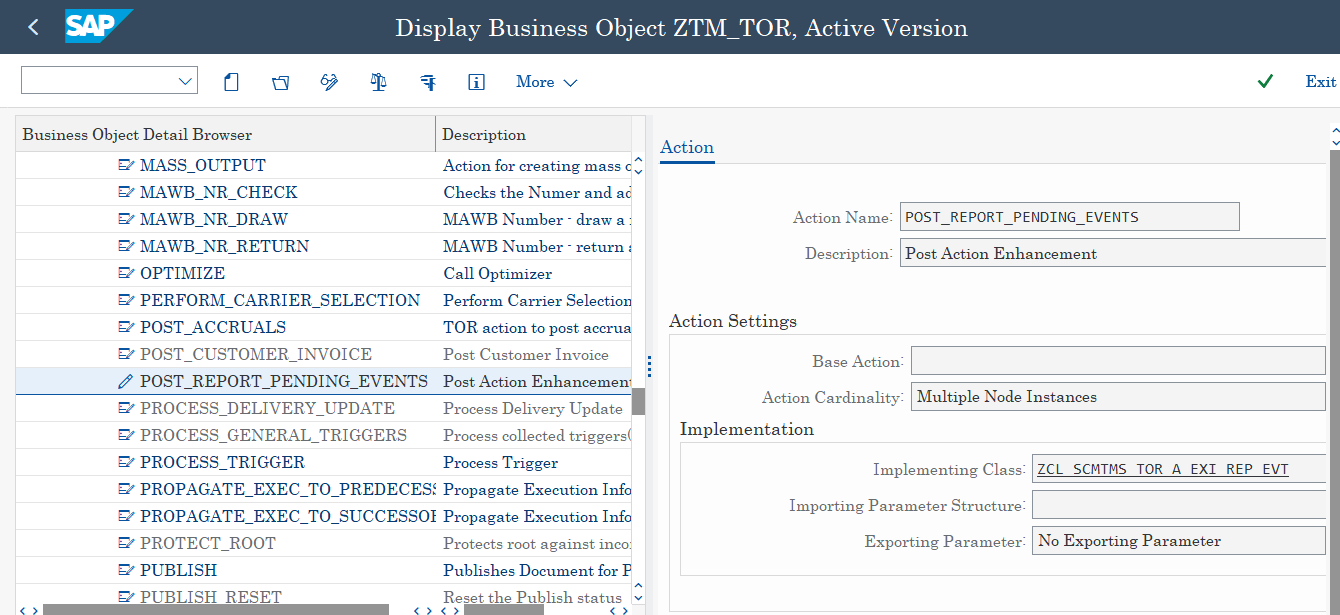
SAP TM – Update POD status in outbound delivery

SAP BO : /SCMTMS/TOR

Scenario: Insert POD event in the execution tab in the Freight order and press save – POD status should be updated as Completed in all outbound deliveries.

Create Post action enhancement:

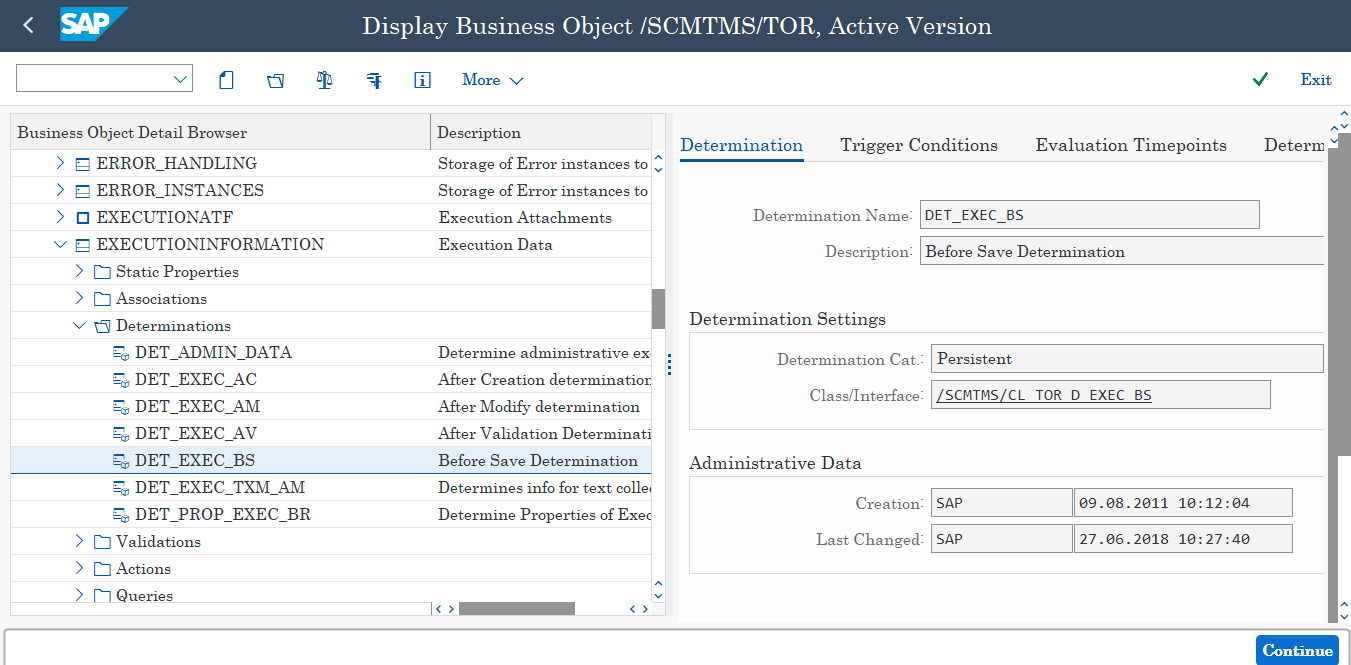


Method : Execute

METHOD /bobf/if\_frw\_action~execute.  
  
  TYPES:BEGIN OF ty\_execution\_info,  
          event\_code   TYPE /scmtms/tor\_event,  
          key          TYPE /bobf/conf\_key,  
          execution\_id TYPE /scmtms/item\_id,  
          status(1)    TYPE c,  
        END OF   ty\_execution\_info.  
  
  DATA:lv\_outbound\_del TYPE vbeln\_vl,  
       lv\_del\_item\_no  TYPE posnr\_vl.  
  
  DATA:ls\_vbkok          TYPE vbkok,  
       ls\_symsg          TYPE symsg,  
       ls\_execution\_info TYPE ty\_execution\_info.  
  
  DATA:lt\_vbpok     TYPE vbpok\_t,  
       lt\_verpo     TYPE tvpodvb\_t,  
       lt\_prot      TYPE tab\_prott,  
       lt\_d\_exec\_tr TYPE /scmtms/t\_tor\_exec\_tr\_k.  
  
  DATA:lr\_d\_exec\_tr      TYPE REF TO /scmtms/s\_tor\_exec\_tr\_k,  
       lr\_param\_execinfo TYPE REF TO /scmtms/s\_tor\_c\_execinfo\_tr.  
  
\*-------------------------------------------------------------------------------------------------  
\*Check if POD event reported  
\*-------------------------------------------------------------------------------------------------  
  
  CREATE DATA lr\_param\_execinfo.  
  lr\_param\_execinfo->do\_not\_create = abap\_true.  
  
  io\_read->retrieve\_by\_association(  
    EXPORTING  
      it\_key         = it\_key  
      iv\_association = /scmtms/if\_tor\_c=>sc\_association-root-executioninformation\_tr  
      iv\_node        = /scmtms/if\_tor\_c=>sc\_node-root  
      iv\_fill\_data   = abap\_true  
      is\_parameters  = lr\_param\_execinfo  
    IMPORTING  
      et\_data        = lt\_d\_exec\_tr ).  
  IF lt\_d\_exec\_tr IS INITIAL.  
    RETURN.  
  ENDIF.  
  
  LOOP AT lt\_d\_exec\_tr REFERENCE INTO lr\_d\_exec\_tr USING KEY to\_be\_reported WHERE to\_be\_reported EQ abap\_true AND  
                                                                                  event\_code     EQ 'POD'.  
    DATA(lv\_upd\_pod\_status) = abap\_true.  
    DATA(lv\_execution\_key)  = lr\_d\_exec\_tr->key.  
    DATA(lv\_root\_key)       = lr\_d\_exec\_tr->root\_key.  
    DATA(lv\_event\_code)     = lr\_d\_exec\_tr->event\_code.  
    DATA(lv\_execution\_id)   = lr\_d\_exec\_tr->execution\_id.  
    EXIT.  
  ENDLOOP.  
  
\*-------------------------------------------------------------------------------------------------  
  
\*-------------------------------------------------------------------------------------------------  
\*Read Root and Item Node data  
\*-------------------------------------------------------------------------------------------------  
  IF lv\_upd\_pod\_status EQ abap\_true.  
  
    /scmtms/cl\_tor\_helper\_read=>get\_tor\_data(  
      EXPORTING  
        it\_root\_key  = it\_key  
        io\_read      = io\_read  
        is\_read\_req  = VALUE #( et\_root      = abap\_true  
                                et\_all\_items = abap\_true )  
      IMPORTING  
        et\_all\_items = DATA(lt\_d\_items)  
        et\_root      = DATA(lt\_d\_root) ).  
  
\*-------------------------------------------------------------------------------------------------  
\*Read current Freight order  
\*-------------------------------------------------------------------------------------------------  
    IF line\_exists( lt\_d\_root[ key      = it\_key[ 1 ]-key  
                               tor\_type = 'ZFO1' ] ).  
      DATA(lv\_freight\_order) = lt\_d\_root[ key      = it\_key[ 1 ]-key  
                                          tor\_type = 'ZFO1' ]-tor\_id.  
      DATA(lv\_key) = lt\_d\_root[ key      = it\_key[ 1 ]-key  
                                tor\_type = 'ZFO1' ]-key.  
    ENDIF.  
  
\*-------------------------------------------------------------------------------------------------  
\*Prepare and Check outbound delivery.  
\*-------------------------------------------------------------------------------------------------  
    IF lv\_freight\_order IS NOT INITIAL.  
      LOOP AT lt\_d\_items ASSIGNING FIELD-SYMBOL(<lfs\_d\_items>) WHERE item\_type  EQ 'PRD'  AND  
                                                                     trq\_cat    EQ '06'   AND  
                                                                     base\_btd\_id IS NOT INITIAL.  
  
        DATA(lv\_base\_btd\_id) = <lfs\_d\_items>-base\_btd\_id.  
        CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_OUTPUT'  
          EXPORTING  
            input  = lv\_base\_btd\_id  
          IMPORTING  
            output = lv\_base\_btd\_id.  
  
        CLEAR lv\_outbound\_del.  
        lv\_outbound\_del = lv\_base\_btd\_id.  
  
        CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
          EXPORTING  
            input  = lv\_outbound\_del  
          IMPORTING  
            output = lv\_outbound\_del.  
  
\*Check if the delivery is not yet processed  
        SELECT SINGLE @abap\_true  
        FROM likp  
        INTO @DATA(lv\_exists)  
        WHERE vbeln EQ @lv\_outbound\_del AND  
              pdstk EQ 'A'.  
        IF sy-subrc EQ 0.  
  
          CLEAR ls\_vbkok.  
          ls\_vbkok = VALUE #( podat =  sy-datum  
                              potim =  sy-uzeit  
                              kzpod =  'B' ).  
  
          CLEAR:lv\_del\_item\_no,lt\_vbpok,lt\_verpo.  
          lv\_del\_item\_no = <lfs\_d\_items>-base\_btditem\_id.  
          lt\_vbpok = VALUE #( ( vbeln\_vl = lv\_outbound\_del  
                                posnr\_vl = lv\_del\_item\_no  
                                kzpod    = 'X') ).  
  
          CALL FUNCTION 'WS\_DELIVERY\_UPDATE'  
            EXPORTING  
              vbkok\_wa                 = ls\_vbkok  
              delivery                 = lv\_outbound\_del  
              if\_error\_messages\_send\_0 = space  
              commit                   = abap\_false  
              if\_database\_update       = '2'  
            TABLES  
              tvpod\_tab                = lt\_verpo  
              vbpok\_tab                = lt\_vbpok  
              prot                     = lt\_prot.  
  
          IF line\_exists( lt\_prot[ msgty = 'E' ] ).  
  
            /scmtms/cl\_msg\_helper=>msg\_helper\_add\_symsg(  
              EXPORTING  
                iv\_bo\_key    = /scmtms/if\_tor\_c=>sc\_bo\_key  
                iv\_node\_key  = /scmtms/if\_tor\_c=>sc\_node-root  
                iv\_key       = lv\_root\_key  
                iv\_probclass = /scmtms/cl\_applog\_helper=>sc\_al\_probclass\_important  
              CHANGING  
                co\_message   = eo\_message ).  
  
            et\_failed\_key = VALUE #( ( key = lv\_root\_key ) ).  
  
            LOOP AT lt\_prot ASSIGNING FIELD-SYMBOL(<lfs\_prot>) WHERE msgty = 'E'.  
  
              CLEAR ls\_symsg.  
              ls\_symsg = VALUE #( msgty = <lfs\_prot>-msgty  
                                  msgid = <lfs\_prot>-msgid  
                                  msgno = <lfs\_prot>-msgno  
                                  msgv1 = <lfs\_prot>-msgv1  
                                  msgv2 = <lfs\_prot>-msgv2  
                                  msgv3 = <lfs\_prot>-msgv3  
                                  msgv4 = <lfs\_prot>-msgv4 ).  
  
              IF eo\_message IS BOUND.  
                eo\_message->add\_message(  
                  is\_msg  = ls\_symsg  
                  iv\_node = /scmtms/if\_tor\_c=>sc\_node-root  
                  iv\_key  = lv\_root\_key ).  
              ENDIF.  
            ENDLOOP.  
  
\*Pass data to buffer to skip the POD event save  
            ls\_execution\_info = VALUE #( event\_code   = lv\_event\_code  
                                         key          = lv\_execution\_key  
                                         execution\_id = lv\_execution\_id  
                                         status       = 'E' ).  
  
            EXPORT ls\_execution\_info FROM ls\_execution\_info TO MEMORY ID 'ZPOD\_EVT'.  
  
            RETURN.  
          ELSE.  
            CALL FUNCTION 'DEQUEUE\_EVVBLKE'  
              EXPORTING  
                mode\_likp = 'E'  
                mandt     = sy-mandt  
                vbeln     = lv\_outbound\_del.  
          ENDIF.  
        ENDIF.  
      ENDLOOP.  
  
\*Dequeue Sales Order  
      SELECT SINGLE vgbel  
      FROM lips  
      INTO @DATA(lv\_sales\_order)  
      WHERE vbeln EQ @lv\_outbound\_del.  
      IF sy-subrc EQ 0.  
        CALL FUNCTION 'DEQUEUE\_EVVBAKE'  
          EXPORTING  
            mode\_vbak = 'E'  
            mandt     = sy-mandt  
            vbeln     = lv\_sales\_order.  
      ENDIF.  
  
\*-------------------------------------------------------------------------------------------------  
\*Prepare and Update outbound delivery POD status as Confirmed.  
\*-------------------------------------------------------------------------------------------------  
      LOOP AT lt\_d\_items ASSIGNING <lfs\_d\_items> WHERE item\_type   EQ 'PRD'  AND  
                                                       trq\_cat     EQ '06'   AND  
                                                       base\_btd\_id IS NOT INITIAL.  
  
        CLEAR lv\_base\_btd\_id.  
        lv\_base\_btd\_id = <lfs\_d\_items>-base\_btd\_id.  
        CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_OUTPUT'  
          EXPORTING  
            input  = lv\_base\_btd\_id  
          IMPORTING  
            output = lv\_base\_btd\_id.  
  
        CLEAR lv\_outbound\_del.  
        lv\_outbound\_del = lv\_base\_btd\_id.  
  
        CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
          EXPORTING  
            input  = lv\_outbound\_del  
          IMPORTING  
            output = lv\_outbound\_del.  
  
\*Check if the delivery is not yet processed  
        CLEAR lv\_exists.  
        SELECT SINGLE @abap\_true  
        FROM likp  
        INTO @lv\_exists  
        WHERE vbeln EQ @lv\_outbound\_del AND  
              pdstk EQ 'A'.  
        IF sy-subrc EQ 0.  
  
          CLEAR ls\_vbkok.  
          ls\_vbkok = VALUE #( podat =  sy-datum  
                              potim =  sy-uzeit  
                              kzpod =  'B' ).  
  
          CLEAR:lv\_del\_item\_no,lt\_vbpok,lt\_verpo.  
          lv\_del\_item\_no = <lfs\_d\_items>-base\_btditem\_id.  
          lt\_vbpok = VALUE #( ( vbeln\_vl = lv\_outbound\_del  
                                posnr\_vl = lv\_del\_item\_no  
                                kzpod    = 'X') ).  
  
          CALL FUNCTION 'ZTM\_OUTBOUND\_DELIVERY\_UPDATE' IN UPDATE TASK  
            EXPORTING  
              is\_vbkok     = ls\_vbkok  
              iv\_delivery  = lv\_outbound\_del  
              iv\_vbeln     = lv\_sales\_order  
            TABLES  
              it\_vbpok\_tab = lt\_vbpok  
              it\_vpod\_tab  = lt\_verpo.  
  
          CALL FUNCTION 'DEQUEUE\_EVVBAKE'  
            EXPORTING  
              mode\_vbak = 'E'  
              mandt     = sy-mandt  
              vbeln     = lv\_sales\_order.  
  
        ENDIF.  
      ENDLOOP.  
  
\*-------------------------------------------------------------------------------------------------  
  
    ENDIF.  
  ENDIF.  
  
ENDMETHOD.

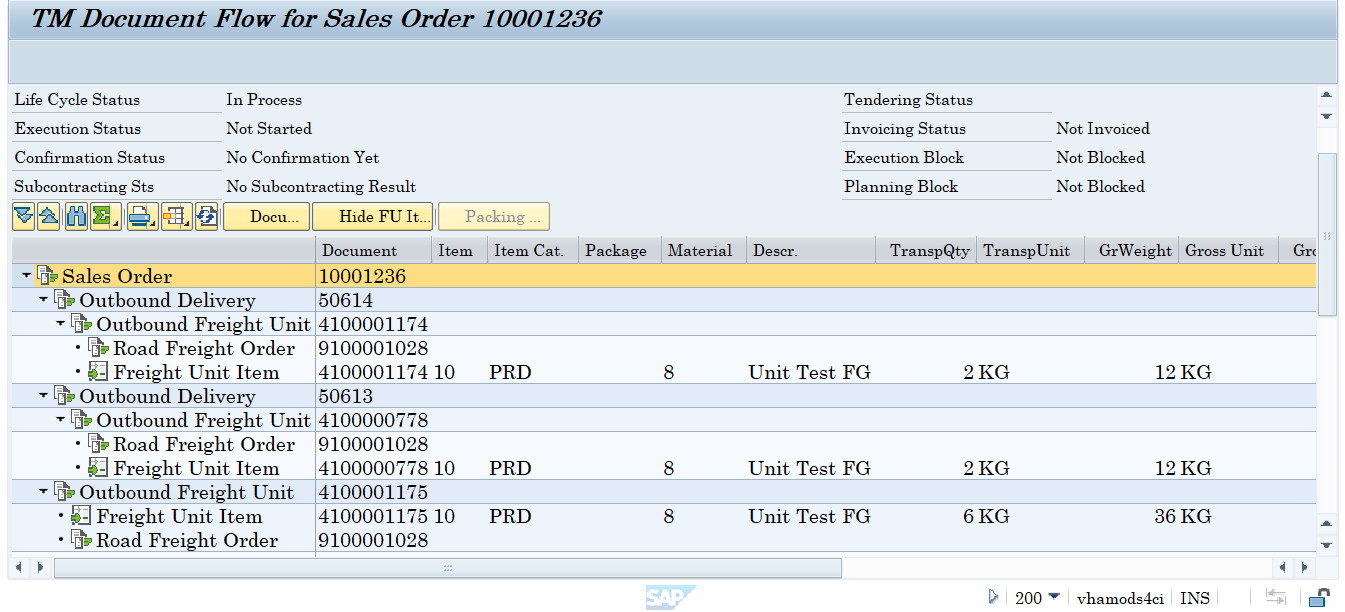
FUNCTION ztm\_outbound\_delivery\_update.  
\*"----------------------------------------------------------------------  
\*"\*"Update Function Module:  
\*"  
\*"\*"Local Interface:  
\*"  IMPORTING  
\*"     VALUE(IS\_VBKOK) LIKE  VBKOK STRUCTURE  VBKOK  
\*"     VALUE(IV\_DELIVERY) LIKE  LIKP-VBELN  
\*"     VALUE(IV\_VBELN) TYPE  VBELN\_VA  
\*"  TABLES  
\*"      IT\_VBPOK\_TAB STRUCTURE  VBPOK OPTIONAL  
\*"      IT\_VPOD\_TAB STRUCTURE  TVPODVB OPTIONAL  
\*"----------------------------------------------------------------------  
  
  CALL FUNCTION 'DEQUEUE\_EVVBAKE'  
    EXPORTING  
      mode\_vbak = 'E'  
      mandt     = sy-mandt  
      vbeln     = iv\_vbeln.  
  
  CALL FUNCTION 'WS\_DELIVERY\_UPDATE'  
    EXPORTING  
      vbkok\_wa  = is\_vbkok  
      delivery  = iv\_delivery  
    TABLES  
      tvpod\_tab = it\_vpod\_tab  
      vbpok\_tab = it\_vbpok\_tab.  
  
ENDFUNCTION.

In order to cancel the save of POD event , enhance determination class and create POST exit on the execute method:

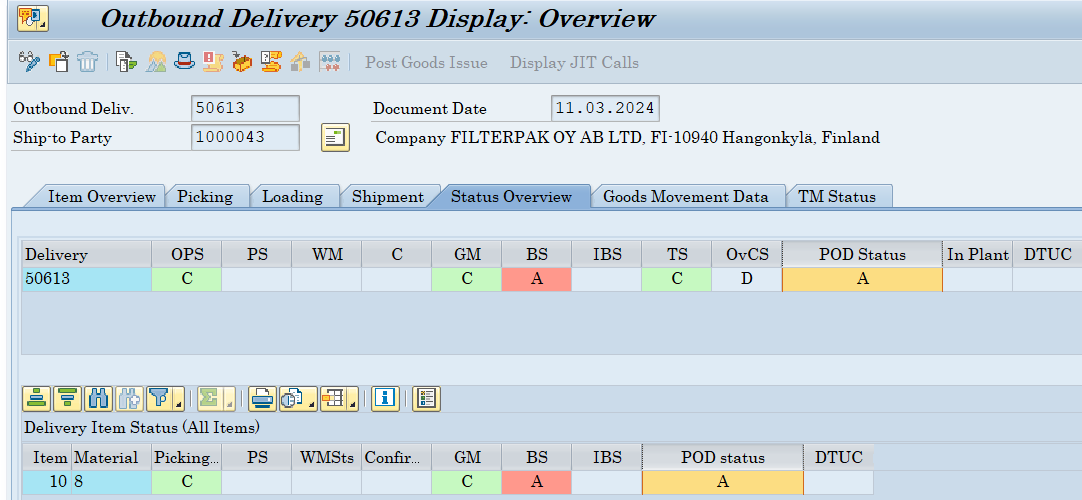


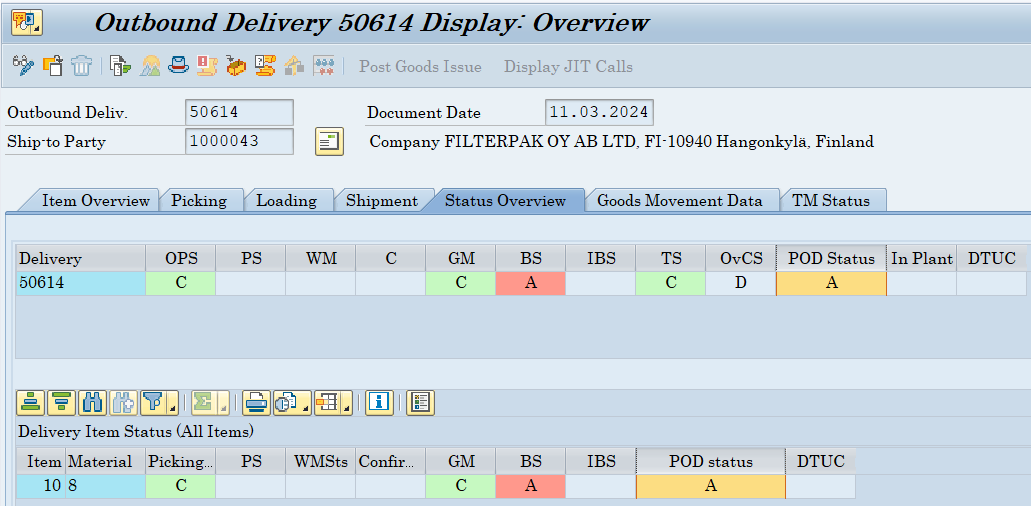
METHOD ipo\_ztm\_ei\_pod\_exec\_bs~execute.  
\*"------------------------------------------------------------------------\*  
\*" Declaration of POST-method, do not insert any comments here please!  
\*"  
\*"methods EXECUTE  
\*"  importing  
\*"    !IS\_CTX type /BOBF/S\_FRW\_CTX\_DET  
\*"    !IT\_KEY type /BOBF/T\_FRW\_KEY  
\*"    !IO\_READ type ref to /BOBF/IF\_FRW\_READ  
\*"    !IO\_MODIFY type ref to /BOBF/IF\_FRW\_MODIFY  
\*"  changing  
\*"    !EO\_MESSAGE type ref to /BOBF/IF\_FRW\_MESSAGE  
\*"    !ET\_FAILED\_KEY type /BOBF/T\_FRW\_KEY  
\*"  raising  
\*"    /BOBF/CX\_FRW .  
\*"------------------------------------------------------------------------\*  
  
    TYPES:BEGIN OF ty\_execution\_info,  
            event\_code   TYPE /scmtms/tor\_event,  
            key          TYPE /bobf/conf\_key,  
            execution\_id TYPE /scmtms/item\_id,  
            status(1)    TYPE c,  
          END OF   ty\_execution\_info.  
  
    DATA:ls\_execution\_info TYPE ty\_execution\_info.  
  
    IMPORT ls\_execution\_info TO ls\_execution\_info FROM MEMORY ID 'ZPOD\_EVT'.  
    IF ls\_execution\_info IS NOT INITIAL.  
      FREE MEMORY ID 'ZPOD\_EVT'.  
  
      IF ls\_execution\_info-event\_code EQ 'POD' AND  
         ls\_execution\_info-status     EQ 'E'.  
        et\_failed\_key = VALUE #( ( key = ls\_execution\_info-key ) ).  
      ENDIF.  
    ENDIF.  
  
  ENDMETHOD.  
ENDCLASS.

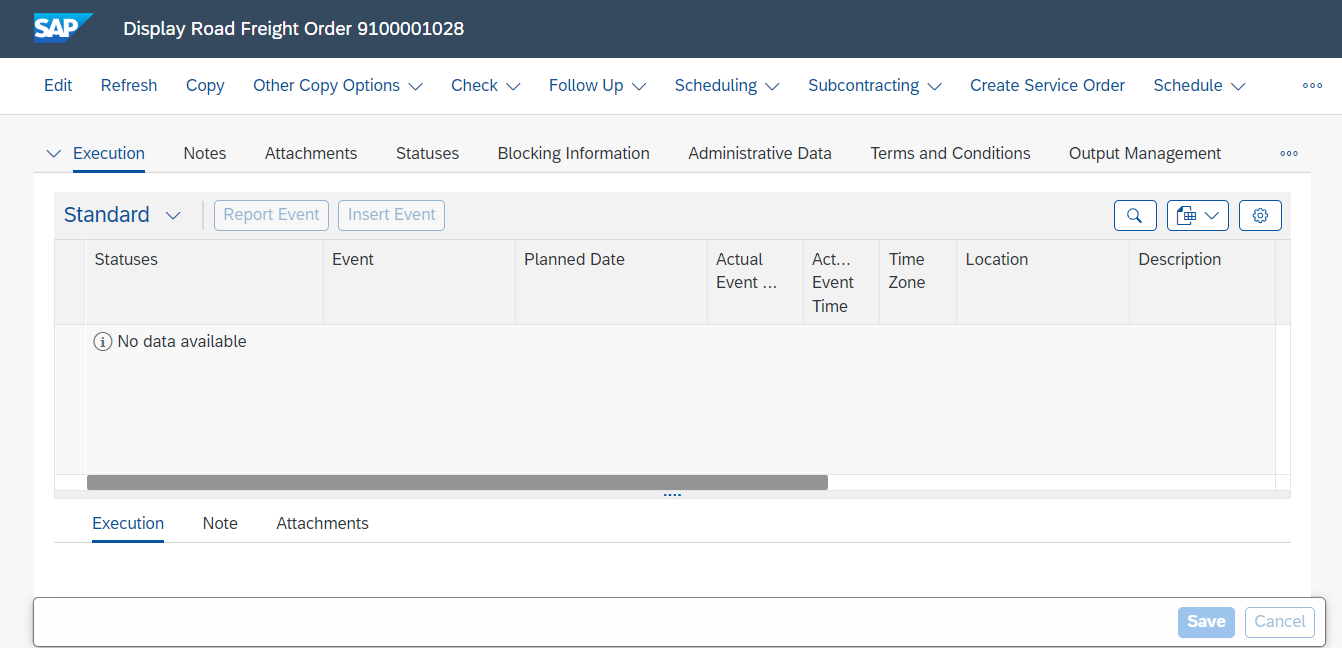
Unit Test Results:



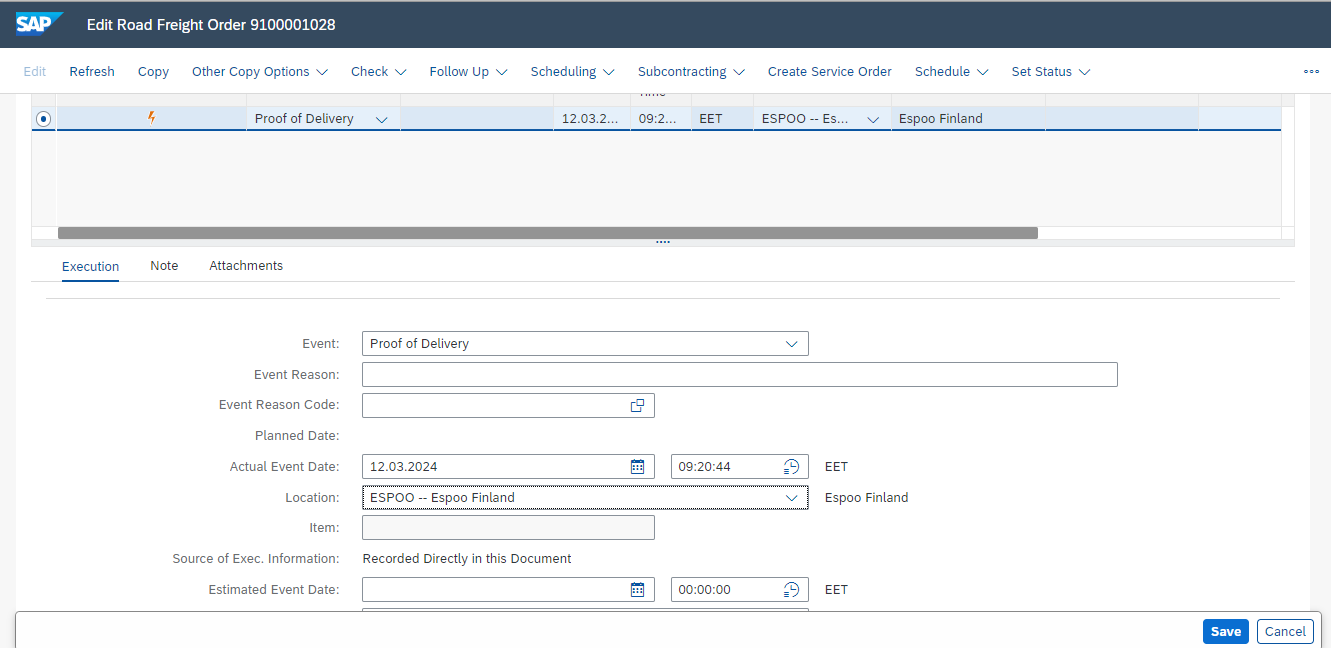
Before POD event report

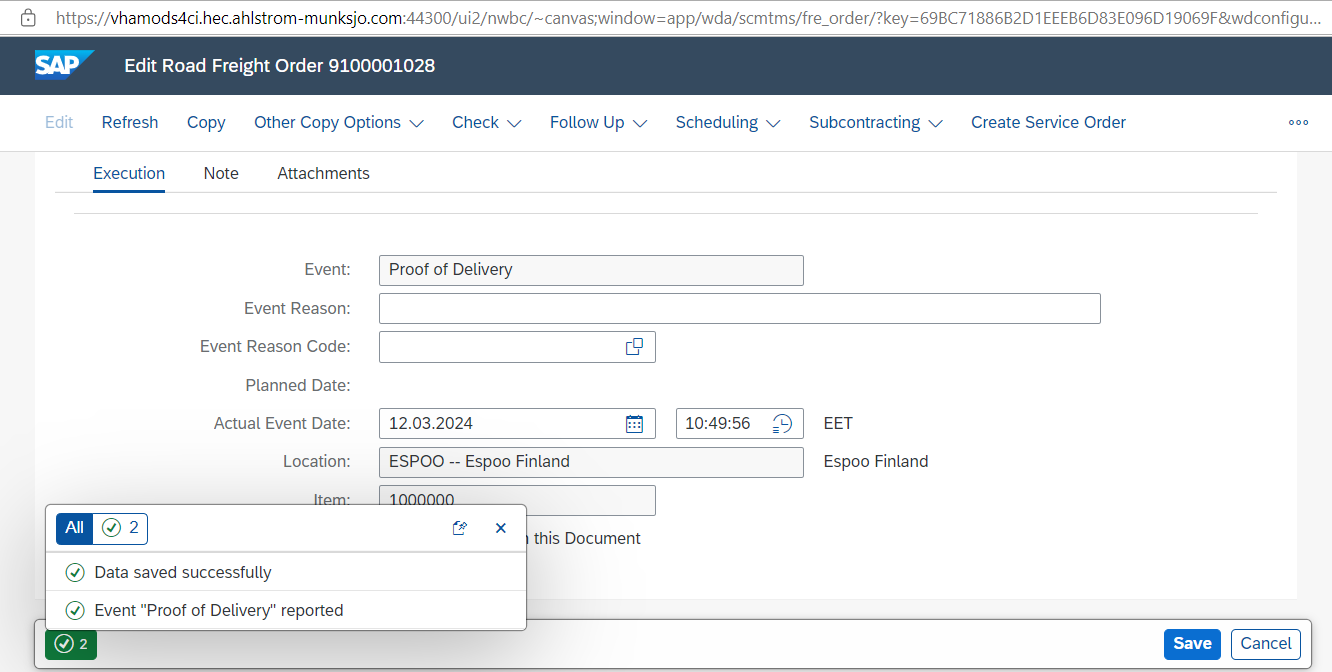






After POD event reported in the Freight order 9100001028





POD status updated as Confirmed in both outbound deliveries

