Type Linkages</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object Category</strong></td>
<td><strong>Object Type</strong></td>
</tr>
<tr>
<td>ABAP Class</td>
<td>CL_HRASR00_WF_PROCESS_OBJECT</td>
</tr>
<tr>
<td>ABAP Class</td>
<td>CL_HRASR00_WF_PROCESS_OBJECT</td>
</tr>
<tr>
<td>ABAP Class</td>
<td>CL_HRASR00_WF_PROCESS_OBJECT</td>
</tr>
</tbody>
</table>

### Change View "Event Type Linkages": Details

**Linkage Setting (Event Receiver)**

**Object Category:** ABAP Class  
**Object Type:** CL_HRASR00_WF_PROCESS_OBJECT  
**Event:** COMPLETED  
**Receiver Type:** HR ADMINISTRATIVE SERVICES  
**Receiver Call:**  
**Receiver Function Module:** HR_ASR_START_PERSFILE_CREATION

**Linkage Setting (Event Receiver)**

**Object Category:** ABAP Class  
**Object Type:** CL_HRASR00_WF_PROCESS_OBJECT  
**Event:** COMPLETED  
**Receiver Function Module:** SWM_WI_CREATE_VIA_EVENT_IDB

**Behavior Upon Error Feedback:** System defaults
Transaction SWEQADM is used to schedule the event queue in every system. By spreading out the events, this job provides a better system performance. Based on the size of your system, you can double or triple the number of events per minute.
Workflow with triggering event

Hint:
- The standard binding is for "_EVT_OBJECT" and "_EVT_CREATOR"
- The release-code is an extra parameter
- The "Aggregate" is an attribute of the object
Workitem with Terminating Event

Hint:
One outcome for each (activated) terminating event

Note:
Binding to bring data back to the task container. Another binding defines the transport from the task to the workflow container.
Workflow for a preliminary posted FI document or invoice

A user outside of this workflow has deleted this invoice - End Workflow

A user has changed the invoice - Restart approval

An approver has rejected the invoice - Inform the initiator
Workflow start conditions allow to implement the evaluation of predefined criteria before a workflow instance is started.

- Conditions are defined by event container elements and logical operators and evaluated every time a start mechanism tries to trigger the corresponding workflow.
- If the condition is not fulfilled, the workflow instance is not started.
- Transaction to access workflow start conditions is SWB_COND.
WORKFLOW START CONDITIONS

Select existing start conditions based on standard criteria

Review and change details of start condition
<table>
<thead>
<tr>
<th>Workflow with events</th>
<th>Workflow without events</th>
</tr>
</thead>
<tbody>
<tr>
<td>No delay for the triggering program. Advantage: No impact on triggering program</td>
<td>Triggering program waits for the workflow instance to start. Problems could have an impact on the outside transaction</td>
</tr>
<tr>
<td>Dialog Workitem is completed, when one of the terminating events has occurred. Advantage: Less risk of completing a workitem too early.</td>
<td>Dialog Workitem is completed, when the user backs out the launched transaction or (if configured) clicks on the ‘Complete Workitem’ button.</td>
</tr>
<tr>
<td>Use the “Wait for event” step to react on object status changes, which occurred outside of the workflow process</td>
<td>Users can only influence the process, by executing a workitem.</td>
</tr>
<tr>
<td>Trigger events to communicate with other workflows</td>
<td></td>
</tr>
</tbody>
</table>
BEST PRACTICES

- Connect your SAP Business Workflow to SAP via events
  - Trigger of a new workflow instance
    - Example: A certain Sales Order status was applied
  - Re-act on object status changes, which were caused by activities outside of your workflow
    - Example: Purchase requisition was changed
- Use events to complete a workitem
  - Example: Specific System or User Status was set on a plant maintenance order
- Use the event queue for events, which could occur too often within a short period of time
Use events where ever possible in your workflow design
- Trigger Workflow
- Terminate a workitem
- Wait for an outside event

Activate the Workflow Event Trigger (customizing transport)

Check the events in the test system

Turn off the event trace in the production system
Follow @ASUG365 and ASUG CEO Bridgette Chambers @BChambersASUG on Twitter to keep up to date with everything at ASUG.

Follow the ASUGNews team of Tom Wailgum: @twailgum and Courtney Bjorlin: @cbjorlin for all things SAP.
THANK YOU FOR PARTICIPATING

Please provide feedback on this session by completing a short survey via the event mobile application.

SESSION CODE: ####

For ongoing education on this area of focus, visit www.ASUG.com