Tutorial on SMART FORMS

Applies To:

SAP R/3 4.6C – ABAP / SMART FORMS.

Summary

SAP Smart Forms are used to create and maintain forms for mass printing in SAP Systems. As an output medium SAP Smart Forms support a printer, a fax, e-mail. Many a times there is requirement to download output of SAP Smart Forms in a PDF file. This piece of code gives users a demo for creating and downloading Smart forms to their local PC.

By: Swaroopa Vishwanath
Company and Title: Wipro Technologies, SAP ABAP/XI Consultant
Date: 11 Feb 2005

Table of Contents

Introduction:

SAP Smart Forms allow us to execute simple modifications to the form and in the form logic by using simple graphical tools; in 90% of all cases, this won't include any programming effort. To print a form, you need a program for data retrieval and a Smart Form that contains the entire from logic. As data retrieval and form logic are separated, you must only adapt the Smart Form if changes to the form logic are necessary. For the purpose of demonstrating the power and download capabilities of Smart form, I have created a Smart from named ZSMARTFORM_SWAR.

Creation of SMART FORM:

From SAP R/3 release 4.6C onwards, SAP has come up with the concept of SMART FORMS. Prior to SAP R/3 4.6C, Sap Scripts were being used for mass printing. To create a smart form, execute the transaction **"SMART FORMS"**. Provide "ZSMARTFORM_SWAR" as the name of the Smart form. Select "Create".

When you create a Smart Form, the tree structure of the Form Painter already contains two root nodes:

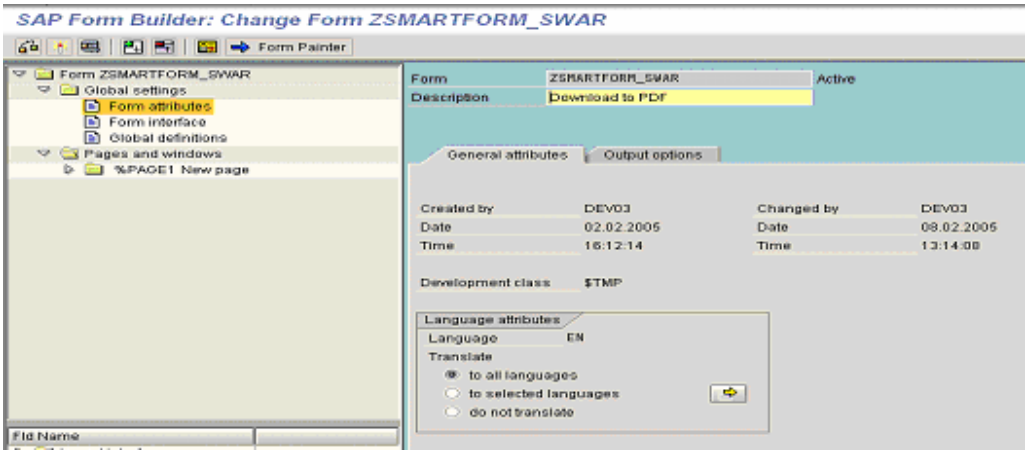
- Global setting
- Pages and Windows

Global Setting

The node *Global Settings* as well as its three successors *Form attributes*, *Form interface,* and *Global definitions* always exist for any newly created forms.

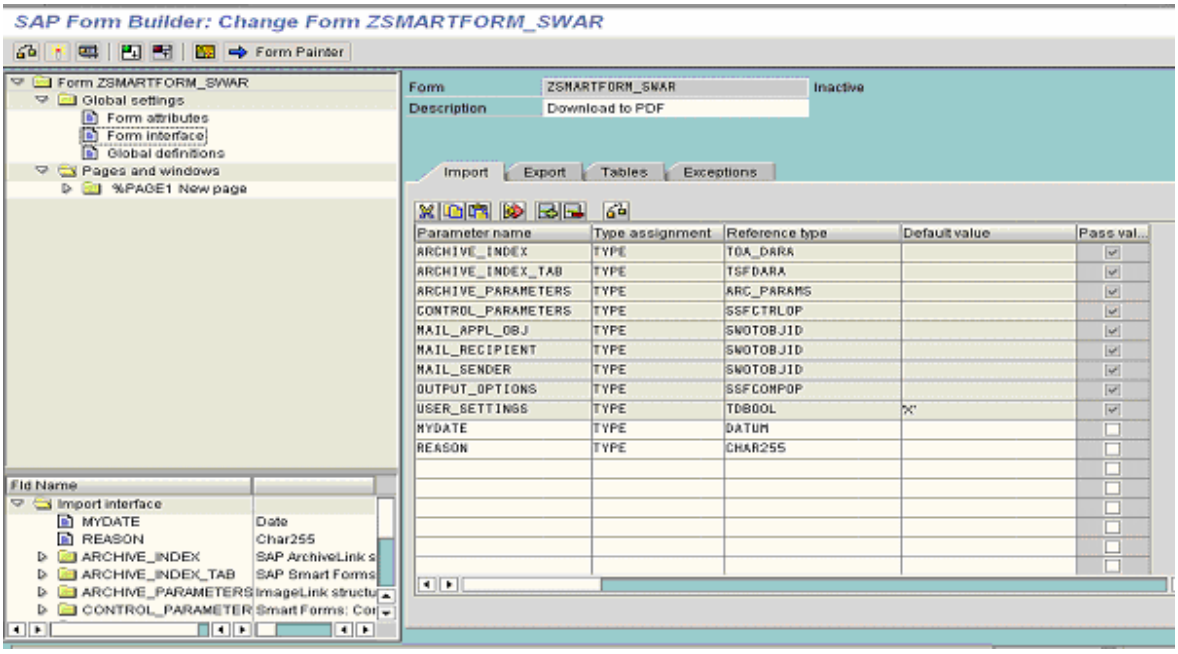
Form attributes

Here you need to provide the description for the smart form. In this case, the description is "Create and Download SMART FORM to PDF". Here, there is a provision to specify whether the Smart form can be translated to other languages in future.



Form Interface

In Form Interface, you can specify the parameters that would be imported and exported to Smart form. In this case, I have imported two parameters "MYDATE" and "REASON". In this case, there are no Export parameters and Tables. These can be used if we intent to pass information from Smart forms to the ABAP program. There are few standard import and export parameters that can be used depending on the functionality.

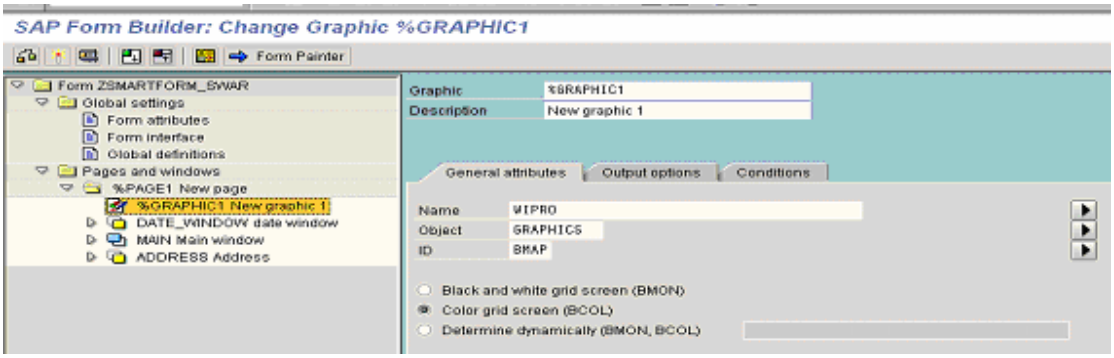


Pages and windows:

A form can consist of pages, output areas, addresses, graphics (such as company logo), and data or text contents. Within an output area, we can use static or dynamic tables to display data or texts in lines, columns, or cells. To further structure and format text and data, we can use paragraphs with the paragraph and character formats. SAP calls output areas "windows". We can position windows freely on a page. On a page, there are two different types of output areas for texts and data: the main window and the sub window.

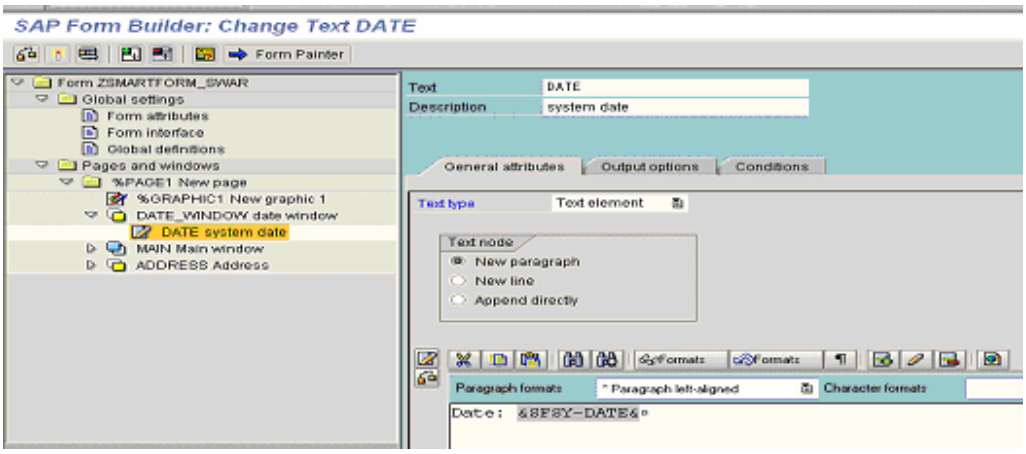
Graphics:

In the Form Builder we describe a Smart Form by a set of nodes. To do this, you build up a tree structure on the left side of the user interface. On the user Interface, specify the name, object and Id of the bitmap image to be placed on the Smart form.



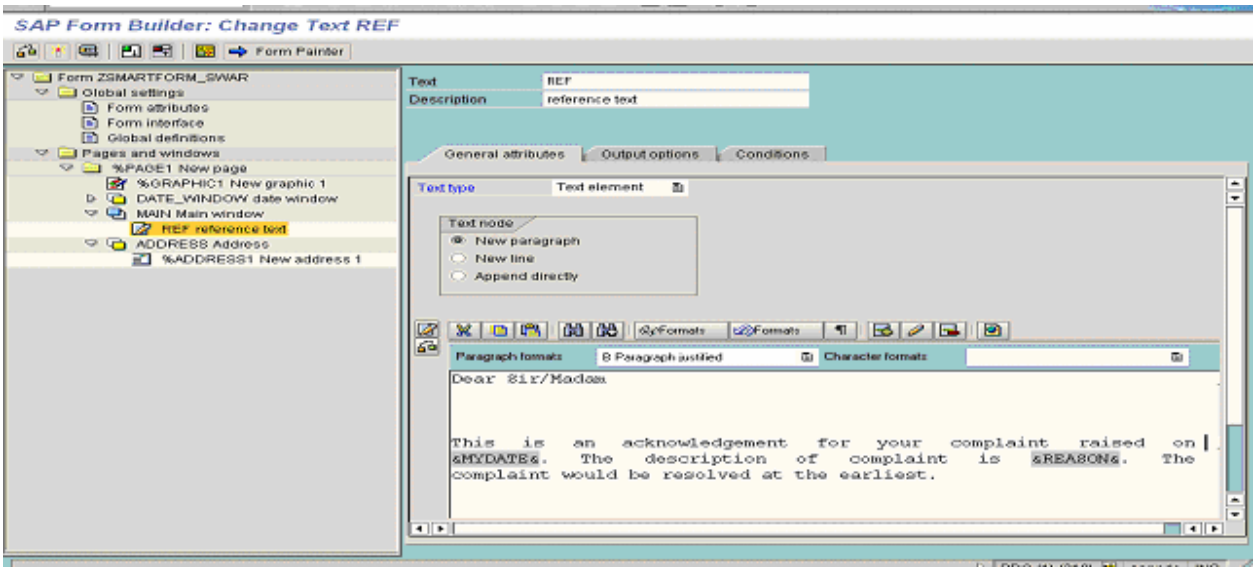
Date Window

In a Date sub window we display text and data in a predetermined output area. If you position a sub window with the same name on several pages, the system displays the contents of this sub window on each page. Text and data that do not fit into the sub window are truncated and not displayed.



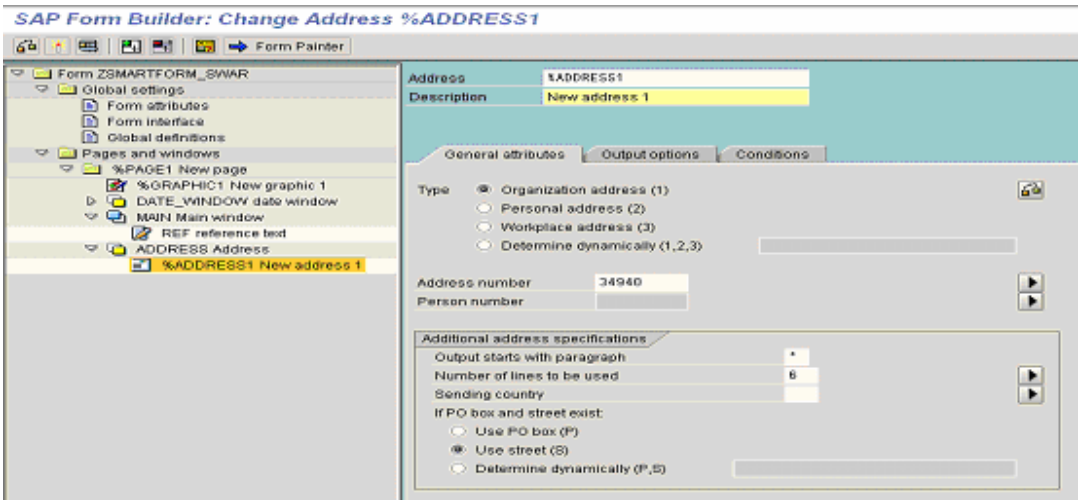
Main window

On a page, there are two different types of output areas for texts and data: the main window and the sub window. In a main window you display text and data, which can cover several pages (flow text). As soon as a main window is completely filled with text and data, the system continues displaying the text in the main window of the next page. It automatically triggers the page break.



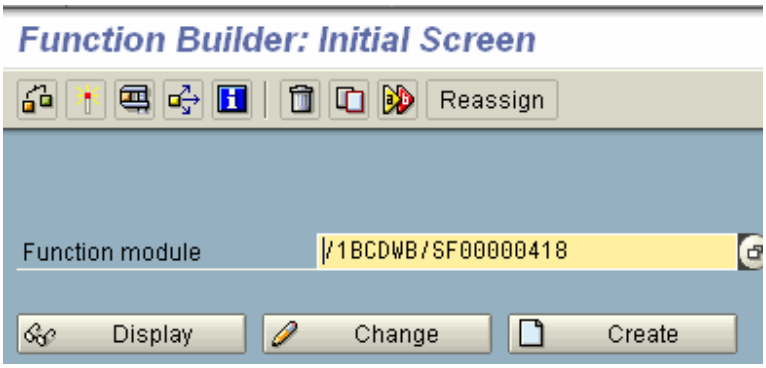
Address

You use this node to include an address. The system reads the address data directly from the database tables and formats them for print output. This guarantees that the address is formatted according to the postal rules of the sender country. You can select the type of address that should be displayed in the output.



Generate Function Module for Smart Form

The system generates a function module from the Smart Form as soon as you activate the Smart form the form interface equals that of the generated function module. For this reason, you can pass the same data in the form interface as you could pass to a function module. The generated function module triggers spool processing.



Downloading Smart Form to PDF from Application

You can pass data retrieved from the application to the Smart form using the Form Interface. The data could also include the data retrieved from database tables. In addition, you can define exceptions to which the application program must react. You trigger form printing by calling only two function modules. The first module uses the name of the form to determine the name of the generated function module. Then you call this Smart form.

Call function module **SSF_FUNCTION_MODULE_NAME**. It returns the name of the generated function module. Call the generated function module. To do this, use the *Insert statement* function for **CALL FUNCTION** in the ABAP Editor. Then replace the function module name with the variable **FM_NAME.** Set the **'GETOTF'** parameter to **'X'** and pass it to **CONTROL_PARAMETERS**. The function module would export the Smart form to an internal table **'T_OTFDATA'**.

Call the function module **'CONVERT_OTF'** to convert the smart form to PDF format. The function module returns the size of downloaded file. Using **'WS_DOWNLOAD'** function module, you can download the Smart form to a PDF file in the required directory.

Sample Code

```
REPORT zswar.

SELECTION-SCREEN: BEGIN OF BLOCK b1 WITH FRAME.
PARAMETER: p_date LIKE sy-datum.
PARAMETER: p_rea TYPE char255.
SELECTION-SCREEN: END OF BLOCK b1.

DATA:      ws_ucomm LIKE sy-ucomm.


INITIALIZATION.
  SET PF-STATUS 'STANDARD' OF PROGRAM 'ZSWAR'.


AT SELECTION-SCREEN.

  ws_ucomm = sy-ucomm.


  CASE ws_ucomm.
    WHEN '&PDF'.
      PERFORM f1000_download_form.
      EXIT.
    WHEN '&BACK'.
      SET SCREEN 0.
      EXIT.
    WHEN '&EXIT'.
      SET SCREEN 0.
      EXIT.
    WHEN '&canc'.
      SET SCREEN 0.
      LEAVE TO SCREEN 0.
  ENDCASE.


*&---------------------------------------------------------------------*
*&      Form  F1000_DOWNLOAD_FORM
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM f1000_download_form.

  DATA: form_name TYPE rs38l_fnam.
  DATA: wa_ctrlop TYPE ssfctrlop,
        wa_outopt TYPE ssfcompop.

  DATA: t_otfdata  TYPE ssfcrescl,
        t_pdf_tab  LIKE tline OCCURS 0 WITH HEADER LINE.
```

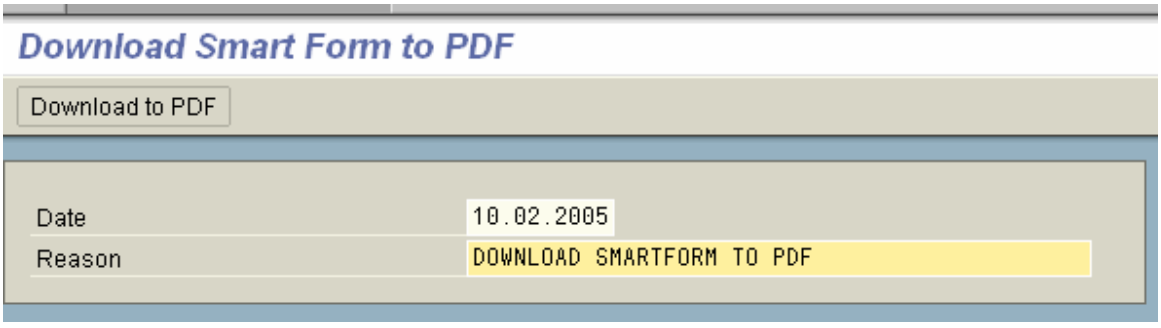
```
  DATA: t_otf TYPE itcoo OCCURS 0 WITH HEADER LINE.
  DATA: w_filesize TYPE i.
  DATA: w_bin_filesize TYPE i.

  CALL FUNCTION 'SSF_FUNCTION_MODULE_NAME'
    EXPORTING
      formname                 = 'ZSMARTFORM_SWAR'
*   VARIANT                  = ' '
*   DIRECT_CALL              = ' '
   IMPORTING
     fm_name                  = form_name
   EXCEPTIONS
     no_form                  = 1
     no_function_module       = 2
     OTHERS                   = 3
           .
  IF sy-subrc <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*         WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
  ENDIF.

  wa_ctrlop-getotf = 'X'.
  wa_ctrlop-no_dialog = 'X'.
  wa_outopt-tdnoprev = 'X'.

  CALL FUNCTION form_name
    EXPORTING
*   ARCHIVE_INDEX              =
*   ARCHIVE_INDEX_TAB          =
*   ARCHIVE_PARAMETERS         =
     control_parameters         = wa_ctrlop
*   MAIL_APPL_OBJ              =
*   MAIL_RECIPIENT             =
*   MAIL_SENDER                =
     output_options             = wa_outopt
     user_settings              = 'X'
     mydate                     = p_date
     reason                     = p_rea
   IMPORTING
*   DOCUMENT_OUTPUT_INFO       =
     job_output_info            = t_otfdata
*   JOB_OUTPUT_OPTIONS         =
   EXCEPTIONS
     formatting_error           = 1
     internal_error             = 2
     send_error                 = 3
     user_canceled              = 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.

  t_otf[] = t_otfdata-otfdata[].

  CALL FUNCTION 'CONVERT_OTF'
   EXPORTING
     format                       = 'PDF'
```

```
   max_linewidth               = 132
*   ARCHIVE_INDEX               = ' '
   IMPORTING
     bin_filesize                = w_bin_filesize
    TABLES
      otf                         = t_otf
      lines                       = t_pdf_tab
   EXCEPTIONS
     err_max_linewidth           = 1
     err_format                  = 2
     err_conv_not_possible       = 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.



  CALL FUNCTION 'WS_DOWNLOAD'
   EXPORTING
     bin_filesize                 = w_bin_filesize
*    CODEPAGE                    = ' '
     filename                     = 'd:\test.PDF'
     filetype                     = 'BIN'
*    MODE                        = ' '
*    WK1_N_FORMAT                = ' '
*    WK1_N_SIZE                  = ' '
*    WK1_T_FORMAT                = ' '
*    WK1_T_SIZE                  = ' '
*    COL_SELECT                  = ' '
*    COL_SELECTMASK              = ' '
*    NO_AUTH_CHECK               = ' '
   IMPORTING
     filelength                   = w_filesize
    TABLES
      data_tab                     = t_pdf_tab
*    FIELDNAMES                  =
   EXCEPTIONS
     file_open_error              = 1
     file_write_error             = 2
     invalid_filesize             = 3
     invalid_type                 = 4
     no_batch                     = 5
     unknown_error                = 6
     invalid_table_width          = 7
     gui_refuse_filetransfer      = 8
     customer_error               = 9
     OTHERS                       = 10
           .
  IF sy-subrc <> 0.
    MESSAGE i003(z00) WITH 'File not downloaded succesfully'.
  ELSE.
    MESSAGE i003(z00) WITH 'File Test.pdf downloaded succesfully '
                                          'under D drive'.
  ENDIF.

ENDFORM.                    " F1000_DOWNLOAD_FORM
```

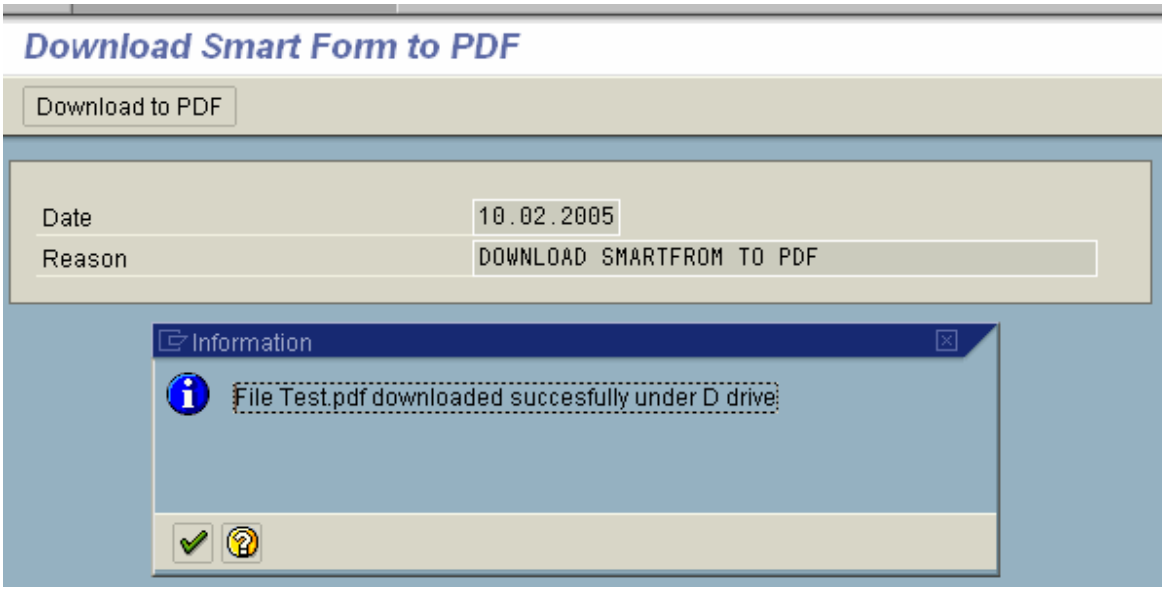
Sample Output of Smart Form Created

Output of application program with screen shots

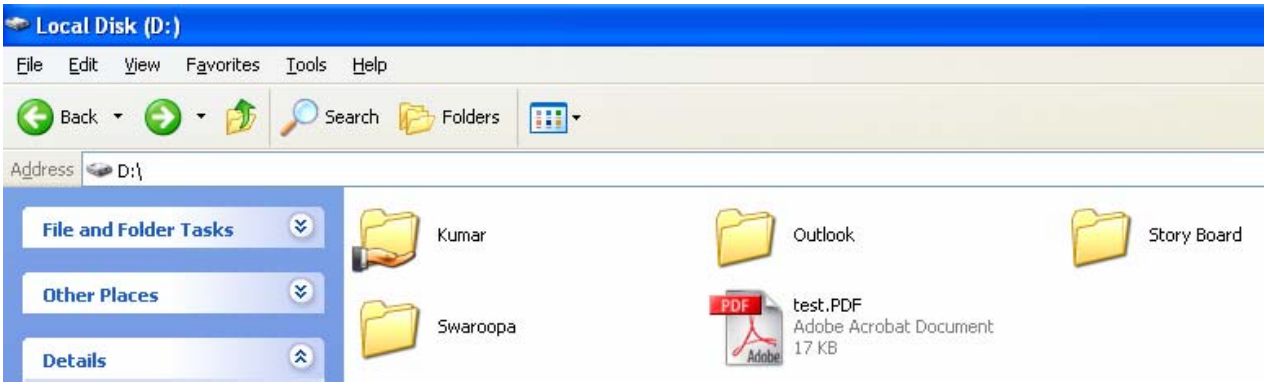
Activate and compile the code specified above. The selection screen is displayed as an output. The selection screen would contain a button named ‘Download to PDF’.



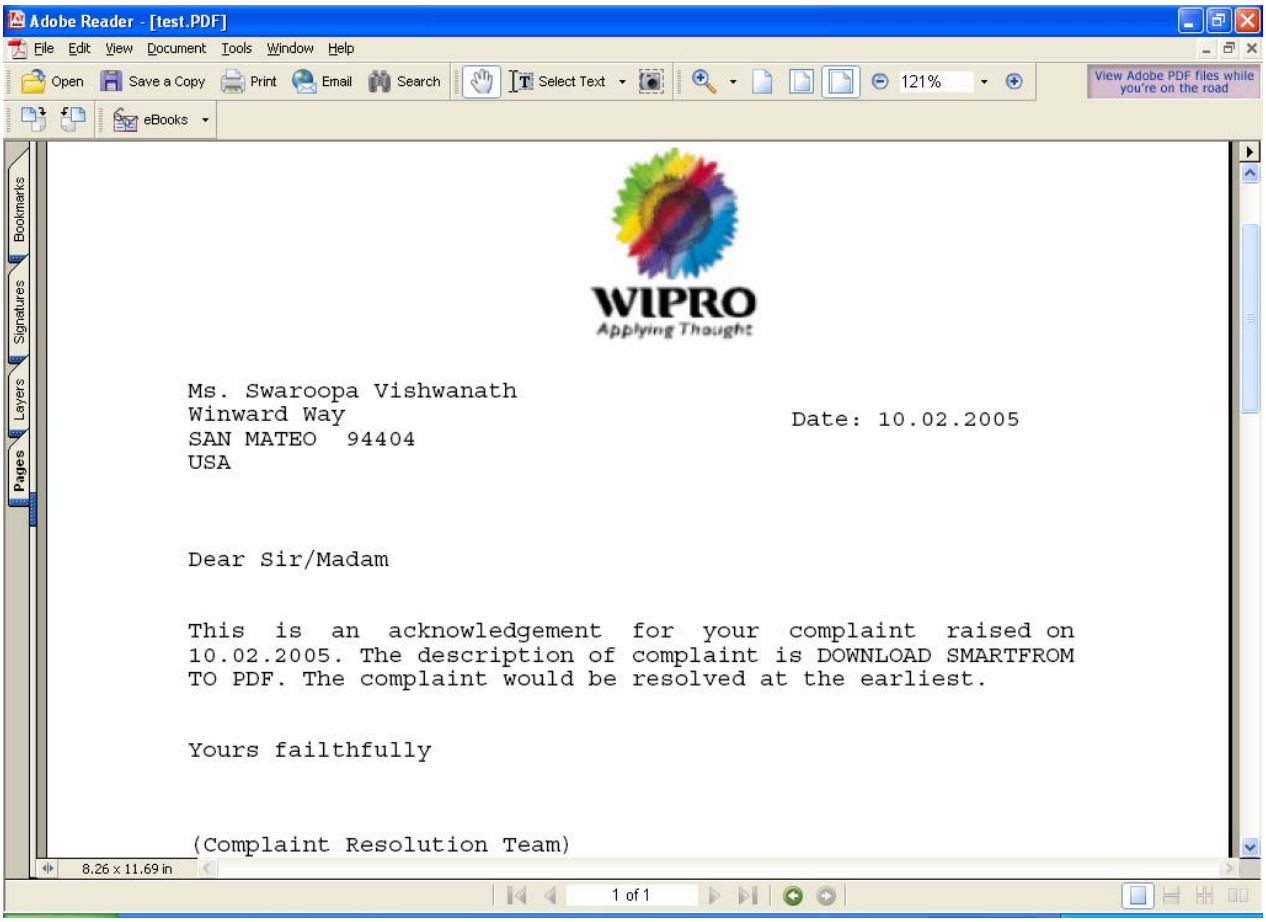
After inserting values for **'Date'** and **'Reason'**, execute **'Download to PDF'** button.



Below is the screen shot of D drive where **'Test.PDF'** file has been created.



The content of the downloaded file is shown below. The Smart form has been downloaded along with the graphics.



Disclaimer & Liability Notice

This document may discuss sample coding, which does not include official interfaces and therefore is not supported. Changes made based on this information are not supported and can be overwritten during an upgrade.
SAP will not be held liable for any damages caused by using or misusing of the code and methods suggested here, and anyone using these methods, is doing it under his/her own responsibility.
SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of the technical article, including any liability resulting from incompatibility between the content of the technical article and the materials and services offered by SAP. You agree that you will not hold SAP responsible or liable with respect to the content of the Technical Article or seek to do so.

Author Bio

I am an SAP ABAP/XI consultant working for Wipro Technologies. This tutorial would assist a starter to develop and download smart forms to PDF.