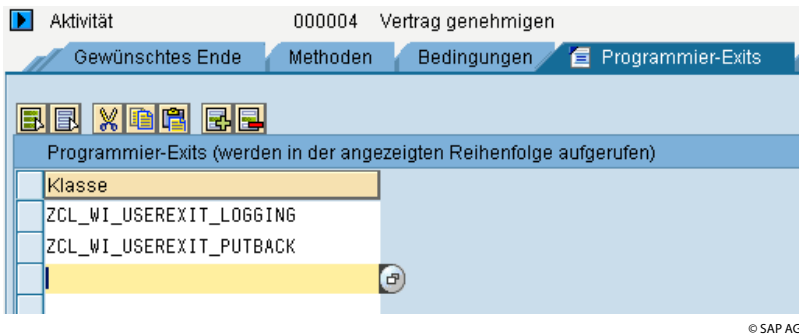


## 10.6 Programmier-Exits für Workitems

### 10.6.1 Das Interface IF\_SWF\_IFS\_WORKITEM\_EXIT

Am Schritt einer Workflow-Definition im Reiter »Programmier-Exits« können verschiedene Klassen eingetragen werden, die alle das Interface IF\_SWF\_IFS\_WORKITEM\_EXIT unterstützen müssen.



**Abb. 10-17**  
Workitem-Exit

Typische Aufgaben, die in Userexits ausgeführt werden, sind:

- anwendungsspezifisches Logging
- Mailversand
- Manipulieren von Workitems

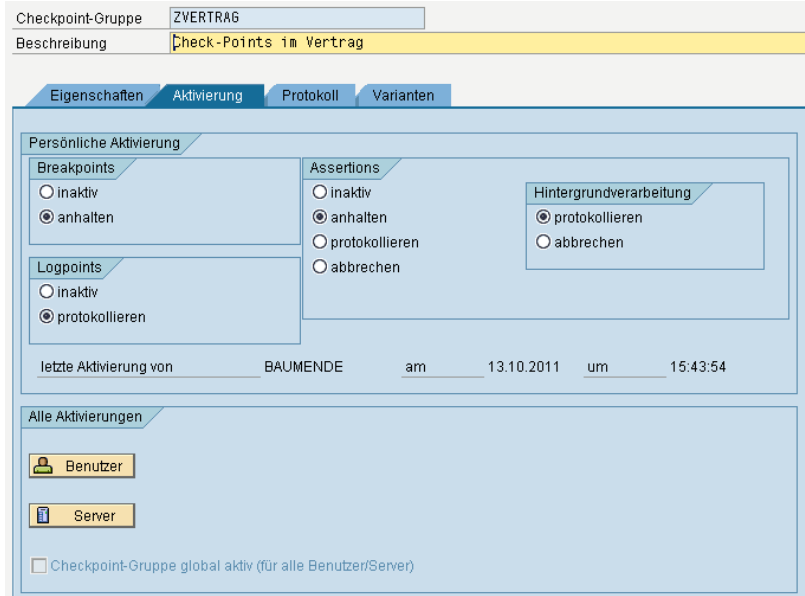
### 10.6.2 Logging mit CHECK-POINT-Gruppe

Das Logging wird am Beispiel einer CHECK-POINT-Gruppe ZVERTRAG demonstriert, die mit TA SAAB angelegt und aktiviert wird (s. Abb. 10-18).

Als Beispiel wird die Klasse ZCL\_WI\_USEREXIT\_LOGGING betrachtet, die verschiedene Statusübergänge an Workitems protokolliert.

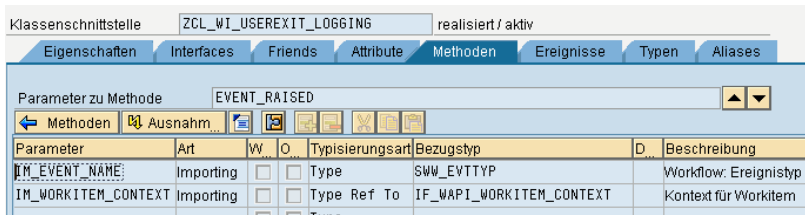
Durch Verwendung des Interface erbt die Klasse die Methode EVENT\_RAISED mit folgenden Importparametern (s. Abb. 10-19).

**Abb. 10-18**  
CHECK-POINT-Gruppe  
ZVERTRAG mit  
Aktivierungen



© SAP AG

**Abb. 10-19**  
Parameter  
im Workitem-Exit



© SAP AG

Die möglichen Events sind:

**Abb. 10-20**  
Mögliche Events für den  
Aufruf des Workitem-Exits



© SAP AG

Alle notwendigen Angaben über das Workitem, den zugehörigen Workflow, die Containerelemente und -objekte usw. bekommt man bei Bedarf aus der Referenz auf den Workitem-Kontext:

| Methoden             | Art      | Nur...                   | M... | Beschreibung  |
|----------------------|----------|--------------------------|------|---|
| DO_COMMIT_WORK       | Insta... | <input type="checkbox"/> |      | 'Commit Work' registrieren                                |
| GET_HEADER           | Insta... | <input type="checkbox"/> |      | Header  |
| GET_PROPERTY         | Insta... | <input type="checkbox"/> |      | Wert einer Workflow-Eigenschaft (Grunddaten+Schrittdaten) |
| GET_RULE_CONTEXT     | Insta... | <input type="checkbox"/> |      | Liefert den aktuellen Regelkontext                        |
| GET_RULE_RESULT      | Insta... | <input type="checkbox"/> |      | Liefert das Ergebnis der aktuellen Regelauflösung         |
| GET_STATE_TRANSITION | Insta... | <input type="checkbox"/> |      | Statusübergang  |
| GET_TASK_ID          | Insta... | <input type="checkbox"/> |      | Name der Aufgabe  |
| GET_WF_CONTAINER     | Insta... | <input type="checkbox"/> |      | Container des Workflow                                    |
| GET_WI_CONTAINER     | Insta... | <input type="checkbox"/> |      | Container des Workitem                                    |
| GET_WORKFLOW_ID      | Insta... | <input type="checkbox"/> |      | Workflow-Kennung  |
| GET_WORKFLOW_TASK_ID | Insta... | <input type="checkbox"/> |      | Name des Workflow   |
| GET_WORKFLOW_VERSION | Insta... | <input type="checkbox"/> |      | Version des Workflow                                      |
| GET_WORKITEM_ID      | Insta... | <input type="checkbox"/> |      | Workitem-Kennung  |
| IS_ENQUEUED          | Insta... | <input type="checkbox"/> |      | Workitem steht exklusiv dem Aufrufer zur Verfügung        |
| SET_DEBUG            | Insta... | <input type="checkbox"/> |      | Debugging ein-/ausschalten                                |
| SET_MESSAGE_TO_LOG   | Insta... | <input type="checkbox"/> |      | Schreibt Meldung ins Workitem-Protokoll                   |
| SET_TRACE            | Insta... | <input type="checkbox"/> |      | Trace ein-/ausschalten                                    |

© SAP AG

**Abb. 10-21**

Interface  
IF\_WAPI\_WORKITEM\_...  
CONTEXT

```

METHOD if_swf_ifs_workitem_exit-event_raised.
  DATA: ls_wi_header TYPE swr_wihdr,
         ls_statra   TYPE sww_statra,
         lv_datum_txt(10) TYPE c,
         lv_zeit_txt(8) TYPE c,
         lv_subkey(60) TYPE c.
  ls_wi_header = im_workitem_context->get_header( ).
  ls_statra   = im_workitem_context->get_state_transition( ).
  WRITE sy-datum TO lv_datum_txt.
  WRITE sy-uzeit TO lv_zeit_txt.
  CONCATENATE ls_wi_header-wi_id
              lv_datum_txt
              lv_zeit_txt
              im_event_name
              INTO lv_subkey SEPARATED BY space.
  if ls_statra-TAR_STAT is initial.
    concatenate lv_subkey
               ls_wi_header-wi_stat
               into lv_subkey separated by space.
  else.
    concatenate lv_subkey
               '-->' ls_statra-TAR_STAT
               into lv_subkey separated by space.
  endif.

```

**Listing 10-8**

Workitem-Exit mit  
Logging

```

wait up to 1 seconds.
LOG-POINT ID      zvertrag
              SUBKEY lv_subkey
              FIELDS ls_wi_header-wi_rh_task
                   ls_wi_header-wi_id
                   im_event_name.
ENDMETHOD.

```

Das folgende Protokoll zeigt, welche Ereignisse für ein Workitem durchlaufen wurden, wobei bei Statusübergängen der neue Status angezeigt wird.

**Abb. 10-22**  
Anzeige der Log-Einträge  
zum Userexit

| ID/SUBKEY/Programm/Prozedur                         | Datum      | Uhrzeit  | Zähler |
|---|------------|----------|--------|
| ZVERTRAG  |            |          | 9      |
| 000000011116  | 23.04.2012 | 15:11:06 | 1      |
| ZCL_WI_USEREXIT_LOGGING=====CP                      |            |          | 1      |
| IF_SWF_IFS_WORKITEM_EXIT-EVENT_RAISED               |            | ZCL_WI_  | 1      |
| Include: ZCL_WI_USEREXIT_LOGGING=====CM00 Zeile: 26 | 23.04.2012 | 15:11:07 | 1      |
| 000000011116  | 23.04.2012 | 15:11:07 | 1      |
| 000000011116  | 23.04.2012 | 15:11:08 | 1      |
| 000000011116  | 23.04.2012 | 15:11:09 | 1      |
| 000000011116  | 23.04.2012 | 15:11:10 | 1      |
| 000000011116  | 23.04.2012 | 15:11:11 | 1      |
| 000000011116  | 23.04.2012 | 15:11:19 | 1      |
| 000000011116  | 23.04.2012 | 15:11:20 | 1      |
| 000000011116  | 23.04.2012 | 15:11:21 | 1      |

© SAP AG

Mit dieser Übersicht ist es einfach, weitere Dinge wie Mailversand zu programmieren.

### 10.6.3 Automatisches Zurücklegen eines Workitems

Wenn Workitems begonnen und abgebrochen werden, dann sind sie im Status STARTED und damit nicht mehr für alle ursprünglichen Empfänger sichtbar. Dies kann zu Arbeitsverzögerungen führen, wenn Mitarbeiter vergessen, begonnene (angenommene), aber nicht abschließend bearbeitete Workitems in den Bearbeiterpool zurückzulegen. Der hier vorgestellte Userexit erledigt das automatisch.

Leider ist das nicht ganz so einfach, wie man vermuten könnte. Ein direkter Aufruf eines FBS zum Zurücklegen eines Workitems (z.B. SWL\_ADM\_BACK) schlägt fehl, da das während eines Statusüberganges nicht erlaubt ist. Es wird daher ein asynchrones Verfahren eingesetzt, wozu die Klasse ZCL\_WI\_USEREXIT\_PUTBACK benutzt wird. Das Verfahren funktioniert für beliebige Dialog-Workitems, da es keinerlei Anwenderdaten liest. Das zurückzulegende Workitem wird in

eine Tabelle ZPUTBACKWI geschrieben, danach wird im Hintergrund der Report ZAO\_PUTBACK\_WI via SIMPLE\_BATCH\_JOB\_SUBMIT gestartet, der alle Einträge dieser Tabelle liest und die Workitems mit o.g. FBS zurücklegt.

```

METHOD IF_SWF_IFS_WORKITEM_EXIT-EVENT_RAISED.
  DATA: ls_wi_header  TYPE swr_wihdr,
         lv_wi_stat   TYPE sww_wistat,
         lv_wi_id     TYPE sww_wiid,
         lo_wi_cont   TYPE REF TO if_swf_ifs_parameter_container,
         ls_witab     TYPE zvmk_put_back_wi.

  lo_wi_cont     = im_workitem_context->get_wi_container( ).

  ls_wi_header   = im_workitem_context->get_header( ).
  lv_wi_id       = ls_wi_header-wi_id.
  lv_wi_stat     = ls_wi_header-wi_stat.

* Zurücklegen des WI asynchron im Batchjob bei Statuswechsel
* zu Status STARTED oder COMMITTED (nach Abbrechen dies WI)

IF im_event_name = 'AFT_EXEC' AND
  ( lv_wi_stat = 'STARTED' OR lv_wi_stat = 'COMMITTED' ).

  ls_witab-wi_id = ls_wi_header-wi_id.

  MODIFY zputbackwi FROM ls_witab.
  CALL FUNCTION 'SIMPLE_BATCH_JOB_SUBMIT'
    EXPORTING
      program          = 'ZAO_PUTBACK_WI'
*     VARIANT          = ' '
    EXCEPTIONS
      submit_failed   = 1
      OTHERS          = 2.

ENDIF.
ENDMETHOD.

```

**Listing 10-9**

*Workitem-Exit für das  
Zurücklegen eines  
Workitems*

```

REPORT ZAO_PUTBACK_WI.

DATA:   lv_wi_stat TYPE sww_wistat,
        ls_witab  TYPE zputbackwi,
        lt_witab  TYPE STANDARD TABLE OF zputbackwi.

*-----
START-OF-SELECTION.
  " Alle zurückzulegenden WIs holen
  SELECT * FROM zputbackwi INTO TABLE lt_witab.
  WAIT UP TO 1 SECONDS.
  LOOP AT lt_witab INTO ls_witab.

```

**Listing 10-10**

*Bacht-Report zum  
Zurücklegen eines  
Workitems*

```

SELECT SINGLE wi_stat INTO lv_wi_stat
FROM swwihead
WHERE wi_id = ls_witab-wi_id.

IF sy-subrc = 0 AND
( lv_wi_stat = 'STARTED' OR lv_wi_stat = 'COMMITTED' ).

CALL FUNCTION 'SWL_WI_ADM_BACK'
EXPORTING
    wi_id                = ls_witab-wi_id
    authorization_checked = 'X'
EXCEPTIONS
    replace_failed      = 1
    enqueue_failed     = 2
    OTHERS              = 3.
IF sy-subrc <> 0.
    " WI ist zwar in der Tabelle und hat den richtigen
    " Status, kann aber nicht zurückgelegt werden,
    " also warten bis zum nächsten Mal
ELSE.
    DELETE FROM zputbackwi
           WHERE wi_id = ls_witab-wi_id.
    commit work.
ENDIF.
ELSE.
    DELETE FROM zputbackwi
           WHERE wi_id = ls_witab-wi_id.
    commit work.
ENDIF.
ENDLOOP.

```

Nach dem Zurücklegen des Workitems, an dessen Schritt die Klasse ZCL\_WI\_USEREXIT\_PUTBACK eingetragen ist, dauert es ein bis zwei Sekunden, dann steht das Workitem wieder auf READY.

Mit diesem Userexit kann man jeden beliebigen Dialogschritt in jedem beliebigen Workflow mit der Funktion des automatischen Zurücklegens versehen.

## 10.7 Fortgeschrittene Workflow-Definition

### 10.7.1 Lokale Ereignisse und Ausnahmen

Unter den versionsabhängigen Grunddaten eines Workflow-Musters lassen sich auf dem Reiter *Lokale Ereignisse* sogenannten lokale Ereignisse und lokale Ausnahmen definieren. Diese sind ausschließlich dem jeweiligen Workflow-Muster ab der Version bekannt, in der sie definiert wurden.