

Je viens de faire évoluer une édition spécifique. Le but est de pouvoir obtenir, à partir d'un ALV, avec l'édition papier, un fichier PDF. Chaque Edition doit déclencher l'écriture du PDF
J'ai donc consulté notre forum, ceux de SAP, et j'ai fini par aboutir à ce que je vous livre ici.

1. La gestion de l'impression

L'appel de l'ALV se fait de cette manière :

```
CALL FUNCTION 'REUSE_ALV_LIST_DISPLAY'
  EXPORTING
    i_callback_program = sy-cprog
    i_structure_name    = 'ST_ALV'
    it_fieldcat        = t_fields[]
    it_sort             = t_sort[]
    i_save              = 'X'
    it_events           = gt_events[]
    is_print            = lst_print
  TABLES
    t_outtab           = t_alv.
```

La structure lst_print a ici son importance car elle donne à l'utilisateur la possibilité de lancer son édition sans aperçu : si "lst_print-print" est initialisé, l'ALV est directement envoyé vers l'imprimante sans être affiché à l'écran.

```
*_paramètres d'impression
IF NOT p_chkpdf IS INITIAL.
  lst_print-print = 'X'.
ENDIF.
lst_print-no_coverpage = 'X'.
lst_print-no_print_selinfos = 'X'.
lst_print-no_print_listinfos = 'X'.
PERFORM get_param CHANGING lw_ok.
```

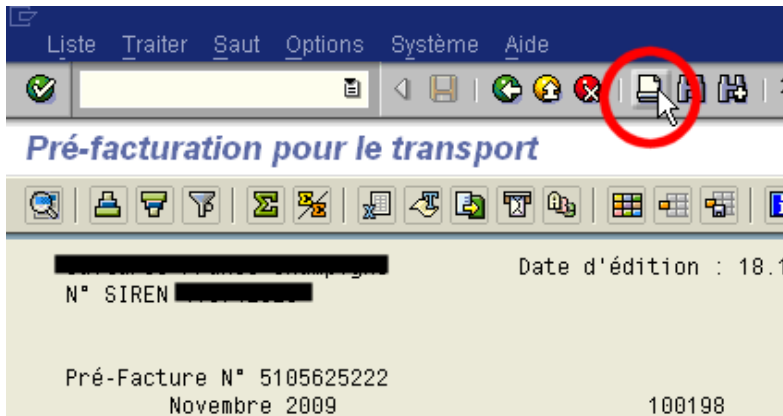
La routine "get_param" va nous permettre de charger les paramètres d'impression qui seront exploités pour le PDF :

```
FORM get_param CHANGING pw_ok TYPE c.
  DATA: lw_days(1) TYPE n VALUE 2,
        lw_count(3) TYPE n VALUE 1.

  CALL FUNCTION 'GET_PRINT_PARAMETERS'
    EXPORTING
      destination      = 'BADS'
      copies            = lw_count
      list_name         = 'PREFAC'
      list_text         = 'Préfac. de Transport'
      immediately      = 'X'
      release           = ' '
      new_list_id       = 'X'
      expiration        = lw_days
      line_size         = 245
      line_count        = 65
      layout            = 'X_65_255'
      sap_cover_page    = ' '
      receiver          = 'SAP*'
      department        = 'System'
      no_dialog         = 'X'
    IMPORTING
      out_parameters    = st_params
      valid              = pw_ok.

ENDFORM.                " get_param
```


On décide d'imprimer :



Et on sort du report.

3. Générer le PDF

En retour vers le spécifique le but est récupérer le No du spool ainsi généré et de le convertir vers une édition PDF : la routine "print_pdf" va nous faire ça

```
CALL FUNCTION 'REUSE_ALV_LIST_DISPLAY'
  EXPORTING
    i_callback_program = sy-cprog
    i_structure_name   = 'ST_ALV'
    it_fieldcat       = t_fields[]
    it_sort           = t_sort[]
    i_save            = 'X'
    it_events         = gt_events[]
    is_print          = lst_print
  TABLES
    t_outtab         = t_alv.
```

```
PERFORM print_pdf.
```

```
*****
```

```
FORM print_pdf .
  DATA: lw_spono LIKE tsp01-rqidnt,
        lw_npono(10) TYPE n.
  DATA: lst_pdf LIKE tline OCCURS 0 WITH HEADER LINE.
  *_Variables à transmettre à GUI_DOWNLOAD
  DATA: lw_filesize TYPE i,
        lw_bin_filesize TYPE i.
  DATA: lw_fullpath TYPE string,
        lw_filename TYPE string.
```

```
lw_spono = sy-spono.
IF lw_spono IS INITIAL.
  EXIT.
ENDIF.
```

```
*_convertir le spool vers du PDF
CALL FUNCTION 'CONVERT_ABAPSPoolJOB_2_PDF'
  EXPORTING
    src_spoolid           = lw_spono
    no_dialog             = 'X'
```

```

    dst_device          = st_params-pdest
    pdf_destination    = 'X'
TABLES
    pdf                 = lst_pdf
EXCEPTIONS
    err_no_abap_spooljob = 1
    err_no_spooljob      = 2
    err_no_permission    = 3
    err_conv_not_possible = 4
    err_bad_destdevice   = 5
    user_cancelled      = 6
    err_spoolerror       = 7
    err_temseerror       = 8
    err_btcjob_open_failed = 9
    err_btcjob_submit_failed = 10
    err_btcjob_close_failed = 11
    OTHERS                = 12.
IF sy-subrc = 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ELSE.
    EXIT.
ENDIF.

```

*_construire le nom du pdf

```

SELECT SINGLE zont1 FROM zparam INTO lw_fullpath
  WHERE appli = 'MC'
    AND codep = 'PREFAC'
    AND clef1 = 'PDF'.
MOVE lw_spono TO lw_npono.
CONCATENATE w_lifnr '_' w_name1 '_' p_belnr '_' lw_npono '.pdf'
  INTO lw_filename.
CONCATENATE lw_fullpath lw_filename INTO lw_fullpath.

```

*_Edition du PDF

```

CALL FUNCTION 'GUI_DOWNLOAD'
  EXPORTING
    bin_filesize      = lw_bin_filesize
    filename           = lw_fullpath
    filetype           = 'BIN'
  IMPORTING
    filelength        = lw_filesize
  TABLES
    data_tab          = lst_pdf
  EXCEPTIONS
    file_write_error  = 1
    no_batch           = 2
    gui_refuse_filetransfer = 3
    invalid_type       = 4
    no_authority       = 5
    unknown_error      = 6
    header_not_allowed = 7
    separator_not_allowed = 8
    filesize_not_allowed = 9
    header_too_long    = 10
    dp_error_create    = 11
    dp_error_send      = 12
    dp_error_write     = 13
    unknown_dp_error   = 14
    access_denied      = 15
    dp_out_of_memory   = 16
    disk_full          = 17
    dp_timeout         = 18
    file_not_found     = 19
    dataprovider_exception = 20
    control_flush_error = 21
    OTHERS              = 22.
IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.

```

```
ENDIF.  
ENDFORM.          " print_pdf
```

Et voilà, le tour est joué.

4. Conclusion

Je ne suis pas sûr que ce cheminement soit optimal, mais c'est un bon début pour ceux qui, comme moi, tâtonne encore autour de traitements aussi particuliers.

Je pense, vu les questions sur le forum, que cette démarche doit pouvoir en aider quelques uns.