**Item 1 – Rotina no infobjeto da transformation**

PROGRAM trans\_routine.  
  
  
*\*---------------------------------------------------------------------\**  
*\*       CLASS routine DEFINITION*  
*\*---------------------------------------------------------------------\**  
*\**  
*\*---------------------------------------------------------------------\**  
CLASS lcl\_transform DEFINITION.  
  PUBLIC SECTION.  
  
*\*  Attributs*  
    DATA:  
      p\_check\_master\_data\_exist  
            TYPE RSODSOCHECKONLY READ-ONLY,  
*\*-    Instance for getting request runtime attributs;*  
*\*     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\_SC\_1,  
*\*      Field: WERKS Centro.*  
        WERKS           TYPE C LENGTH 4,  
*\*      Field: PLNBEZ Material.*  
        PLNBEZ           TYPE C LENGTH 18,  
*\*      Field: FEVOR Resp.ctrl.prod..*  
        FEVOR           TYPE C LENGTH 3,  
*\*      Field: SAKTO Conta do Razão.*  
        SAKTO           TYPE C LENGTH 10,  
*\*      Field: RECORD.*  
        RECORD           TYPE RSARECORD,  
      END   OF \_ty\_s\_SC\_1.  
    TYPES:  
      BEGIN OF \_ty\_s\_TG\_1,  
*\*      InfoObject: 0COSTCENTER Centro de custo.*  
        COSTCENTER           TYPE /BI0/OICOSTCENTER,  
      END   OF \_ty\_s\_TG\_1.  
  
*\*$\*$ 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  
      compute\_0COSTCENTER  
        IMPORTING  
          request                  type rsrequest  
          datapackid               type rsdatapid  
          SOURCE\_FIELDS              type \_ty\_s\_SC\_1  
          segid                    type RSBK\_SEGID  
        EXPORTING  
          RESULT                   type \_ty\_s\_TG\_1-COSTCENTER  
          monitor                  type rstr\_ty\_t\_monitor  
        RAISING  
          cx\_rsrout\_abort  
          cx\_rsrout\_skip\_record  
          cx\_rsrout\_skip\_val  
          cx\_rsbk\_errorcount.  
    METHODS  
      invert\_0COSTCENTER  
        IMPORTING  
          i\_th\_fields\_outbound         TYPE rstran\_t\_field\_inv  
          i\_r\_selset\_outbound          TYPE REF TO cl\_rsmds\_set  
          i\_is\_main\_selection          TYPE rs\_bool  
          i\_r\_selset\_outbound\_complete TYPE REF TO cl\_rsmds\_set  
          i\_r\_universe\_inbound         TYPE REF TO cl\_rsmds\_universe  
        CHANGING  
          c\_th\_fields\_inbound          TYPE rstran\_t\_field\_inv  
          c\_r\_selset\_inbound           TYPE REF TO cl\_rsmds\_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*  
  
*\*   Busca a tabela do infobjeto Taxa Real Dolar*  
    tables: /BIC/MTXDOL.  
  
    data: v\_taxa  type /BIC/MTXDOL-/BIC/TXVAL.  
*\*   data: v\_mes   type /BIC/MTXDOL-/BIC/TXDOL.*  
  
*\*$\*$ end of 2nd part global - insert your code only before this line   \**  
  
*\*---------------------------------------------------------------------\**  
*\*       CLASS routine IMPLEMENTATION*  
*\*---------------------------------------------------------------------\**  
*\**  
*\*---------------------------------------------------------------------\**  
CLASS lcl\_transform IMPLEMENTATION.  
  
*\*----------------------------------------------------------------------\**  
*\*       Method compute\_0COSTCENTER*  
*\*----------------------------------------------------------------------\**  
*\*       This subroutine allows the mapping from source to target fields*  
*\*       of a transformation rule using ABAP for application specific*  
*\*       coding.*  
*\*----------------------------------------------------------------------\**  
*\*       Customer comment:*  
*\*----------------------------------------------------------------------\**  
  METHOD compute\_0COSTCENTER.  
  
*\*   IMPORTING*  
*\*     request     type rsrequest*  
*\*     datapackid  type rsdatapid*  
*\*     SOURCE\_FIELDS-WERKS TYPE C LENGTH 000004*  
*\*     SOURCE\_FIELDS-PLNBEZ TYPE C LENGTH 000018*  
*\*     SOURCE\_FIELDS-FEVOR TYPE C LENGTH 000003*  
*\*     SOURCE\_FIELDS-SAKTO TYPE C LENGTH 000010*  
*\*    EXPORTING*  
*\*      RESULT type \_ty\_s\_TG\_1-COSTCENTER*  
  
    DATA:  
      MONITOR\_REC    TYPE rsmonitor.  
  
*\*$\*$ 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*  
... *"to cancel the update process*  
*\*    raise exception type CX\_RSROUT\_ABORT.*  
... *"to skip a record"*  
*\*    raise exception type CX\_RSROUT\_SKIP\_RECORD.*  
... *"to clear target fields*  
*\*    raise exception type CX\_RSROUT\_SKIP\_VAL.*  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002001' AND SOURCE\_FIELDS-WERKS =  
    'LVAQ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = '3505413'.  
        WHEN 'AUT'.  
          RESULT = '3505413'.  
        WHEN 'CP'.  
          RESULT = '3505411'.  
        WHEN 'LAB'.  
          RESULT = '3505411'.  
        WHEN ' '.  
          RESULT = '3505413'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002001' AND SOURCE\_FIELDS-WERKS =  
    'BISJ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = '3505451'.  
        WHEN 'AUT'.  
          RESULT = '3505451'.  
        WHEN 'CP'.  
          RESULT = '3505451'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002001' AND SOURCE\_FIELDS-WERKS =  
    'BIBG'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = '3505471'.  
        WHEN 'AUT'.  
          RESULT = '3505471'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002001' AND SOURCE\_FIELDS-WERKS =  
    'PTX'.  
      RESULT = '9600009'.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002001' AND SOURCE\_FIELDS-WERKS =  
    'AAAQ'  
    .  
      CASE SOURCE\_FIELDS-PLNBEZ.  
        WHEN '2000103'.  
          RESULT = '3505234'.  
        WHEN '2000137'.  
          RESULT = '3505234'.  
        WHEN '2000104'.  
          RESULT = '3505232'.  
        WHEN '2000187'.  
          RESULT = '3505212'.  
        WHEN '2000135'.  
          RESULT = '3505222'.  
        WHEN '2000188'.  
          RESULT = '3505214'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002001' AND SOURCE\_FIELDS-WERKS =  
    'AASJ'  
    .  
      CASE SOURCE\_FIELDS-PLNBEZ.  
        WHEN '2000103'.  
          RESULT = '1105234'.  
        WHEN '2000137'.  
          RESULT = '1105234'.  
        WHEN '2000104'.  
          RESULT = '1105232'.  
        WHEN '2000187'.  
          RESULT = '1105212'.  
        WHEN '2000135'.  
          RESULT = '1105222'.  
        WHEN '2000188'.  
          RESULT = '1105214'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002001' AND SOURCE\_FIELDS-WERKS =  
    'AABG'  
    .  
      CASE SOURCE\_FIELDS-PLNBEZ.  
        WHEN '2000103'.  
          RESULT = '2305234'.  
        WHEN '2000137'.  
          RESULT = '2305234'.  
        WHEN '2000104'.  
          RESULT = '2305232'.  
        WHEN '2000187'.  
          RESULT = '2305212'.  
        WHEN '2000135'.  
          RESULT = '2305222'.  
        WHEN '2000188'.  
          RESULT = '2305214'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002003' AND SOURCE\_FIELDS-WERKS =  
    'LVAQ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = '3505413'.  
        WHEN 'AUT'.  
          RESULT = '3505413'.  
        WHEN 'CP'.  
          RESULT = '3505413'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002003' AND SOURCE\_FIELDS-WERKS =  
    'BISJ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = '3505451'.  
        WHEN 'AUT'.  
          RESULT = '3505471'.  
        WHEN 'CP'.  
          RESULT = '3505471'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002003' AND SOURCE\_FIELDS-WERKS =  
    'BIBG'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = '3505471'.  
        WHEN 'AUT'.  
          RESULT = '3505471'.  
        WHEN 'CP'.  
          RESULT = '3505471'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002003' AND SOURCE\_FIELDS-WERKS =  
    'PTX'.  
      RESULT = '9600009'.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030003002' AND SOURCE\_FIELDS-WERKS =  
    'LVAQ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = '3505413'.  
        WHEN 'AUT'.  
          RESULT = '3505413'.  
        WHEN 'CP'.  
          RESULT = '3505413'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030003002' AND SOURCE\_FIELDS-WERKS =  
    'BISJ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = '3505451'.  
        WHEN 'AUT'.  
          RESULT = '3505451'.  
        WHEN 'CP'.  
          RESULT = '3505451'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030005002' AND SOURCE\_FIELDS-WERKS =  
    'LVAQ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = '3505413'.  
        WHEN 'AUT'.  
          RESULT = '3505413'.  
        WHEN 'CP'.  
          RESULT = '3505413'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030005005' AND SOURCE\_FIELDS-WERKS =  
    'LVAQ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = '3505413'.  
        WHEN 'AUT'.  
          RESULT = '3505413'.  
        WHEN 'CP'.  
          RESULT = '3505413'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002015' AND SOURCE\_FIELDS-WERKS =  
    'PTX'.  
      RESULT = '9600018'.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002025' AND SOURCE\_FIELDS-WERKS =  
    'PTX'.  
      RESULT = '9600017'.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002026' AND SOURCE\_FIELDS-WERKS =  
    'PTX'.  
      RESULT = '9600009'.  
    ENDIF.  
  
*\*$\*$ end of routine - insert your code only before this line         \*-\**  
  ENDMETHOD.                    *"compute\_0COSTCENTER*  
*\*----------------------------------------------------------------------\**  
*\*       Inverse method invert\_0COSTCENTER*  
*\*----------------------------------------------------------------------\**  
*\*       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.*  
*\*----------------------------------------------------------------------\**  
*\*       Customer comment:*  
*\*----------------------------------------------------------------------\**  
  METHOD invert\_0COSTCENTER.  
  
*\*   IMPORTING*  
*\*     i\_r\_selset\_outbound          TYPE REF TO cl\_rsmds\_set*  
*\*     i\_th\_fields\_outbound         TYPE HASHED TABLE*  
*\*     i\_r\_selset\_outbound\_complete TYPE REF TO cl\_rsmds\_set*  
*\*     i\_r\_universe\_inbound         TYPE REF TO cl\_rsmds\_universe*  
*\*   CHANGING*  
*\*     c\_r\_selset\_inbound           TYPE REF TO cl\_rsmds\_set*  
*\*     c\_th\_fields\_inbound          TYPE HASHED TABLE*  
*\*     c\_exact                      TYPE rs\_bool*  
  
*\*$\*$ 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.                    *"invert\_0COSTCENTER*  
ENDCLASS.                    *"routine IMPLEMENTATION*

**Item 2 – Rotina no infobjeto da transformation**

PROGRAM trans\_routine.  
  
  
*\*---------------------------------------------------------------------\**  
*\*       CLASS routine DEFINITION*  
*\*---------------------------------------------------------------------\**  
*\**  
*\*---------------------------------------------------------------------\**  
CLASS lcl\_transform DEFINITION.  
  PUBLIC SECTION.  
  
*\*  Attributs*  
    DATA:  
      p\_check\_master\_data\_exist  
            TYPE RSODSOCHECKONLY READ-ONLY,  
*\*-    Instance for getting request runtime attributs;*  
*\*     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\_SC\_1,  
*\*      Field: WERKS Centro.*  
        WERKS           TYPE C LENGTH 4,  
*\*      Field: PLNBEZ Material.*  
        PLNBEZ           TYPE C LENGTH 18,  
*\*      Field: FEVOR Resp.ctrl.prod..*  
        FEVOR           TYPE C LENGTH 3,  
*\*      Field: SAKTO Conta do Razão.*  
        SAKTO           TYPE C LENGTH 10,  
*\*      Field: RECORD.*  
        RECORD           TYPE RSARECORD,  
      END   OF \_ty\_s\_SC\_1.  
    TYPES:  
      BEGIN OF \_ty\_s\_TG\_1,  
*\*      InfoObject: 0CO\_ITEM\_TX Texto da linha de lançamento CO.*  
        CO\_ITEM\_TX           TYPE /BI0/OICO\_ITEM\_TX,  
      END   OF \_ty\_s\_TG\_1.  
  
*\*$\*$ 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  
      compute\_0CO\_ITEM\_TX  
        IMPORTING  
          request                  type rsrequest  
          datapackid               type rsdatapid  
          SOURCE\_FIELDS              type \_ty\_s\_SC\_1  
          segid                    type RSBK\_SEGID  
        EXPORTING  
          RESULT                   type \_ty\_s\_TG\_1-CO\_ITEM\_TX  
          monitor                  type rstr\_ty\_t\_monitor  
        RAISING  
          cx\_rsrout\_abort  
          cx\_rsrout\_skip\_record  
          cx\_rsrout\_skip\_val  
          cx\_rsbk\_errorcount.  
    METHODS  
      invert\_0CO\_ITEM\_TX  
        IMPORTING  
          i\_th\_fields\_outbound         TYPE rstran\_t\_field\_inv  
          i\_r\_selset\_outbound          TYPE REF TO cl\_rsmds\_set  
          i\_is\_main\_selection          TYPE rs\_bool  
          i\_r\_selset\_outbound\_complete TYPE REF TO cl\_rsmds\_set  
          i\_r\_universe\_inbound         TYPE REF TO cl\_rsmds\_universe  
        CHANGING  
          c\_th\_fields\_inbound          TYPE rstran\_t\_field\_inv  
          c\_r\_selset\_inbound           TYPE REF TO cl\_rsmds\_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*  
  
*\*   Busca a tabela do infobjeto Taxa Real Dolar*  
    tables: /BIC/MTXDOL.  
  
    data: v\_taxa  type /BIC/MTXDOL-/BIC/TXVAL.  
*\*   data: v\_mes   type /BIC/MTXDOL-/BIC/TXDOL.*  
  
*\*$\*$ end of 2nd part global - insert your code only before this line   \**  
  
*\*---------------------------------------------------------------------\**  
*\*       CLASS routine IMPLEMENTATION*  
*\*---------------------------------------------------------------------\**  
*\**  
*\*---------------------------------------------------------------------\**  
CLASS lcl\_transform IMPLEMENTATION.  
  
*\*----------------------------------------------------------------------\**  
*\*       Method compute\_0CO\_ITEM\_TX*  
*\*----------------------------------------------------------------------\**  
*\*       This subroutine allows the mapping from source to target fields*  
*\*       of a transformation rule using ABAP for application specific*  
*\*       coding.*  
*\*----------------------------------------------------------------------\**  
*\*       Customer comment:*  
*\*----------------------------------------------------------------------\**  
  METHOD compute\_0CO\_ITEM\_TX.  
  
*\*   IMPORTING*  
*\*     request     type rsrequest*  
*\*     datapackid  type rsdatapid*  
*\*     SOURCE\_FIELDS-WERKS TYPE C LENGTH 000004*  
*\*     SOURCE\_FIELDS-PLNBEZ TYPE C LENGTH 000018*  
*\*     SOURCE\_FIELDS-FEVOR TYPE C LENGTH 000003*  
*\*     SOURCE\_FIELDS-SAKTO TYPE C LENGTH 000010*  
*\*    EXPORTING*  
*\*      RESULT type \_ty\_s\_TG\_1-CO\_ITEM\_TX*  
  
    DATA:  
      MONITOR\_REC    TYPE rsmonitor.  
  
*\*$\*$ 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*  
... *"to cancel the update process*  
*\*    raise exception type CX\_RSROUT\_ABORT.*  
... *"to skip a record"*  
*\*    raise exception type CX\_RSROUT\_SKIP\_RECORD.*  
... *"to clear target fields*  
*\*    raise exception type CX\_RSROUT\_SKIP\_VAL.*  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002001' AND SOURCE\_FIELDS-WERKS =  
    'LVAQ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = 'Produto Químico e Embalagem LVAQ SEC'.  
        WHEN 'AUT'.  
          RESULT = 'Produto Químico e Embalagem LVAQ AUT'.  
        WHEN 'CP'.  
          RESULT = 'Produto Químico e Embalagem LVAQ CP'.  
        WHEN 'LAB'.  
          RESULT = 'Produto Químico e Embalagem LVAQ LAB'.  
        WHEN ' '.  
          RESULT = 'Produto Químico e Embalagem LVAQ'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002001' AND SOURCE\_FIELDS-WERKS =  
    'BISJ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = 'Produto Químico e Embalagem BISJ SEC'.  
        WHEN 'AUT'.  
          RESULT = 'Produto Químico e Embalagem BISJ AUT'.  
        WHEN 'CP'.  
          RESULT = 'Produto Químico e Embalagem BISJ CP'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '000030002001' AND SOURCE\_FIELDS-WERKS =  
    'BIBG'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = 'Produto Químico e Embalagem BIBG SEC'.  
        WHEN 'AUT'.  
          RESULT = 'Produto Químico e Embalagem BIBG AUT'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002001' AND SOURCE\_FIELDS-WERKS =  
    'PTX'.  
      RESULT = 'Produto Químico PTX'.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002001' AND SOURCE\_FIELDS-WERKS =  
    'AAAQ'  
    .  
      CASE SOURCE\_FIELDS-PLNBEZ.  
        WHEN '2000103'.  
          RESULT = 'Água Tratada AAAQ'.  
        WHEN '2000137'.  
          RESULT = 'Água Desmineralizada AAAQ'.  
        WHEN '2000104'.  
          RESULT = 'Vapor AAAQ'.  
        WHEN '2000187'.  
          RESULT = 'Caldo Tratado p/ Açúcar AAAQ'.  
        WHEN '2000135'.  
          RESULT = 'Mosto AAAQ'.  
        WHEN '2000188'.  
          RESULT = 'Xarope AAAQ'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002001' AND SOURCE\_FIELDS-WERKS =  
    'AASJ'  
    .  
      CASE SOURCE\_FIELDS-PLNBEZ.  
        WHEN '2000103'.  
          RESULT = 'Água Tratada AASJ'.  
        WHEN '2000137'.  
          RESULT = 'Água Desmineralizada AASJ'.  
        WHEN '2000104'.  
          RESULT = 'Vapor AASJ'.  
        WHEN '2000187'.  
          RESULT = 'Caldo Tratado p/ Açúcar AASJ'.  
        WHEN '2000135'.  
          RESULT = 'Mosto AASJ'.  
        WHEN '2000188'.  
          RESULT = 'Xarope AASJ'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002001' AND SOURCE\_FIELDS-WERKS =  
    'AABG'  
    .  
      CASE SOURCE\_FIELDS-PLNBEZ.  
        WHEN '2000103'.  
          RESULT = 'Água Tratada AABG'.  
        WHEN '2000137'.  
          RESULT = 'Água Desmineralizada AABG'.  
        WHEN '2000104'.  
          RESULT = 'Vapor AABG'.  
        WHEN '2000187'.  
          RESULT = 'Caldo Tratado p/ Açúcar AABG'.  
        WHEN '2000135'.  
          RESULT = 'Mosto AABG'.  
        WHEN '2000188'.  
          RESULT = 'Xarope AABG'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002003' AND SOURCE\_FIELDS-WERKS =  
    'LVAQ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = 'Produto Químico e Embalagem LVAQ SEC'.  
        WHEN 'AUT'.  
          RESULT = 'Produto Químico e Embalagem LVAQ AUT'.  
        WHEN 'CP'.  
          RESULT = 'Produto Químico e Embalagem LVAQ CP'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002003' AND SOURCE\_FIELDS-WERKS =  
    'BISJ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = 'Produto Químico e Embalagem BISJ SEC'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002003' AND SOURCE\_FIELDS-WERKS =  
    'BIBG'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = 'Produto Químico e Embalagem BIBG SEC'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002003' AND SOURCE\_FIELDS-WERKS =  
    'PTX'.  
      RESULT = 'Embalagem PTX'.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030003002' AND SOURCE\_FIELDS-WERKS =  
    'LVAQ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = 'Produto Químico e Embalagem LVAQ SEC'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030003002' AND SOURCE\_FIELDS-WERKS =  
    'BISJ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = 'Produto Químico e Embalagem BISJ SEC'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030005002' AND SOURCE\_FIELDS-WERKS =  
    'LVAQ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = 'Produto Químico e Embalagem LVAQ SEC'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030005005' AND SOURCE\_FIELDS-WERKS =  
    'LVAQ'  
    .  
      CASE SOURCE\_FIELDS-FEVOR.  
        WHEN 'SEC'.  
          RESULT = 'Produto Químico e Embalagem LVAQ SEC'.  
      ENDCASE.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002015' AND SOURCE\_FIELDS-WERKS =  
    'PTX'.  
      RESULT = 'Gás Natural PTX'.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002025' AND SOURCE\_FIELDS-WERKS =  
    'PTX'.  
      RESULT = 'Energia PTX'.  
    ENDIF.  
  
    IF SOURCE\_FIELDS-SAKTO = '0030002026' AND SOURCE\_FIELDS-WERKS =  
    'PTX'.  
      RESULT = 'Água e Efluentes PTX'.  
    ENDIF.  
  
*\*$\*$ end of routine - insert your code only before this line         \*-\**  
  ENDMETHOD.                    *"compute\_0CO\_ITEM\_TX*  
*\*----------------------------------------------------------------------\**  
*\*       Inverse method invert\_0CO\_ITEM\_TX*  
*\*----------------------------------------------------------------------\**  
*\*       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.*  
*\*----------------------------------------------------------------------\**  
*\*       Customer comment:*  
*\*----------------------------------------------------------------------\**  
  METHOD invert\_0CO\_ITEM\_TX.  
  
*\*   IMPORTING*  
*\*     i\_r\_selset\_outbound          TYPE REF TO cl\_rsmds\_set*  
*\*     i\_th\_fields\_outbound         TYPE HASHED TABLE*  
*\*     i\_r\_selset\_outbound\_complete TYPE REF TO cl\_rsmds\_set*  
*\*     i\_r\_universe\_inbound         TYPE REF TO cl\_rsmds\_universe*  
*\*   CHANGING*  
*\*     c\_r\_selset\_inbound           TYPE REF TO cl\_rsmds\_set*  
*\*     c\_th\_fields\_inbound          TYPE HASHED TABLE*  
*\*     c\_exact                      TYPE rs\_bool*  
  
*\*$\*$ 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.                    *"invert\_0CO\_ITEM\_TX*  
ENDCLASS.                    *"routine IMPLEMENTATION*

**Rotina inicial**  
  
PROGRAM trans\_routine.  
  
  
*\*---------------------------------------------------------------------\**  
*\*       CLASS routine DEFINITION*  
*\*---------------------------------------------------------------------\**  
*\**  
*\*---------------------------------------------------------------------\**  
CLASS lcl\_transform DEFINITION.  
  PUBLIC SECTION.  
  
*\*  Attributs*  
    DATA:  
      p\_check\_master\_data\_exist  
            TYPE RSODSOCHECKONLY READ-ONLY,  
*\*-    Instance for getting request runtime attributs;*  
*\*     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\_SC\_1,  
*\*      Field: MLVNR Operação LM.*  
        MLVNR           TYPE C LENGTH 12,  
*\*      Field: VGABE Tp.operação.*  
        VGABE           TYPE C LENGTH 1,  
*\*      Field: GJAHR Ano DocMaterial.*  
        GJAHR           TYPE N LENGTH 4,  
*\*      Field: BELNR Doc.material.*  
        BELNR           TYPE C LENGTH 10,  
*\*      Field: BUZEI Item doc.mat..*  
        BUZEI           TYPE N LENGTH 4,  
*\*      Field: KALNR Nº cálc.csts..*  
        KALNR           TYPE N LENGTH 12,  
*\*      Field: BUDAT Data lançamento.*  
        BUDAT           TYPE D,  
*\*      Field: MENGE Quantidade.*  
        MENGE           TYPE P LENGTH 7 DECIMALS 3,  
*\*      Field: MEINS UM básica.*  
        MEINS           TYPE C LENGTH 3,  
*\*      Field: POPER Período contáb..*  
        POPER           TYPE N LENGTH 3,  
*\*      Field: KSTAR Classe de custo.*  
        KSTAR           TYPE C LENGTH 10,  
*\*      Field: BWART Tipo movimento.*  
        BWART           TYPE C LENGTH 3,  
*\*      Field: AUFNR Ordem.*  
        AUFNR           TYPE C LENGTH 12,  
*\*      Field: POSNR Nº item.*  
        POSNR           TYPE N LENGTH 4,  
*\*      Field: SAPRL Release SAP.*  
        SAPRL           TYPE C LENGTH 4,  
*\*      Field: CPUDT Entrado em.*  
        CPUDT           TYPE D,  
*\*      Field: XWOFF Cálc.valor aberto.*  
        XWOFF           TYPE C LENGTH 1,  
*\*      Field: WERKS Centro.*  
        WERKS           TYPE C LENGTH 4,  
*\*      Field: STORNO Campo de seleção.*  
        STORNO           TYPE C LENGTH 1,  
*\*      Field: BSTMG Qtd.do pedido.*  
        BSTMG           TYPE P LENGTH 7 DECIMALS 3,  
*\*      Field: BSTME UM pedido.*  
        BSTME           TYPE C LENGTH 3,  
*\*      Field: GRIR\_ACTIVE Utilização múl..*  
        GRIR\_ACTIVE           TYPE C LENGTH 1,  
*\*      Field: PRCTR Centro de lucro.*  
        PRCTR           TYPE C LENGTH 10,  
*\*      Field: KZBEW Cód.movimento.*  
        KZBEW           TYPE C LENGTH 1,  
*\*      Field: RSNUM Reserva.*  
        RSNUM           TYPE N LENGTH 10,  
*\*      Field: RSPOS Item.*  
        RSPOS           TYPE N LENGTH 4,  
*\*      Field: SJAHR Ano DocMaterial.*  
        SJAHR           TYPE N LENGTH 4,  
*\*      Field: SMBLN Doc.material.*  
        SMBLN           TYPE C LENGTH 10,  
*\*      Field: SMBLP Item doc.mat..*  
        SMBLP           TYPE N LENGTH 4,  
*\*      Field: MATNR Material.*  
        MATNR           TYPE C LENGTH 18,  
*\*      Field: CHARG Lote.*  
        CHARG           TYPE C LENGTH 10,  
*\*      Field: SHKZG Débito/Crédito.*  
        SHKZG           TYPE C LENGTH 1,  
*\*      Field: DMBTR Montante MI.*  
        DMBTR           TYPE P LENGTH 7 DECIMALS 2,  
*\*      Field: WAERS Moeda.*  
        WAERS           TYPE C LENGTH 5,  
*\*      Field: ERFMG Qtd.UM registro.*  
        ERFMG           TYPE P LENGTH 7 DECIMALS 3,  
*\*      Field: ERFME UM registro.*  
        ERFME           TYPE C LENGTH 3,  
*\*      Field: PLNBEZ Material.*  
        PLNBEZ           TYPE C LENGTH 18,  
*\*      Field: FEVOR Resp.ctrl.prod..*  
        FEVOR           TYPE C LENGTH 3,  
*\*      Field: IGMNG Qtd.confirmada.*  
        IGMNG           TYPE P LENGTH 7 DECIMALS 3,  
*\*      Field: GMEIN UM básica.*  
        GMEIN           TYPE C LENGTH 3,  
*\*      Field: CURTP Tpo moeda/aval..*  
        CURTP           TYPE C LENGTH 2,  
*\*      Field: PEINH Unid.preço.*  
        PEINH           TYPE P LENGTH 3 DECIMALS 0,  
*\*      Field: VPRSV Controle preço.*  
        VPRSV           TYPE C LENGTH 1,  
*\*      Field: STPRS Preço-padrão.*  
        STPRS           TYPE P LENGTH 6 DECIMALS 2,  
*\*      Field: WAERS002 Moeda.*  
        WAERS002           TYPE C LENGTH 5,  
*\*      Field: PVPRS Preço int.per..*  
        PVPRS           TYPE P LENGTH 6 DECIMALS 2,  
*\*      Field: WAERS003 Moeda.*  
        WAERS003           TYPE C LENGTH 5,  
*\*      Field: BWART001 Tipo movimento.*  
        BWART001           TYPE C LENGTH 3,  
*\*      Field: MJAHR Ano DocMaterial.*  
        MJAHR           TYPE N LENGTH 4,  
*\*      Field: MBLNR Doc.material.*  
        MBLNR           TYPE C LENGTH 10,  
*\*      Field: ZEILE Item doc.mat..*  
        ZEILE           TYPE N LENGTH 4,  
*\*      Field: SAKTO Conta do Razão.*  
        SAKTO           TYPE C LENGTH 10,  
*\*      Field: VGART Tp.operação.*  
        VGART           TYPE C LENGTH 2,  
*\*      Field: BLART Tp.doc..*  
        BLART           TYPE C LENGTH 2,  
*\*      Field: USNAM Nome do usuário.*  
        USNAM           TYPE C LENGTH 12,  
*\*      Field: XBLNR Referência.*  
        XBLNR           TYPE C LENGTH 16,  
*\*      Field: BKTXT Txt.cab.doc..*  
        BKTXT           TYPE C LENGTH 25,  
*\*      Field: TCODE2 Código transação.*  
        TCODE2           TYPE C LENGTH 20,  
*\*      Field: BISMT Nº mat.antigo.*  
        BISMT           TYPE C LENGTH 18,  
*\*      Field: MATKL GrpMercads..*  
        MATKL           TYPE C LENGTH 9,  
*\*      Field: MTART Tipo material.*  
        MTART           TYPE C LENGTH 4,  
*\*      Field: MBRSH SetorIndustrial.*  
        MBRSH           TYPE C LENGTH 1,  
*\*      Field: AUART Tipo de ordem.*  
        AUART           TYPE C LENGTH 4,  
*\*      Field: AUTYP Categoria ordem.*  
        AUTYP           TYPE N LENGTH 2,  
*\*      Field: RECORD Nº de registro.*  
        RECORD           TYPE RSARECORD,  
      END   OF \_ty\_s\_SC\_1.  
    TYPES:  
      \_ty\_t\_SC\_1        TYPE STANDARD TABLE OF \_ty\_s\_SC\_1  
                        WITH NON-UNIQUE DEFAULT KEY.  
  
*\*$\*$ 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  
      start\_routine  
        IMPORTING  
          request                  type rsrequest  
          datapackid               type rsdatapid  
          segid                    type rsbk\_segid  
        EXPORTING  
          monitor                  type rstr\_ty\_t\_monitors  
        CHANGING  
          SOURCE\_PACKAGE              type \_ty\_t\_SC\_1  
        RAISING  
          cx\_rsrout\_abort  
          cx\_rsbk\_errorcount.  
    METHODS  
      inverse\_start\_routine  
        IMPORTING  
          i\_th\_fields\_outbound         TYPE rstran\_t\_field\_inv  
          I\_R\_SELSET\_OUTBOUND          TYPE REF TO CL\_RSMDS\_SET  
          i\_is\_main\_selection          TYPE rs\_bool  
          i\_r\_selset\_outbound\_complete TYPE REF TO cl\_rsmds\_set  
          i\_r\_universe\_inbound         TYPE REF TO CL\_RSMDS\_UNIVERSE  
        CHANGING  
          c\_th\_fields\_inbound          TYPE rstran\_t\_field\_inv  
          c\_r\_selset\_inbound           TYPE REF TO CL\_RSMDS\_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*  
*\*$\*$ end of 2nd part global - insert your code only before this line   \**  
  
*\*---------------------------------------------------------------------\**  
*\*       CLASS routine IMPLEMENTATION*  
*\*---------------------------------------------------------------------\**  
*\**  
*\*---------------------------------------------------------------------\**  
CLASS lcl\_transform IMPLEMENTATION.  
  
*\*----------------------------------------------------------------------\**  
*\*       Method start\_routine*  
*\*----------------------------------------------------------------------\**  
*\*       Calculation of source package via start routine*  
*\*----------------------------------------------------------------------\**  
*\*   <-> source package*  
*\*----------------------------------------------------------------------\**  
  METHOD start\_routine.  
*\*=== Segments ===*  
  
    FIELD-SYMBOLS:  
      <SOURCE\_FIELDS>    TYPE \_ty\_s\_SC\_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*  
... *"to cancel the update process*  
*\*    raise exception type CX\_RSROUT\_ABORT.*  
  
    data: e\_s\_result   type \_ty\_s\_SC\_1.  
    data: e\_t\_result   type \_ty\_t\_SC\_1.  
    data: v\_budat      type c length 6.  
*\**  
  
    v\_budat = sy-datum(6).  
  
    BREAK-POINT.  
    loop at SOURCE\_PACKAGE into e\_s\_result where  
      ( SAKTO = '0030002001' OR  
      SAKTO = '0030002003' OR  
      SAKTO = '0030003002' OR  
      SAKTO = '0030005002' OR  
      SAKTO = '0030005005' OR  
      SAKTO = '0030002015' OR  
      SAKTO = '0030002025' OR  
      SAKTO = '0030002026') AND  
      ( BWART = '261' OR  
      BWART = '262').  
  
*\*      e\_s\_result-BUDAT(6) = '201708'.*  
  
      if e\_s\_result-WERKS = 'PTX'.  
        e\_s\_result-CURTP = '40'.  
      else.  
        e\_s\_result-CURTP = '10'.  
      endif.  
  
      COLLECT e\_s\_result into e\_t\_result.  
    endloop.  
  
    clear SOURCE\_PACKAGE.  
    move e\_t\_result[] to SOURCE\_PACKAGE[].  
  
  
*\*$\*$ end of routine - insert your code only before this line         \*-\**  
  ENDMETHOD.                    *"start\_routine*  
*\*----------------------------------------------------------------------\**  
*\*       Inverse method inverse\_start\_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.*  
*\*----------------------------------------------------------------------\**  
*\*       Customer comment:*  
*\*----------------------------------------------------------------------\**  
  METHOD inverse\_start\_routine.  
  
*\*   IMPORTING*  
*\*     i\_r\_selset\_outbound          TYPE REF TO cl\_rsmds\_set*  
*\*     i\_th\_fields\_outbound         TYPE HASHED TABLE*  
*\*     i\_r\_selset\_outbound\_complete TYPE REF TO cl\_rsmds\_set*  
*\*     i\_r\_universe\_inbound         TYPE REF TO cl\_rsmds\_universe*  
*\*   CHANGING*  
*\*     c\_r\_selset\_inbound           TYPE REF TO cl\_rsmds\_set*  
*\*     c\_th\_fields\_inbound          TYPE HASHED TABLE*  
*\*     c\_exact                      TYPE rs\_bool*  
  
*\*$\*$ 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\_start\_routine*  
ENDCLASS.                    *"routine IMPLEMENTATION*

**Rotina final**

PROGRAM trans\_routine.  
  
  
*\*---------------------------------------------------------------------\**  
*\*       CLASS routine DEFINITION*  
*\*---------------------------------------------------------------------\**  
*\**  
*\*---------------------------------------------------------------------\**  
CLASS lcl\_transform DEFINITION.  
  PUBLIC SECTION.  
  
*\*  Attributs*  
    DATA:  
      p\_check\_master\_data\_exist  
            TYPE RSODSOCHECKONLY READ-ONLY,  
*\*-    Instance for getting request runtime attributs;*  
*\*     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: SID.*  
        SID           TYPE RSSID,  
*\*      Field: DATAPAKID.*  
        DATAPAKID           TYPE RSDATAPID,  
*\*      Field: RECORD.*  
        RECORD           TYPE RSARECORD,  
*\*      InfoObject: 0CO\_DOC\_NO Nº doc.do documento CO.*  
        CO\_DOC\_NO           TYPE /BI0/OICO\_DOC\_NO,  
*\*      InfoObject: 0COSTCENTER Centro de custo.*  
        COSTCENTER           TYPE /BI0/OICOSTCENTER,  
*\*      InfoObject: 0COSTELMNT Classe de custo.*  
        COSTELMNT           TYPE /BI0/OICOSTELMNT,  
*\*      InfoObject: 0PLANT Centro.*  
        PLANT           TYPE /BI0/OIPLANT,  
*\*      InfoObject: 0CO\_AREA Área de contabilidade de custos.*  
        CO\_AREA           TYPE /BI0/OICO\_AREA,  
*\*      InfoObject: 0CALYEAR Ano civil.*  
        CALYEAR           TYPE /BI0/OICALYEAR,  
*\*      InfoObject: 0CALMONTH2 Mês calendário.*  
        CALMONTH2           TYPE /BI0/OICALMONTH2,  
*\*      InfoObject: 0CO\_ITEM\_NO Linha de lançamento do documento CO.*  
        CO\_ITEM\_NO           TYPE /BI0/OICO\_ITEM\_NO,  
*\*      InfoObject: 0MATERIAL Material.*  
        MATERIAL           TYPE /BI0/OIMATERIAL,  
*\*      InfoObject: 0RSPL\_ORG Unidade responsável.*  
        RSPL\_ORG           TYPE /BI0/OIRSPL\_ORG,  
*\*      InfoObject: 0MATL\_TYPE Tipo de material.*  
        MATL\_TYPE           TYPE /BI0/OIMATL\_TYPE,  
*\*      InfoObject: 0CO\_ITEM\_TX Texto da linha de lançamento CO.*  
        CO\_ITEM\_TX           TYPE /BI0/OICO\_ITEM\_TX,  
*\*      InfoObject: 0CURRENCY Código da moeda.*  
        CURRENCY           TYPE /BI0/OICURRENCY,  
*\*      InfoObject: 0USER\_NAME Nome do usuário.*  
        USER\_NAME           TYPE /BI0/OIUSER\_NAME,  
*\*      InfoObject: 0CALDAY Dia de calendário.*  
        CALDAY           TYPE /BI0/OICALDAY,  
*\*      InfoObject: 0AMOUNT Montante.*  
        AMOUNT           TYPE /BI0/OIAMOUNT,  
*\*      InfoObject: 0RECORDMODE Processo delta BW: modo de atualização.*  
        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.  
  
*\*$\*$ 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  
          segid                    type rsbk\_segid  
        EXPORTING  
          monitor                  type rstr\_ty\_t\_monitors  
        CHANGING  
          RESULT\_PACKAGE              type \_ty\_t\_TG\_1  
        RAISING  
          cx\_rsrout\_abort  
          cx\_rsbk\_errorcount.  
    METHODS  
      inverse\_end\_routine  
        IMPORTING  
          i\_th\_fields\_outbound         TYPE rstran\_t\_field\_inv  
          I\_R\_SELSET\_OUTBOUND          TYPE REF TO CL\_RSMDS\_SET  
          i\_is\_main\_selection          TYPE rs\_bool  
          i\_r\_selset\_outbound\_complete TYPE REF TO cl\_rsmds\_set  
          i\_r\_universe\_inbound         TYPE REF TO CL\_RSMDS\_UNIVERSE  
        CHANGING  
          c\_th\_fields\_inbound          TYPE rstran\_t\_field\_inv  
          c\_r\_selset\_inbound           TYPE REF TO CL\_RSMDS\_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*  
*\*$\*$ 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*  
... *"to cancel the update process*  
*\*    raise exception type CX\_RSROUT\_ABORT.*  
  
    data: e\_s\_result   type \_ty\_s\_TG\_1.  
    data: e\_t\_result   type \_ty\_t\_TG\_1.  
    data: v\_budat      type c length 6.  
*\**  
  
    v\_budat = sy-datum(6).  
  
    BREAK-POINT.  
    loop at RESULT\_PACKAGE into e\_s\_result where COSTCENTER NE ' '.  
  
      COLLECT e\_s\_result into e\_t\_result.  
    endloop.  
  
    clear RESULT\_PACKAGE .  
    move e\_t\_result[] to RESULT\_PACKAGE[].  
  
  
*\*$\*$ end of routine - insert your code only before this line         \*-\**  
  ENDMETHOD.                    *"end\_routine*  
*\*----------------------------------------------------------------------\**  
*\*       Inverse 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.*  
*\*----------------------------------------------------------------------\**  
*\*       Customer comment:*  
*\*----------------------------------------------------------------------\**  
  METHOD inverse\_end\_routine.  
  
*\*   IMPORTING*  
*\*     i\_r\_selset\_outbound          TYPE REF TO cl\_rsmds\_set*  
*\*     i\_th\_fields\_outbound         TYPE HASHED TABLE*  
*\*     i\_r\_selset\_outbound\_complete TYPE REF TO cl\_rsmds\_set*  
*\*     i\_r\_universe\_inbound         TYPE REF TO cl\_rsmds\_universe*  
*\*   CHANGING*  
*\*     c\_r\_selset\_inbound           TYPE REF TO cl\_rsmds\_set*  
*\*     c\_th\_fields\_inbound          TYPE HASHED TABLE*  
*\*     c\_exact                      TYPE rs\_bool*  
  
*\*$\*$ 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*