

Reporting 6 – Liste des devises

Introduction :

Un tableau A.L.V. (Abap List Viewer) est une méthode pratique pour afficher les données tout en laissant la possibilité à l'utilisateur de trier les données suivant un critère choisi, de sommer, de réarranger les colonnes, et de sauvegarder ces propriétés dans un format d'affichage (*Layout*)...

Le programme ZTZ_XXX_CURRENCY affichera un tableau A.L.V. contenant

- les codes devises, choisis à l'écran de sélection,
- leurs libellés,
- leur code numérique alternatif (utilisé par les banques)
- le nombre de décimales des montants exprimés dans cette devise
- le nom du pays où cette devise est en cours (si ce pays existe et est unique)

Une coche à l'écran de sélection permet de ne pas lister les devises déjà obsolètes à une date donnée à l'écran de sélection, par défaut avant le début de l'année.

Tables nécessaires :

TCURC Devises

(+ *Text table* correspondante)

TCURX Nombre de décimales si différent de 2 (deux décimales par défaut)

T005 Pays

T005T Nom des pays

Pour savoir si une devise est obsolète, utilisez le function module

'CURRENCY_EXPIRATION_CHECK' avec OBJECT = 'BKPF'

(pour les tests, les devises utilisées avant l'EUR sont évidemment obsolètes)

A.L.V. function modules

'REUSE_ALV_GRID_DISPLAY' est le function module à utiliser pour afficher l'ALV grid.

CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'

EXPORTING

is_layout = wa_layout "wa_layout-colwidth_optimize = 'X'.

it_fieldcat = wt_fieldcat

TABLES

t_outtab =

Il contient beaucoup de paramètres optionnels, structures ou tables qu'on peut laisser vides.

Pour remplir les champs du field catalog IT_FIELDCAT,

il suffit de remplir cette table avec les noms des champs dans notre table interne (fieldname)

et les champs de référence correspondant du data dictionary (ref_tabname, ref_fieldname)

Notre table interne à afficher sera envoyée comme paramètre T_OUTTAB

Note : quelques autres function modules (testables en SE37, **non** utilisé dans le programme présent)

'REUSE_ALV_FIELDCATALOG_MERGE' permet de construire le field catalog d'après une structure DDIC

'DDIF_FIELDINFO_GET' permet de connaître les propriétés d'un champ du data dictionary (déjà vu)

Reporting 7 – Liste des commandes

Introduction

Nous allons appeler une transaction standard depuis un tableau ALV.

Un utilisateur SAP peut appeler une transaction (dans l'ok-code ou par le Menu) et ainsi entrer des données ou consulter un document, ou encore changer un paramétrage...

Ces actions sont aussi possibles depuis un programme abap.

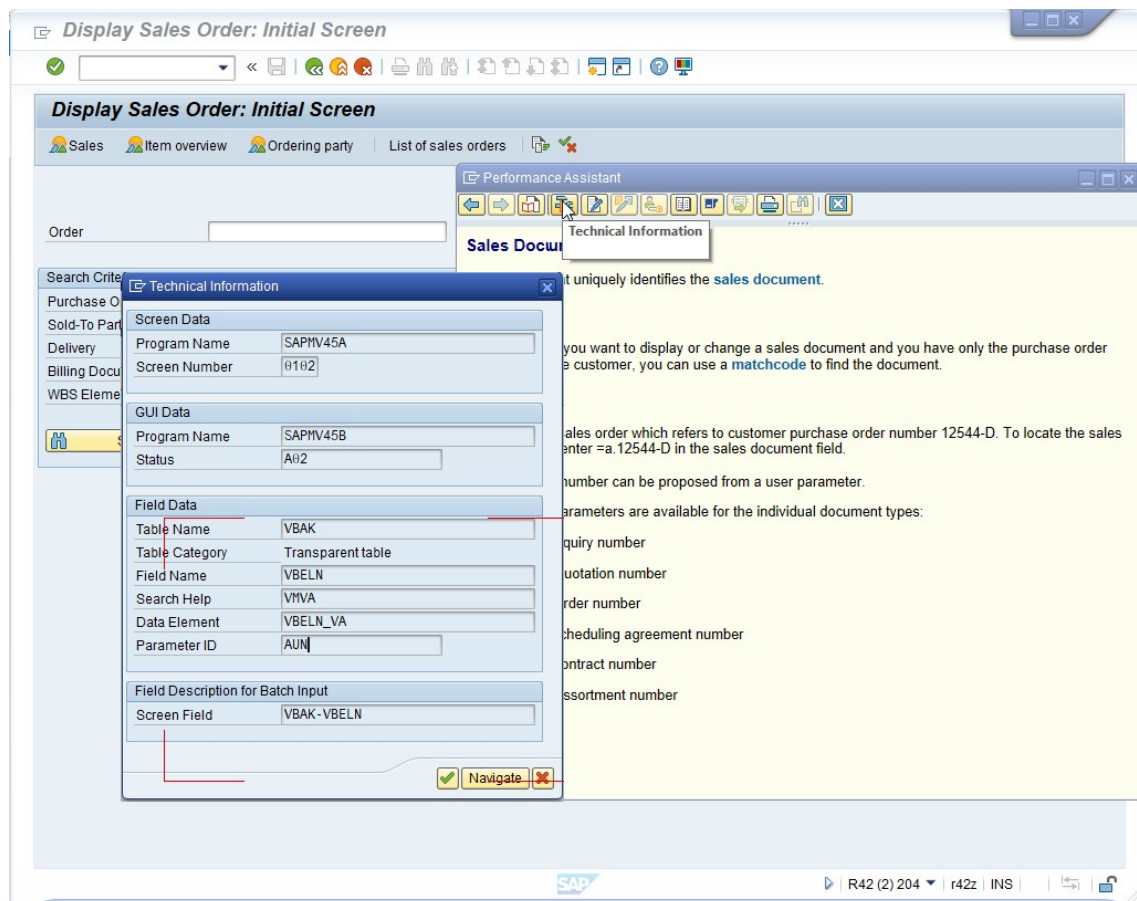
- Entrer des données : Nous n'avons pas vu comment faire (call transaction using bdcdata)
- Consulter un document : Généralement, il suffit de remplir quelques champs du premier écran de la transaction (ensuite il n'y a plus rien à remplir). C'est ce cas qui nous intéresse.

Ci-dessous un exemple avec la transaction VA03 « Display Sales Order ».

Il y a un seul champ à remplir : Le numéro de commande « Order ».

Pour préremplir le champ depuis un abap avant de lancer la transaction, il faut connaître son 'Parameter ID'.

Comment faire ? Lancer la transaction (ici VA03), appuyer sur la touche <F1> « Aide » et comme d'habitude, immédiatement cliquer sur <Information technique>. On y voit le nom du champ (mais ce n'est pas ce qui nous intéresse, cette fois) et aussi le 'Parameter ID'.



```
REPORT ZZT_000_ALV_CALL_TRAN.
parameters p_vbeln type vbak-vbeln.
```

```
START-OF-SELECTION.
set parameter id 'AUN' field p_vbeln.
call transaction 'VA03' and skip first screen.
```

Ensuite, le double-clic est un des évènements de l'ALV provenant d'une commande de l'utilisateur.

On a le choix de faire appel à une FORM donnée du programme lors du double-clic.

```
CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
  EXPORTING
    i_callback_program      = sy-repid
*    I_CALLBACK_PF_STATUS_SET =
    i_callback_user_command = 'USER_COMMAND' "Name of the form inside our program that will be called
*    i_structure_name       = ''
*    ... (other parameters as usual)
*
*-----*
FORM user_command USING f_ucomm      TYPE sy-ucomm
                    fs_selfield TYPE slis_selfield.
* Parameters of this FORM: See Documentation of REUSE_ALV_GRID_DISPLAY
CLEAR wa_order.
CASE f_ucomm.
  WHEN '&IC1'. "&IC1 while double click, deducted by debugging
    READ TABLE wt_order INTO wa_order INDEX fs_selfield-tabindex.
    IF sy-subrc = 0.
*
      ENDIF.
    ENDCASE.
ENDFORM.                    "user_command
```

Enoncé

Ecrivez un programme ZZT_XXX_ORDERS qui affiche un tableau A.L.V. contenant la liste des commandes attribuées à une Organisation de vente (*Sales Organization*) donnée.

L'écran de sélection contient le choix d'une Organisation de vente (obligatoire), de numéros de commande (*Sales document*), et de clients donneurs d'ordre (*sold-to party*).

Le tableau contient, à part les éléments cités, le nom 1 du client, la date de création de la commande, le type de commande (*Sales Document Type*) et sa description.

Un double-clic sur une ligne permet l'affichage de la commande (transaction VA03).

Tables :

VBAK Commande de vente
KNA1 Client
TVAKT ...