

Documentation technique

HU (Handling Unit)

Contexte d'utilisation : Utilisation des HU lors d'une livraison sortante

Plan

Contexte d'utilisation

I – Création d'un HU

II – Alimentation d'un HU

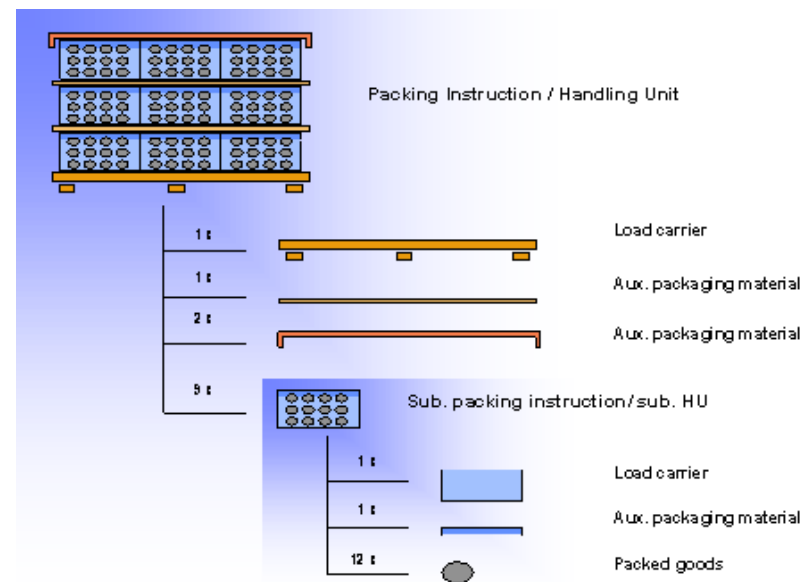
III – Assignation d'un HU à une livraison

IV – Modification de l'entête d'un HU

Contexte d'utilisation

Le flux consiste à utiliser les hu afin d'emballer/déballer les articles de la livraison sortante.

On utilise un hu palette afin de stocker les articles de la livraison et le hu conteneur permet de conserver les palettes.



Source : SAP Help Portal

Il y a donc une notion hiérarchique dans notre emballage, le conteneur comprend une ou plusieurs palettes.

I – Création d'un HU

- BAPI à utiliser : **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.
```

Dans cet exemple, on crée un hu à l'aide d'un article correspondant à une palette.

I – Création d'un HU

```
"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.
```

On crée un hu qui va contenir un autre hu à l'aide d'un article correspondant à un container.

II – Alimentation d'un HU

- BAPI à utiliser : **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.
```

Dans cet exemple, on alimente un hu correspondant à la palette. On lui fournit les données de l'article à emballer.

Pour déballer un hu, il suffit simplement de renseigner une quantité négative.

II – Alimentation d'un HU

```
"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.
```

On alimente un hu correspondant au conteneur (emballage supérieur).

On lui fournit les données correspondant au hu palette à emballer (emballage inférieur).

II – Alimentation d'un HU

```
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.
```

On appelle la BAPI avec les données fournies précédemment.

III – Assignment d'un HU à une livraison

- BAPI à utiliser : **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.
```

On associe notre hu palette à notre livraison sortante.

III – Assignment d'un HU à une livraison

```
* 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.
```

On associe notre hu conteneur à notre livraison sortante.

IV – Modification de l'entête d'un HU

- BAPI à utiliser : **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.
```

L'appel de cette bapi permet de remplir une table interne gt_xvekp (entête des hu).

IV – Modification de l'entête d'un HU

```
"Champ à mettre à jour
ls_new_values-hdl_unit_itid = p_um-venum. " Internal Handling Unit Number
ls_new_values-hdl_unit_exidv = 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.
```

Valeurs modifiées

```
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 et remplissage de la table interne vekp

Mise à jour de la table de base de données