

[ABAP] Écran de sélection (partie 2/2)

Bonjour à toi cher lecteur.

Voici donc mon deuxième tutoriel pour sap-integration.net et il a pour but d'être la continuité de la partie 1 écrit par Jérôme Fortias (alias jfo).

Les thèmes abordés dans ce tutoriel seront beaucoup plus poussés que dans la partie 1 car nous verront comment rendre un écran de sélection "vivant", c'est à dire de pouvoir conditionner l'affichage des composants selon les actions faites à l'écran.

Donc si ça t'intéresse et que t'es prêt, alors passons à la suite.

Avant toute chose, on va commencer par se faire un écran de sélection donc on crée un programme, qui sera ZMODIF_SCREEN pour moi, et on va y placer des composants.

```
*&-----*
*& Report  ZMODIF_SCREEN
*&
*&-----*
*& Auteur : Celdrøn
*&
*&-----*
```

```
REPORT  zmodif_screen.
```

```
* Données pour écran de sélection.
```

```
DATA : vbeln TYPE vbak-vbeln,
       auart TYPE vbak-auart,
       pstyv TYPE vbap-pstyv,
```

```
* BLOCK 1
```

```
SELECTION-SCREEN BEGIN OF BLOCK b11 WITH FRAME TITLE text-b11.
```

```
SELECT-OPTIONS : so_vbeln FOR vbeln.
```

```
SELECT-OPTIONS : so_auart FOR auart.
```

```
SELECTION-SCREEN END OF BLOCK b11.
```

```
* BLOCK 2
```

```
SELECTION-SCREEN BEGIN OF BLOCK b12 WITH FRAME TITLE text-b12.
```

```
PARAMETERS : cb1 TYPE c AS CHECKBOX USER-COMMAND cb1.
```

```
SELECT-OPTIONS : so_pstyv FOR pstyv MODIF ID b12.
```

```
PARAMETERS : p_file TYPE fileextern MODIF ID b12.
```

```
SELECTION-SCREEN END OF BLOCK b12.
```

Ce qui nous donne à l'affichage.

Block 1

Document de vente à

Type document vente à

Block 2

Checkbox

Type de poste à

fich. physique

Voilà, et maintenant je suis grillé car certains vont se rendre compte au premier coup d'œil qu'en ce moment je bosse sur SD. ^^

Alors si tu te demandes comment j'ai eu les descriptions de champs sans les mettre dans le code, il suffit d'aller dans Saut=>"Éléments de texte"=>"Texte de sélection" et de cocher la case de sélection qui signifie que la description de zone fait référence à celle du DDIC (Dictionnaire de donnée).

Et pour la case à cocher (Checkbox) sur l'écran, et bien elle va tout simplement nous servir à conditionner les zones du Bloque 2 (BL2). A noter la présence du **USER-COMMAND fcode** qui est nécessaire pour indiquer qu'une action faite sur ce composant déclenche un traitement, sans ça les actions faites sur le composants ne seront pas détectés.

Sinon pour les deux éléments où j'ai rajouté MODIF ID, c'est juste pour montrer, à titre d'exemple, que l'on peut regrouper des zones. C'est utile pour ne mettre qu'une condition plutôt que de conditionner selon chaque zone. L'ID se trouvera dans la zone SCREEN-GROUP1.

Exemple :

```
IF screen-group1 = 'BL2'
*   Traitement
ENDIF.
```

Plutôt que :

```
IF screen-name = 'SO_PSTYV' OR
   screen-name = 'P_FILE'.
*   Traitement
ENDIF.
```

A noter pour finir que j'ai un principe de base qui est d'utiliser juste la mémoire dont j'ai besoin c'est pour cela que je n'utilise pas l'instruction TABLES qui déclare une structure qui ne sera sûrement pas utilisé mais que je déclare des DATA typé qui me servent pour les SELECT-OPTIONS.

1 Rendre des zones non-saisissables

Disons que les zones seront saisissables si la case à cocher est coché et non saisissables si elle ne l'est pas.

Il faut donc conditionner les zones avant l'affichage de l'écran et cela se passe dans l'évènement : **AT SELECTION-SCREEN OUTPUT.**

* PBO de l'écran de sélection

```
AT SELECTION-SCREEN OUTPUT.
```

```
LOOP AT SCREEN.
```

```
IF cb1 IS INITIAL.
```

```
IF screen-group1 = 'BL2'.
```

```
screen-input = 0.
```

```
ENDIF. "screen-name = 'P_FILE'.
```

```
ELSE.
```

```
IF screen-group1 = 'BL2'.
```

```
screen-input = 1.
```

```
ENDIF. "screen-group1 = 'BL2'.
```

```
ENDIF. "cb1 IS INITIAL.
```

```
MODIFY SCREEN.
```

```
ENDLOOP. "AT SCREEN.
```

The screenshot shows a SAP selection screen with two blocks. Block 1 contains two rows of input fields: 'Document de vente' and 'Type document vente', each followed by a 'à' label and another input field. Block 2 contains a 'Checkbox' field, 'Type de poste', and 'fich. physique'.

Et normalement :

The screenshot shows the same SAP selection screen as above, but with the 'Checkbox' field checked.

et on doit revenir à l'écran initial quand on décoche la case à cocher.

2 Rendre des zones invisibles

Pour ça, ce sera pareil que la partie précédente, c'est à dire que tout se passe dans le PBO de l'écran de sélection.

* PBO de l'écran de sélection

```
AT SELECTION-SCREEN OUTPUT.  
  
LOOP AT SCREEN.  
  
  IF cb1 IS INITIAL.  
  
    IF screen-group1 = 'BL2'.  
      screen-input      = 0.  
      screen-invisible = 1.  
    ENDIF. "screen-name = 'P_FILE'.  
  
  ELSE.  
  
    IF screen-group1 = 'BL2'.  
      screen-input      = 1.  
      screen-invisible = 0.  
    ENDIF. "screen-group1 = 'BL2'.  
  
  ENDIF. "cb1 IS INITIAL.  
  
  MODIFY SCREEN.  
  
ENDLOOP. "AT SCREEN.
```

The screenshot shows two blocks of a SAP selection screen. Block 1 contains two rows of fields: 'Document de vente' and 'Type document vente'. Each row has a text input field, a 'à' label, another text input field, and a yellow arrow button. Block 2 contains a single row with a checkbox and the label 'Checkbox'.

Et voilà le travail :

The screenshot shows the same two blocks as above, but with changes. In Block 1, the text input fields are now empty. In Block 2, the checkbox is checked, and there are two new rows: 'Type de poste' and 'fich. physique'. The 'Type de poste' row has a text input field, a 'à' label, another text input field, and a yellow arrow button. The 'fich. physique' row has a long text input field.

Attention !!! il faut quand même gérer l'affichage des zones (INPUT) sous peine d'avoir des surprises à l'affichage.

Sinon, il est possible d'utiliser la zone SCREEN-ACTIVE.

3 Rendre des zones obligatoires ↔ non-obligatoires

Voilà une partie qui pourrait être intéressante pour toi si tu es dans le cas où tu as été confronté au cas et que tu as abandonné car ça ne marchait pas comme tu le voulais à cause du standard qui te bloquait.

Nota : je me souviens que j'avais confronté ce cas, à mes début en ABAP (2 ans et quelques à ce jour), à un consultant expérimenté mais il n'avait pu me donner une réponse à l'époque et je suis assez content de moi car j'ai trouvé une solution de contournement... Sans rancune Yann. ^_^

Alors illustrons tout d'abord la situation dans laquelle on tombe en rendant les zones obligatoires.

* PBO de l'écran de sélection

```
AT SELECTION-SCREEN OUTPUT.

LOOP AT SCREEN.

    IF cb1 IS INITIAL.

        IF screen-group1 = 'BL2'.
            screen-input = 0.
        ENDIF. "screen-name = 'P_FILE'.

        IF screen-name = 'P_FILE'.
            screen-required = 0.
        ENDIF. "screen-name = 'P_FILE'.

    ELSE.

        IF screen-group1 = 'BL2'.
            screen-input = 1.
        ENDIF. "screen-group1 = 'BL2'.

        IF screen-name = 'P_FILE'.
            screen-required = 1.
        ENDIF. "screen-name = 'P_FILE'.

    ENDIF. "cb1 IS INITIAL.

    MODIFY SCREEN.

ENDLOOP. "AT SCREEN.
```

Et voilà ce qu'on obtient

Block 1

Document de vente à

Type document vente à

Block 2

Checkbox

Type de poste à

fich. physique

Et après que la case à cocher soit coché :

Block 1

Document de vente à

Type document vente à

Block 2

Checkbox

Type de poste à

fich. physique

Bien, ça ressemble à ce que l'on voudrait obtenir mais seulement à peu près car si on essaie de décocher la case à cocher voilà ce qu'on obtient :

Block 1

Document de vente à

Type document vente à

Block 2

Checkbox

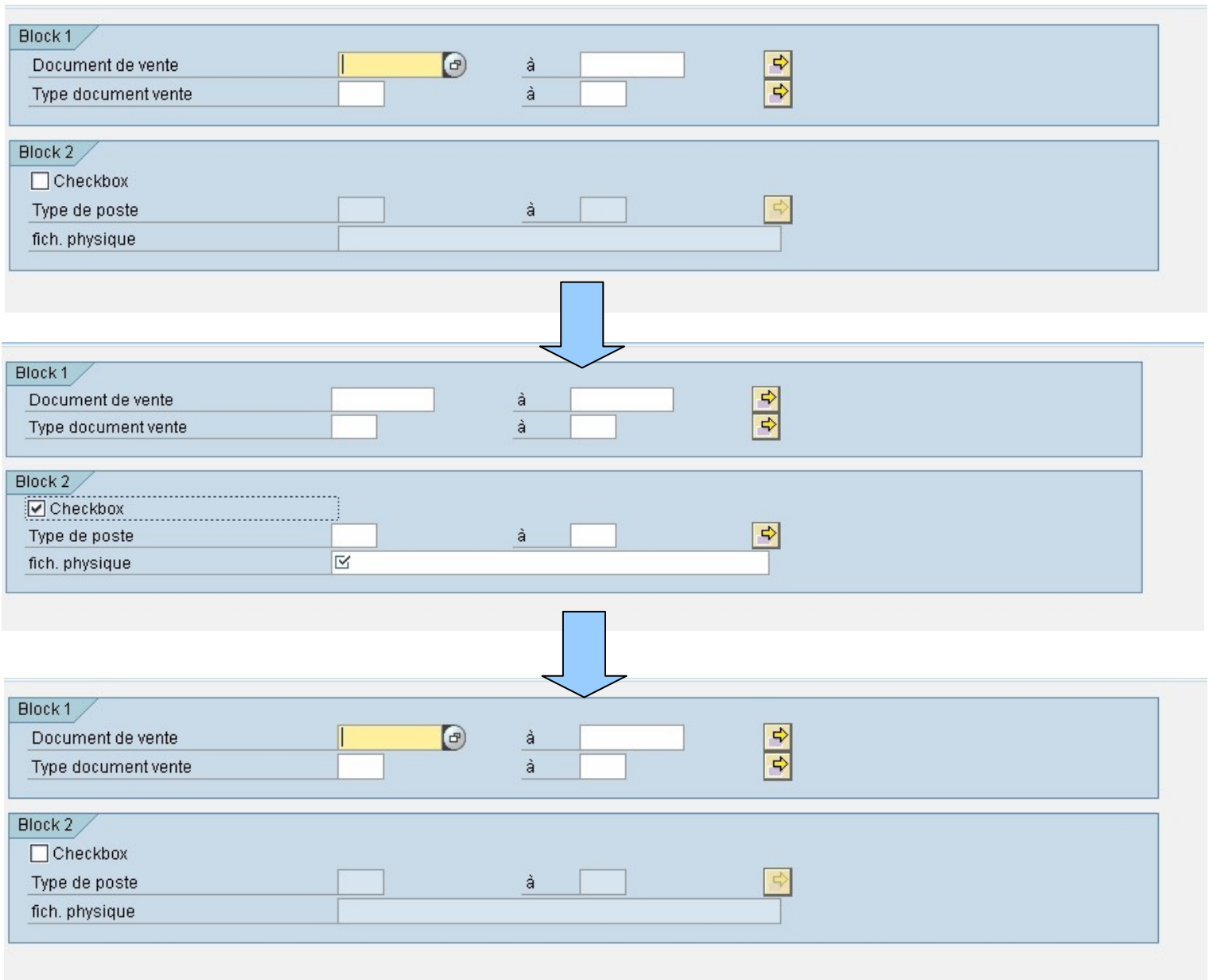
Type de poste à

fich. physique

Avec le message :

Renseignez toutes les zones obligatoires

Et voilà où ça pourrait s'arrêter car je pense que si tu as lus jusqu'ici, tu as pu être confronté à cette situation et que tu as abandonné pensant que c'était impossible alors que tu aurais sûrement aimé pouvoir faire ceci.



Et bien saches que la succession d'écran que tu vois au dessus a été faite pendant le même traitement ...

Et maintenant que je ris juste à l'idée de t'imaginer en train de saliver, je te donne la solution que j'ai trouvé et ça me fait d'autant plus plaisir que de la partager avec toi... par contre pense à dire que c'est moi qui te l'ai appris. ^_^

* PBO de l'écran de sélection

```
AT SELECTION-SCREEN OUTPUT.
```

```
LOOP AT SCREEN.
```

```
IF cb1 IS INITIAL.
```

```
IF screen-group1 = 'BL2'.
```

```
Screen-input = 0.
```

```
ENDIF. "screen-name = 'P_FILE'.
```

```
IF screen-name = 'P_FILE'.
```

```
screen-required = 0.
```

```

ENDIF. "screen-name = 'P_FILE'.

ELSE.

  IF screen-group1 = 'BL2'.
    screen-input = 1.
  ENDIF. "screen-group1 = 'BL2'.

  IF screen-name = 'P_FILE'.
    screen-required = 2.
  ENDIF. "screen-name = 'P_FILE'.

ENDIF. "cb1 IS INITIAL.

MODIFY SCREEN.

ENDLOOP. "AT SCREEN.

```

Tu dois te demander ce qui change par rapport à ce qu'il y avait précédemment. Alors je t'invite à bien regarder cette ligne de code :
`screen-required = 2.`


Avant on mettait REQUIRED à 1 ce qui avait pour conséquence de rendre la zone obligatoire, alors que si on met REQUIRED à 2, ça a pour conséquence de la rendre recommandée... c'est-à-dire qu'elle va apparaître comme obligatoire à l'écran mais le standard ne bloquera pas les traitements si elle n'est pas remplie. Donc en procédant ainsi, on obtient bien le résultat attendu...

Après que tu aies testé, je m'attends à ta remarque :
 C'est nul ton truc, la zone est bien obligatoire mais je peux quand même lancer le traitement alors que la zone n'est pas remplie !!!

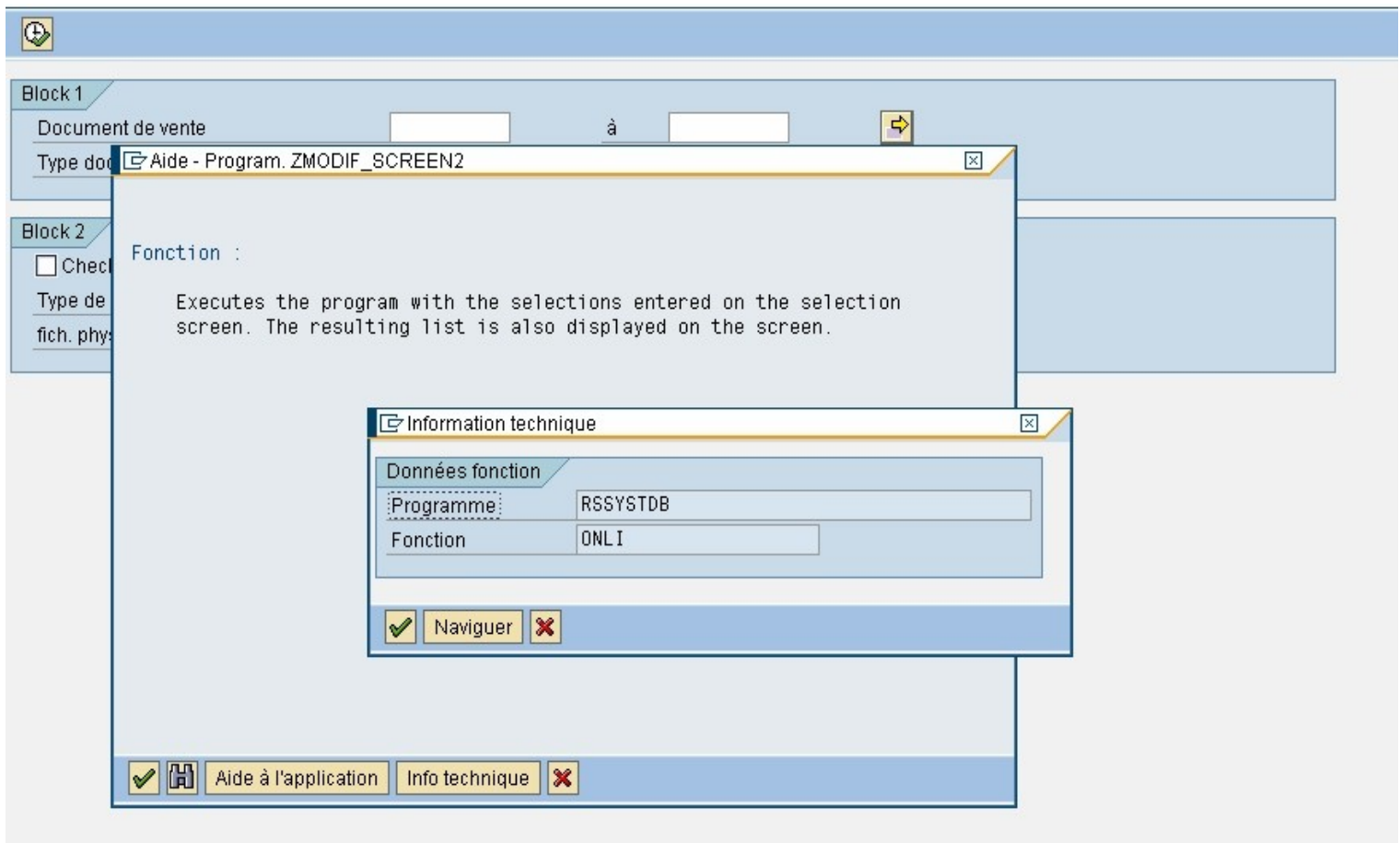
Et je vais te répondre tout simplement "Oui", mais je t'ai dit que le standard ne bloquait plus si la zone n'était pas remplie, cette partie du traitement est aussi à mettre en place. Et pour cela, il va nous falloir un nouvel événement pour les traitements du PAI et ça se passe dans le :

AT SELECTION-SCREEN.

Mais avant toute chose, il faut réfléchir à la manière de procéder. On ne va pas balancer un blocage à chaque action, le mieux serait de lancer le blocage que dans un, ou plusieurs, cas particuliers.

Pour l'instant, je n'ai besoin que d'un traitement de base, c'est à dire de bloquer que dans le cas où je lance le traitement c'est à dire sur l'action du bouton  ou sur l'appui de la touche F8.

La grande question reste de détecter que le traitement est lancé... bonne question et je vais te mettre sur la piste de la réponse.



Tu bloques encore ??

Ce que tu vois en face de Fonction c'est le code fonction du bouton "Exécuter".

Nota : Pour afficher l'aide sur un bouton, il suffit de cliquer sur le bouton en maintenant appuyé, de sortir du composant et de relâcher. Après faire F1 pour afficher l'aide et ensuite F9 pour afficher les Infos techniques.

Donc, on sait maintenant que le code fonction du bouton "Exécuter" est 'ONLI' (pour Online surement). Maintenant que l'on a le code fonction du bouton, il ne nous reste plus qu'à capturer cet événement et il n'y a rien de plus simple.

Rajouter à la suite du programme :

* PAI de l'écran de sélection.

AT SELECTION-SCREEN.

* On test si l'action a été de lancer le traitement
IF sy-ucomm = 'ONLI'.

* On test si bien sur la case à cocher est coché
car c'est l'une des conditions
IF NOT cb1 IS INITIAL.

* Et bien sur la dernière condition est que la zone
soit vide
IF p_file IS INITIAL.

MESSAGE ID '38' TYPE 'E' NUMBER '000'
WITH 'Veuillez renseigner un nom de fichier'(e01).

ENDIF. "p_file IS INITIAL.

ENDIF. "NOT cb1 IS INITIAL.

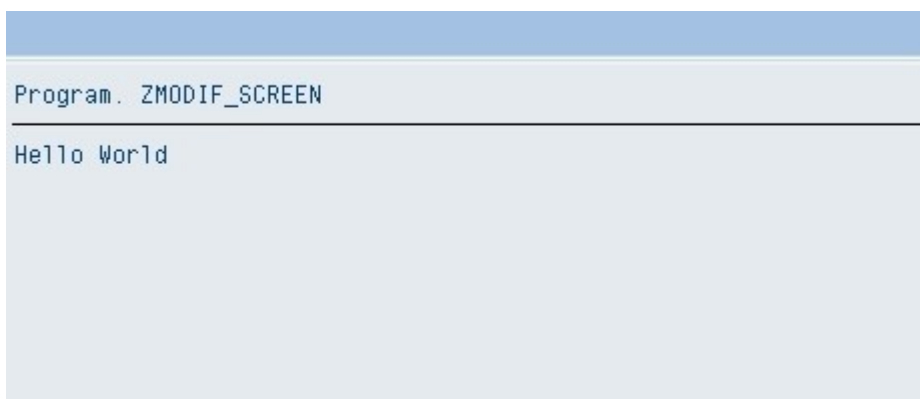
```
ENDIF. "sy-ucomm = 'ONLI'.
```

Et voilà, maintenant quand la case à cocher CB1 sera coché, que la zone P_FILE sera vide et que l'action sera de lancer le traitement, un message de type 'E' (Erreur) sera lancé. A noter le coté avantageux de pouvoir lancer le message souhaité ou de lancer le message standard, donc on peut ainsi customiser à sa guise selon différents cas précis.

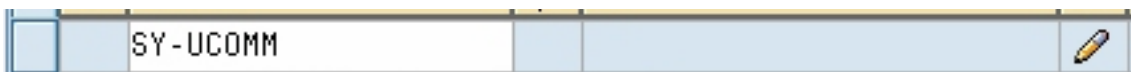


Mais ce n'est pas tout, en effet tu t'es peut-être rendu compte que si on lance l'exécution du traitement avec la zone obligatoire manquante, le traitement mis en place nous bloque... ça c'est OK seulement si tu as essayé de renseigner la zone et de l'effacer et faire Entrée juste après... le traitement se lance.

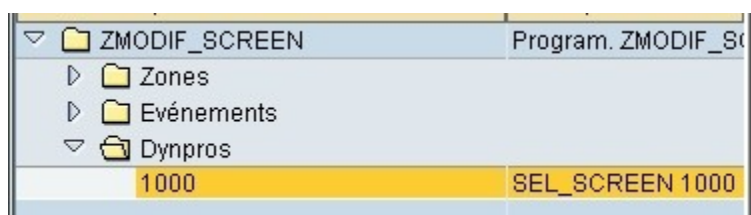
Pour preuve :




Et si on va voir en debug :



Alors on pourrait se demander pourquoi l'exécution se lance et c'est vrai que c'est vicieux mais si on va voir comment est généré l'écran de sélection par le standard :



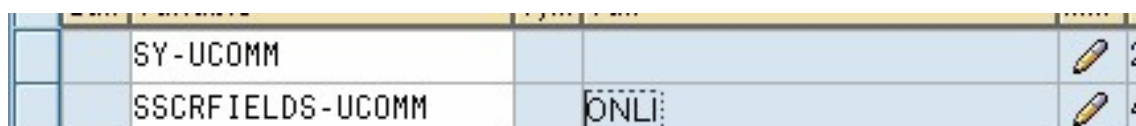
Nota : Pour afficher la liste d'objet, tu as le bouton  ou alors de faire Ctrl + Maj + F5.

Si tu n'as pas le répertoire Dynpros, il faut que tu clic droit sur le répertoire du programme et de choisir "Autres fonctions"=>"Restructurer la liste des objets".

Voilà ce qu'on trouve :



Et si on va voir en debug :



Et on y voit tout de suite plus clair...

Pour palier à ce petit problème, il faut annuler l'OK_CODE seulement le compilateur renvoie une erreur comme quoi la variable SSCRFIELDS-UCOMM n'existe pas et quand on essaie de la déclarer le compilateur nous dit que la zone existe déjà... Alors la solution c'est d'utiliser un field-symbol.

Au niveau de la déclaration des données :

```
FIELD-SYMBOLS : <ucomm> TYPE sy-ucomm.
```

Au niveau du PAI de l'écran de sélection.

```
*      Annulation de l'execution
```

```
      CLEAR sy-ucomm.
```

```
      ASSIGN ('SSCRFIELDS-UCOMM') TO <ucomm>.
```

```
      IF <ucomm> IS ASSIGNED.
```

```
          CLEAR <ucomm>.
```

```
          UNASSIGN <ucomm>.
```

```
      ENDIF. "<ucomm> IS ASSIGNED.
```

```
      MESSAGE ID '38' TYPE 'E' NUMBER '000'
```

```
          WITH 'Veuillez renseigner un nom de fichier'(e01).
```

Et voilà, ça résout définitivement notre problème.

Voilà pour cette partie bien laborieuse. ^_^

4 Mettre des zones en surbrillance

Cette partie est surtout destinée à ceux qui veulent faire du tape à l'œil donc si t'es l'un des leur alors la suite pourra peut-être t'intéresser.

Reprenons l'exemple de la partie précédente, c'est à dire de rendre la zone P_FILE obligatoire et de mettre en surbrillance la zone quand la zone est vide.

Cela va se faire avec la zone SCREEN-INTENSIFIED.

* PAI de l'écran de sélection.

```
AT SELECTION-SCREEN.
```

```
IF sy-ucomm = 'ONLI'.
```

```
IF NOT cb1 IS INITIAL.
```

```
IF p_file IS INITIAL.
```

```
LOOP AT SCREEN.
```

```
IF screen-name = 'P_FILE'.
```

```
screen-input = 1.  
screen-required = 2.  
screen-intensified = 1.
```

```
ELSE.
```

```
screen-input = 1.  
screen-required = 0.  
screen-intensified = 0.
```

```
ENDIF. "screen-name = 'P_FILE'.
```

```
MODIFY SCREEN.
```

```
ENDLOOP. "AT SCREEN
```

```
MESSAGE ID '38' TYPE 'E' NUMBER '000'  
WITH 'Veuillez renseigner un nom de fichier'(e01).
```

```
ENDIF. "p_file IS INITIAL.
```

```
ENDIF. "NOT cb1 IS INITIAL.
```

```
ENDIF. "sy-ucomm = 'ONLI'.
```

On obtient donc :

Document de vente	<input type="text"/>	à	<input type="text"/>	<input type="text"/>	
Type document vente	<input type="text"/>	à	<input type="text"/>	<input type="text"/>	
<input checked="" type="checkbox"/> Checkbox					
Type de poste	<input type="text"/>	à	<input type="text"/>	<input type="text"/>	
fich. physique	<input checked="" type="checkbox"/>				

Et la zone est en surbrillance quand on lance le traitement, si si je l'assure... bon pour être crédible, on va interdire de mettre 'Toto' dans la zone

```
IF p_file IS INITIAL OR  
   p_file = 'Toto'.
```

Et on obtient :

The screenshot shows a SAP screen with two blocks. Block 1 has two rows: 'Document de vente' and 'Type document vente'. Each row has an input field and an arrow button. Block 2 has three rows: a checked checkbox, 'Type de poste' with an input field, and 'fich. physique' with the text 'Toto' in red. A dashed box highlights the checkbox.

On voit bien avec le fait que le contenu de la zone soit écrit en rouge que la zone est en surbrillance. :-)

Par contre, pense à annuler la surbrillance dans le PBO sinon la zone restera en rouge :

* PBO de l'écran de sélection

```
AT SELECTION-SCREEN OUTPUT.
```

```
LOOP AT SCREEN.
```

```
IF cb1 IS INITIAL.
```

```
IF screen-group1 = 'BL2'.
```

```
screen-input = 0.
```

```
ENDIF. "screen-name = 'P_FILE'.
```

```
IF screen-name = 'P_FILE'.
```

```
screen-required = 0.
```

```
ENDIF. "screen-name = 'P_FILE'.
```

```
ELSE.
```

```
IF screen-group1 = 'BL2'.
```

```
screen-input = 1.
```

```
ENDIF. "screen-group1 = 'BL2'.
```

```
IF screen-name = 'P_FILE'.
```

```
screen-required = 2.
```

```
ENDIF. "screen-name = 'P_FILE'.
```

```
ENDIF. "cb1 IS INITIAL.
```

* Annulation de la surbrillance

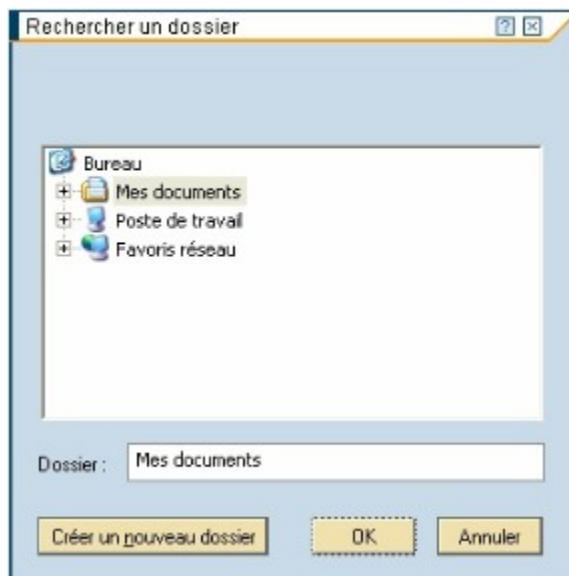
```
screen-intensified = 0.
```

```
MODIFY SCREEN.
```

```
ENDLOOP. "AT SCREEN.
```

5 Recherche d'un répertoire

On va voir ici comment avoir une fenêtre modal (voir ci-dessous) pour pouvoir choisir un emplacement de répertoire pour enregistrer un fichier généré automatiquement par exemple.



Pour cela, on va avoir besoin de faire apparaître un matchcode comme ceci.

Block 1			
Document de vente	<input type="text"/>	à	<input type="text"/>
Type document vente	<input type="text"/>	à	<input type="text"/>
<input type="button" value="→"/>			
Block 2			
<input type="checkbox"/> Checkbox			
Type de poste	<input type="text"/>	à	<input type="text"/>
fich. physique	<input type="text"/>		<input type="button" value="→"/>

Pour cela, il n'est pas nécessaire de changer la déclaration du paramètre mais juste de lui allouer un événement particulier..

AT SELECTION-SCREEN ON VALUE-REQUEST FOR param.

Pour nous ce sera :

AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_file.

Avec cet événement, la zone va afficher un matchcode et réagira à l'appui sur le bouton ou l'appui sur la touche F4.

Et on obtient ceci :

Block 1			
Document de vente	<input type="text"/>	à	<input type="text"/>
Type document vente	<input type="text"/>	à	<input type="text"/>
<input type="button" value="→"/>			
Block 2			
<input type="checkbox"/> Checkbox			
Type de poste	<input type="text"/>	à	<input type="text"/>
fich. physique	<input type="text"/>		<input type="button" value="→"/>

ou encore cela :

Block 1

Document de vente	<input type="text"/>	à	<input type="text"/>	
Type document vente	<input type="text"/>	à	<input type="text"/>	

Block 2

<input checked="" type="checkbox"/> Checkbox				
Type de poste	<input type="text"/>	à	<input type="text"/>	
fich. physique	<input checked="" type="checkbox"/>			



Reste à mettre en place le traitement pour afficher la fenêtre. Afin d'afficher une fenêtre de navigation dans les répertoires, on va utiliser une méthode de la classe CL_GUI_FRONTEND_SERVICES : **DIRECTORY_BROWSE**.

Nota : Je rappelle que pour générer les méthode proprement dans ton code, il est préférable de passer par les modèles objets, au même que les Modules Fonctions

```
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_file.  
  DATA: pickedfolder TYPE string.  
  
  CHECK NOT cb1 IS INITIAL.  
  
  CALL METHOD cl_gui_frontend_services=>directory_browse  
    EXPORTING  
  *   WINDOW_TITLE           =  
  *   INITIAL_FOLDER        =  
    CHANGING  
      selected_folder       = pickedfolder  
    EXCEPTIONS  
      cntl_error            = 1  
      error_no_gui         = 2  
      not_supported_by_gui = 3  
      OTHERS                = 4  
    .  
  
  IF sy-subrc <> 0.  
  *   MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO  
  *           WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.  
  ENDIF.  
  
  IF sy-subrc IS INITIAL.  
    p_file = pickedfolder.  
  
  ELSE.  
    WRITE: / 'Une erreur est survenue lors du choix du dossier'.  
  ENDIF.
```


Et on obtient :

Block 1

Document de vente	<input type="text"/>	à	<input type="text"/>	
Type document vente	<input type="text"/>	à	<input type="text"/>	

Block 2

Checkbox

Type de poste	<input type="text"/>	à	<input type="text"/>	
fich. physique	<input checked="" type="checkbox"/>			



6 Recherche/Sauvegarde d'un fichier

Le traitement va être quasiment identique sauf que l'on va utiliser une autre méthode de la classe CL_GUI_FRONTEND_SERVICES : **FILE_OPEN_DIALOG**.

```
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_file.

DATA : w_rc TYPE i.
DATA : t_filetable TYPE TABLE OF file_table,
      s_filetable LIKE LINE OF t_filetable.

CHECK NOT cb1 IS INITIAL.

CALL METHOD cl_gui_frontend_services=>file_open_dialog
*   EXPORTING
*   WINDOW_TITLE           =
*   DEFAULT_EXTENSION     =
*   DEFAULT_FILENAME     =
*   FILE_FILTER           =
*   WITH_ENCODING         =
*   INITIAL_DIRECTORY    =
*   MULTISELECTION       =
  CHANGING
    file_table           = t_filetable
    rc                   = w_rc
*   USER_ACTION          =
*   FILE_ENCODING       =
EXCEPTIONS
  file_open_dialog_failed = 1
  cntl_error              = 2
  error_no_gui            = 3
  not_supported_by_gui   = 4
  OTHERS                  = 5
  .

IF sy-subrc <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*   WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

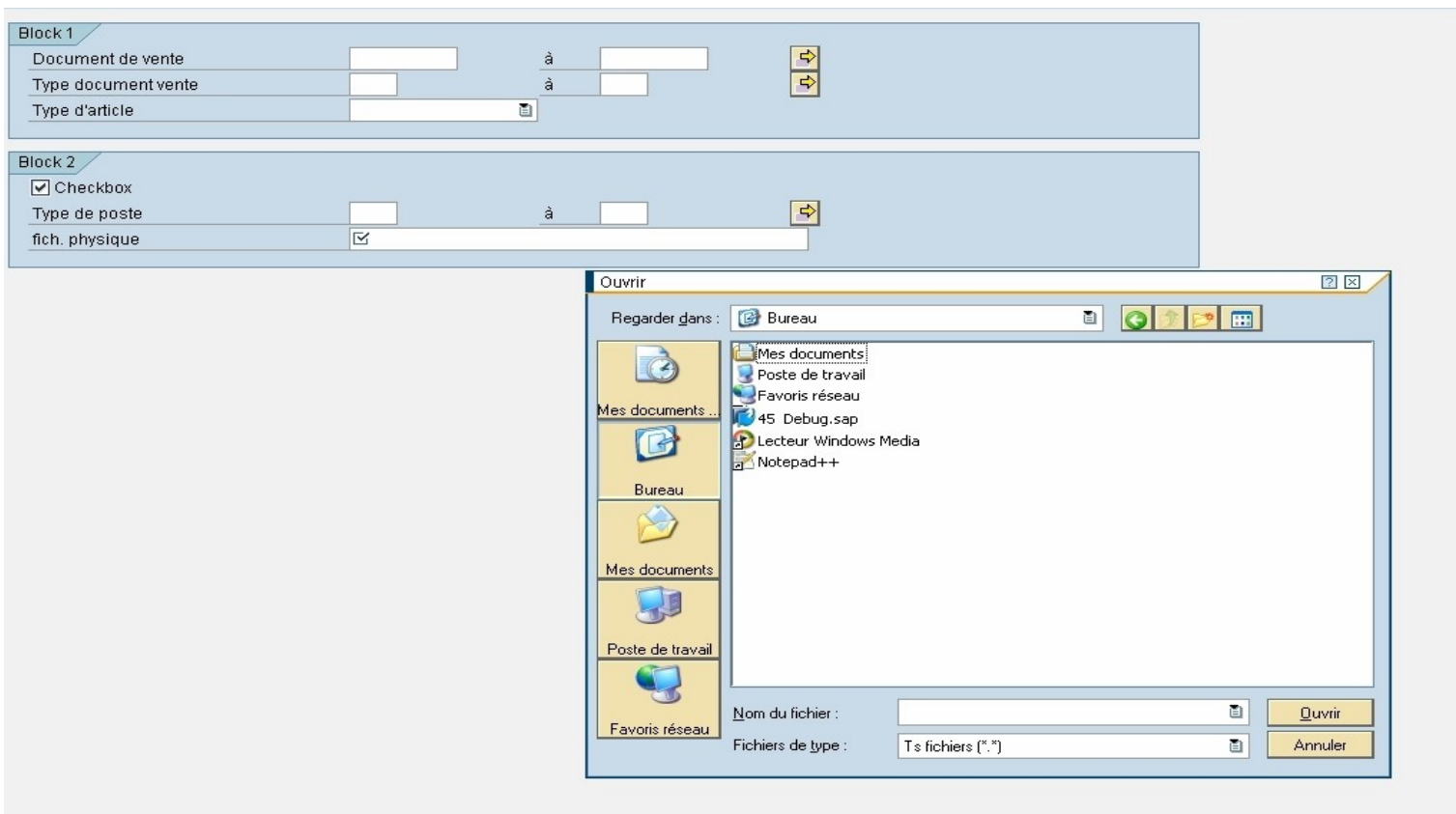
IF sy-subrc IS INITIAL.

  CLEAR s_filetable.
  READ TABLE t_filetable INTO s_filetable
    INDEX 1.

  p_file = s_filetable-filename.

ELSE.
  WRITE: / 'Une erreur est survenue lors du choix du fichier'.
ENDIF.
```

Et on obtient :



Pour enregistrer sous forme de fichier, il suffit d'utiliser la méthode FILE_SAVE_DIALOG mais je pense qu'il est inutile de donner un exemple.

7 Petit point sur certains composants

Le SELECT-OPTIONS :

Je voudrais juste montrer ici les possibilité que nous offre l'ABAP avec ce composant. En effet il est possible de l'afficher différemment selon le besoin.

L'affichage standard est celui-ci et ne nécessite qu'une simple déclaration :

```
SELECT-OPTIONS : so_vbeln FOR vbeln.
```

The screenshot shows two blocks of a SAP Select-Options dialog. Block 1 contains two rows: 'Document de vente' and 'Type document vente'. The first row has a yellow selection bar, a lock icon, and a right-pointing arrow button. The second row has a white selection bar and a right-pointing arrow button. Block 2 contains three rows: 'Checkbox', 'Type de poste', and 'fich. physique'. The first row has a checkbox. The second row has a white selection bar and a right-pointing arrow button. The third row has a white selection bar.

Mais si par exemple on ne veut autoriser qu'une valeur ou qu'une plage de valeur, dans ce cas il faut enlever le bouton de droite qui se traduit par EXTENSION en ABAP.

```
SELECT-OPTIONS : so_vbeln FOR vbeln NO-EXTENSION.
```

The screenshot shows two blocks of a SAP Select-Options dialog. Block 1 contains two rows: 'Document de vente' and 'Type document vente'. The first row has a yellow selection bar, a lock icon, and no right-pointing arrow button. The second row has a white selection bar and a right-pointing arrow button. Block 2 contains three rows: 'Checkbox', 'Type de poste', and 'fich. physique'. The first row has a checkbox. The second row has a white selection bar and a right-pointing arrow button. The third row has a white selection bar.

Si par contre, on ne veut pas autoriser les intervalles, il faut exclure la seconde zone du composant, mais il restera la possibilité de rentrer plusieurs valeurs.

```
SELECT-OPTIONS : so_vbeln FOR vbeln NO INTERVALS.
```

The screenshot shows two blocks of a SAP Select-Options dialog. Block 1 contains two rows: 'Document de vente' and 'Type document vente'. The first row has a yellow selection bar, a lock icon, and a right-pointing arrow button. The second row has a white selection bar and a right-pointing arrow button. Block 2 contains three rows: 'Checkbox', 'Type de poste', and 'fich. physique'. The first row has a checkbox. The second row has a white selection bar and a right-pointing arrow button. The third row has a white selection bar.

Et pour finir, on peut bien sûr n'autoriser qu'une seule valeur ainsi :

```
SELECT-OPTIONS : so_vbeln FOR vbeln NO INTERVALS NO-EXTENSION.
```

The screenshot shows two blocks of code in an SAP ABAP editor. Block 1 contains a SELECT statement with a red box highlighting the 'Document de vente' field and a yellow box highlighting the 'so_vbeln' field. Block 2 contains a SELECT statement with a checkbox and a 'fich. physique' field.

Pour ce dernier exemple, je m'attends à la remarque suivant :

Ça n'a aucun intérêt, autant déclarer un paramètre (PARAMETERS), ça ira plus vite.

Et j'ai envie de répondre oui et non, mais il faut se souvenir de comment fonctionne un SELECT-OPTIONS dans une requête (SELECT ...).

Si aucune valeur n'est renseignée dans le SELECT-OPTIONS, la requête récupérera toutes les valeurs comme si que la condition **WHERE champ IN so_champ** était ignoré.

Alors qu'avec un paramètre, s'il n'est pas renseigné, la requête va chercher les valeurs correspondant à la valeur '' (initial) et la requête ne retournera surement rien, donc un paramètre qui est utilisé dans une requête a de très grandes chances de devoir être mis obligatoire (ou recommandé ^_^).

Je voudrais finir sur l'intérêt de privilégier les SELECT-OPTIONS dans des requêtes SQL. Ce composant, et au même titre les RANGES, permettent de modifier les conditions en debug sur un environnement ou le programme ne peut être modifier :

Exemple :

Solution sans SELECT-OPTIONS :

```
l_low = '20080101'.  
l_high = '20081231'.
```

```
SELECT *  
  FROM vbak  
  INTO TABLE t_vbak  
 WHERE vbeln IN so_vbeln  
    AND vdatu >= l_low  
    AND vdatu <= l_high.
```

Solution avec SELECT-OPTIONS :

* Alimentation du range LR_VDATU.

```
ls_vdatu-sign = 'I'.  
ls_vdatu-option = 'BT'.  
ls_vdatu-low = '20080101'.  
ls_vdatu-high = '20081231'.  
APPEND ls_vdatu TO lr_vdatu.
```

```
SELECT *  
  FROM vbak  
  INTO TABLE t_vbak  
 WHERE vbeln IN so_vbeln  
    AND vdatu IN lr_vdatu.
```

Et qui c'est qui sera content de pouvoir prendre une autre plage de valeur en debug??

La Listbox (Liste déroulante) :

Un composant de type **listbox** peut-être placé sur un écran de sélection.

```
PARAMETERS : p_mtart TYPE mara-rtart AS LISTBOX VISIBLE LENGTH 18  
USER-COMMAND mta.
```

J'ai mis un USER-COMMAND juste pour indiquer que pour faire un traitement particulier suite à la sélection d'une valeur, il faut mettre un USER-COMMAND fcode.

L'évènement à mettre en place pour les traitements sur ce composant est :
AT SELECTION-SCREEN ON param.

The screenshot shows a SAP selection screen with two blocks. Block 1 contains three fields: 'Document de vente', 'Type document vente', and 'Type d'article'. Block 2 contains a 'Checkbox', 'Type de poste', and 'fich. physique'. Arrows indicate relationships between fields in Block 1 and Block 2.

On peut consulter les valeurs.

The screenshot shows the same SAP selection screen as above, but with the 'Type d'article' dropdown menu open, showing a list of values including 'Coupon', 'Déchet', 'Emballage', 'Emballages consignés', 'Matière première', 'Mat./pdt consommable', 'Mode (saisonnière)', 'Moyen publicitaire', 'Outillage', and 'Pièce fabricant'.

Et en choisir une.

The screenshot shows the same SAP selection screen as above, but with the 'Type d'article' dropdown menu closed and the value 'Boissons' selected.

PS : Oui je cherchais le bar le plus proche. ^_^

A noter qu'il est impératif que le paramètre fasse référence à une zone de table ayant une table de contrôle lui étant associé ou une liste de valeur (dans le domaine) sans quoi la Listbox sera vide.

Pour vérifier que la zone est lié à une table de contrôle, il suffit de consulter la table en SE11 et d'aller dans l'onglet "Aide/Contrôle à la saisie" et de regarder dans la colonne "Table de contrôle".

Pour les listes de valeurs, il suffit d'aller dans l'Élément de donnée de la zone, rentrer dans le domaine s'il en possède un, et d'aller dans l'onglet "Plage de valeur".

8 Annexe

Je met mon code source en annexe :

```
*&-----*
*& Report  ZMODIF_SCREEN
*&
*&-----*
*& Auteur : Celdrøn
*&
*&-----*

REPORT  zmodif_screen.

* Données pour écran de sélection.
DATA : vbeln TYPE vbak-vbeln,
       auart TYPE vbak-auart,
       pstyv TYPE vbap-pstyv,

* BLOCK 1
SELECTION-SCREEN BEGIN OF BLOCK b11 WITH FRAME TITLE text-b11.

SELECT-OPTIONS : so_vbeln FOR vbeln. "NO INTERVALS NO-EXTENSION.

SELECT-OPTIONS : so_auart FOR auart.

PARAMETERS : p_mtart TYPE mara-mtart AS LISTBOX VISIBLE LENGTH 18
              USER-COMMAND mta.

SELECTION-SCREEN END OF BLOCK b11.

* BLOCK 2
SELECTION-SCREEN BEGIN OF BLOCK b12 WITH FRAME TITLE text-b12.

PARAMETERS : cb1 TYPE c AS CHECKBOX USER-COMMAND cb1.

SELECT-OPTIONS : so_pstyv FOR pstyv MODIF ID b12.

PARAMETERS : p_file TYPE fileextern MODIF ID b12.

SELECTION-SCREEN END OF BLOCK b12.

* Field_symbols pour pointer vers la zone SSCRFIELDS-UCOMM
FIELD-SYMBOLS : <ucomm> TYPE sy-ucomm.

* Action sur le paramètre P_FILE
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_file.
* DATA: pickedfolder TYPE string.

DATA : w_rc TYPE i.
DATA : t_filetable TYPE TABLE OF file_table,
       s_filetable LIKE LINE OF t_filetable.

CHECK NOT cb1 IS INITIAL.

* CALL METHOD cl_gui_frontend_services=>directory_browse
* EXPORTING
** WINDOW_TITLE           =
** INITIAL_FOLDER        =
* CHANGING
```

```

*      selected_folder      = pickedfolder
*      EXCEPTIONS
*      cntl_error           = 1
*      error_no_gui        = 2
*      not_supported_by_gui = 3
*      OTHERS               = 4
*
*
CALL METHOD cl_gui_frontend_services=>file_open_dialog
*      EXPORTING
*      WINDOW_TITLE         =
*      DEFAULT_EXTENSION    =
*      DEFAULT_FILENAME     =
*      FILE_FILTER          =
*      WITH_ENCODING        =
*      INITIAL_DIRECTORY    =
*      MULTISELECTION       =
      CHANGING
      file_table            = t_filetable
      rc                    = w_rc
*      USER_ACTION          =
*      FILE_ENCODING        =
EXCEPTIONS
      file_open_dialog_failed = 1
      cntl_error              = 2
      error_no_gui            = 3
      not_supported_by_gui    = 4
      OTHERS                  = 5
      .

IF sy-subrc <> 0.
*      MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*              WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

IF sy-subrc IS INITIAL.

*      p_file = pickedfolder.

      CLEAR s_filetable.
      READ TABLE t_filetable INTO s_filetable
          INDEX 1.

      p_file = s_filetable-filename.

ELSE.
*      WRITE: / 'Une erreur est survenue lors du choix du dossier'.

      WRITE: / 'Une erreur est survenue lors du choix du fichier'.
ENDIF.

* PBO de l'écran de sélection
AT SELECTION-SCREEN OUTPUT.

LOOP AT SCREEN.

      IF cb1 IS INITIAL.

          IF screen-group1 = 'BL2'.
              screen-input = 0.
          .
      .

```

```

ENDIF. "screen-group1 = 'BL2'.

IF screen-name = 'P_FILE'.
    screen-required = 0.
ENDIF. "screen-name = 'P_FILE'.

ELSE.

    IF screen-group1 = 'BL2'.
        screen-input = 1.
    ENDIF. "screen-group1 = 'BL2'.

    IF screen-name = 'P_FILE'.
        screen-required = 2.
    ENDIF. "screen-name = 'P_FILE'.

ENDIF. "cb1 IS INITIAL.

* Annulation de la surbrillance
screen-intensified = 0.

MODIFY SCREEN.

ENDLOOP. "AT SCREEN.

* PAI de l'écran de sélection.
AT SELECTION-SCREEN.

IF sy-ucomm = 'ONLI'.

    IF NOT cb1 IS INITIAL.

        IF p_file IS INITIAL OR
            p_file = 'Toto'.

            LOOP AT SCREEN.

                IF screen-name = 'P_FILE'.

                    screen-input = 1.
                    screen-required = 2.
                    screen-intensified = 1.

                ELSEIF screen-name = 'CB1'.
                    screen-input = 1.
                    screen-required = 0.
                    screen-intensified = 0.

                ELSE.
                    screen-input = 0.
                    screen-required = 0.
                    screen-intensified = 0.

                ENDIF. "screen-name = 'P_FILE'.

            MODIFY SCREEN.

        ENDLOOP. "AT SCREEN

* Annulation de l'exécution
CLEAR sy-ucomm.

```

```
ASSIGN ('SSCRFIELDS-UCOMM') TO <ucomm>.

IF <ucomm> IS ASSIGNED.
  CLEAR <ucomm>.

  UNASSIGN <ucomm>.

ENDIF. "<ucomm> IS ASSIGNED.

MESSAGE ID '38' TYPE 'E' NUMBER '000'
  WITH 'Veuillez renseigner un nom de fichier'(e01).

ENDIF. "p_file IS INITIAL OR ...

ENDIF. "NOT cb1 IS INITIAL.

ENDIF. "sy-ucomm = 'ONLI'.

START-OF-SELECTION.

WRITE : 'Hello World'.
```