11.11

Business Activity Monitoring – Overview

Getting Started

SAP NetWeaver Product Management
Business Process Management

11.11

THE BEST-RUN BUSINESSES RUN SAP™

Events and Event Infrastructure

What are events?

Object status changes, which are ‘published’ throughout the system. Examples: *employee termination*, *purchase order released, etc.*

Available events are defined by an object type in the Business Object Builder. Customers can extend the list of events.

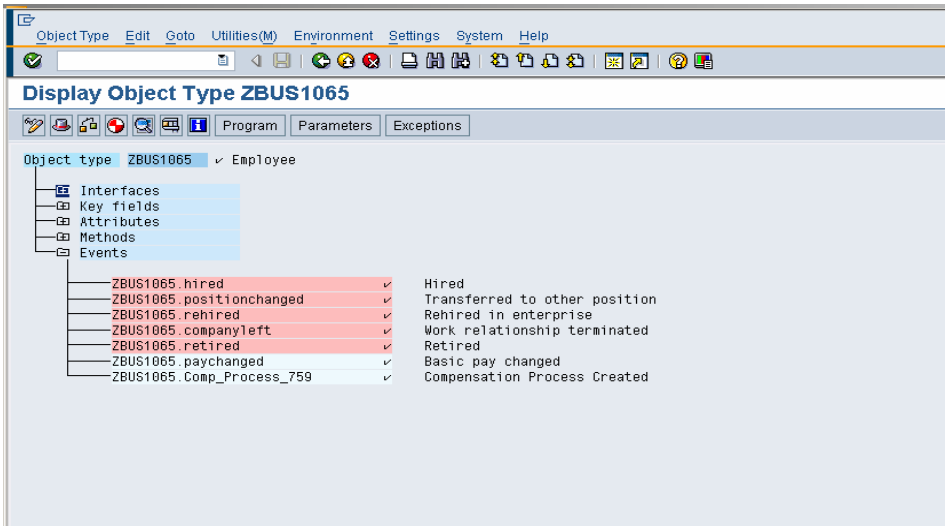
Why are events used?

Each event triggered carries information from its creation context in its event container. This information is available to the receiver of the event and can be used for event-driven control and communication mechanisms.

An event may start, terminate, or continue tasks and workflows.

Events link SAP system applications, external applications, and SAP Business Workflow together in an efficient and flexible manner.

The Events Location

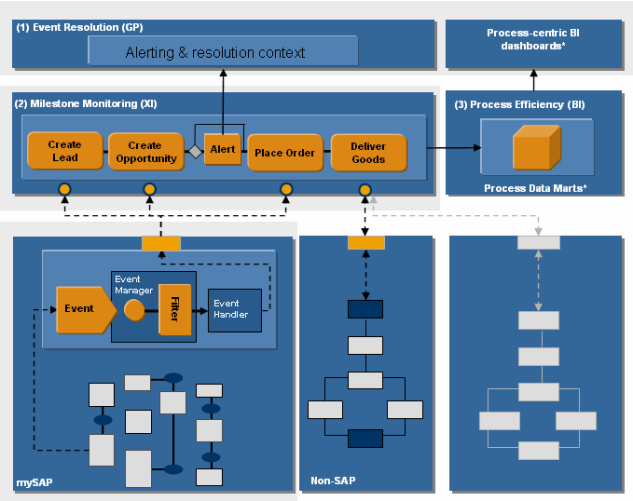


Business event definitions represent Business Objects aspects and are located in the Business Object Builder.

In the graphic the Business Object BUS1065 (‘Employee’) has been extended to include Business Object sub type ZBUS1065 with all the inheritance from and delegation back to the parent object:

- Standard events available to this object are highlighted in pink.
- Extended events are highlighted in white.

Process Milestone Monitoring

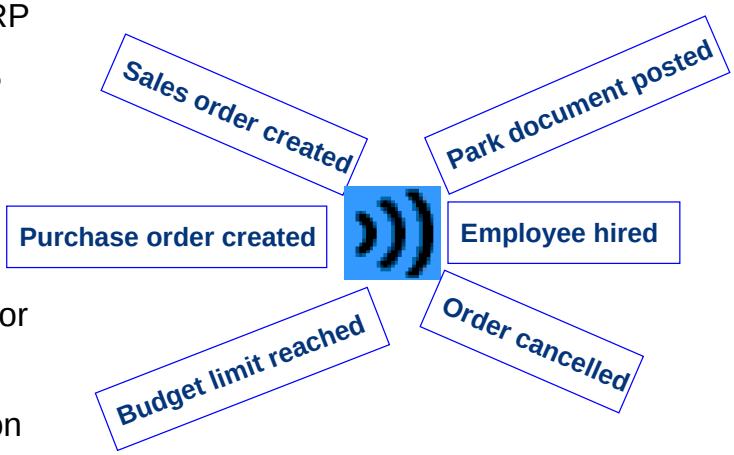


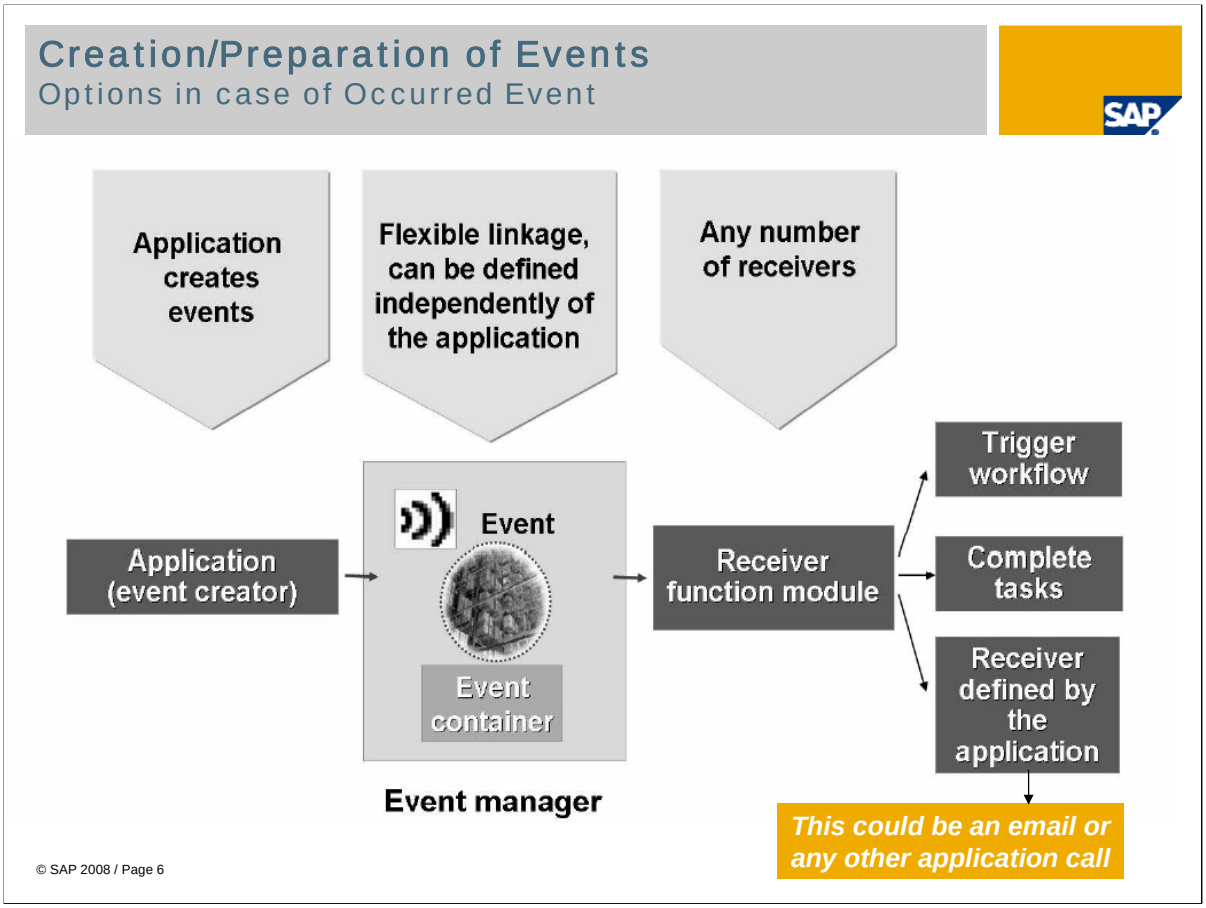
Event Infrastructure

- Infrastructure embedded in SAP Business Suite applications
- Enables collecting, pre-filtering and publishing of events for cross-system use (across SAP systems)

Creation/Preparation of Events
Published Events

- Events occur within SAP applications, such as ERP
- Events provided by SAP occur by default, no configuration is required
- Events are published without consideration of subscribed applications or processes
- If a process or application ‘wants’ to react to certain events, it must ‘subscribe’ to ‘be notified’ for these events





When events are occurring they could trigger workflows, a complete task, an email, or an application defined function call and so on.

Each event carries event container data.

Existing Events

Customers can use local events already in-place to propagate events across system boundaries using messages

Modification-free event-enabling of SAP

Local in-place events are triggered throughout the SAP system:

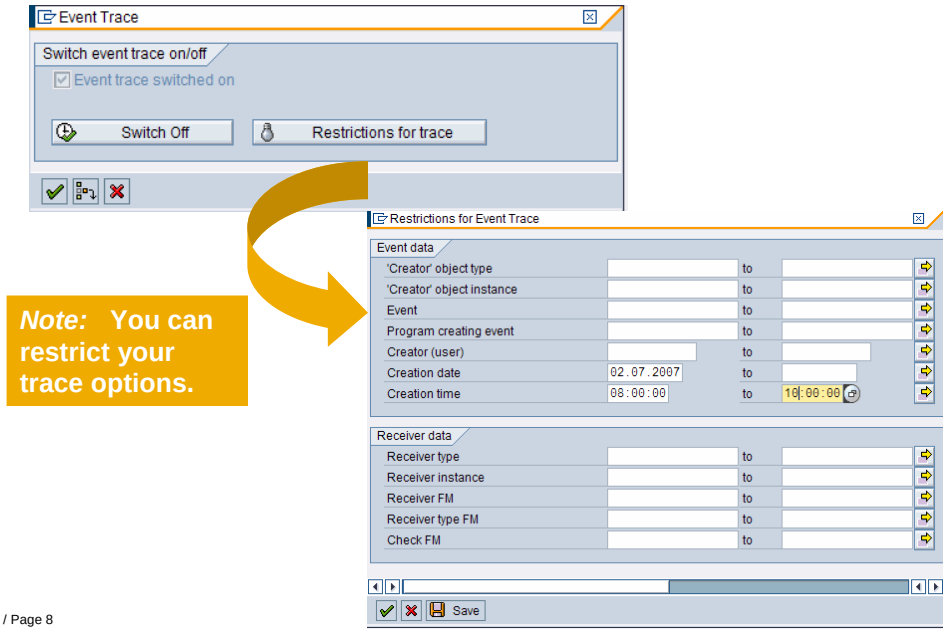
- ECC600: ~1000 Events defined
- SRM/CRM: ~500 Events defined
- SCM: ~250 Events defined

Almost 1750 events are already available throughout SAP systems.

Customers can use these out-of-the-box events without touching SAP core functionality.

Steps for Finding Existing Events

1. Select an SAP application system and turn on the event trace via SWELS



Note: **You can restrict your trace options.**

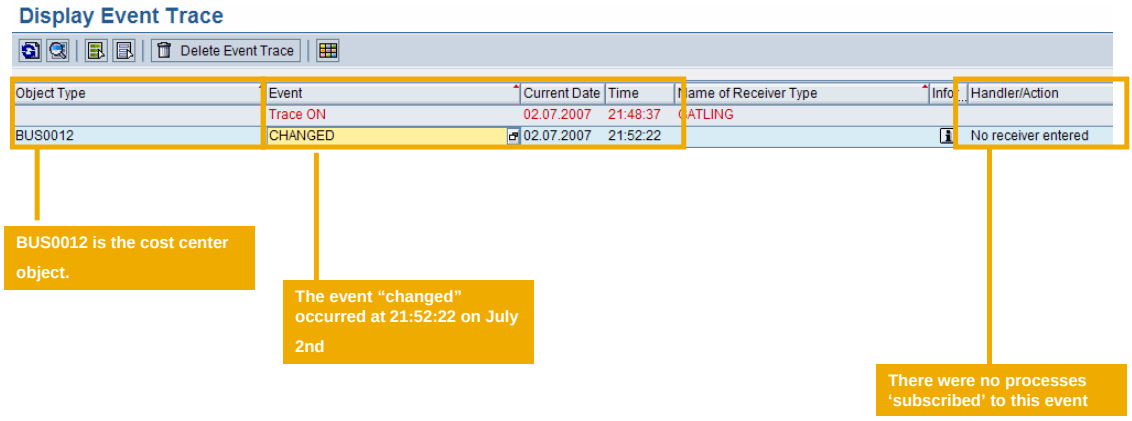
Before events are propagated, customers need to find out where these events are located.

Event trace SWELS can be turned on to find out which transactions can raise which events respectively.

After turning on this trace customers have to check the result soon after entering the transaction code, otherwise the connection between the transaction code and events will be lost.

Steps for Finding Existing Events

2. Once the trace is on, users should execute application functions then review the trace in transaction SWEL.

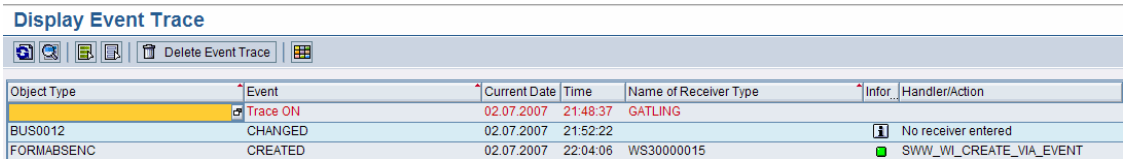


With the event trace result customers can see:

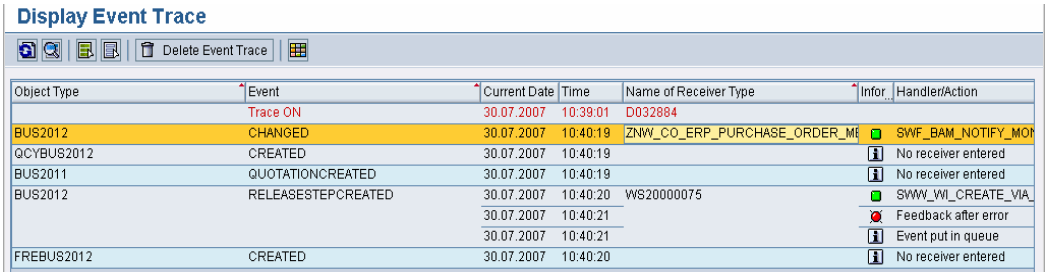
- User Gatling changed the cost center (with transaction SWO; here BUS0012 is the cost center object)
- An entry in this table can be seen, which means changing the cost center can raise an event
- No receiver is currently 'subscribed' to this event, this means an event is just raised but no workflow/task/function call and so on is triggered

Events Tracing

Example 1: the CREATED event for object FORMABSENC was raised. WS30000015 is a process that was listening and this process was started when the event occurred



Example 2: the Changed Event for object BUS2012 (purchase order) was raised. ZNW_CO_ERP_PURCHASE_ORDER_MIT is the proxy which is defined as the receiver for this event



Example 1:

Once the Notification of Absence is created, a workflow will be triggered as a receiver for this event.

Example 2:

Here is another example:

- Event trace is switched on. Purchase order is created with transaction ME21.
- In the result table can be seen the following:
  - Event ‘purchase order changed’ is raised. The receiver is the proxy ‘ZNW_CO_ERP___’ which is defined with transaction SWF_BAM (SWF_BAM is the transaction which can publish local events to global events –> for details please refer to the next chapter).
  - Event ‘Quotation created’ is raised. No receiver is defined for this event.
  - Event ‘Release step created’ is raised. A workflow ‘WS20000075’ is defined as receiver for this event
  - Event ‘Purchase order created (FREBUS2012 is the subtype of BUS2012)’ is raised. No receiver is defined for this event.

How to Create New Events

New events can be added by configuration, for example:

- Change documents or message control (e.g. Sales and Purchasing)
- Status management (e.g. Plant Maintenance)
- Business transaction documents (e.g. Financial Accounting)
- Post processing framework (CRM)
- Logistic information system
- Links to object types, information types and sub types (e.g. Human Resources), etc.

New events can also be added using custom code.

Pre-defined events are triggered throughout the SAP system.

Custom events can be generated from anywhere customers require in code:

- ABAP Programs
- Function Modules
- User Exits
- BAPIs
- BADIs
- Classes
- Business Object Methods

This slide, and those following, give information about how to create events.

If customers need an event (for example, a goods receipt) which does not exist yet in an SAP system, there are different options to add them by configuring **without coding**.

How to Trigger Events – Function Module

SWE_EVENT_CREATE

- → OBJTYPE – the triggering business object
- → OBJKEY – the object key
- → EVENT – the event to be raised
- ← EVENT_ID – ID number of event when event f(x) established
- Explicit **Commit Work** must be executed for the event to be recognized – (allowing asynchronous RFC calling the receiver function module to execute)

SAP_WAPI_CREATE_EVENT

Special Cases

- SWE_EVENT_CREATE_IN_UPD_TASK
- SWE_EVENT_CREATE_FOR_UPD_TASK

SWE_EVENT_CREATE is the current incumbent, the champ – the most widely used.

Import parameters for SWE_EVENT_CREATE:

OBJTYPE SWETYPECOU-OBJTYPE - Type of the triggering business object.

OBJKEY SWEINSTCOU-OBJKEY - Concatenated, object type-specific key of the triggering object. The reference to the triggering object is created internally from this information and written to the event container under the element ID _Evt_Object.

EVENT SWETYPECOU-EVENT - ID of the event. The event must be defined for the triggering object type.

Other parameters are available. CREATOR has been known to be a tricky one. A local variable of type SWHACTOR should be used to pass on any user IDs. Passing a user ID directly to the function module parameter has been known to cause problems in the past.

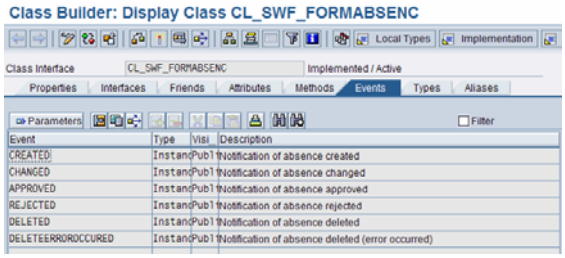
Import tables:

EVENT_CONTAINER SWCONT - Persistent event container of the event.

The event container is only passed if event parameters have been defined in addition to the predefined elements.

Triggering Events Using an ABAP Class

Requires explicit Commit Work



CL_SWF_EVT_EVENT→GET_INSTANCE

IF_SWF_EVT_EVENT→RAISE

Set Parameters with:

- **IF_SWF_IFS_PARAMETER_CONTAINER→SET**

Special Case:

- **CL_SWF_EVT_EVENT→RAISE_IN_UPDATE_TASK**

For the 'do-it-yourselfer', customers can call the event class or interface directly.

Customers need to process specific instances of the object with parameters then raise the event or raise_in_update_task.

*---- *get a instance for the event*

```
CALL METHOD cl_swf_evt_event=>get_instance
EXPORTING
    im_objcateg          = g_obj_category          (SWF_CLSTYP → workflow object category like BO or CL)
    im_objtype           = l_objtype               (SWO_OBJTYP → Object or Class name)
    im_event             = l_event                 (SWO_EVENT → the event you want to raise)
    im_objkey            = l_objkey                (SIBFINSTID – Key of object or class)
    im_event_container   = l_event_container (interface if_swf_ifs_parameter_container → container data elements)

RECEIVING
    re_event                        = l_event_ref    -→ (interface IF_SWF_EVT_EVENT)
```

Triggering Events Using Change Documents

Creating Change Document Linkage:

- Transaction SWEC
- Establishes a relationship between a change document object and a Workflow through the Business Object event

Change View "Events for Change Document": Overview

New Entries

Events for Change Document

Change doc. object	ObjectCate...	Business obj. type	Event	On Create	On Change	On Delete
KRED	BOR Obj ...	LFA1	RLSFLAGFORDELETION	○	◉	○
MATERIAL	BOR Obj ...	ZAB_MARA	WRKSTCHANGED	○	◉	○
MATERIAL	BOR Obj ...	A00MARA	CHANGED	○	◉	○
OBJKT	BOR Obj ...	ECO		◉	○	○
OBJKT	BOR Obj ...	ECO		◉	○	○
PPESCMPV	BOR Obj ...	BUS1176002	CHANGE	○	◉	○
PPESGENN	BOR Obj ...	BUS1197001	CHANGE	○	◉	○

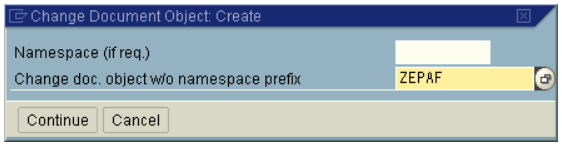
The following system behavior can be set by making entries in the relevant control tables of change document management: Status changes of objects that are logged as change documents are reported automatically as events as well.

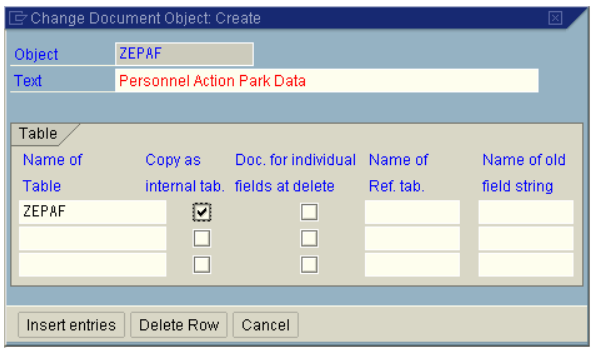
In the first view Change "Events for Change Document": Overview, create a new entry. To do this:

- Select the function **New entries**.
- Specify a **change document object** in the field Change doc. object.
- In the field Business Object type, enter the **object type whose status change is to be indicated by the event created**.
  - Use the F4 input help. The object types offered by the system in the F4 input help usually have the same key structure as the change document object.
- Enter the **ID of the event to be created in the field Event**. The event must be defined for the specified object type. The name is taken automatically from the object type definition.
- Select whether the event is to be **created upon change, create or delete actions**.

Changes to fields in various tables are generally logged with a change document object. Only one of these tables is the main table in the sense that the relevant change document is written with its key. The create, change and delete actions are always associated with the main table. Depending on how the change documents are used in the application, it may be that **no** change document is written upon create or delete actions.

Triggering Events Using Change Documents





Create Change Document

- Transaction SCDO
- Create
- Enter change doc. name
- Continue
- Enter object name
- Enter description
- Enter table name (s)
- Check flags (if necessary)
- Insert entries

■Choose the menu option **Create**.

■Enter a **name for the change document object** which is to be created. It can be any name starting with "Y" or "Z" (customer name area).

■Choose **Continue**. A new window for inputting the associated tables appears.

■Enter a **descriptive short text** for the change document object.

■Make the following entries for each table whose changes are to be logged in the change document for this change document object:

- **Table name** - Name of the table, as defined in the Dictionary

- **Copy as internal table flag** –

If the change data is to be passed in an internal table (multiple case), this field should be marked.

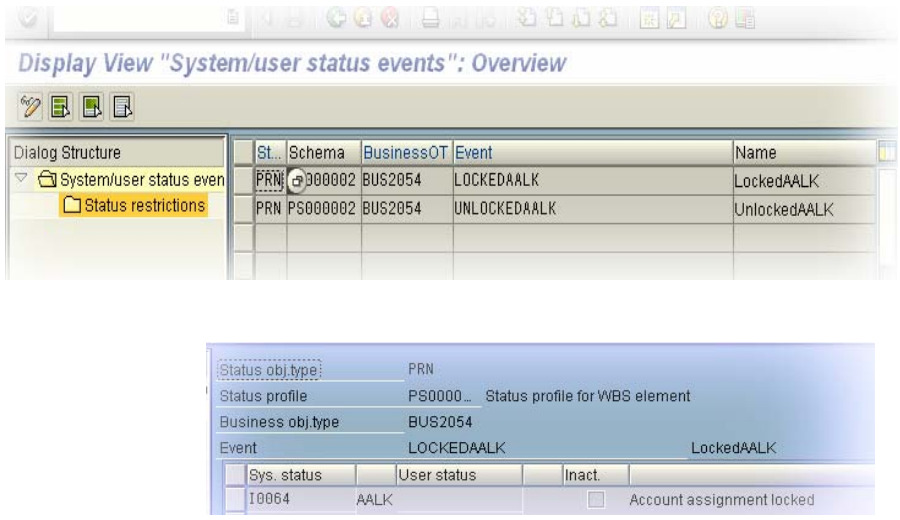
If it is not marked, the change data is passed in a work area (single case).

- **Document for individual fields at delete flag** –

If separate log entries are needed or each field when data are deleted, this field should be marked.

If it is not marked, the deletion of all relevant fields is entered in one document item.

Triggering Events Using Status Change



Creates an association between an object status and a workflow

Transaction BSVW (BSVV, BSVX, BSVY, BSVZ)

Creates an association between an object status (per example a production order release or a creation of a maintenance order) and a workflow.

Column Headings (BSVZ) (BSVY omits the schema column)

Status Object Type/Schema/Business Object Type/Business Object Event/Business Object Event Name

Before setting up a status change event, the following is needed:

1. The event set up on the target Business Object
2. The Business Object and the status object must have the same key field(s)

To set up a status change event:

1. Status object type
2. Schema (only for event creation using application status).
3. Business Object type
4. Event ID
5. Status restrictions (if necessary) – so that the event is not triggered every time an object's status changes

In this situation, Project Systems WBS element events LOCKEDAALK and UNLOCKEDAALK

Triggering Events Using Message Control

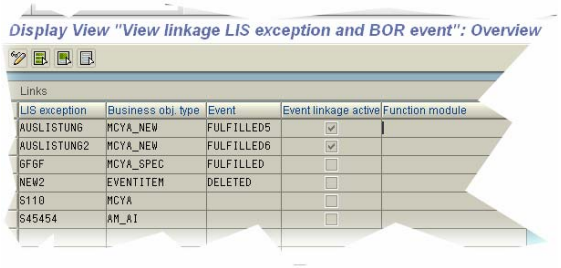
- **Create an output type and specify the transmission medium as type 9 (workflow event) in transaction NACE**
- **Set up an output type (i.e. EVEN for "trigger event") in an application's output control**
- **Use program RVNSWE01 with form routine CREATE_EVENT. This is the standard messaging processing program for generating the workflow event or creating a customized one that will generate the event**
- **The event ID and object type that describe the event created need to be assigned to a message application in the table NAST**

Triggering Events Using Logistics Information System Exceptions

Trigger a workflow from LIS exceptions

- Tolerance levels are defined in LIS
- Exceeding the tolerance levels generates an exception
- LIS exception triggers a workflow

Set up linkage with transaction AWUW



In conjunction with the event creation, the system fills the event container with the following information about the exception at runtime:

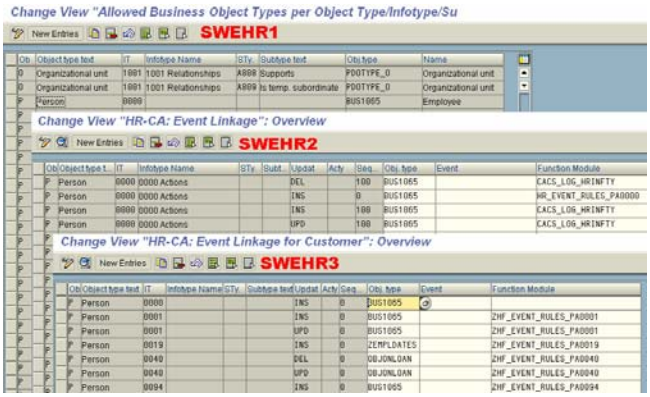
■The characteristic values when the *exception* was triggered

■The key figures when the *exception* was triggered

Triggering Events Using HR Master Data

Key transactions

- SWEHR1 – links object type/info type/sub type to BO
- SWEHR2 – OT/IT/ST to BO event or function module
- SWEHR3 – Customer OT/IT/ST to BO event or function



For customers who are not familiar with HR, *infotypes* are HR's way of storing data. These basically segment the table data to an object level.

Transaction SWEHR1 creates the linkage between object, *infotype* and *subtype* to a business object. The data get stored in table T777IBO.

Transaction SWEHR2 creates the linkage between:

object type

infotype

subtype

update operation (processing at the database level)

activity ID (primarily used with training and event management)

activity sequence to a Business Object and event or function module (which can allows to put criteria on event triggering like a check function module).

- Activity IDs
- INS 'Insert' Create record
- DEL 'Delete' Delete record
- UPD 'Update' Change record
- REQ 'Request' Create locked record
- APP 'Approve' Unlock locked record
- APC 'ApproveCancel' Lock valid record
- RQC 'RequestCancel' Delete locked record
- * 'All other opcodes' All other operations

Transaction SWEHR3 is similar to SWEHR2 except it allows customers to override the settings of SWEHR2.

Triggering Events Using Business Transaction Events

Financial applications have subscribe and publish interfaces to alert of specific financial activity

Business Object events can be generated from this interface

Business Transaction Events and Business Object Events are already defined

- Just need to be linked together
- The workflow wizard for this can be used
- After wizard creation, the event status needs to be set to “released” in the Business Object Builder

■**G/L accounting** and **accounts receivable/payable** accounting provide publish and subscribe interfaces. Open FI

■When a business transaction event (alert) occurs, the business object event function is called and a BO event is published.

■Transaction **BF01** Displays the **business transaction events** in the system

■Linked together with function module SWE_OPENFI_COUPLING_CREATE with data stored in table TBE24.

■Existing business products can be viewed with transaction BF24.

Event Creation Wizards

Generating event entries with a wizard interface

- It is possible to generate event creation processes using a wizard interface.
- The wizard attempts to simplify the processes of creating new events.

A few wizards are available
Please note: they do not cover all aspects of event creation

- **Change documents (Transaction SWU_EWCD)**
- **Logistics Information System (Transaction SWU_EWLIS)**
- **Business Transaction Events (Transaction SWU_EWBTE)**

Menu Path

- ***Tools → Business Workflow → Development → Definition Tools → Events → Event Creation → Set Up with Wizard***

Summary

- An event represents a status change of object, for example, *sales order created*, *employee hired*, etc.
- Event infrastructure is embedded in SAP applications for collection, pre-filtering and publication of events for cross-system usage.
- Local events can be generated all over the SAP system directly or using configuration or customer code.