HU (Handling Unit) technical note

Context : HU usage for outbound delivery

Louange Mukuna

Plan

Context

I – HU creation

II – HU filling

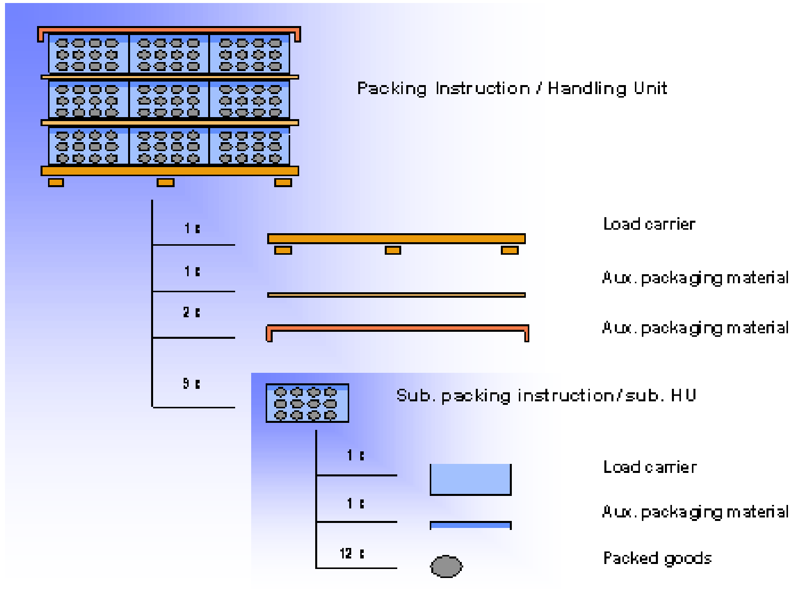
III – Assign HU to a delivery

IV – HU header modification

Context

The process consists to use handling unit therewith pack and unpack the material of our outbound delivery.

We use a pallet handling unit to store materials of the delivery and a container handling unit to store paletts.



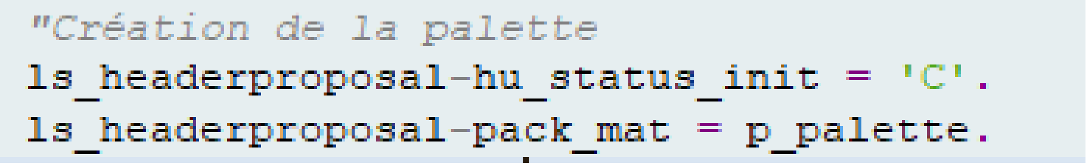
Source : SAP Help Portal

Consequently, there are a hierarchical concept in our packing, container can consolidate one or many pallets.

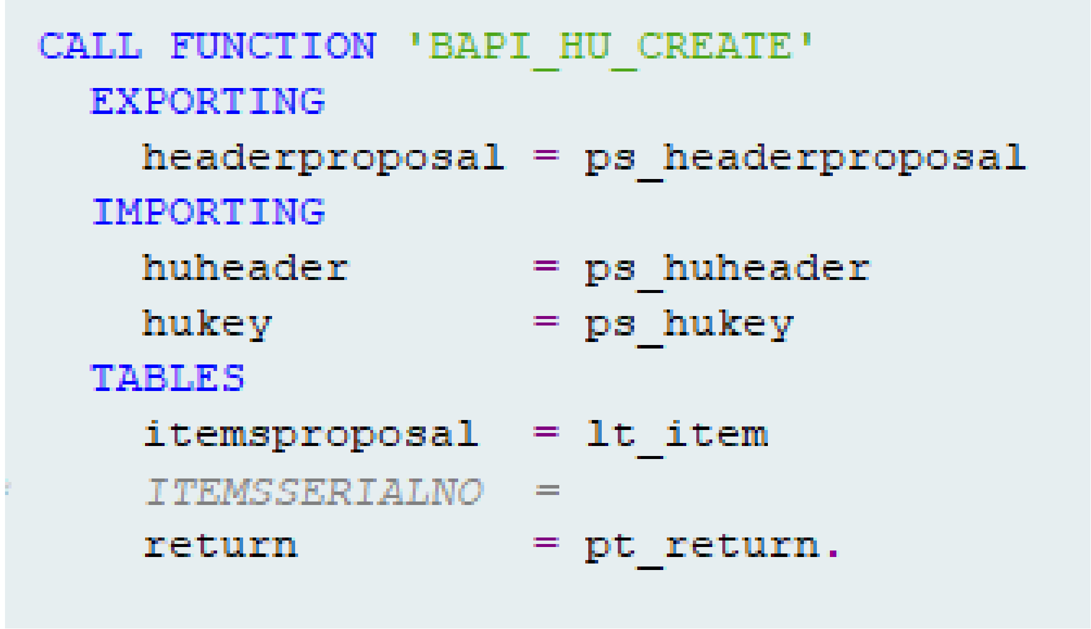
I – HU creation

- BAPI to use : **BAPI_HU_CREATE**

Exemple :



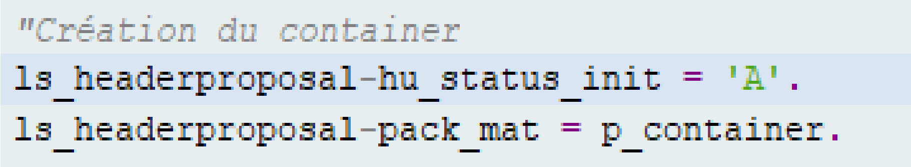
```
"Création de la palette
ls_headerproposal-hu_status_init = 'C'.
ls_headerproposal-pack_mat = p_palette.
```



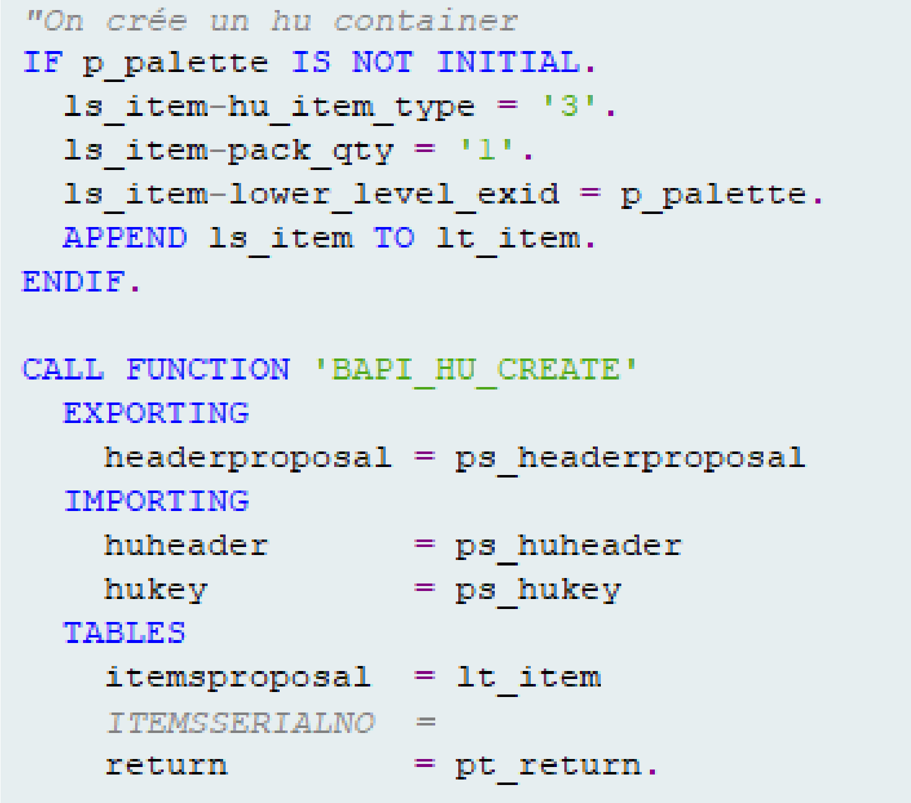
```
CALL FUNCTION 'BAPI_HU_CREATE'
  EXPORTING
    headerproposal = ps_headerproposal
  IMPORTING
    huheader       = ps_huheader
    hukey          = ps_hukey
  TABLES
    itemsproposal  = lt_item
    ITEMSSERIALNO  =
    return         = pt_return.
```

In this sample, we create a hu thanks to a material corresponding to a pallet.

I – HU creation



```
"Création du container
ls_headerproposal-hu_status_init = 'A'.
ls_headerproposal-pack_mat = p_container.
```



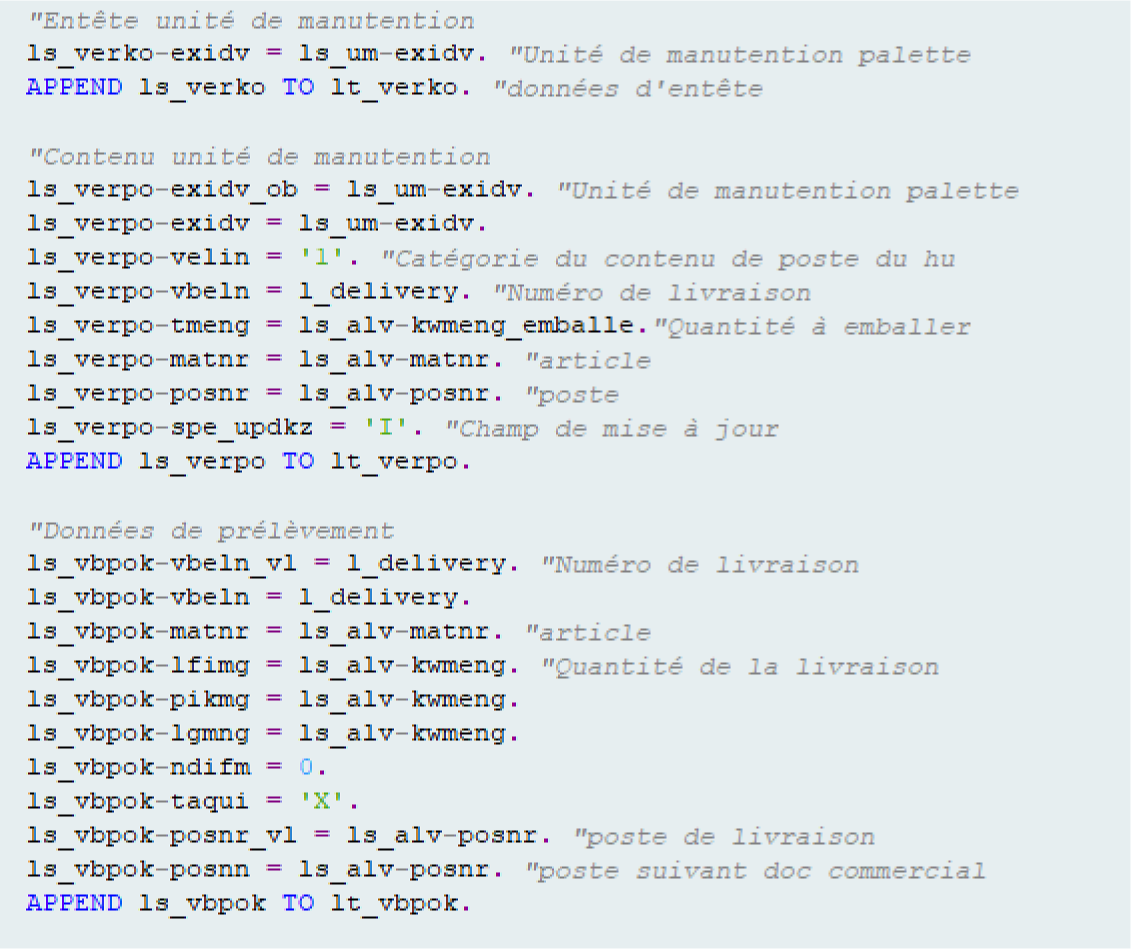
```
"On crée un hu container
IF p_palette IS NOT INITIAL.
  ls_item-hu_item_type = '3'.
  ls_item-pack_qty = '1'.
  ls_item-lower_level_exid = p_palette.
  APPEND ls_item TO lt_item.
ENDIF.

CALL FUNCTION 'BAPI_HU_CREATE'
  EXPORTING
    headerproposal = ps_headerproposal
  IMPORTING
    huheader       = ps_huheader
    hukey          = ps_hukey
  TABLES
    itemsproposal  = lt_item
    ITEMSSERIALNO  =
    return         = pt_return.
```

We create a hu that will contain another hu thanks to a packaging material corresponding to a container.

II – HU filling

- BAPI to use : **WS_DELIVERY_UPDATE**

Exemple :

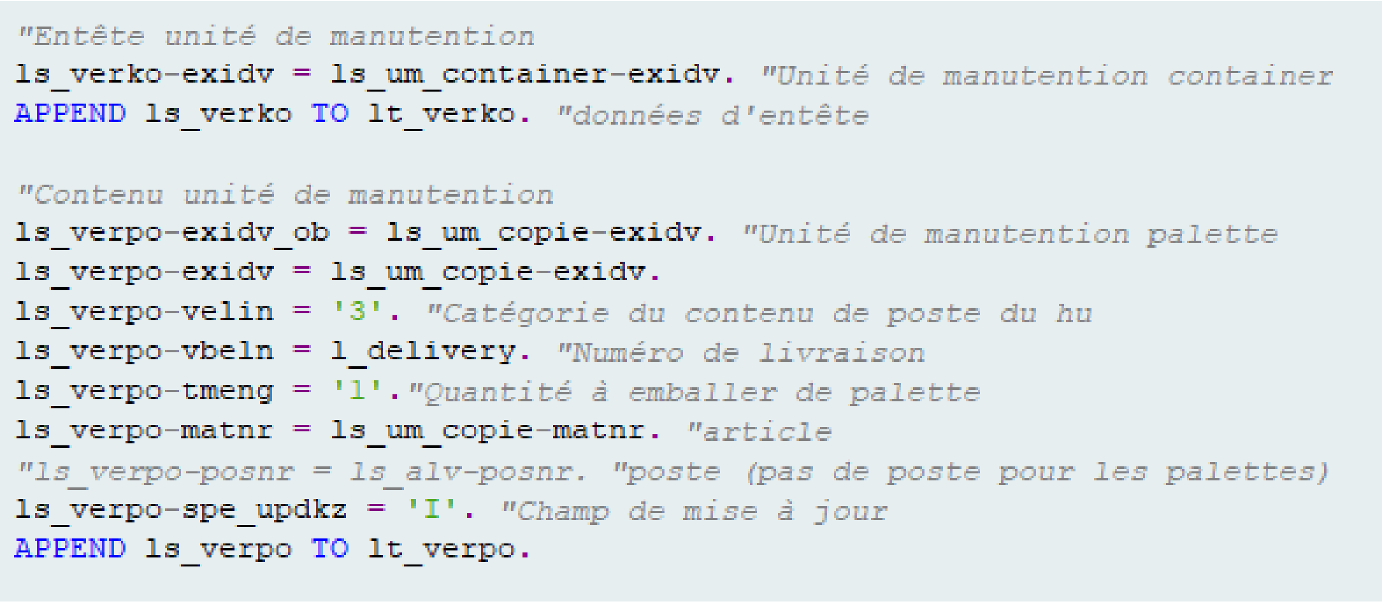


```
"Entête unité de manutention
ls_verko-exidv = ls_um-exidv. "Unité de manutention palette
APPEND ls_verko TO lt_verko. "données d'entête

"Contenu unité de manutention
ls_verpo-exidv_ob = ls_um-exidv. "Unité de manutention palette
ls_verpo-exidv = ls_um-exidv.
ls_verpo-velin = '1'. "Catégorie du contenu de poste du hu
ls_verpo-vbeln = l_delivery. "Numéro de livraison
ls_verpo-tmeng = ls_alv-kwmeng_emballe."Quantité à emballer
ls_verpo-matnr = ls_alv-matnr. "article
ls_verpo-posnr = ls_alv-posnr. "poste
ls_verpo-spe_updkz = 'I'. "Champ de mise à jour
APPEND ls_verpo TO lt_verpo.

"Données de prélèvement
ls_vbpok-vbeln_vl = l_delivery. "Numéro de livraison
ls_vbpok-vbeln = l_delivery.
ls_vbpok-matnr = ls_alv-matnr. "article
ls_vbpok-lfimg = ls_alv-kwmeng. "Quantité de la livraison
ls_vbpok-pikmg = ls_alv-kwmeng.
ls_vbpok-lgmng = ls_alv-kwmeng.
ls_vbpok-ndifm = 0.
ls_vbpok-taqui = 'X'.
ls_vbpok-posnr_vl = ls_alv-posnr. "poste de livraison
ls_vbpok-posnn = ls_alv-posnr. "poste suivant doc commercial
APPEND ls_vbpok TO lt_vbpok.
```

In this sample, we put data on a handling unit corresponding to the pallet.
We import material data to pack.

To unpack a handling unit, we need to indicate a negative quantity.

II – HU filling

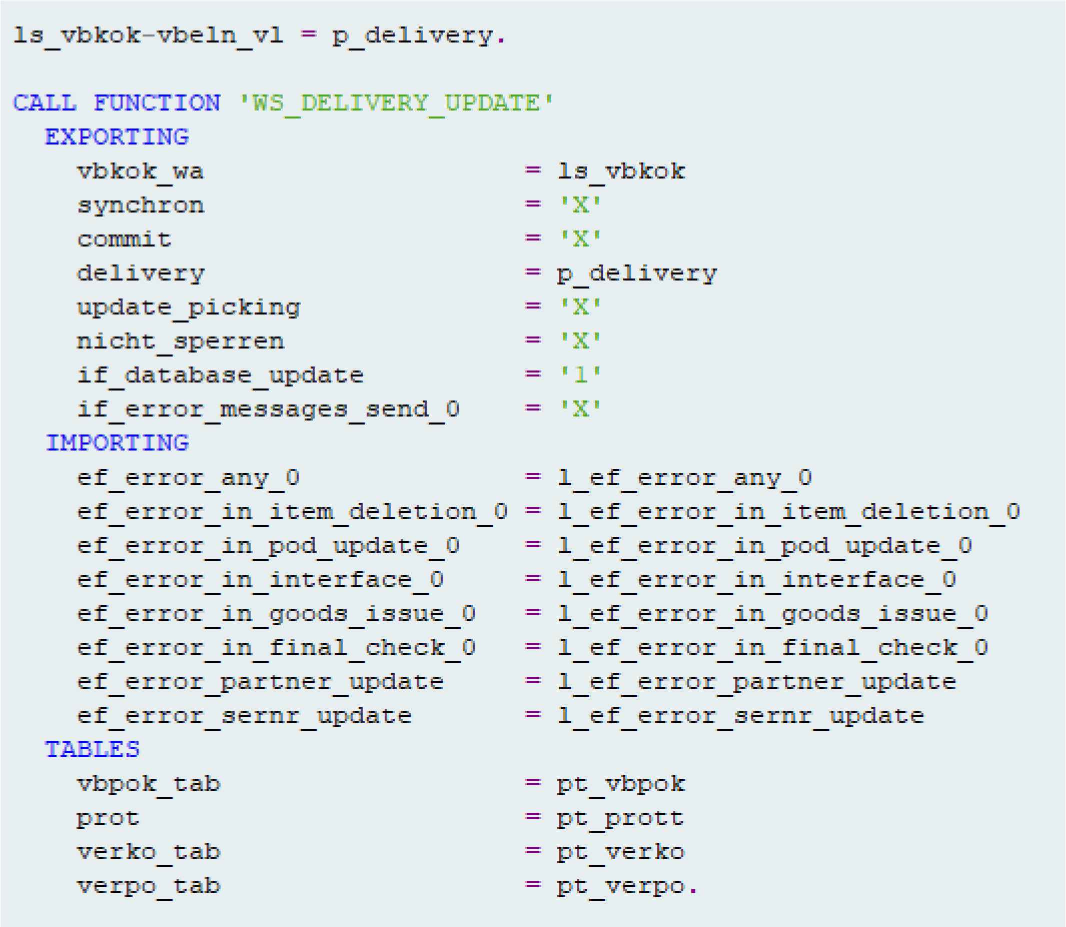


```
"Entête unité de manutention
ls_verko-exidv = ls_um_container-exidv. "Unité de manutention container
APPEND ls_verko TO lt_verko. "données d'entête

"Contenu unité de manutention
ls_verpo-exidv_ob = ls_um_copie-exidv. "Unité de manutention palette
ls_verpo-exidv = ls_um_copie-exidv.
ls_verpo-velin = '3'. "Catégorie du contenu de poste du hu
ls_verpo-vbeln = l_delivery. "Numéro de livraison
ls_verpo-tmeng = '1'."Quantité à emballer de palette
ls_verpo-matnr = ls_um_copie-matnr. "article
"ls_verpo-posnr = ls_alv-posnr. "poste (pas de poste pour les palettes)
ls_verpo-spe_updkz = 'I'. "Champ de mise à jour
APPEND ls_verpo TO lt_verpo.
```

We fill the handling unit that will be our container (superior packaging).

We provide data regarding the pallet to pack (inferior packaging).

II – HU filling



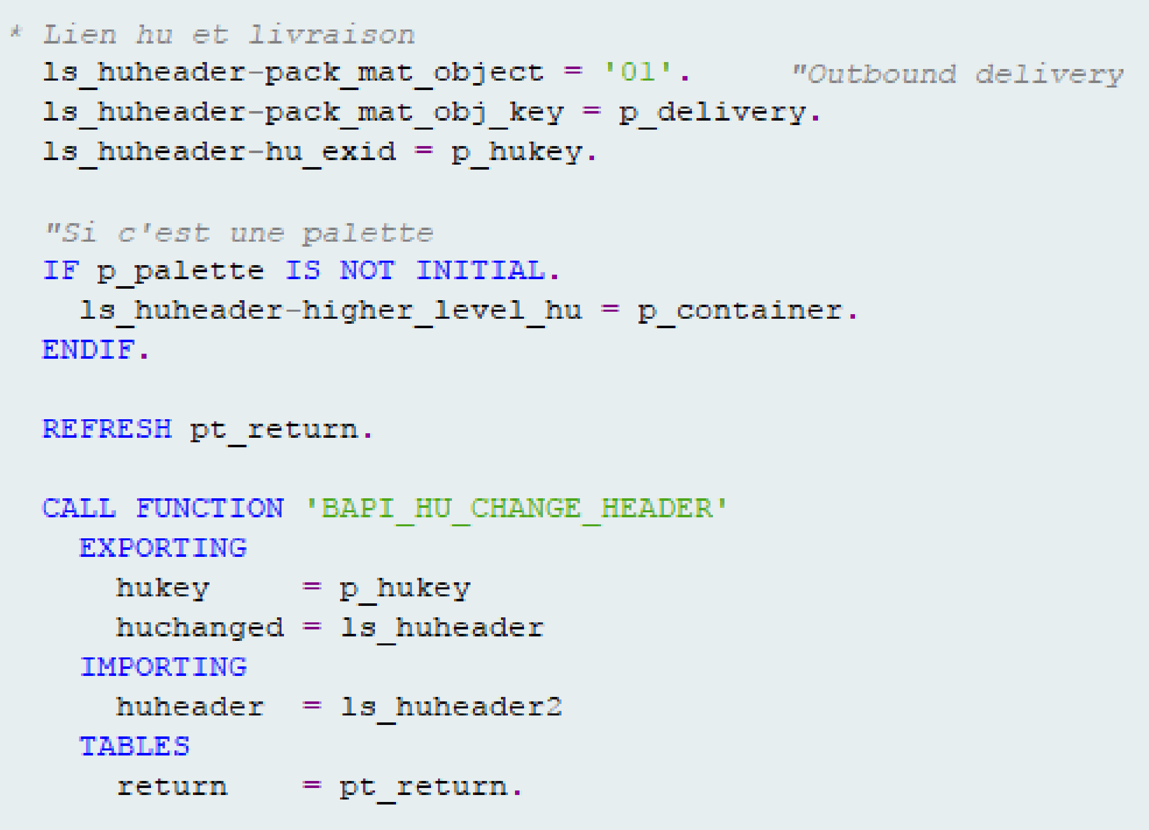
```
ls_vbkok-vbeln_vl = p_delivery.

CALL FUNCTION 'WS_DELIVERY_UPDATE'
  EXPORTING
    vbkok_wa                    = ls_vbkok
    synchron                    = 'X'
    commit                      = 'X'
    delivery                    = p_delivery
    update_picking              = 'X'
    nicht_sperren               = 'X'
    if_database_update          = '1'
    if_error_messages_send_0    = 'X'
  IMPORTING
    ef_error_any_0              = l_ef_error_any_0
    ef_error_in_item_deletion_0 = l_ef_error_in_item_deletion_0
    ef_error_in_pod_update_0    = l_ef_error_in_pod_update_0
    ef_error_in_interface_0     = l_ef_error_in_interface_0
    ef_error_in_goods_issue_0   = l_ef_error_in_goods_issue_0
    ef_error_in_final_check_0   = l_ef_error_in_final_check_0
    ef_error_partner_update     = l_ef_error_partner_update
    ef_error_sernr_update       = l_ef_error_sernr_update
  TABLES
    vbpok_tab                   = pt_vbpok
    prot                        = pt_prott
    verko_tab                   = pt_verko
    verpo_tab                   = pt_verpo.
```

We call the BAPI with the previous data.

III – Assign a HU to a delivery

- BAPI to use : **BAPI_HU_CHANGE_HEADER**

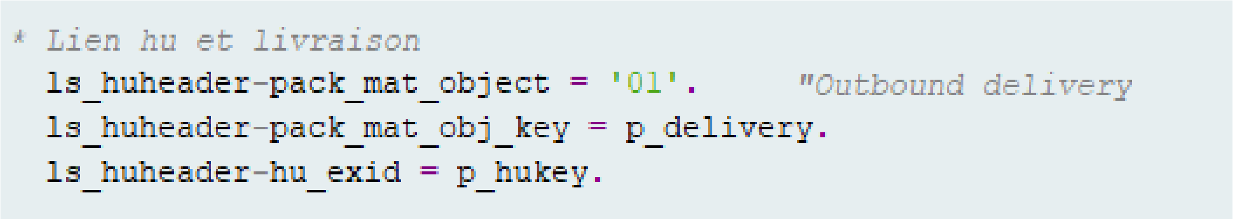
Exemple :



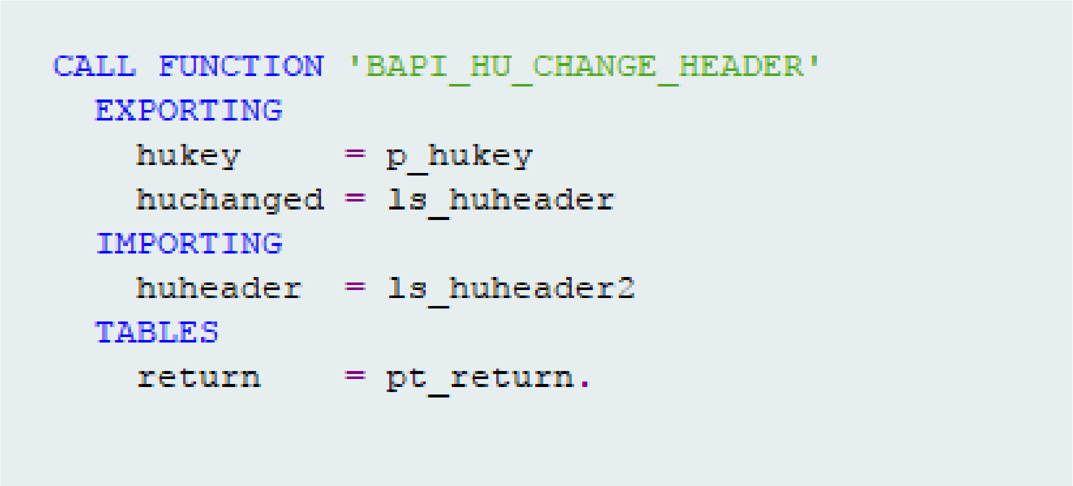
```
* Lien hu et livraison
  ls_huheader-pack_mat_object = '01'.     "Outbound delivery
  ls_huheader-pack_mat_obj_key = p_delivery.
  ls_huheader-hu_exid = p_hukey.

  "Si c'est une palette
  IF p_palette IS NOT INITIAL.
    ls_huheader-higher_level_hu = p_container.
  ENDIF.

  REFRESH pt_return.

  CALL FUNCTION 'BAPI_HU_CHANGE_HEADER'
    EXPORTING
      hukey     = p_hukey
      huchanged = ls_huheader
    IMPORTING
      huheader   = ls_huheader2
    TABLES
      return     = pt_return.
```

We associate our pallet hu to our outbound delivery.

III – Assign a HU to a delivery



```
* Lien hu et livraison
  ls_huheader-pack_mat_object = '01'.      "Outbound delivery
  ls_huheader-pack_mat_obj_key = p_delivery.
  ls_huheader-hu_exid = p_hukey.
```



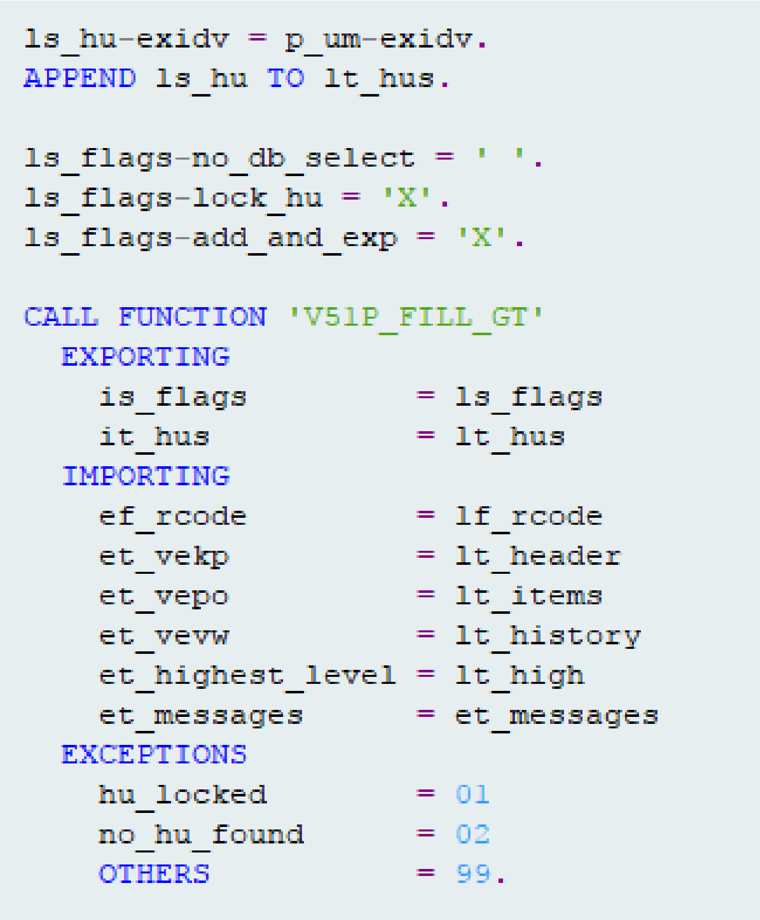
```
  CALL FUNCTION 'BAPI_HU_CHANGE_HEADER'
    EXPORTING
      hukey       = p_hukey
      huchanged = ls_huheader
    IMPORTING
      huheader   = ls_huheader2
    TABLES
      return     = pt_return.
```

We associate our container hu to our outbound delivery.

IV – HU header modification

- BAPI to use : **V51P_FILL_GT / HU_HEADER_UPDATE / HU_PACKING_UPDATE**

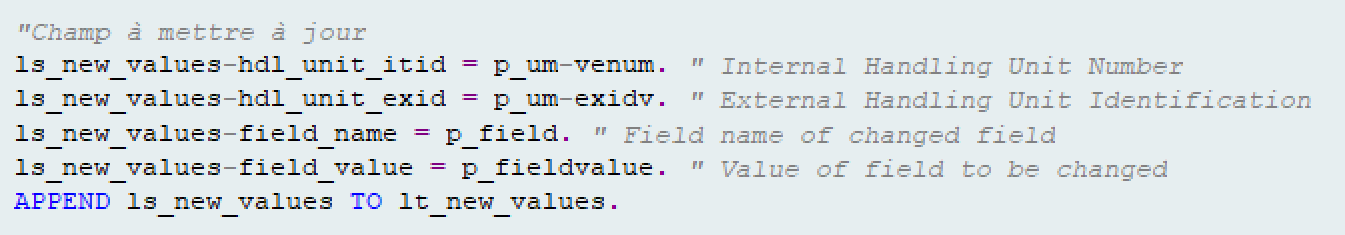
Exemple :



```
ls_hu-exidv = p_um-exidv.
APPEND ls_hu TO lt_hus.

ls_flags-no_db_select = ' '.
ls_flags-lock_hu = 'X'.
ls_flags-add_and_exp = 'X'.

CALL FUNCTION 'V51P_FILL_GT'
  EXPORTING
    is_flags          = ls_flags
    it_hus            = lt_hus
  IMPORTING
    ef_rcode          = lf_rcode
    et_vekp           = lt_header
    et_vepo           = lt_items
    et_vevw           = lt_history
    et_highest_level  = lt_high
    et_messages       = et_messages
  EXCEPTIONS
    hu_locked         = 01
    no_hu_found       = 02
    OTHERS            = 99.
```

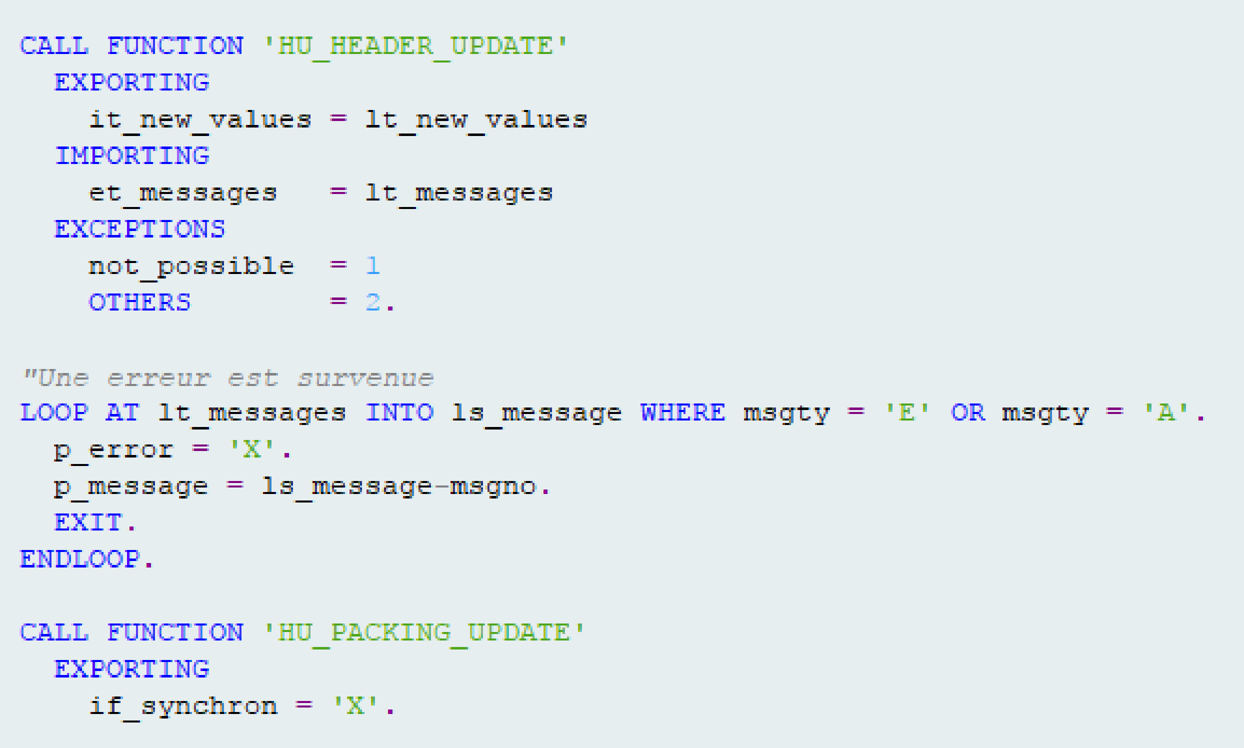
BAPI call will fill an internal table called gt_xvekp (hu header).

IV – HU header modification



```
"Champ à mettre à jour
ls_new_values-hdl_unit_itid = p_um-venum. " Internal Handling Unit Number
ls_new_values-hdl_unit_exid = p_um-exidv. " External Handling Unit Identification
ls_new_values-field_name = p_field. " Field name of changed field
ls_new_values-field_value = p_fieldvalue. " Value of field to be changed
APPEND ls_new_values TO lt_new_values.
```

Modify data



```
CALL FUNCTION 'HU_HEADER_UPDATE'
  EXPORTING
    it_new_values = lt_new_values
  IMPORTING
    et_messages   = lt_messages
  EXCEPTIONS
    not_possible  = 1
    OTHERS        = 2.

"Une erreur est survenue
LOOP AT lt_messages INTO ls_message WHERE msgty = 'E' OR msgty = 'A'.
  p_error = 'X'.
  p_message = ls_message-msgno.
  EXIT.
ENDLOOP.

CALL FUNCTION 'HU_PACKING_UPDATE'
  EXPORTING
    if_synchron = 'X'.
```

Validation and filling of internal table VEKP

Update of database table