Cookbook SAP REA Workflow

Applies to:
SAP Recycling Administration (SAP REA) on EhP6 or higher

Summary
This document explains how to use SAP REA’s build in workflows to foster process integration and to ensure master data quality.

Author: Bernd Roedel
Company: SAP AG
Created on: 1 July 2012

Author Bio
Dr. Bernd Roedel joined SAP SI in 2000. Later he moved to SAP AG and became a Development Architect. His responsibilities include the technical governance of SAP Recycling Administration. He has also worked on the SAP Enterprise Portal and in Java and Objective C projects.
# Table of Contents

Preliminary Remark .................................................................................................................. 3
Changes ...................................................................................................................................... 3
Introduction to the REA master data model ........................................................................... 4
Workflows in REA ...................................................................................................................... 6
  Business Objects ..................................................................................................................... 6
  Customizing ............................................................................................................................. 6
  Templates ................................................................................................................................. 7
Integrating Material Master and REA Master Data ................................................................. 7
Keeping BOMs and REA Master Data in Sync ........................................................................ 9
Ensuring Data Consistency in REA .......................................................................................... 14
Appendix .................................................................................................................................. 14
Error! Bookmark not defined.
Related Content ...................................................................................................................... 17
Copyright ................................................................................................................................. 18
Preliminary Remark

The following document is a cookbook for using workflows in REA to ensure data integrity and tight integration between REA and ERP processes (Figure 1). It begins with a short introduction to “Workflows in REA”. The document neither explains REA customizing/configuration nor a basic workflow setup. In addition to the master data REA uses terms like recycling partner, price list, license fee or splitting that are all part of a contract between the REA user and a compliance scheme. Those contract management tasks are not explained in this document. This document addresses user familiar with workflows and their configuration. Whereas workflows serve as a asynchronous process integrator, the direct integration between material master in REA master data is described in [1].

After reading this document you should be able to understand how to use workflows in REA.

![Figure 1: Process Integration](image)

Changes

This is the initial version.
Introduction to the REA master data model

The REA master data consists of two main objects: REA article and REA (packaging) component.

A REA (packaging) component is based on REA internal fractions. Figure 2 depicts the master data model schematically. The colors in Figure 2 will be used consistently throughout this document.

**Figure 2: REA Master data scheme**

REA article and REA components refer to the material number in the ERP Material master (MARA-MATNR). Hence REA article cannot exist without a corresponding MM entry. REA packaging components may exist without a material master entry, if configured appropriately. In addition it also is possible that a REA article and a REA component refer to the identical MM entry (Figure 3).

**Figure 3: Relationship to ERP Material master**

REA internal fractions are maintained in the REA customizing and assigned to one or many recycling partner fractions (Figure 4). It is also possible to assign an internal fraction to a recycling partner fraction without specifying a partner fraction. In this case there is no settlement with this particular recycling partner for that internal fraction. This mechanism is called cancellation of partner fraction requirement. This fraction assignment is out of scope for this document.
An internal fraction in REA is a packaging material that can be assigned 1 to n times to a REA packaging with a defined weight / weight unit.

A REA component consists of one or many internal fractions and that can be settled with one or many recycling partners. Both assignments are time dependent, so that several packaging versions with a non-overlapping timeframes can be created. A REA component can be of type **consumed packaging**, which can be directly identified in material movements by the declaration system, or of type **sales packaging**, which can be assigned to one or many REA articles as a component.

A REA article represents finished product that must be reported to a recycling partner due to the legal obligations of the REA user. In addition to the material number, a REA article is identified by the key values company code, country and sales unit. REA components and recycling partners are assigned to REA articles in a time dependent manner. A REA article is directly identified in billing documents and/or material movements by the declaration system.

Figure 5 depicts the REA master data structuring. The REA article and the REA components are the two main components that form the REA master data. In addition to the keys and relationships explained so far, REA articles and REA components can hold various attribute values that are generally dependent on the assigned recycling partner.

**Figure 4: Assignment internal fraction to recycling partner fraction by Customizing**

**Figure 5: REA Master data**

REA components are maintained by the transactions J7L5/J7L6/J7L7. The tab **fraction** is used to maintain the internal fraction assignment. The tab **partner** is used to maintain the recycling partner assignment. A REA component can only be settled with partners that are assigned to the packaging in a particular timeframe.
REA articles are maintained by transactions J7L1/J7L2/J7L3. The tab **packaging** is used to maintain the REA component assignment. The tab partner is used to maintain the recycling partner assignment. A REA article can only be settled with partners that are assigned to the article in a particular timeframe. In case there are multiple recycling partners assigned in the same timeframe, **license fee splitting** rules enable the user to split the license fee of the packaging between the assigned recycling partners dependent on the business process.

REA articles can alternatively be maintained as **reference articles** by transaction J7L0. Reference articles are settled exactly as their referenced article. Reference articles are not discussed in detail in this document.

Figure 6 summarizes the REA master data maintenance process.

![Figure 6: Process view](image)

### Workflows in REA

#### Business Objects

The master data entities described in section above are represented by the business objects listed in Table 1. These business objects are the basis for workflows encompassing REA processes.

<table>
<thead>
<tr>
<th>Technical name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/J7L/BUS01</td>
<td>REA article</td>
</tr>
<tr>
<td>/J7L/BUS02</td>
<td>REA packaging</td>
</tr>
<tr>
<td>/J7L/BUS03</td>
<td>REA reference article</td>
</tr>
</tbody>
</table>

**Table 1: REA Business Objects in Transaction SWO1**

The workflow templates in section “Integrating Material Master and REA Master Data”, “Keeping BOMs and REA Master Data in Sync” and “Ensuring Data Consistency in REA” use these business objects when accessing REA master data.

#### Customizing

Triggering of REA workflow events must be explicitly activated in REA customizing → general control → define general control. If the appropriate checkbox is not set, no REA transaction will trigger the corresponding event.
If the checkbox is set, you may be prompted to explicitly trigger a workflow event on selection screens during mass maintenance.

Templates

REA is delivered with 5 templates covering three main processes in ERP, such as Integrating Material Master and REA Master Data, Keeping BOMs and REA Master Data in Sync and Ensuring Data Consistency in REA. None of these templates supports an agent determination. The examples showing the activation of the workflows in section “Integrating Material Master and REA Master Data”, “Keeping BOMs and REA Master Data in Sync” and “Ensuring Data Consistency in REA” define the workflow as general task, so that the user triggering the event receives the workitem in his Workplace inbox. The next three chapters will cover each supported process and the underlying workflow templates in depth.

Integrating Material Master and REA Master Data

One task of REA users is to ensure that REA master data is created in case new packaging or new articles are created in the ERP material master. Each of both task are supported by a separate workflow template that is triggered in the moment a material is created in ERP material master.

Figure 7 displays the workflow /J7L/WS_3 that links the material creation process to the article creation process in REA. Prerequisite for starting the workflow is the linkages of the triggering event of the BUS BUS1001006 to the REA workflow (event linkage). Proceed as follow:

a) Go to ABAP workbench (TA SE80) and open package /J7L/REA_WORKFLOW_TEMPLATES
b) Open workflow standard task 76200008
c) Go to menu additional data -- agent assignment -- maintain and set the attributes of the entry to general task (save your settings in a transport request). Generate the changes.
d) Open workflow template 76200007
e) On tab triggering events activate the trigger for BO BUS1001006 (save your settings in a transport request)
f) Go to transaction SWETYPV (Type Linkage) and select the entry with BOR BUS1001006 event VIEWCREATED and receiver type WS76200007. In the detail screen enter /J7L/WF_CHECK_ARTICLE as check function module.
g) The workflow is not executed if the data filter for material type REA article is active and does not include the material type of the material record created in MM. The same applies to the data filter for sales area.

Figure 8 displays the workflow /J7L/WS_4 that links the material creation process to the packaging creation process in REA. Prerequisite for starting the workflow is the linkages of the triggering event of the BUS BUS1001006 to the REA workflow (event linkage). Proceed as follow:
a) Go to ABAP workbench (TA SE80) and open package /J7L/REA_WORKFLOW_TEMPLATES
b) Open workflow standard task 76200010
c) Go to menu additional data -> agent assignment --> maintain and set the attributes of the entry to general task (save your settings in a transport request). Generate the changes.
d) Open workflow template 76200008
e) On tab triggering events activate the trigger for BO BUS1001006 (save your settings in a transport request)
f) Go to transaction SWETYPV (type linkage) and select the entry with BOR BUS1001006 event VIEWCREATED and receiver type WS76200008. In the detail screen enter /J7L/WF_CHECK_PACKAGING as check function module.

The workflow is not executed if the data filter for material type REA packaging is active and does not include the material type of the material record created in MM. The same applies to the data filter for sales area.

![Diagram of workflow](image)

Figure 8: Workflow /J7L/WS_4: Create REA Packaging Triggered by Material Record Creation

If you like to test both workflows just create a material record with the material type / sales area included in the REA data filters and navigate to your workflow inbox within the SAP business workplace (Figure 9). When opening the workitem, you may directly navigate to the article/packaging creation transaction in REA.

![Table of workitems](image)

Figure 9: Workitem in Workflow inbox. Material Number of Article is depicted (134)
When returning to your Workflow inbox, the workitem has to be set to completed explicitly (see Figure 10).

**Figure 10: Setting Completion Status Explicitly**

### 134: Create Article

<table>
<thead>
<tr>
<th>Description</th>
<th>Objects and attachments</th>
</tr>
</thead>
<tbody>
<tr>
<td>No description available</td>
<td>- <strong>Standard material:</strong> 00000000000000134</td>
</tr>
</tbody>
</table>

**Keeping BOMs and REA Master Data in Sync**

Bills of material (BOM) can be leveraged to create REA Master data if they contain packaging information. In this case REA is customized to display all packaging related components of a BOM on a separate tab in REA article maintenance transaction. The BOM can be displayed filtered according to the data filter defined in REA customizing. Each of its components can be selected for assignment to the REA article. Thus the BOM serves as a kind of temporary template from which the REA user copies the relevant data to REA. There is no permanent link between the BOM and The REA article. The challenge for all REA users is to keep this template in sync with the REA article and to ensure that BOM changes are reflected by corresponding REA article changes if they affect the packaging composition. Figure 11 displays the process steps.
A BOM change is executed in transaction CS02

While saving a BADI Method (BOM Update) is executed and checks if the BOM change affects any REA master data

A workflow event IsInvolved of the affected REA articles is triggered

A background task tries to adjust BOM an REA article components automatically

If automatic adjustment fails, user interaction is requested with the option to repeat automatic adjustment or to initiate manual adjustment

---

**Figure 11: Process During BOM Adjustment**

By virtue of a missing business object event to link to (contrary to section “Integrating Material Master and REA Master Data”) the BOM adjustment must be triggered by a BAdI implementation executed, when a BOM is saved (Figure 12). The BAdI implementation also serves as a checkpoint, which evaluates the nature of the BOM changes with respect to its REA relevance. In case the workflow is started the first step is to try an automatic adjustment. If this fails, the automatic adjustment attempt might be repeated (if the article was locked against changes during execution) or the adjustment might be executed manually by the user. In order activate the workflow proceed as follows:

a) Go to ABAP workbench (TA SE80) and open package /J7L/REA_WORKFLOW_TEMPLATES
b) Open workflow standard task 76200013
c) Go to menu additional data --> agent assignment --> maintain and set the attributes of the entry to general task (save your settings in a transport request). Generate the changes.
d) Open workflow template 76200009
e) On tab triggering events activate the trigger for BO /J7L/BUS01 (save your settings in a transport request)
f) Go to menu additional data --> agent assignment --> maintain and set the attributes of the entry to general task (save your settings in a transport request). Generate the changes.

In order to create the BAdI implementation proceed as follows:

a) Enter transaction SE19 and create an implementation of the classical BADI BOM_UPDATE. Of the interface IF_EX_BOM_UPDATE implement the method CHANGE_AT_SAVE

The following code section contains an example implementation, which execute the workflow in case a BOM is directly linked to a REA article or contains components, which are REA packaging. In both cases workflows with all involved REA articles are created.
method IF_EX_BOM_UPDATE-CHANGE_AT_SAVE.
data: trigger_event type boolean,
exists type c,
tabname type tabname,
MATRAM_TAB type BAPIMATRAM_TAB,
wa_matram like line of matram_tab,
article_list type J_7LM03_TAB,
wa_article like line of article_list.
data: object_type TYPE swr_struct-object_typ,
object_key TYPE swr_struct-object_key,
event TYPE swr_struct-event,
input_container type standard table of SWR_CONT,
input_line type SWR_CONT.
field-symbols: <wa_mast> like line of DELTA_MASTB,
<wa_stpo> like line of DELTA_STPOB.

trigger_event = abap_false.
read table DELTA_MASTB assigning <wa_mast> index 1. "check BOM header for article"
if sy-subrc = 0.
  CALL FUNCTION 'J_7L_ARTICLE_EXISTS'
    EXPORTING
      I_MATNR = <wa_mast>-matnr
      * I_BUKRS =
      * I_LAND1 =
      * I_VRKME =
      * I_VARIANTE =
    IMPORTING
      E_FLG_ARTICLE_EXITS = exists
      E_TABNAME = tabname.
    if not exists is initial and tabname = 'J_7LM01'.
      trigger_event = abap_true.
      move <wa_mast>-matnr to wa_matram-matnr_low.
    endif.
  CALL FUNCTION 'J_7L_PACK_EXISTS'
    EXPORTING
      I_matnr = <wa_mast>-matnr
    IMPORTING
      E_FLG_PACK_EXITS = exists.
    if not exists is initial.
      trigger_event = abap_true.
    endif.
  endif.
** If header is not interesting -> loop at items
if trigger_event = abap_false.
  loop at DELTA_STPOB assigning <wa_stpo>.
    CALL FUNCTION 'J_7L_PACK_EXISTS'
      EXPORTING
        I_matnr = <wa_stpo>-IDNRK
      IMPORTING
        E_FLG_PACK_EXITS = exists.
      if not exists is initial.
        trigger_event = abap_true.
        exit.
      endif.
  endloop.
* trigger event?
  
  if trigger_event = abap_true.
    wa_matram-sign = 'I'.
    wa_matram-option = 'EQ'.
    append wa_matram to matram_tab.
  
  CALL FUNCTION 'J_7L_GET_ARTICLE_LIST'
    EXPORTING
      * I_MAXROWS = 0
      I_MATRAM_TAB = matram_tab
    * I_BUKRERASO_TAB =
    * I_LAND1RASO_TAB =
    * I_VRKMERASO_TAB =
    * I_VARIANTERASO_TAB =
    * I_IGNORE_DELETED = 'X'
  
  IMPORTING
    * RETURN =
    O_ARTICLE_LIST = article_list.
  
  loop at article_list into wa_article.
    object_key = /j7l/rea_bo=>get_rea_article_object_key(
      in_matnr = wa_article-matnr
      in_bukrs = wa_article-bukrs
      in_land1 = wa_article-land1
      in_vrkme = wa_article-vrkme
      in_variante = wa_article-variante ).
    
    object_type = /j7l/rea_bo=>rea_article.
    refresh input_container.
    clear input_line.
    input_line-ELEMENT = 'DATE'.
    input_line-VALUE = I_DATUV.
    append input_line to input_container.
  
  CALL FUNCTION 'SAP_WAPI_CREATE_EVENT'
    EXPORTING
      object_type = object_type
      object_key = object_key
      event = 'IsInvolved'
      commit_work = space
    * EVENT_LANGUAGE = SY-LANGU
    * LANGUAGE = SY-LANGU
    * USER = SY-UNAME
    * IFS_XML_CONTAINER = IFS_XML_CONTAINER
    * IMPORTING
      return_code = return_code
      EVENT_ID = EVENT_ID
    
    TABLES
      INPUT_CONTAINER = INPUT_CONTAINER.
    
    MESSAGE_LINES = MESSAGE_LINES
    MESSAGE_STRUCT = MESSAGE_STRUCT
  
endloop.

endif.

endmethod.
Figure 12: Workflow /J7L/WS_5: The Workflow is Started in this Example by a BAdI Implementation

In case an automatic adjustment is possible, no workitem will appear in the workflow inbox. In case the automatic adjustment fails, a workitem will appear as in Figure 13.

Figure 13: Workitem if BOM Adjustment Fails

The user has got three options, when clicking on the workitem:

1. Retry (in case the adjustment failed if the article was locked in a different LUW)
2. Exit workflow and enter manual article maintenance (TA J7L2)
3. Keep workitem for later processing

Figure 14: Options of a Failed BOM Adjustment
The following use cases will lead to a failed adjustment:

1. Component is added to BOM, which fulfills criteria as REA packaging, but has not been created as REA packaging yet
2. Article is locked in a different LUW against changes
3. Saving of article fails due to inconsistent data
4. 

The following use case will lead to a successful adjustment:

1. Existing component’s quantity is changed
2. Component is deleted from BOM
3. Component already created as REA packaging is added to the BOM

Ensuring Data Consistency in REA

REA validates its own master data when a legal relevant process is initiated. The process can be completed only when the REA master data is in a consistent state. Therefore it is desirable for the REA user to ensure data consistency prior to process start. There are two main tools that support the REA user in achieving data consistency. One tool is the collective consistency check; the other is the declaration log evaluation. Both tools can be used to feed a workflow for all inconsistent master data.

Figure 15 displays the workflow that supports the user to identify inconsistent article. The trigger is set by REA collective consistency check (event is inconsistent), which initiates one dialog step (change REA article). This workflow helps to start and track the correction of inconsistent REA articles. The triggering program’s selection screen can be used to define the scope of the trigger events.

In order to activate the workflow proceed as follows:

- Go to ABAP workbench (TA SE80) and open package /J7L/REA_WORKFLOW_TEMPLATES
- Open workflow template 76200005
- On tab triggering events activate the trigger for BO /J7L/BUS01 (Save your settings in a transport request)
- Go to menu additional data --> agent assignment --> maintain and set the attributes of the entry to general task (save your settings in a transport request). Generate the changes.

One option to trigger the workflow /J7L/WS_1 is transaction J7LUTILITY24. Figure 16 shows selection criteria to identify inconsistent articles in company code 0001 assigned to the recycling partner ISD in timeframe 01.01.2001 until 31.12.9999 and trigger the isInconsistent event.
Cookbook SAP REA Workflow

**Figure 16: J7LUTILITY24 Selection Screen to Trigger /J7L/WS_1**

**Figure 17: Workflow inbox when editing REA Articles (Material Number, Company Code and Country are Displayed in Workitem Title)**

Figure 18 displays the workflow that supports the user to find articles that have not been created in REA yet, but that show up as missing in the declaration's log file. The workflow is started by the REA log file analyser, which submits a container to the workflow containing the keys of the REA article to be created via dialog transaction REA create article. This workflow helps to identify articles that have been not been created in REA yet by analyzing the log file of a REA declaration. The creation of the missing article can be tracked by the workflow. The triggering program's selection screen can be used to define the scope of the trigger events.

**Figure 18: Workflow /J7L/WS_2: Creating Articles Logged as Missing**

In order to activate the workflow proceed as follows:

a) Go to ABAP workbench (TA SE80) and open package /J7L/REA_WORKFLOW_TEMPLATES
b) Open workflow template 76200006
c) Go to menu additional data --> agent assignment --> maintain and set the attributes of the entry to general task (save your settings in a transport request). Generate the changes.
One option to trigger the workflow /J7L/WS_2 is transaction J7LUTILITY26. Figure 19 shows selection criteria to identify missing articles in declaration 5000000000. It is important to select execute workflow and subsequently enter a sales unit when creating (PC) and the task 76200006.

![Analysis of Declaration Log File](image)

**Figure 19: Finding Missing Articles by Log Analysis of J7LUTILITY24**

Choosing execute workflow will cause J7LUTILITY26 to call the workflow task with the keys extracted out of the declaration log. Often the sales unit cannot be determined due to referencing. Therefore a default sales unit has to be set (sales unit when creating).

With different settings J7LUTILITY24 can be used to trigger /J7L/WS_1, too.
Related Content

[1] Cookbook REA MM Integration for Industries
[2] Cookbook REA MM Integration for Retail