



# Massengenehmigung mit der UWL(Universal- Worklist)

18 December 2009 - B. Sc. Alexander Graf

**Applies to:**

SAP ECC 6.0 Level 2 Backendsystem and SAP NetWeaver Portal 7.0. Patchlevel 19 with EHP 3 package

**Einleitung:**

Über die UWL(Universal-Worklist) des SAP NetWeaver Portals ist es möglich Anträge wie z.B. einen Abwesenheitsantrag zu genehmigen. Wird ein Abwesenheitsantrag im SAP Standard erstellt und ist im Customizing ein Workflow zur Abwesenheitsart hinterlegt, wird ein Workitem erstellt und dieses z.B. dem Vorgesetzten zugesendet. Dieses Workitem wird in der UWL des Vorgesetzten angezeigt und kann dort direkt bearbeitet werden.

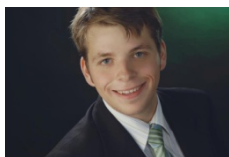
Beim SAP Standardworkflow für Belege(WF12300111) wird dazu ein WebDynPro geöffnet und der Vorgesetzte kann sich den Abwesenheitsantrag zur Entscheidungsfindung detaillierter betrachten. Bei einer Masse von zu genehmigen Anträgen nimmt diese Vorgehensweise aber schnell viel Zeit in Anspruch. Aus diesem Grund bietet sich eine Massengenehmigung für Belege wie die Abwesenheit(Englisch: Leave Request) oder Zeitbuchungskorrektur(Englisch: Clock-In / Out Corrections) an.

Über die UWL(Universal Worklist)-Massengenehmigung soll Vorgesetzten ermöglicht werden, mehrere Abwesenheitsanträge bzw. Zeitbuchungskorrekturanträge gleichzeitig zu bearbeiten.

**Author(s):** Alexander Graf

**Company:** mindsquare GmbH ([www.mindsquare.de](http://www.mindsquare.de))

**Created on:** 18 December 2009

**Author Bio**

**Alexander Graf** ist bei der **mindsquare GmbH - SAP NetWeaver Development Consultant** - neben seiner technischen Expertise im ABAP- und Java-Umfeld hat er sich fachlich unter anderem auf Portalthemen, LO, FI, CO, HCM und Workflow Management bei der mindsquare spezialisiert.



**Inhaltsverzeichnis:**

1 PORTAL-CUSTOMIZING ..... 4

2 CUSTOMIZING UWL; XML-DATEI: UWL.STANDARD ..... 4

3 CUSTOMIZING UWL; XML-DATEI: UWL.WEBFLOW ..... 5

4 FUNKTIONSWEISE DES FUNKTIONSBAUSTEIN ZHR\_LR\_SET\_STATUS..... 7

## 1 Portal-Customizing

Die UWL kann im SAP NetWeaver Portal über XML-Dateien gecustomized werden. Diese können zur Bearbeitung aus dem Portal unter dem Navigationspfad: "System Administration -> System Configuration -> Universal Worklist & Workflow -> Universal Worklist - Administration -> Manage Item-Typen and Definitionen " heruntergeladen werden.

Das Customizing für die Massengenehmigung muss in den XML-Dateien uwl.standard und uwl.webflow vorgenommen werden.

Alle in den Workflow-Schritten verarbeiteten Aufgaben, welche in der UWL dargestellt werden sollen, müssen der UWL in den XML-Dateien bekannt gemacht werden.

## 2 Customizing UWL; XML-Datei: uwl.standard

In der Standard-XML-Datei wird das Erscheinungsbild bzw. das Layout der UWL definiert. Das Standard-Layout ist die "Default-View". Damit dort die Spalten zur Massengenehmigung und ein entsprechender Button, um die Massenabfertigung durchzuführen, erscheint, muss die "Default-View" in der XML-Datei erweitert werden. Um zusätzliche Spalten mit Checkboxen für "genehmigen" bzw. "ablehnen" anzuzeigen, müssen diese als Attribute in der "Default View" definiert werden:

```
[..]
<DisplayAttribute name="rejectCol" type="checkbox" width="" sortable="no" format="default" actionRef="rejectedAction" hAlign="CENTER" vAlign="TOP" maxTextWidth="0" headerVisible="yes">
<Descriptions default="">
  <ShortDescriptions>
    <Description Language="de" Description="Ablehnen"/>
  </ShortDescriptions>
  <LongDescriptions>
    <Description Language="de" Description="Ablehnen"/>
  </LongDescriptions>
</Descriptions>
</DisplayAttribute>
<DisplayAttribute name="approveCol" type="checkbox" width="" sortable="no" format="default" actionRef="approveAction" hAlign="CENTER" vAlign="TOP" maxTextWidth="0" headerVisible="yes">
<Descriptions default="">
  <ShortDescriptions>
    <Description Language="de" Description="Genehmigen"/>
  </ShortDescriptions>
  <LongDescriptions>
    <Description Language="de" Description="Genehmigen"/>
  </LongDescriptions>
</Descriptions>
```

```
</DisplayAttribute>  
[..]
```

Anschließend müssen die neuen Attribute als Spalten in der Eigenschaft "columnOrder" deklariert werden.

```
[..]  
<View name="DefaultView" selectionMode="SINGLESELECT" width="98%" supportedItem-  
Types="uwl.task" columnOrder="subject, isEscalated, creatorId, createdDate, priority, attachment-  
Count, dueDate, status, approveCol, rejectCol" .....  
[..]
```

Damit die Entscheidungen zu jedem einzelnen Workitem(jedem einzelnen Antrag) auch in der Masse durchgeführt werden können, muss ein zusätzlicher Button in der UWL definiert werden. Dazu wird unter der Eigenschaft Actions der "Default View" die folgende, bereits global in der XML-Datei definierte Action, eingebunden.

```
[..]  
<Actions>  
<Action reference="submitUserDecisions"/>  
[..]
```

Über diese Aktionen werden automatisch von der UWL die in der XML-Datei "uwl.webflow" definierten Actions "approveAction" sowie "rejectedAction" der Aufgabe ausgeführt (siehe Customizing XML-Datei uwl.webflow).

### 3 Customizing UWL; XML-Datei: uwl.webflow

In der Webflow-XML-Datei wird die Verarbeitung der einzelnen Workflow-Aufgaben gecustomized. Hier kann das Verhalten jeder Aufgabe individuell angepasst werden. Um Abwesenheitsanträge bzw. Zeitbuchungskorrekturanträge über die Massengenehmigung zu genehmigen bzw. abzulehnen, müssen die entsprechenden Aufgaben(Genehmigungsschritt im Workflow) den Funktionsbaustein ZHR\_LR\_SET\_STATUS aufrufen. Die folgende Definition muss für alle Aufgaben vorgenommen werden, die über die Massengenehmigung bearbeitet werden sollen:

```
[..]  
<ItemType name="uwl.task.webflow.TS90000006.SAP_ECC_HumanResources" connec-  
tor="WebFlowConnector" defaultView="DefaultView" defaultAction="launchWebDynPro" execu-  
tionMode="default">  
<ItemTypeCriteria systemId="SAP_ECC_HumanResources" externalType="TS90000006" connec-  
tor="WebFlowConnector"/>  
    <Actions>
```

```

<Action name="approveAction" groupAction="yes" handler="FunctionModuleActionHandler">
  <Properties>
    <Property name="FunctionModule" value="ZHR_LR_SET_STATUS"/>
    <Property name="IM_WORKFLOW_ID" value="{item.externalId}"/>
    <Property name="IM_NEW_STATUS" value="APPROVED"/>
  </Properties>
  <Descriptions default="">
    <ShortDescriptions>
      <Description Language="de" Description="Genehmigen"/>
    </ShortDescriptions>
    <LongDescriptions>
      <Description Language="de" Description="Genehmigen"/>
    </LongDescriptions>
  </Descriptions>
</Action>
<Action name="rejectedAction" groupAction="yes" handler="FunctionModuleActionHandler">
  <Properties>
    <Property name="FunctionModule" value="ZHR_LR_SET_STATUS"/>
    <Property name="IM_WORKFLOW_ID" value="{item.externalId}"/>
    <Property name="IM_NEW_STATUS" value="REJECTED"/>
  </Properties>
  <Descriptions default="">
    <ShortDescriptions>
      <Description Language="de" Description="Ablehnen"/>
    </ShortDescriptions>
    <LongDescriptions>
      <Description Language="de" Description="Ablehnen"/>
    </LongDescriptions>
  </Descriptions>
</Action>
</Actions>
</ItemType>
[..]

```

Werden die Buttons "Genehmigen" bzw. "Ablehnen" im Workitem betätigt, ruft die UWL den Funktionsbaustein ZHR\_LR\_SET\_STATUS auf, der den Abwesenheitsantrag bzw. Zeitbuchungskorrekturantrag in den entsprechenden Status versetzt.

Nachdem die XML-Dateien angepasst wurden, müssen diese unter Angabe einer hohen Priorität in das Portal hochgeladen werden. Anschließend muss über die Cache-Verwaltung (zu finden unter der Seite: Zentraler Arbeitsvorrat - Administration) der UWL-Cache gelöscht werden. Somit werden vom Portal die XML-Dateien neu gelesen und die UWL auf die neuen Customizing Einstellungen angepasst.

## 4 Funktionsweise des Funktionsbaustein ZHR\_LR\_SET\_STATUS

Der Funktionsbaustein wird – wie oben zu sehen ist – beim Ausführen über "Genehmigen" und "Ablehnen" eines Abwesenheitsantrags bzw. Zeitbuchungskorrekturantrags aufgerufen. Dieser setzt den Status eines Antrags auf den jeweiligen Status(genehmigt/abgelehnt). Die Statusänderung ist direkt im Workflow ersichtlich. Des Weiteren setzt der Funktionsbaustein auch den Bearbeiter bzw. Genehmiger des Antrags. Nachdem der Funktionsbaustein ausgeführt wurde, ist das Workitem in der UWL bearbeitet und der Workflow wird vorgesetzt.

```

FUNCTION ZHR_LR_SET_STATUS.
* "-----
* " * "Lokale Schnittstelle:
* "   IMPORTING
* "       VALUE(IM_WORKFLOW_ID) TYPE   SWW_WIID
* "       VALUE(IM_NEW_STATUS) TYPE   TIM_REQ_STATUS
* "-----

DATA: ls_obj2                TYPE SWR_OBJ_2,
      request_id            TYPE tim_req_id,
      event_approve        TYPE tim_req_xfer_event VALUE cl_pt_req_const=
>c_reqtrans_approve,
      event_reject         TYPE tim_req_xfer_event VALUE cl_pt_req_const=
>C_REQTRANS_REJECT,
      request              TYPE REF TO if_pt_req_request,
      v_new_status         TYPE tim_req_status,
      v_re_responsible     TYPE PTREQ_ACTOR_STRUC_FLAT,
      v_re_n_processor     TYPE PTREQ_ACTOR_STRUC_FLAT.

CALL FUNCTION 'SAP_WAPI_GET_OBJECTS'
  EXPORTING
    WORKITEM_ID            = im_workflow_id
*   LANGUAGE              = SY-LANGU
*   USER                  = SY-UNAME
  IMPORTING
*   LEADING_OBJECT        =
*   RETURN_CODE           =
    LEADING_OBJECT_2      = ls_obj2
*   TABLES
*   OBJECTS                =
*   MESSAGE_LINES         =
*   MESSAGE_STRUCT        =
*   OBJECTS_2             =
.

request_id = ls_obj2-INSTID.

CALL FUNCTION 'ENQUEUE_EPTREQ'
  EXPORTING
    mode_ptreq_header = 'E'
    request_id         = request_id

```



```
EXCEPTIONS
  foreign_lock      = 1
  system_failure   = 2
  OTHERS           = 3.

CALL METHOD ca_pt_req_header=>agent->get_request
EXPORTING
  im_request_id = request_id
IMPORTING
  ex_request    = request.

CALL METHOD REQUEST->GET_NEXT_PROCESSOR
RECEIVING
  RE_NEXT_PROC = v_re_n_processor.

IF IM_NEW_STATUS EQ 'APPROVED'.

  CALL METHOD request->initiate_state_transition
  EXPORTING
    im_transfer_event = event_approve
  IMPORTING
    ex_new_status     = v_new_status.

ELSEIF IM_NEW_STATUS EQ 'REJECTED'.

  CALL METHOD request->initiate_state_transition
  EXPORTING
    im_transfer_event = event_reject
  IMPORTING
    ex_new_status     = v_new_status.

ENDIF.

CALL METHOD REQUEST->SET_RESPONSIBLE
EXPORTING
  IM_ACTOR_TYPE = v_re_n_processor-ACTOR_TYPE
  IM_PLVAR      = v_re_n_processor-PLVAR
  IM_OTYPE     = v_re_n_processor-OTYPE
  IM_OBJID     = v_re_n_processor-OBJID.

CALL METHOD request->set_status
EXPORTING
  im_status = v_new_status.

CALL FUNCTION 'DEQUEUE_EPTREQ'
EXPORTING
  request_id = request_id.

COMMIT WORK AND WAIT.
```



ENDFUNCTION.