

Step by Step Approach for End Routine with Look Up



Applies to:

SAP BW 3.x & SAP BI Net Weaver 2004s. For more information, visit the [EDW homepage](#).

Summary

This document will give the reader detailed information with respect to implementation of an end routine for a business scenario where Header to Item look up is required.

Author: Vikram Srivastava

Company: Infosys Technologies Limited

Created on: 22 August 2010

Author Bio



Vikram Srivastava is working as Technology Analyst with Infosys Technologies Limited. He has got rich experience on various BW Implementation/Support Projects in both SAP BW 3.5 and SAP BW 7.0.

Table of Contents

Business Scenario	3
Data Flow	3
Understanding the Scenario with Example	4
Data in Header DSO:	4
Data in Item DSO:	4
Output After Merging:	4
Step by Step Approach	5
Step 1:	5
Step 2:	5
Step 3:	5
Step 4:	6
Sample end Routine	6
Step 5:	10
Step 6:	10
Related Content	11
Disclaimer and Liability Notice	12

Business Scenario

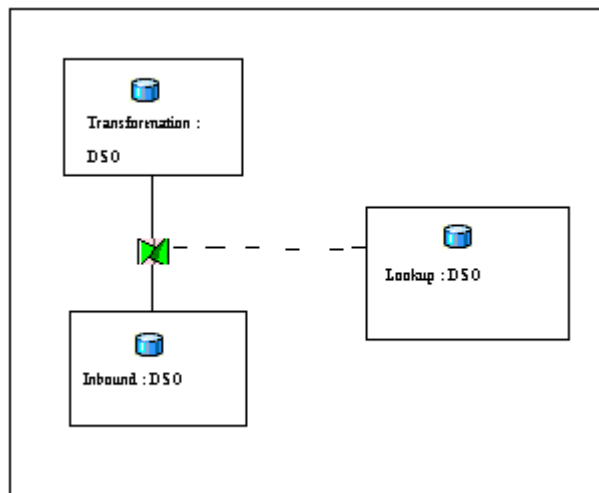
This article explains how to write an end routine for a business scenario where we want to merge the data or item and header by doing a lookup from header to item DSO.

This is a common business scenario especially in the occasions where we want to show the article details along with the header information.

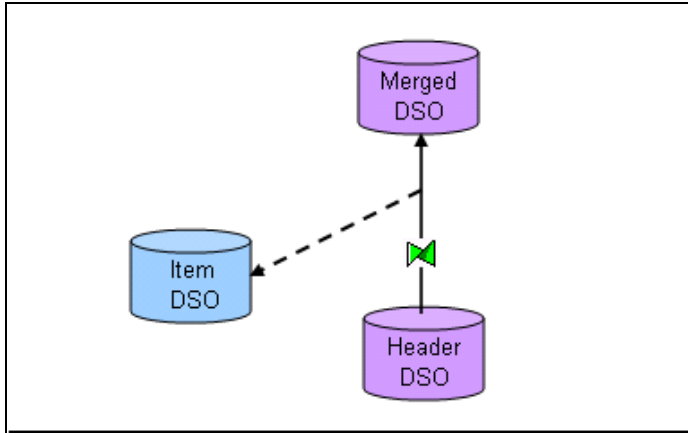
One such example could be merging Sales Header and Item Data in one DSO.

Data Flow

Below shown the typical data flow which will be seen when we try to achieve the business scenario explained above.



Understanding the Scenario with Example



Data in Header DSO:

Order Number	Doc. type	Request.dlv.dt
S1	A	01.01.2010
S2	B	01.05.2010

Data in Item DSO:

Order Number	Item Number	Article	Item type	Order quantity
S1	10	A1	X	10
S1	20	A2	Y	20
S1	30	A5	X	15
S2	10	A3	Y	6
S2	20	A1	Y	15

Output After Merging:

Order Number	Item Number	Doc. type	Request.dlv.dt	Article	Item type	Order quantity
S1	10	A	01.01.2010	A1	X	10
S1	20	A	01.01.2010	A2	Y	20
S1	30	A	01.01.2010	A5	X	15
S2	10	B	01.05.2010	A3	Y	6
S2	20	B	01.05.2010	A1	Y	15

Here as we can see we are trying to show Document Type and Request Delivery Date from the header table along with the item information. In this scenario business wants the delta to be based on header instead of item.

Note: While loading the data from Header DSO to the merged DSO the total number of records increases.

Let's assume the name of the DSO's to be ZHEADER, ZITEM and ZMERGE.

Step by Step Approach

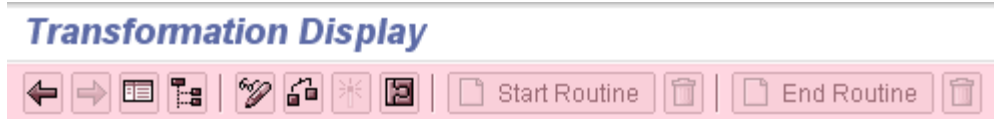
Step 1:

Go to TCode RSA1



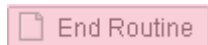
Step 2:

Double Click on the Transformation where you want to write the routine. You get the below screen on the top.



Step 3:

Click on Create End Routine



Below screen appears:



Step 4:

Write the end routine as shown in the below sample after the section:

```

*$$$ begin of routine - insert your code only below this line      ***
... "insert your code here
*-- fill table "MONITOR" with values of structure "MONITOR_REC"
*- to make monitor entries

```

Sample end Routine

Below is the code which can be used when trying to implement such scenario.

```

PROGRAM trans_routine.

*-----*
*   CLASS routine DEFINITION
*-----*
*
*-----*
CLASS lcl_transform DEFINITION.
  PUBLIC SECTION.

* Attributs
  DATA:
    p_check_master_data_exist
      TYPE RSODSOCHECKONLY READ-ONLY,

```

This is the declaration done by the SAP while defining the class

```

*- Instance for getting request runtime attributes;
* Available information: Refer to methods of
* interface 'if_rsbk_request_admintab_view'
p_r_request
  TYPE REF TO if_rsbk_request_admintab_view READ-ONLY.

```

PRIVATE SECTION.

```
TYPE-POOLS: rsd, rstr.
```

```

* Rule specific types
TYPES:
BEGIN OF _ty_s_TG_1,
* Field: RECORD.
  RECORD      TYPE RSARECORD,
* InfoObject: ZORDNUM Order Number.
  /BIC/ZORDUM  TYPE /BIC/OIZORDNUM,
* InfoObject: ZDOCTYP Document Type.
  /BIC/ZDOCTYP TYPE /BIC/OIDOCTYP,
* InfoObject: ZDELDATE Request Delivery Date.
  /BIC/ZDELDATE TYPE /BIC/OIZDELDATE,
* InfoObject: 0RECORDMODE BW Delta Process: Update Mode.
  RECORDMODE  TYPE RODMUPDMOD,
END OF _ty_s_TG_1.
TYPES:
_ty_t_TG_1    TYPE STANDARD TABLE OF _ty_s_TG_1
              WITH NON-UNIQUE DEFAULT KEY.

```

This is the declaration done by the SAP while defining the source structure (Header)

```

*$$$ begin of global - insert your declaration only below this line *-
... "insert your code here

```

```

*$$$ end of global - insert your declaration only before this line *-

```

METHODS

```

new_record__end_routine
IMPORTING
  source_segid      type rstran_segid
  source_record     type sytabix
EXPORTING
  record_new        type sytabix.

```

METHODS

```

end_routine
IMPORTING
  request           type rsrequest
  datapackid       type rsdatapid
EXPORTING
  monitor          type rstr_ty_t_monitors
CHANGING
  RESULT_PACKAGE   type _ty_t_TG_1
RAISING
  cx_rsrount_abort.

```

METHODS

```

inverse_end_routine
IMPORTING
  i_th_fields_outbound TYPE rstran_t_field_inv
  i_r_selset_outbound  TYPE REF TO cl_rsmnds_set
  i_is_main_selection  TYPE rs_bool
  i_r_selset_outbound_complete TYPE REF TO cl_rsmnds_set
  i_r_universe_inbound TYPE REF TO cl_rsmnds_universe
CHANGING
  c_th_fields_inbound TYPE rstran_t_field_inv
  c_r_selset_inbound  TYPE REF TO cl_rsmnds_set
  c_exact             TYPE rs_bool.
ENDCLASS.          "routine DEFINITION

```

*\$\$ begin of 2nd part global - insert your code only below this line *
 ... "insert your code here"

```
TABLES: /BIC/AZITEM00.
DATA: ITAB2 TYPE STANDARD TABLE OF /BIC/AZITEM00.
      WITH HEADER LINE.
DATA: WA LIKE LINE OF ITAB2.
```

We declare some internal tables and work area of type of Item table

*\$\$ end of 2nd part global - insert your code only before this line *

```
*-----*
*   CLASS routine IMPLEMENTATION
*-----*
*
*-----*
CLASS lcl_transform IMPLEMENTATION.

*-----*
*   Method end_routine
*-----*
*   Calculation of result package via end routine
*   Note: Update of target fields depends on rule assignment in
*   transformation editor. Only fields that have a rule assigned,
*   are updated to the data target.
*-----*
*   <-> result package
*-----*
METHOD end_routine.
*=== Segments ===

FIELD-SYMBOLS:
  <RESULT_FIELDS> TYPE _ty_s_TG_1.

DATA:
  MONITOR_REC TYPE rstmonitor.

*$$ begin of routine - insert your code only below this line   *-*
... "insert your code here"
*-- fill table "MONITOR" with values of structure "MONITOR_REC"
*- to make monitor entries
```

```
DATA: ITAB_REFRESH TYPE STANDARD TABLE OF _TY_S_TG_1.
DATA: ITAB_TARGET TYPE STANDARD TABLE OF _TY_S_TG_1.
DATA: WA_TARGET LIKE LINE OF ITAB_TARGET.
DATA : ICOUNT TYPE RSARECORD.
```

We declare some local variables


```

*****
* Start of Routine
*****

* Ensuring the source package is not empty

IF RESULT_PACKAGE IS NOT INITIAL.
  ICOUNT = 1.
  REFRESH ITAB2.

*Selecting records from Item DSO for which
* Document no matches with the same of source package

  SELECT * FROM /BIC/AZITEM00 INTO TABLE ITAB2
  FOR ALL ENTRIES IN RESULT_PACKAGE WHERE
  /BIC/ZORDNUM = RESULT_PACKAGE-/BIC/ZORDNUM .

* Sorting for optimizing readaccess
  SORT ITAB2 BY /BIC/ZORDNUM.

* Processing

  Refresh itab_target.

  SORT RESULT_PACKAGE BY /BIC/ZORDNUM /BIC/ZITEMNUM.

  LOOP AT RESULT_PACKAGE ASSIGNING <RESULT_FIELDS>.

  LOOP AT ITAB2 INTO WA
  WHERE /BIC/ZORDNUM = <RESULT_FIELDS>-/BIC/ZORDNUM.

  <RESULT_FIELDS>-/BIC/ZITEMNUM = WA-/BIC/ZITEMNUM.
  <RESULT_FIELDS>-MATERIAL = WA-MATERIAL.
  <RESULT_FIELDS>-/BIC/ZITMTYP = WA-/BIC/ZITMTYP.
  <RESULT_FIELDS>-QUANT_B = WA-QUANT_B.
  <RESULT_FIELDS>-BASE_UOM = WA-BASE_UOM.
  <RESULT_FIELDS>-RECORD = ICOUNT.

  ICOUNT = ICOUNT + 1.

  APPEND <RESULT_FIELDS> TO ITAB_TARGET.

  ENDLOOP.

  ENDLOOP.

  SORT ITAB_TARGET BY /BIC/ZINHUNUM /BIC/ZHUIITEM.

  REFRESH RESULT_PACKAGE.

  RESULT_PACKAGE[] = ITAB_TARGET[].

  ENDIF.

... "to cancel the update process
* raise exception type CX_RSROUT_ABORT.

*$$$ end of routine - insert your code only before this line    *-*
  ENDMETHOD.           "end_routine

```

This is the sample end routine. Here as we can see we are incrementing the record number. This is very crucial as if we do not do this we will face problems in activating the data from new table to active table as the record number had been the same.

```

*-----*
*   Method inverse_end_routine
*-----*
*
*   This subroutine needs to be implemented only for direct access
*   (for better performance) and for the Report/Report Interface
*   (drill through).
*   The inverse routine should transform a projection and
*   a selection for the target to a projection and a selection
*   for the source, respectively.
*   If the implementation remains empty all fields are filled and
*   all values are selected.
*-----*
*
*-----*
METHOD inverse_end_routine.

*$$$ begin of inverse routine - insert your code only below this line*-*
... "insert your code here
*$$$ end of inverse routine - insert your code only before this line *-*

ENDMETHOD.           "inverse_end_routine

METHOD new_record__end_routine.

***** IMPLEMENTATION is only visible in generated program *****

ENDMETHOD.
ENDCLASS.           "routine IMPLEMENTATION

```

Step 5:

Save and Activate.

Step 6:

Load the data, to get the desired output.

Related Content

<http://wiki.sdn.sap.com/wiki/display/BI/Business+Intelligence+Home>

http://help.sap.com/saphelp_nw04/helpdata/en/44/b4a0137acc11d1899e0000e829fbbd/content.htm

<http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/e73bfc19-0e01-0010-23bc-ef0ad53f2fab>

For more information, visit the [EDW homepage](#)

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.