



1

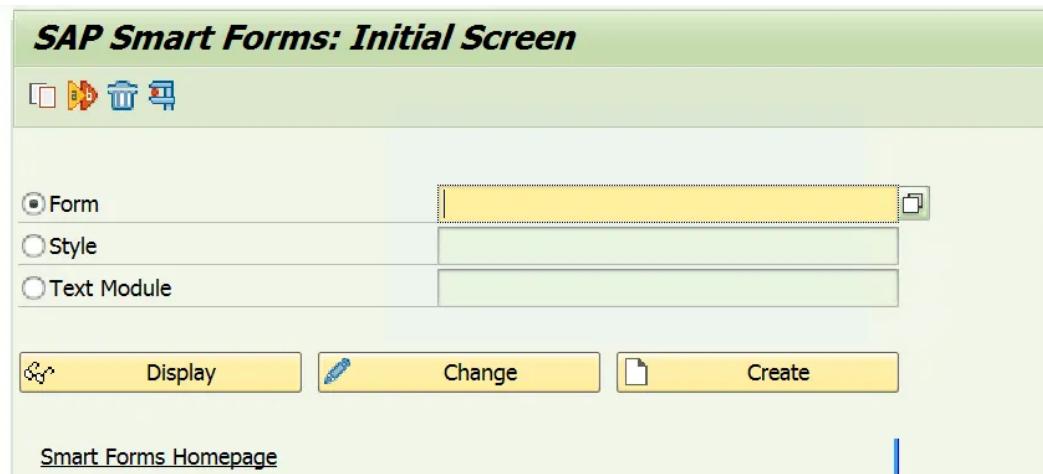
Introduction to Smartforms

1. Introduction

- Smartforms are layout in SAP which are used to create business documents.
 - For example, Amazon sends us document for purchase details, similar kind of details we can create with smartfoms.
-

2. Transaction Code

- The transaction code for smartforms is :- SMARTFORMS



- This is the initial screen for smartforms though which we can create our smartforms.

3. Features of Smartform

Form	Form ZAR_SMARTFORM
	- Global Settings
	* Form Attributes (Selected)
	* Form Interface
	* Global Definitions
	- Pages and Windows
	%PAGE1 New Page

Form	ZAR_SMARTFORM
Description	New Form
General Attributes	
Created By	S419AH02
Date	08.04.2024
Time	06:48:26
Language Attributes	
Language	EN

1. Form Attributes

- Form attributes gives us the generic information for our smartform.

2. Form Interface

SAP Form Builder: Change Form ZAR_SMARTFORM

Form

ZAR_SMARTFORM

Global Settings

- Form Attributes
- Form Interface**
- Global Definitions

Pages and Windows

%PAGE1 New Page

Form

ZAR_SMARTFORM

Description

New Form

Import Export Tables Exceptions

Parameter Name	Type Assignment	Associated Type
ARCHIVE_INDEX	TYPE	TOA_DARA
ARCHIVE_INDEX_TAB	TYPE	TSFDARA
ARCHIVE_PARAMETERS	TYPE	ARC_PARAMS
CONTROL_PARAMETERS	TYPE	SSFCTRLOP
MAIL_APPL_OBJ	TYPE	SWOTOBJID
MAIL_RECIPIENT	TYPE	SWOTOBJID

- In form Interface, we can see we have 4 important tabs :-
 - Import :- Use to import required things into the smartform.
 - Export :- Use for exporting.
 - Tables :- We can take tables though tables tab.
 - Exceptions :- We can pass exception to our smartform using exception tab.

3. Global Definition

SAP Form Builder: Change Form ZAR_SMARTFORM

Form

ZAR_SMARTFORM

Global Settings

- Form Attrit
- Form Inter
- Global Def

Pages and Windows

%PAGE1

Form

ZAR_SMARTFORM

Description

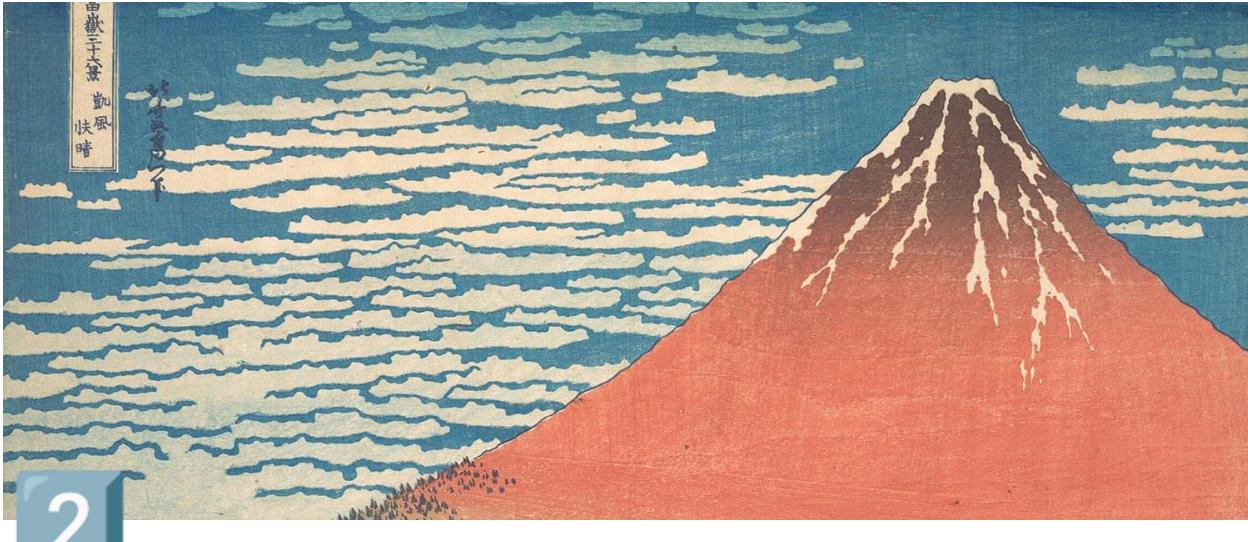
New Form

Global Data Types Field Symbols Initialization Form Routines

Variable Name	Typing	Associated Type
NAME		
TYPE		

- In **global definition** what ever action we perform (for e.g. writing codes, defining variable etc.), these things are **accessible through out the entire smartform.**

Amrit Raj



Designing of Smartform Layout

- Suppose, I want to design a smartform like below.

Employee Details			
<i>Employee Id</i>			
<i>Employee</i>			
Employee Id	103	Employee Name	ABHISHEK YADAV
Manager	AMRIT RAJ	Salary	10000,00
<i>Project Details</i>			
Employee Id	Project Id	Project Name	
103	100	SAP ABAP DEVELOPMENT	
103	200	SAP ABAP DEVELOPMENT	
103	300	SAP ABAP DEVELOPMENT	

- The most important thing while designing a smartform is that we should decide how many sections are there and what action we need to perform on that.
 - So, when I look in the above smartform, What basically I see is that we have 6 sections in our smartform.
-

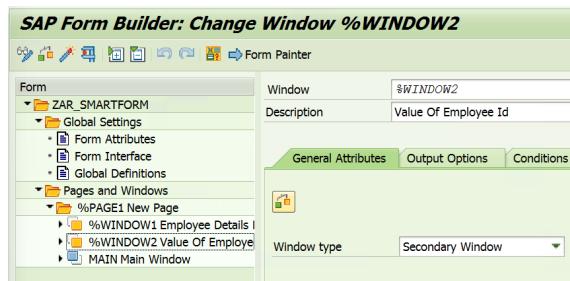
1. Windows in Smartforms

- For every section, we will create a window.
 - There's 6 windows for 6 sections.

1. Window 1 :- For Heading of our smartform



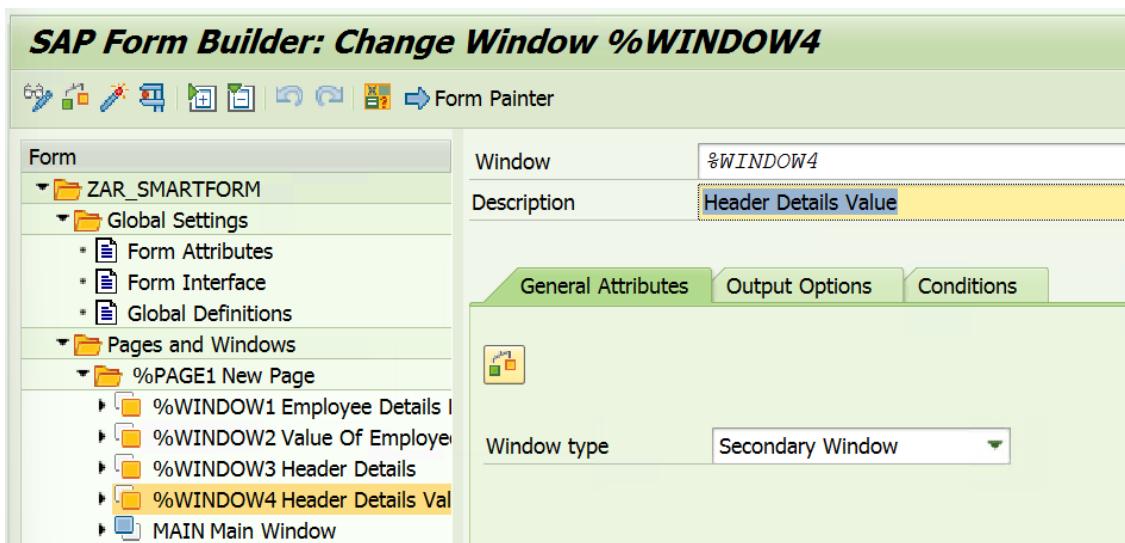
2. For Second Window, I want to display Employee Id and its value



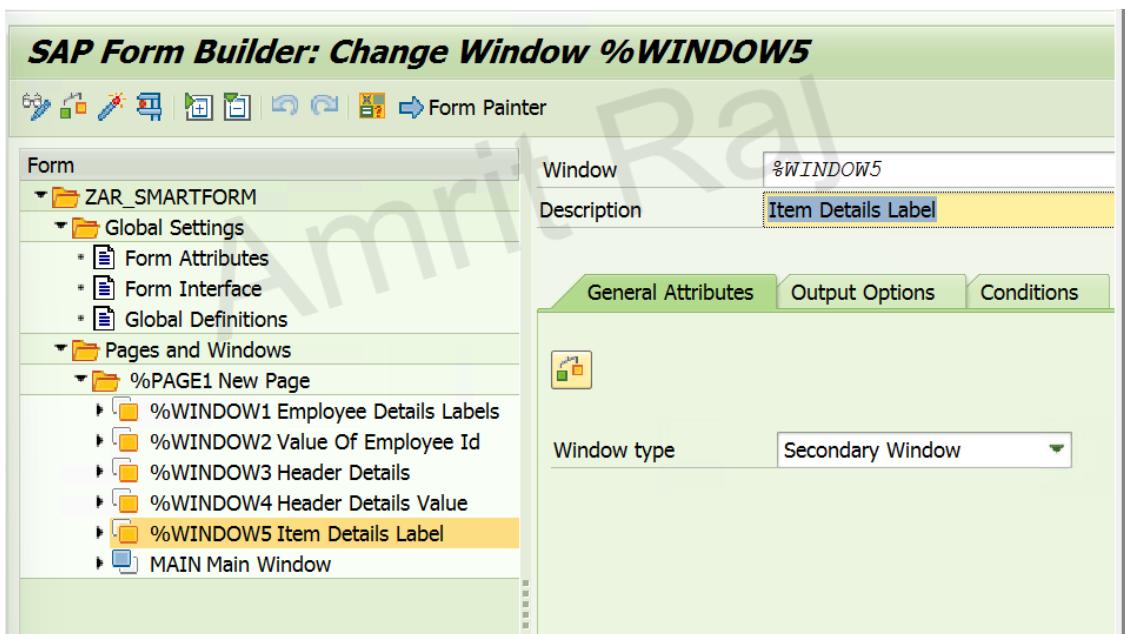
3. For Third Window, I have to display, the Header details Label from Employee Table



4. For fourth window, We will display Header Details from our employee table in form of template.



5. In the 5th window, I will display the Project details label.



6. Our 6th window is the main window, where I will display multiple data from my item table which is (Project Details for Employees)

2. Dimension of Windows

- We generally have 4 parts in the dimensions of our window in any smartform.

SAP Form Builder: Change Window %WINDOW1

Window	%WINDOW1				
Description	Employee Details Labels				
<input checked="" type="radio"/> General Attributes <input type="radio"/> Output Options <input type="radio"/> Conditions					
Position and Size					
Left Margin	5,00	CM	Width	5,00	CM
Top Margin	1,00	CM	Height	2,00	CM
<input type="checkbox"/> Box and Shading					

1. Left Margin :- what left margin I want to give
 2. Width :- Width of any window
 3. Top Margin :- Margin from top
 4. Height :- Height for the Window
-
- So, we will describe height for our each window.

SAP Form Builder: Change Window %WINDOW2

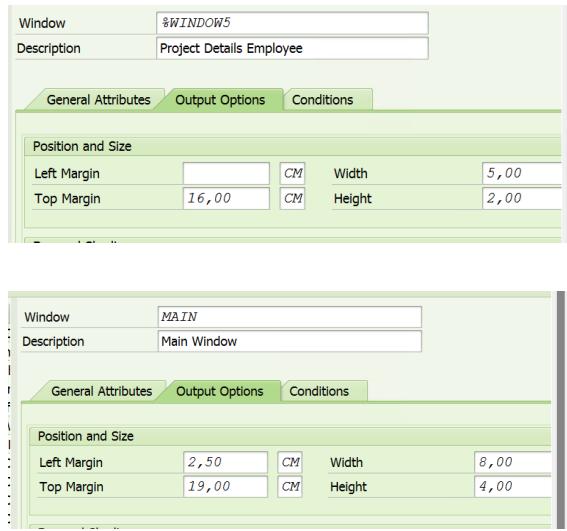
Window	%WINDOW2			
Description	Value Of Employee Id			
<input checked="" type="radio"/> General Attributes <input type="radio"/> Output Options <input type="radio"/> Conditions				
Position and Size				
Left Margin		CM	Width	8,00
Top Margin	4,00	CM	Height	2,00

SAP Form Builder: Change Window %WINDOW3

Window	%WINDOW3			
Description	Employee Details Label			
<input checked="" type="radio"/> General Attributes <input type="radio"/> Output Options <input type="radio"/> Conditions				
Position and Size				
Left Margin		CM	Width	5,00
Top Margin	7,00	CM	Height	2,00

SAP Form Builder: Change Window %WINDOW4

Window	%WINDOW4			
Description	Employee Details Value for Emp			
<input checked="" type="radio"/> General Attributes <input type="radio"/> Output Options <input type="radio"/> Conditions				
Position and Size				
Left Margin		CM	Width	10,00
Top Margin	10,00	CM	Height	5,00



3. Static Data vs Dynamic Data

- Since, we already know that in our header table we only have single data and in our item table we have multiple data for each Employee Id.

1. Template

- Template is used to display the static data which is fixed.
- We already know that we have fixed amount of data in Employee Table, so we will use template for it.

2. Table

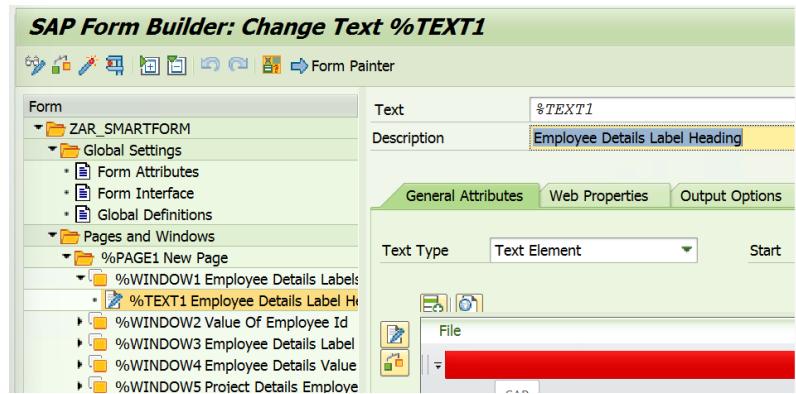
- Table is used to display the dynamic data.
- Since, We don't know how many data can be there in our item table, so for that purpose we will use Table to display the dynamic data.

4. Texts in Smartforms

- Texts are used to display the data on the screen of smartform.

How to use text ?

- Right click on the window and select text.



Note :-

- While creating text, you might see that MS Office gets opened, you can use it for writing your text, or you can use Change Editor to write your texts.
 - Click on goto → Change Editor

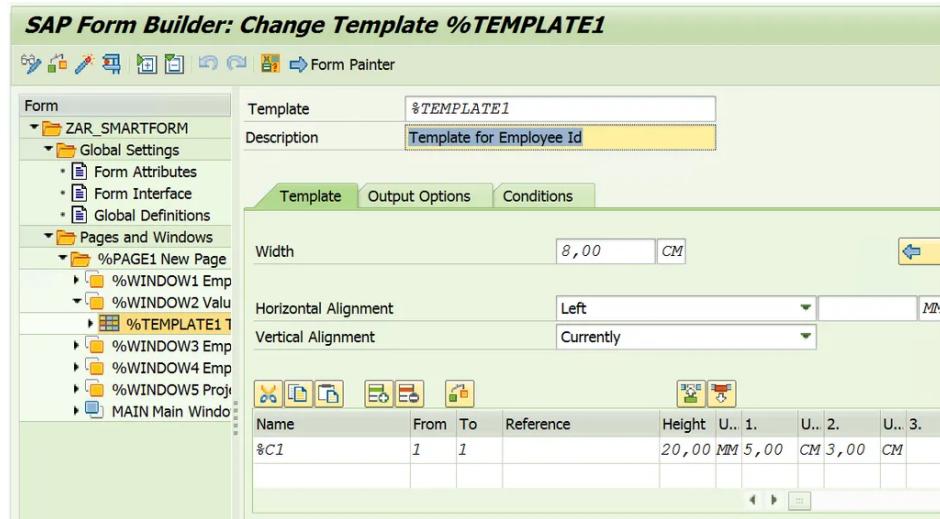


For Window 2 :-

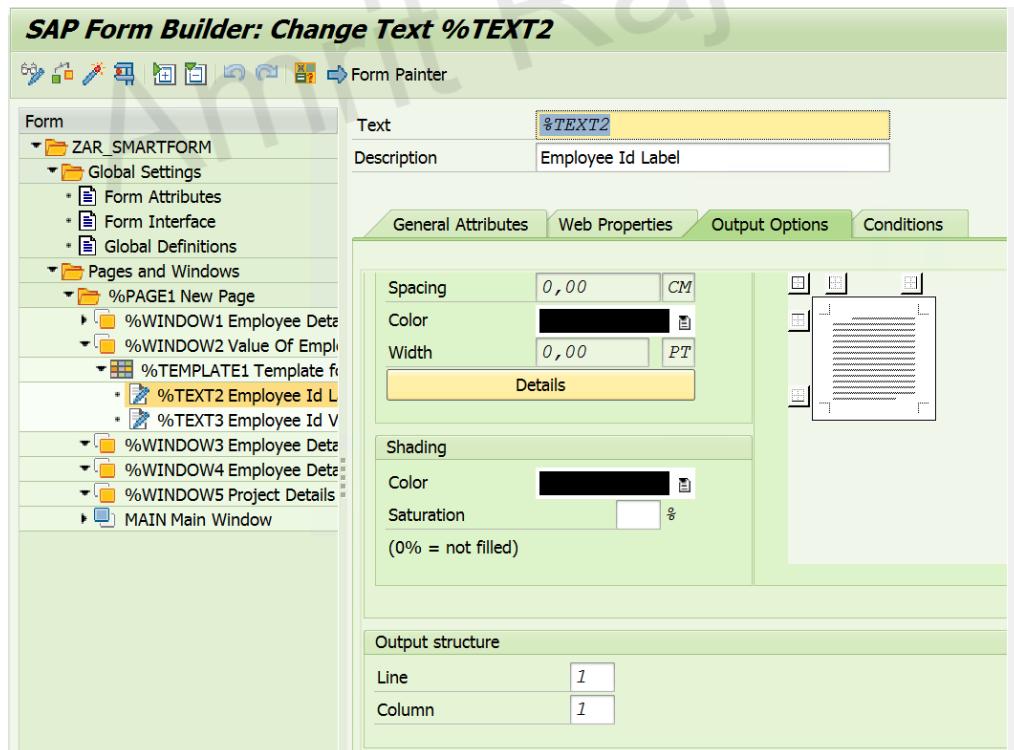
- We need to give employee Id and value of Employee Id.
- So for static data we will have to create a template.

Template :-

- Template is used for fixed number of rows and columns.
- Template is for static data or fixed data.
- For Template it, is mandatory that the size of the template should not exceed the size of the window.
- So, I will right click on the window and will create a template.



- Since, the width of my window 2 was 8 cm, so I am dividing into two columns for storing employee id label and its value.
- Now, right click on the template and create two texts for these two columns of template.

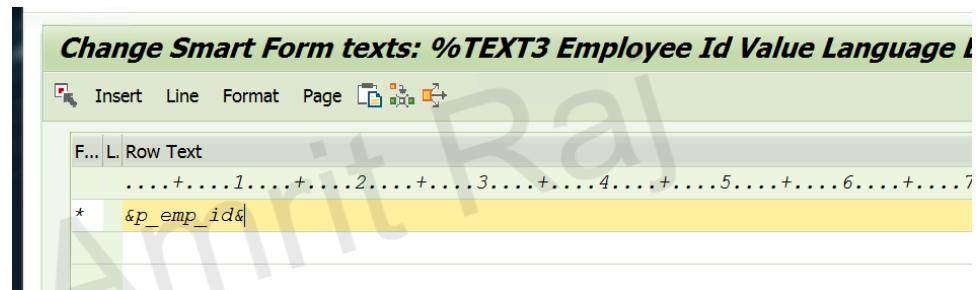


- For first text pass row and column as 1 and 1 in the Output structure.

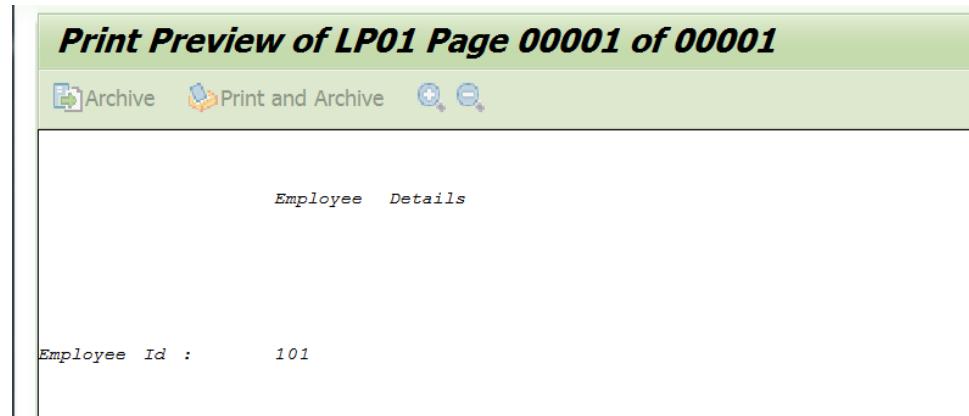
- and for the second we will pass 1 and 2 as columns
- Now, For Employee Id Value we will ne taking it as an input from the user, so we will import it from the FORM interface tab.

Parameter Name	Type Assignment	Associated Type
ARCHIVE_INDEX	TYPE	TOA_DARA
ARCHIVE_INDEX_TAB	TYPE	TSFDARA
ARCHIVE_PARAMETERS	TYPE	ARC_PARAMS
CONTROL_PARAMETERS	TYPE	SSFCTRIOP
MAIL_APPL_OBJ	TYPE	SWTOBJID
MAIL_RECIPIENT	TYPE	SWTOBJID
MAIL_SENDER	TYPE	SWTOBJID
OUTPUT_OPTIONS	TYPE	SSFCOMPPOP
USER_SETTINGS	TYPE	TDBBOOL
P_EMP_ID	TYPE	ZAR_EMPLOYEE_ID

- now pass it to the second text of the template of window 2.

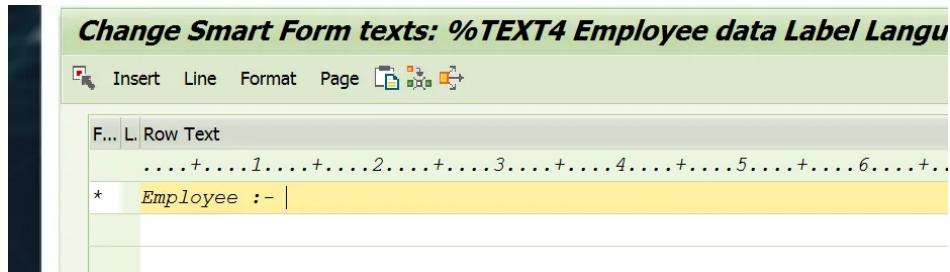


- Now, if i try to execute my smartform for 101 employee id, it should be displayed like below.



For Window 3 :-

- We just need to give a text for Employee Data Label.

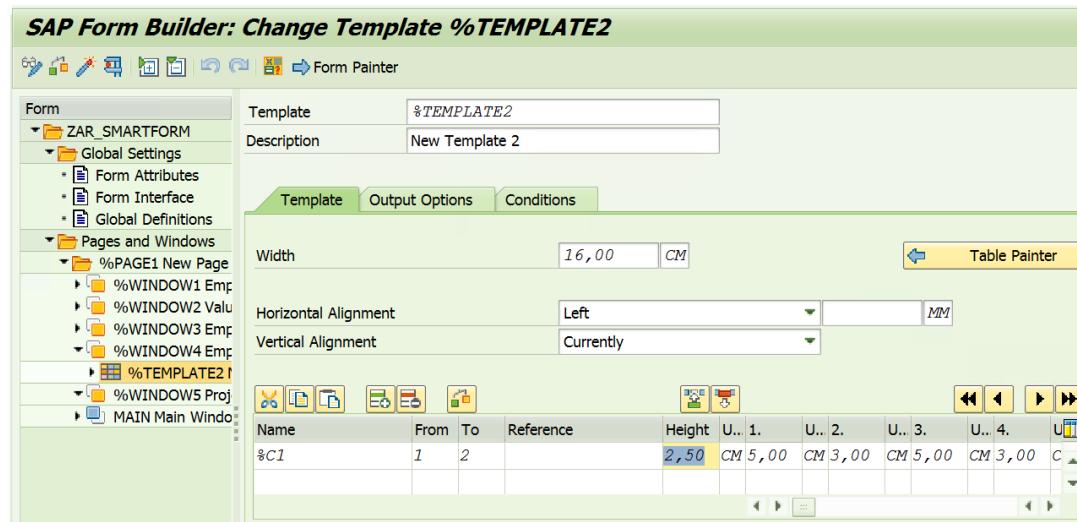


For Window 4 :-

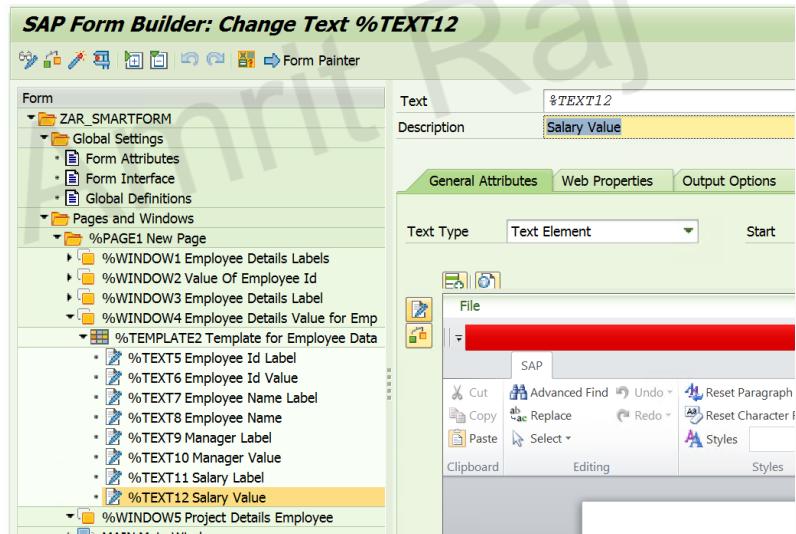
- We will display Employee data which is fixed so we will create a template.
- As discussed earlier template dimensions cannot exceed the window dimensions. now for the employee data.

Field	Key	Initi...	Data element	Data Type	Length	Decima...	Coordinate	Short
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZAR_MANDT	CLNT	3	0		0Clien
EMP_ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZAR_EMPLOYEE_ID	CHAR	3	0		0Empl
EMP_NAME	<input type="checkbox"/>	<input type="checkbox"/>	ZAR_EMPLOYEE_NAM	CHAR	40	0		0Empl
DEPARTMENT	<input type="checkbox"/>	<input type="checkbox"/>	ZAR_DEPARTMENT	CHAR	40	0		0Depa
MANAGER	<input type="checkbox"/>	<input type="checkbox"/>	ZAR_MANAGER	CHAR	30	0		0Man
SALARY	<input type="checkbox"/>	<input type="checkbox"/>	ZAR_SALARY	CURR	7	2		0Salar

- Now suppose for, employee data, we want to display Employee id, employee name, manager and salary.
- So we will use four columns and two rows in each row we will have their label and value for 2 fields.



- Here, we will have 2 rows.
- Now we will create total 8 texts in this template for four fields of employee table.

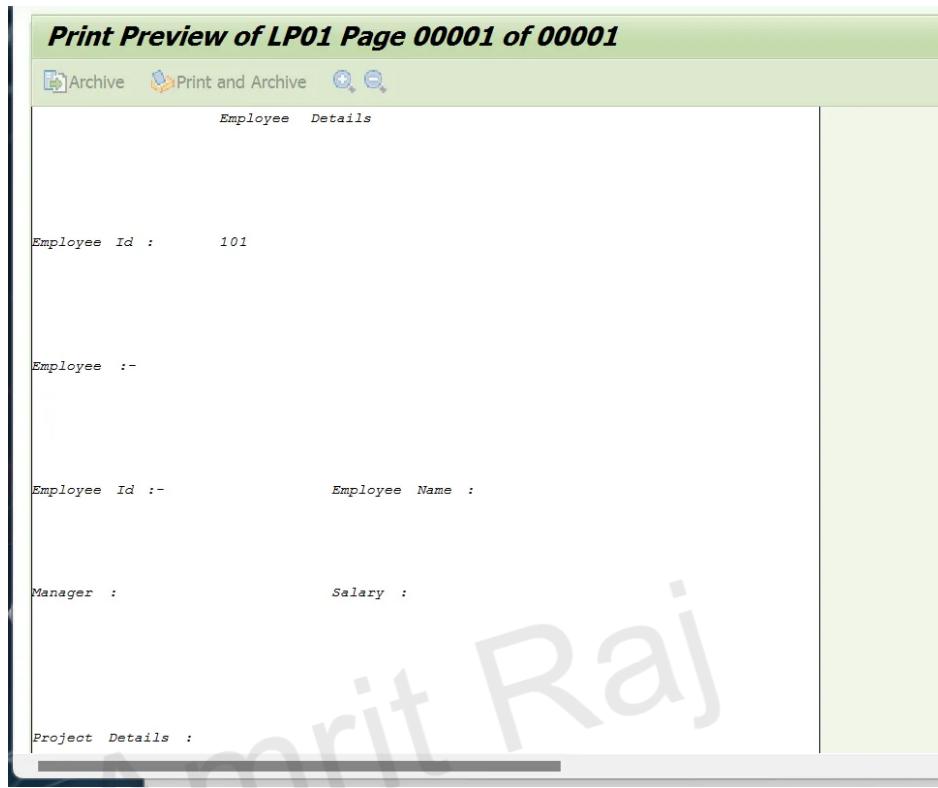


- So, now our data is in the custom employee table and we need to bring data from table to the smartform.
- we will do it at last, firstly we need to create text for our all windows.

For Window 5

- Create a text for Project details heading.

- Now, if I try to execute my smartform, I can see my smartform will be looking like below,

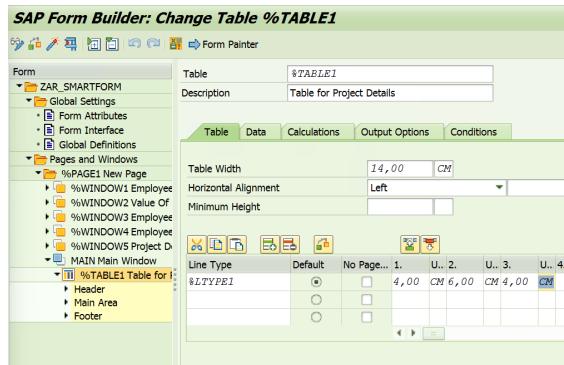


For Main Window :-

- I have to display the entire project details for the corresponding employees.
- Suppose, I will be displaying, Employee Id, project id, and Project Name for the below data.

Transparent Table		ZAR_PROJECT_DET	Active				
Short Description		Project Details					
		Attributes	Delivery and Maintenance				
		Search	Built-In Type				
Field	Key	Initi...	Data element	Data Type	Length	Decimal...	C
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	
EMP_ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZAR_EMP_ID	CHAR	3	0	
PROJECT_ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZAR_PROJECT_ID	CHAR	3	0	
EMP_NAME	<input type="checkbox"/>	<input type="checkbox"/>	ZAR_EMP_NAME	CHAR	40	0	
PROJECT_NAME	<input type="checkbox"/>	<input type="checkbox"/>	ZAR_PROJECT_NAME	CHAR	40	0	

- So, for multiple data, I need to use Table for that, So, I will create a table in the main window with 3 columns.



- Now, we will pass the header in table, in header I will create texts to display the labels for table data.
 - i.e. 3 labels.
 - Create table line and corresponding texts.

5. Writing Logic for Smartform

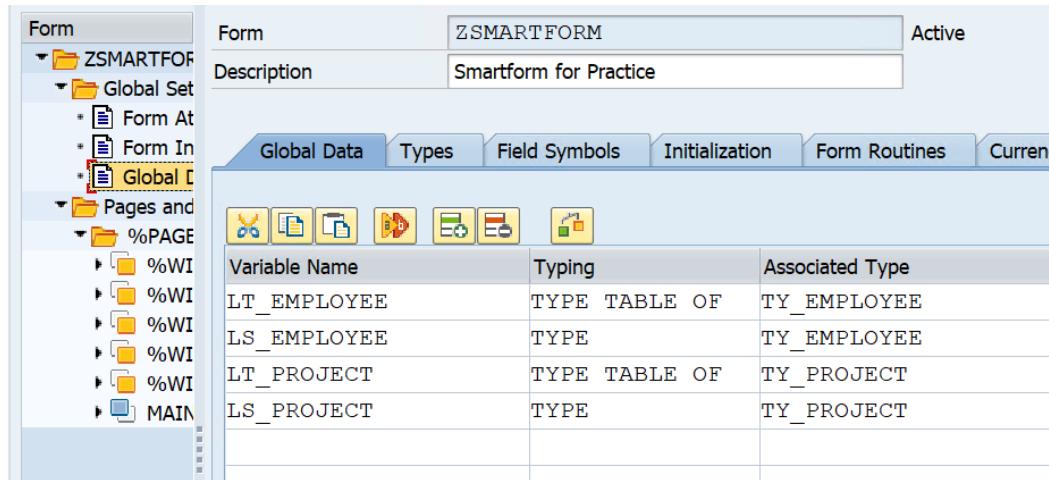
- Step 1 :- Go to the Global definition and in the Types Tab we will create Type Structure for both our Employee and Project Tables.

```

1  TYPES : BEGIN OF TY_EMPLOYEE,
2    EMP_ID      TYPE ZAR_EMP_ID,
3    EMP_NAME    TYPE ZAR_EMP_NAME,
4    MANAGER     TYPE ZAR_MANAGER,
5    SALARY      TYPE ZAR_SALARY,
6  END OF TY_EMPLOYEE.
7
8  TYPES : BEGIN OF TY_PROJECT,
9    EMP_ID      TYPE ZAR_EMP_ID,
10   PROJECT_ID  TYPE ZAR_PROJECT_ID,
11   PROJECT_NAME TYPE ZAR_PROJECT_NAME,
12 END OF TY_PROJECT.
13
14

```

- Step 2 :- In Global data, We will create internal table and work area for both the structures.



- Step 3 :- In Initialization Tab, We will write the logic.

The screenshot shows the SAP Smartform interface with the 'Initialization' tab selected. The code area contains the following SQL logic:

```

1 Select EMP_ID EMP_NAME MANAGER SALARY
2   from ZAR_EMP_TAB
3   into table lt_employee
4   where EMP_ID = P_EMP_ID.
5
6 IF LT_EMPLOYEE is NOT INITIAL.
7   Select EMP_ID PROJECT_ID PROJECT_NAME
8     from ZAR_PROJECT_DET
9     into table LT_PROJECT
10    FOR ALL ENTRIES IN LT_EMPLOYEE
11      where EMP_ID = LT_EMPLOYEE-EMP_ID.
12
13 Read TABLE lt_employee into ls_employee INDEX 1.
14
15 ENDIF.

```

6. Passing data to Windows of Smartform

- Step 1 :- Simply go to the 4th window and I will pass the employee details into the texts of the template.

Change Smart Form texts: %TEXT10 Manager Value Language EN

F... L. Row Text	
....+....1....+....2....+....3....	
*	&ls_employee-EMP_ID&

Change Smart Form texts: %TEXT10 Manager Value Language EN

F... L. Row Text	
....+....1....+....2....+....3....+....4....+....5....+....6....	
*	&ls_Employee-MANAGER&

Change Smart Form texts: %TEXT12 Salary Value Language EN

F... L. Row Text	
....+....1....+....2....+....3....+....4....+....5....+....6....+....7.	
*	&ls_Employee-SALARY&

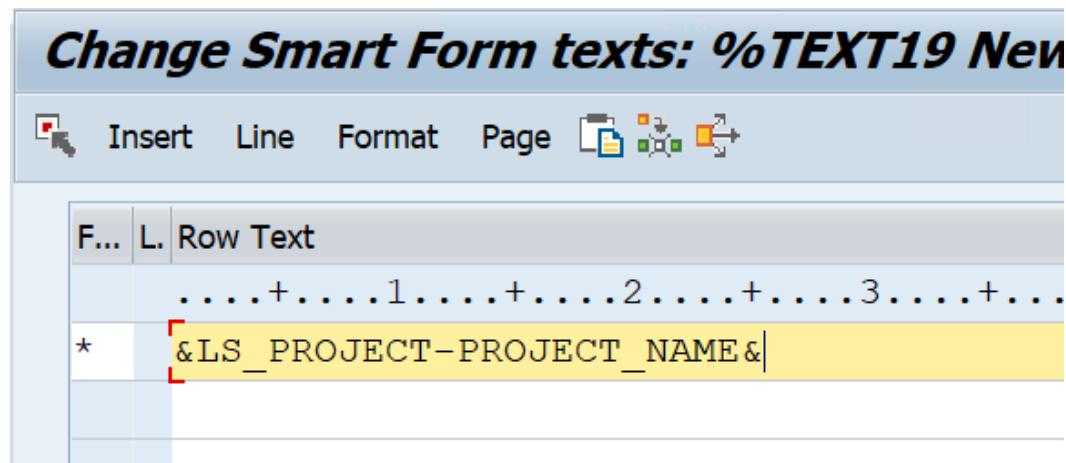
- Step 2 :- For Project Details, We will go to main area of our table and we will create texts accordingly and pass the data.

Change Smart Form texts: %TEXT17 New 1

F... L. Row Text	
....+....1....+....2....+....3....+....4	
*	&LS_PROJECT-EMP_ID&

Change Smart Form texts: %TEXT18 New 1

F... L. Row Text	
....+....1....+....2....+....3....+...	
*	&LS_PROJECT-PROJECT_ID&



7. Execute the Smartform

Employee Details

Employee Id 104

Employee

Employee Id	104	Employee Name	MUDITA GUPTA
Manager	AMRIT RAJ	Salary	10000,00

Project Details

Print Preview of LP01 Page 00001 of 00001

Archive

Print and Archive



Employee Id	104	Employee Name	MUDITA GUPTA
Manager	AMRIT RAJ	Salary	10000,00

Project Details

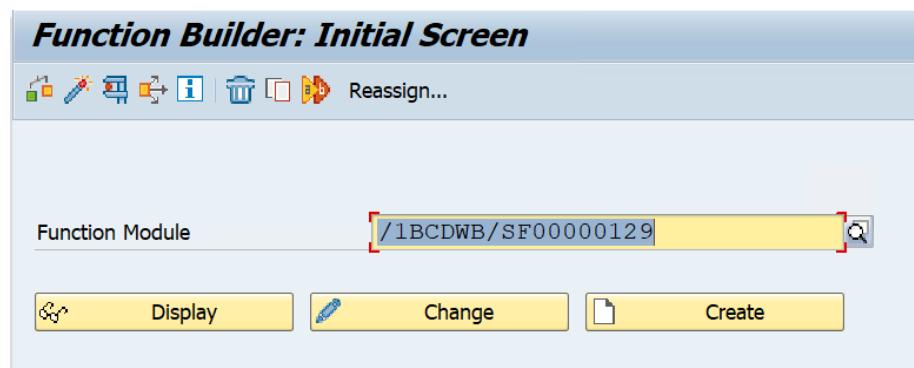
Employee Id	Project Id	Project Name
104	100	SAP ABAP DEVELOPMENT
104	200	SAP ABAP DEVELOPMENT
104	300	SAP ABAP DEVELOPMENT



Calling Smartform into Report (SE38)

1. Introduction

- Whenever we execute our Smartform, It directly navigates to SE37(Function Builder).



- For Every Smartform, SAP will automatically generate a function module at the runtime.

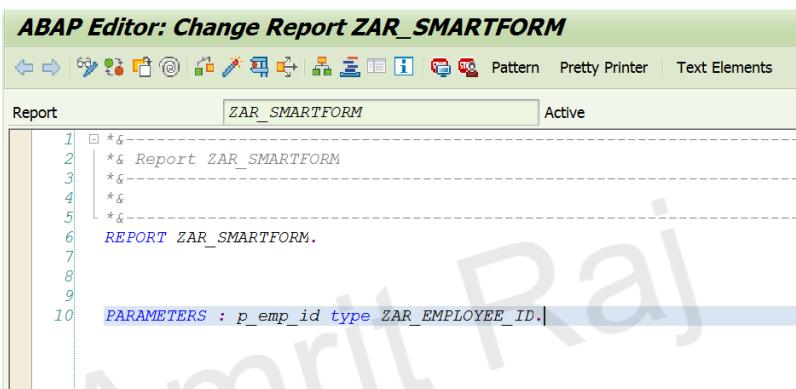
Note :-

- If you change the contents of the smartform, then it is possible that the name of the function module may change at the runtime, so it is not a good practice to use the same function module of the smartform every time.
-

2. Calling Smartform into the Program

- Step 1 :- Create a executable program in ABAP Editor (SE38) transaction code.
- Step 2 :- Since, we are passing employee id as an input to our smartform so, for that we will create a parameter for the same.

ABAP Editor: Change Report ZAR_SMARTFORM



```

Report      ZAR_SMARTFORM      Active
1  *->-
2  *-> Report ZAR_SMARTFORM
3  *->-
4  *->-
5  *->-
6  REPORT ZAR_SMARTFORM.
7
8
9
10 PARAMETERS : p_emp_id type ZAR_EMPLOYEE_ID.

```

- Step 3 :- Since, we have already discussed that for our smartform, SAP automatically creates a function module, so we can use it directly in our program.

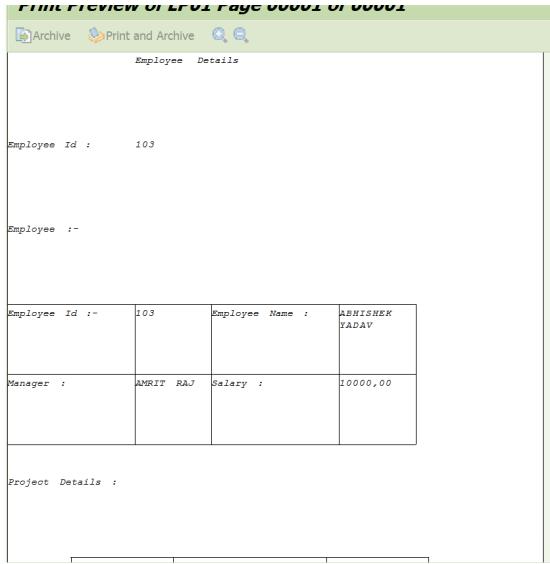
ABAP Editor: Change Report ZAR_SMARTFORM

Report ZAR_SMARTFORM Active

```
7
8
9
10 PARAMETERS : p_emp_id type ZAR_EMPLOYEE_ID.
11
12 CALL FUNCTION '/1BCDWB/SF00000176'
13   EXPORTING
14     * ARCHIVE_INDEX           =
15     * ARCHIVE_INDEX_TAB       =
16     * ARCHIVE_PARAMETERS      =
17     * CONTROL_PARAMETERS      =
18     * MAIL_APPL_OBJ          =
19     * MAIL_RECIPIENT          =
20     * MAIL_SENDER             =
21     * OUTPUT_OPTIONS          =
22     * USER_SETTINGS            = 'X'
23     p_emp_id                 = p_emp_id
24   IMPORTING
25     * DOCUMENT_OUTPUT_INFO    =
26     * JOB_OUTPUT_INFO         =
27     * JOB_OUTPUT_OPTIONS      =
28   EXCEPTIONS
29     FORMATTING_ERROR          = 1
30     INTERNAL_ERROR            = 2
31     SEND_ERROR                = 3
32     USER_CANCELED             = 4
33     OTHERS                     = 5
34
```

Execute the Program :-

- Press F8 to execute the code and pass employee number = 103 as input and press f8.



How can we know the name of our function module using the name of the smartform ?

- It is possible that someone can ask you that how can I know the name of my function module, if I have the name of smartform.
- For knowing the name of the function module, we use

SSF_FUNCTION_MODULE_NAME

```

14  -----
15 CALL FUNCTION 'SSF_FUNCTION_MODULE_NAME'
16   EXPORTING
17     formname          = 'ZAR_SMARTFORM'
18   * VARIANT          = '
19   * DIRECT_CALL      = '
20   IMPORTING
21     FM_NAME           = lv_function_module
22   EXCEPTIONS
23     NO_FORM           = 1
24     NO_FUNCTION_MODULE = 2
25     OTHERS             = 3
26   .
27

```

- Here, using FM_NAME we can import the name of our function module.
- and we can pass it instead of the function module for our smartform.

Code

```

PARAMETERS : p_emp_id type ZAR_EMP_ID.

DATA : lv_function_module type RS38L_FNAM.

CALL FUNCTION 'SSF_FUNCTION_MODULE_NAME'
  EXPORTING
    FORMNAME                = 'ZSMARTFORM'
  *  VARIANT                 = ''
  *  DIRECT_CALL              = ''
  IMPORTING
    FM_NAME                  = LV_FUNCTION_MODULE
EXCEPTIONS
  NO_FORM                   = 1
  NO_FUNCTION_MODULE        = 2
  OTHERS                     = 3

CALL FUNCTION LV_FUNCTION_MODULE
  EXPORTING
  *  ARCHIVE_INDEX            =
  *  ARCHIVE_INDEX_TAB         =
  *  ARCHIVE_PARAMETERS        =
  *  CONTROL_PARAMETERS        =
  *  MAIL_APPL_OBJ             =
  *  MAIL_RECIPIENT            =
  *  MAIL_SENDER                =
  *  OUTPUT_OPTIONS              =
  *  USER_SETTINGS               = 'X'
    P_EMP_ID                  = P_EMP_ID
  *  IMPORTING
  *  DOCUMENT_OUTPUT_INFO       =
  *  JOB_OUTPUT_INFO             =
  *  JOB_OUTPUT_OPTIONS          =
  EXCEPTIONS

```

```
FORMATTING_ERROR          = 1
INTERNAL_ERROR             = 2
SEND_ERROR                 = 3
USER_CANCELED              = 4
OTHERS                      = 5
```

Output

The screenshot shows a SAP ERP interface with a title bar "Employee Details". Below it, there's a section labeled "Employee" containing the text "Employee Id 104". A large watermark "Amrit Raj" is diagonally across the page. Below this, there's a table with two rows. The first row has four columns: "Employee Id" (104), "Employee Name" (MUDITA GUPTA), and two empty columns. The second row has four columns: "Manager" (AMRIT RAJ), "Salary" (10000,00), and two empty columns. At the bottom, there's a section labeled "Project Details".

Employee Id	104	Employee Name	MUDITA GUPTA
Manager	AMRIT RAJ	Salary	10000,00

Project Details

Employee Id	Project Id	Project Name
104	100	SAP ABAP DEVELOPMENT
104	200	SAP ABAP DEVELOPMENT
104	300	SAP ABAP DEVELOPMENT

Amrit Raj



4

Types of Windows in Smartforms

There are mainly four types of Window in Smartforms :-

1. Main Window :-

- Main window is the default window, which you will see each time you will create any smartform.
- Main window is used for the continuous output such as table output. That means when you have continuous flow of data you can use main window, It will automatically shift the data to next page, If the amount of data will increase.

2. Secondary Window :-

- This window is mainly used to display the output for a fixed length.
- For example, displaying output in form of template we can use secondary window.

Note :- We should never use secondary window, if we have continuous flow of data.

3. Copies Window :-

- It is also a kind of special secondary window which is used for making pages as copy or original.

4. Final Window :-

- This window will process at the last of your smartform.
- This window will get executed only when all your smartform process gets completed.

Amrit Raj



5

Types of Texts in Smartforms

1. Introduction

- Texts are used to display data on the smartform layout.

2. Types of Texts in Smartforms

1. Text Element

- They are not reusable texts.
- They are dedicated to one smartform only.
- That means, once we can use them in one smartform, we cannot reuse them again.

2. Text Module

- It is a reusable text.
- It is a reusable text for smartforms.

3. Include Text

- It is also a reusable text.
- But comparing it with text module, we can reuse it in smartforms, programs etc.

Transaction code for Include Text :-

- SO10.

Function Module to display Include Text :-

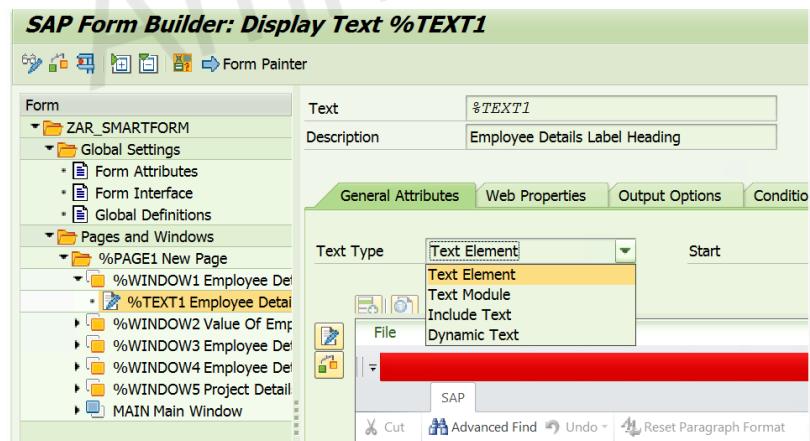
- READ_TEXT

4. Dynamic Texts

- This text is used to display the text at runtime in smartform.
- Pre - defined table type for Dynamic Texts - TSFTEXT

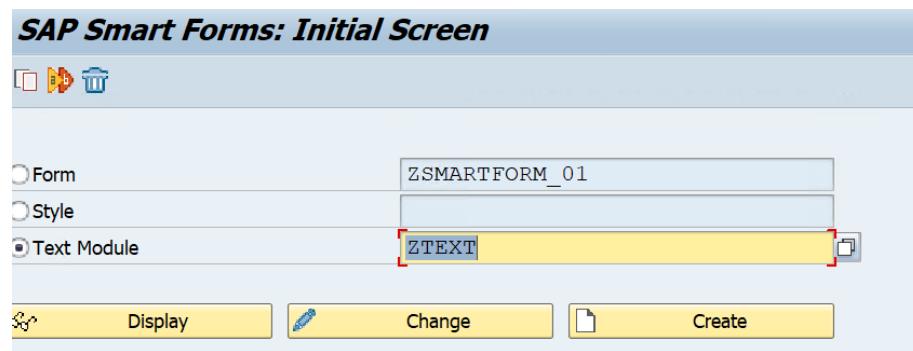
3. Implementation of Text Element

- We have already implemented text elements in the smartform which we have created.

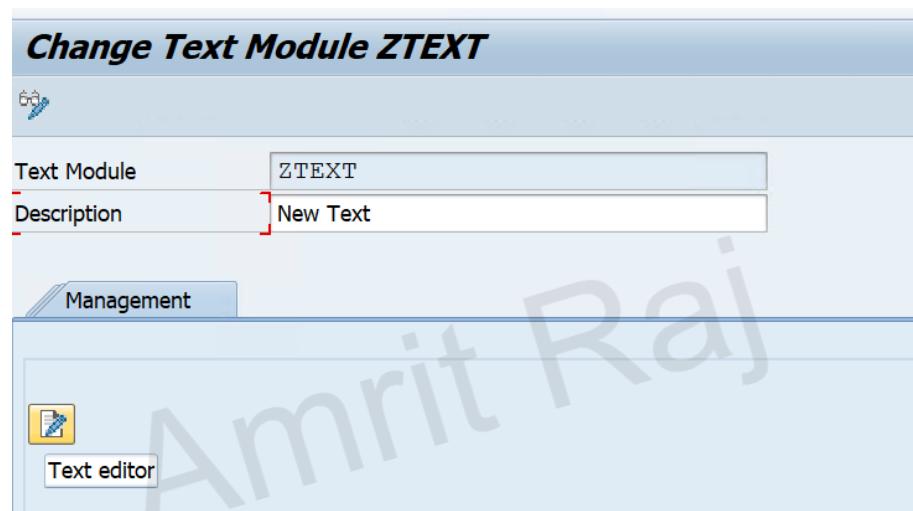


4. Implementation of Text Module

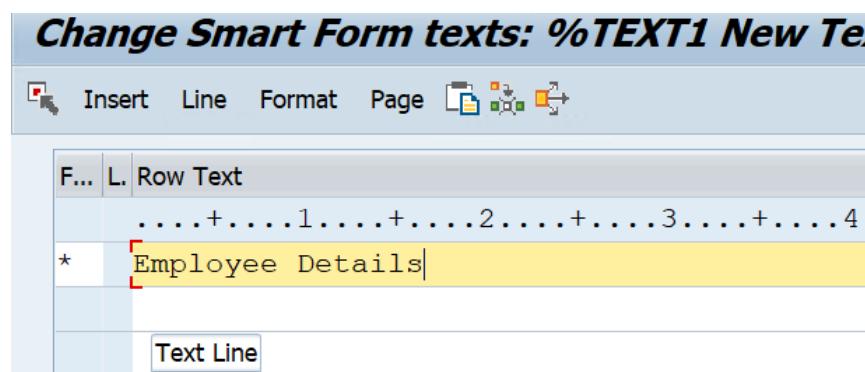
- The transaction code for Text Module is SMARTFORMS.



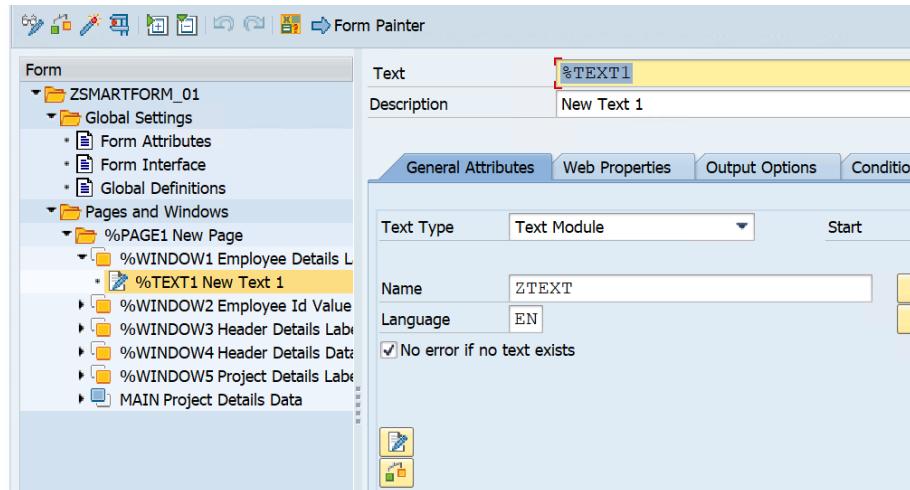
- Click on Create button



- Click on Text Editor and write the text.

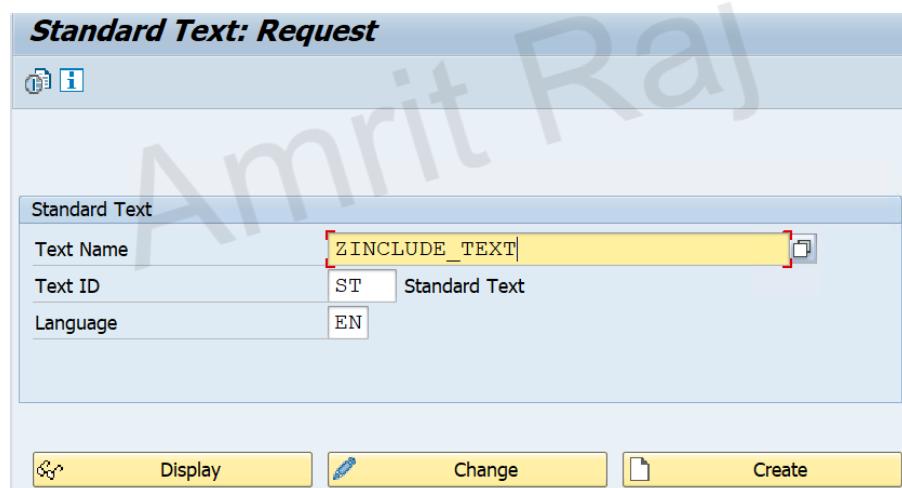


- Then, we can use it in our Smartform.

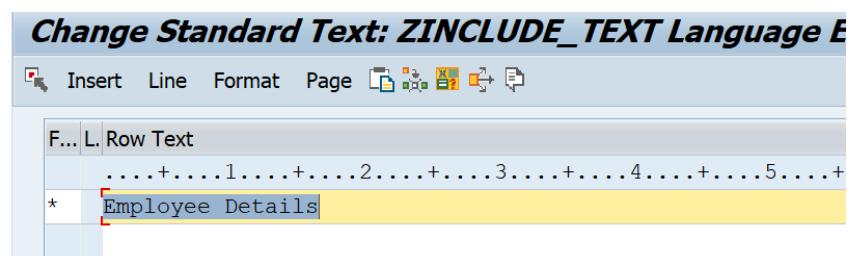


5. Implementation of Include Text

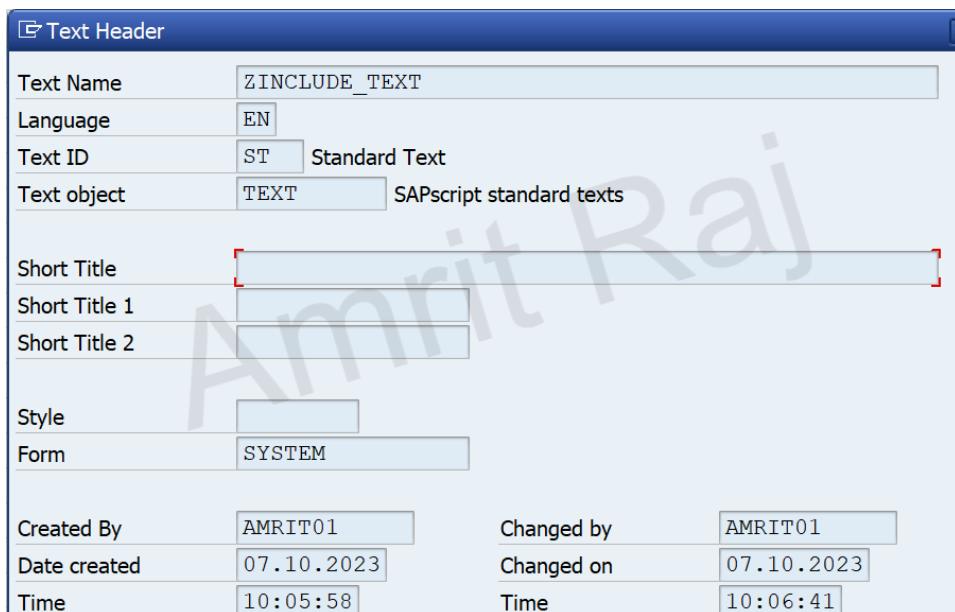
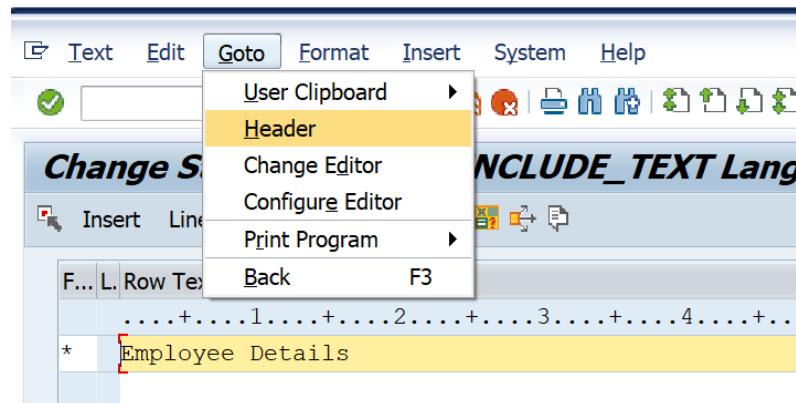
- Step 1 :- Go to SO10 transaction code and give a name for the text.



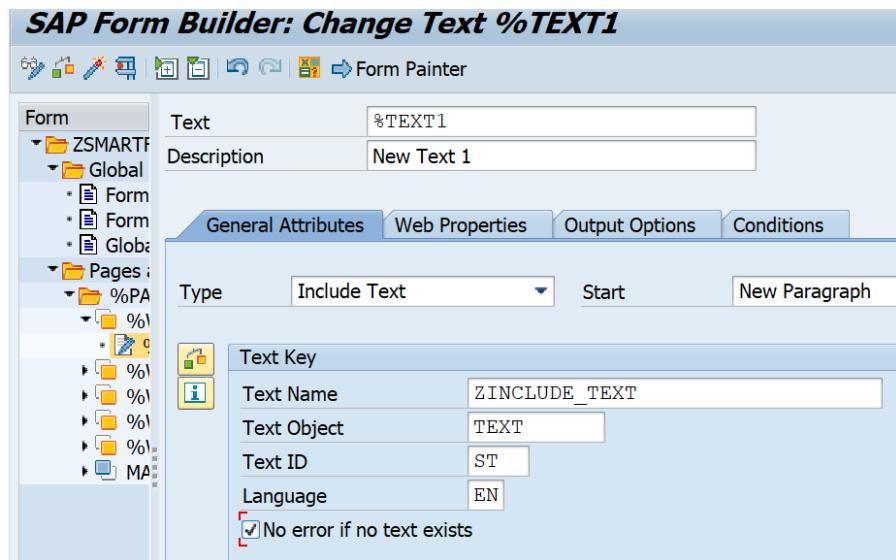
- Click on create button and write the text and click on save button.



- Click on GOTO → Header and you will get all the information of the data that you have created.



- Step 2 :- Now go to the smartform and use the include text.



Calling Include text in Program

- We can use READ_TEXT function module to display the include text.

```

DATA : LT_LINE type table of TLINE,
       LS_LINE LIKE LINE OF LT_LINE.

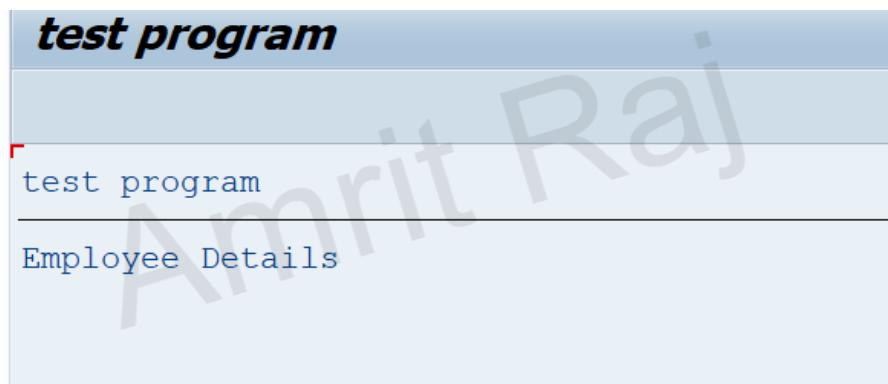
CALL FUNCTION 'READ_TEXT'
  EXPORTING
    *     CLIENT                  = SY-MANDT
    ID                      = 'ST'
    LANGUAGE                = SY-LANGU
    NAME                    = 'ZINCLUDE_TEXT'
    OBJECT                  = 'TEXT'
    *     ARCHIVE_HANDLE          = 0
    *     LOCAL_CAT               = ''
    *     IMPORTING
    *     HEADER
    *     OLD_LINE_COUNTER        =
  TABLES
    LINES                   = LT_LINE
  EXCEPTIONS
    ID                      = 1
    LANGUAGE                = 2
  
```

```
NAME          = 3
NOT_FOUND     = 4
OBJECT        = 5
REFERENCE_CHECK = 6
WRONG_ACCESS_TO_ARCHIVE = 7
OTHERS         = 8
```

```
LOOP AT LT_LINE into LS_LINE.
  WRITE :/ LS_LINE-TDLINE.

ENDLOOP.
```

Output



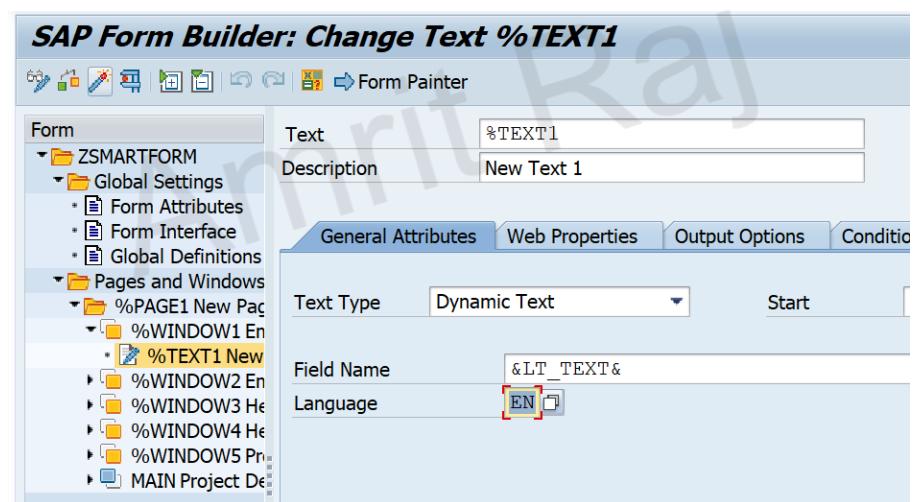
6. Implementation of Dynamic Text

- By default table type for Dynamic Text is TSFTEXT.
- Step 1 :- Go to Import tab of Form Interface and give a parameter of table type TSFTEXT.

The screenshot shows the SAP Smartform configuration interface. On the left, a tree view shows the form structure under 'ZSMARTFORM' with 'Global Settings' expanded, specifically 'Form Interface'. On the right, a table lists parameters with their type assignments and associated types. The row for 'LT_TEXT' is highlighted with a yellow background and has a red border around its 'Associated Type' field, which contains the value 'TSFTEXT'.

Parameter Name	Type Assignment	Associated Type
ARCHIVE_INDEX	TYPE	TOA_DARA
ARCHIVE_INDEX_TAB	TYPE	TSFDARA
ARCHIVE_PARAMETERS	TYPE	ARC_PARAMS
CONTROL_PARAMETERS	TYPE	SSFCTRLOP
MAIL_APPL_OBJ	TYPE	SWOTOBJID
MAIL_RECIPIENT	TYPE	SWOTOBJID
MAIL_SENDER	TYPE	SWOTOBJID
OUTPUT_OPTIONS	TYPE	SSFCOMPPOP
USER_SETTINGS	TYPE	TDBOOL
P_EMP_ID	TYPE	ZAR_EMP_ID
LT_TEXT	TYPE	TSFTEXT

- Step 2 :- Then, go to the text in which you want to use it and write the parameter name.



- Step 3 :- Activate and execute the smartform and write the text which you want to display.

Structure Editor: Change LT_TEXT from Entry

The screenshot shows the SAP Structure Editor interface. The title bar reads "Structure Editor: Change LT_TEXT from Entry". The toolbar includes icons for Save, Undo, Redo, Column, Entry, New Line, Double Line, and More. A message box at the top says "0 Entries". Below is a table structure:

TD	TDLINE
*	Employee Data

Amrit Raj



6

Displaying Logo/Graphics in Smartforms

1. Introduction

- Displaying graphics is one of the important features of Smartforms.
- For displaying graphics, either we can use the uploaded graphics, or we can upload them into server from our local system.
- The transaction code for uploading graphics into server is SE78.

Note :-

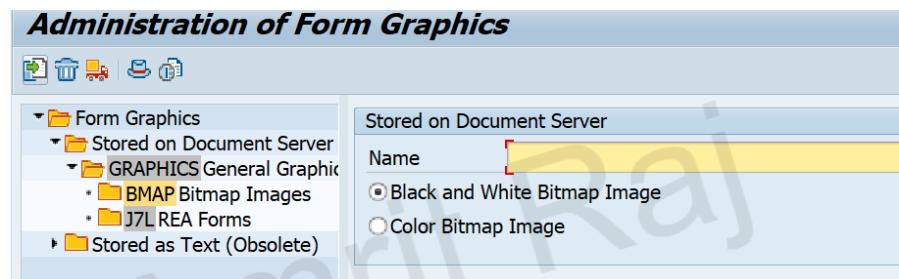
- We can only upload a bitmap image into SAP

2. Uploading BITMAP Image

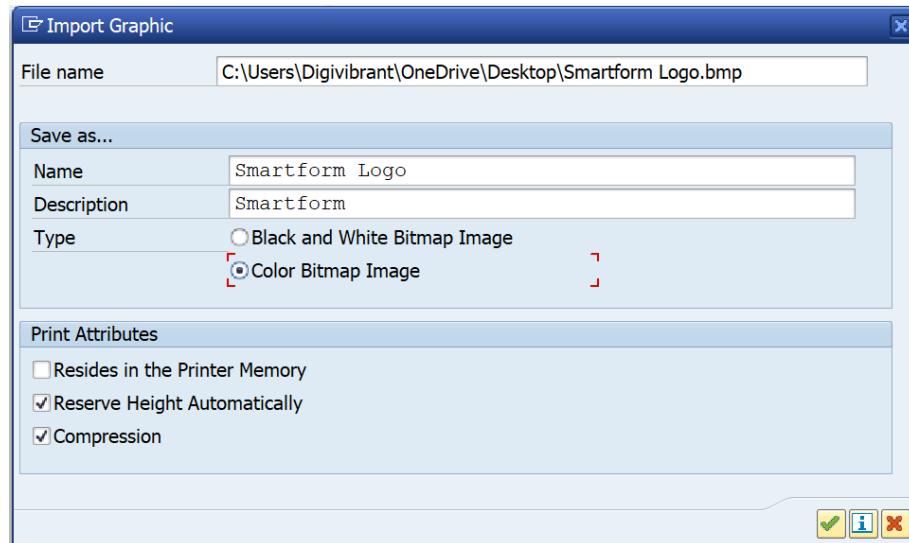
- Suppose, I have this image in bitmap format which I want upload into the server.



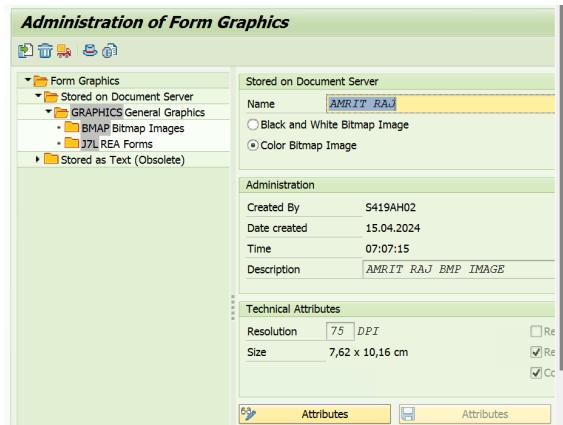
- Step 1 :- Go to SE78 transaction code.



- Step 2 :- Click on import button. It will navigate you to the local system, select the file.



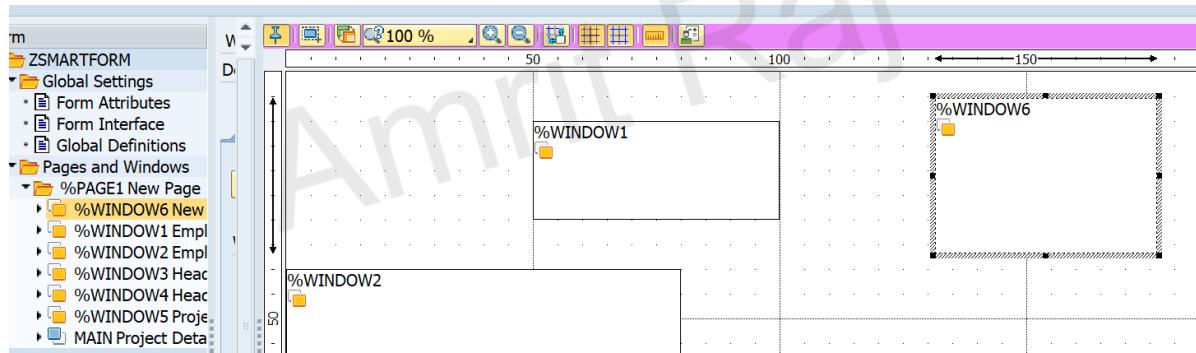
- Press enter.



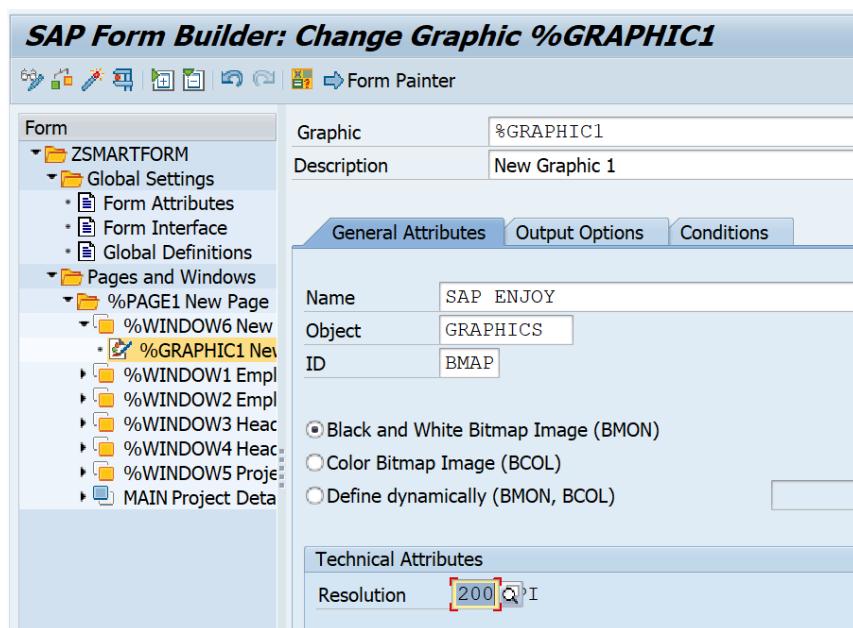
- You can check the preview by clicking on the print preview.

3. Adding Logo into our Smartform

- Step 1 :- Create a separate window for same purpose.



- Step 2 :- Right click on the window and create a graphic.
 - In graphic provide the name of the logo and provide the resolution accordingly.



- Step 3 :- Activate and execute the smartform, we can see that, our logo is added.

Print Preview of LP01 Page 00001 of 00001

Archive Print and Archive

Employee Details

Employee Id 105

SAP Logo

Employee

Employee Id	105	Employee Name	SHIKHAR SRIVASTAV
Manager	AMRIT RAJ	Salary	10000,00

Project Details

Amrit Raj

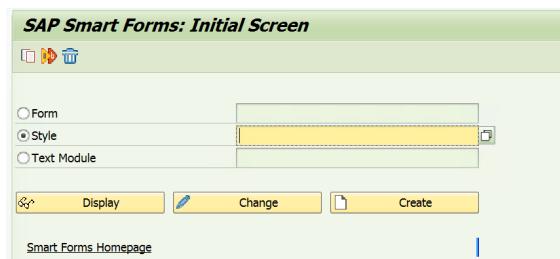


7

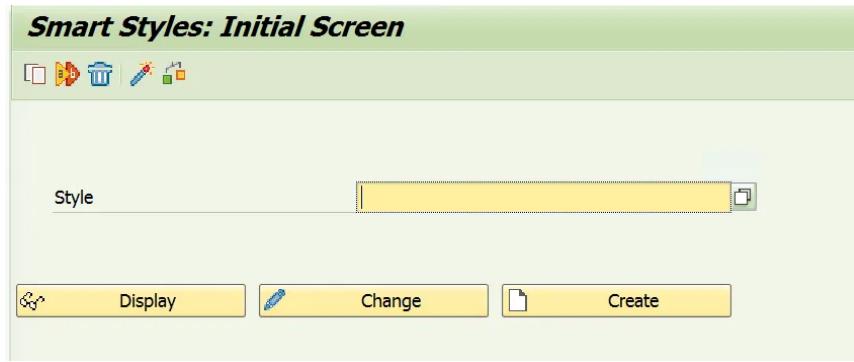
Creation of SmartStyles in Smartforms

1. Introduction

- Smartstyles are used to give proper font and alignment for our texts in smartforms.
- We can create smartstyles either by SMARTFORMS transaction code.



or by SMARTSTYLES transaction code.



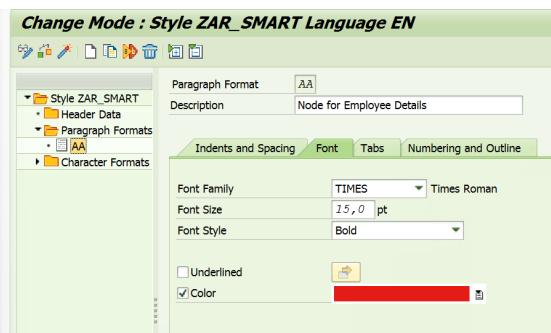
2. Creation of SmartStyles

1. Employee Details Red and bold

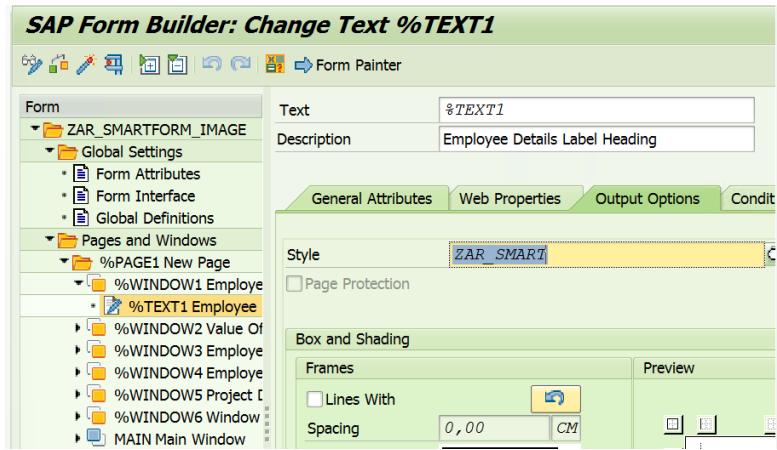
- Step 1 :- Go to SMARTSTYLES transaction code.



- Step 2 :- Right click on the paragraph format and click on create node.



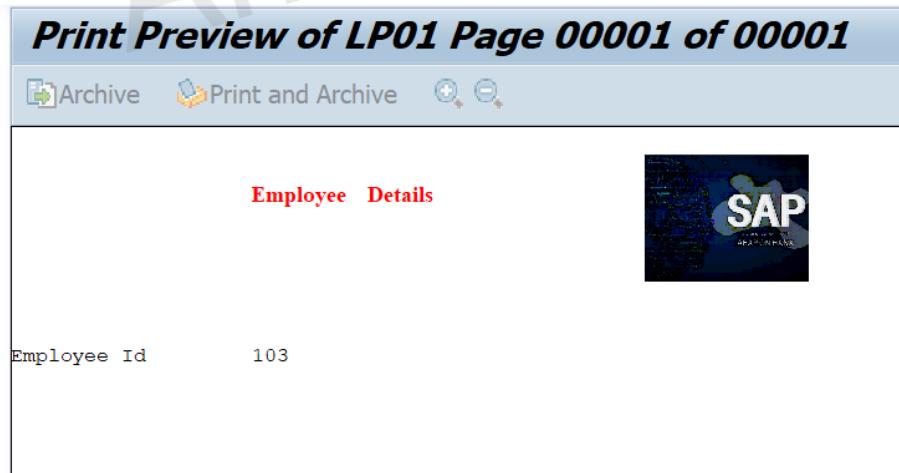
- Step 3 :- Activate the smartstyles and use it in the smartform.



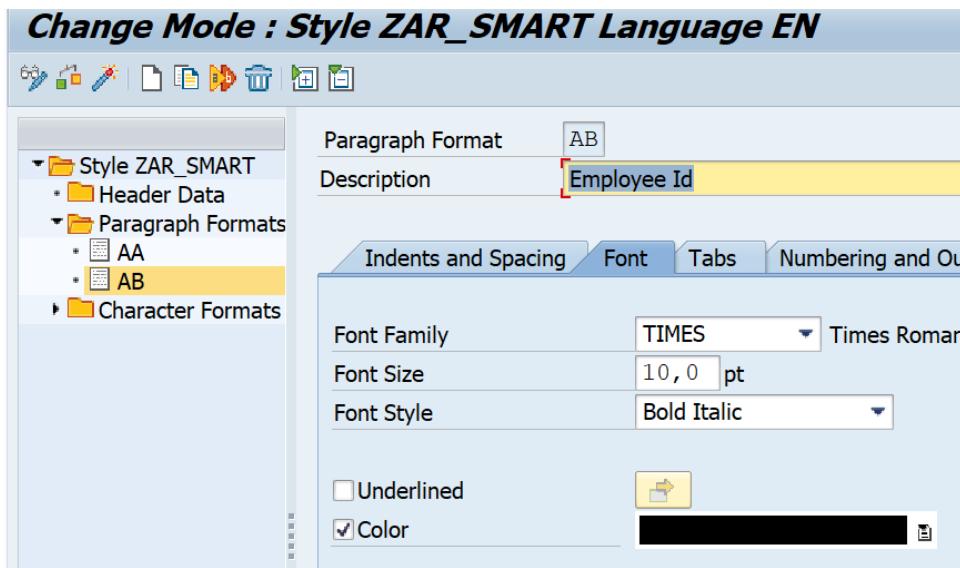
- Step 4 :- Use the Created node inside the text of the heading.



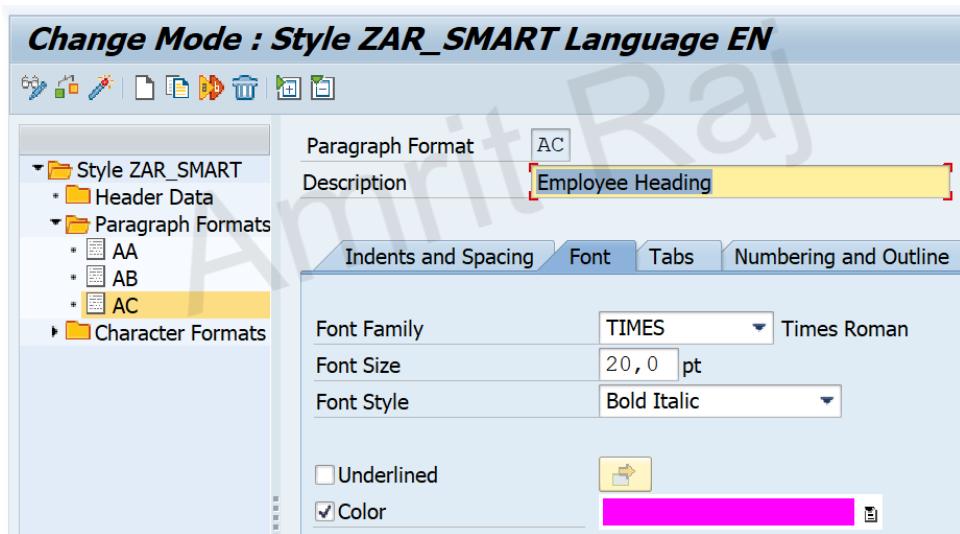
- Step 5 :- Now activate and execute the smartform.



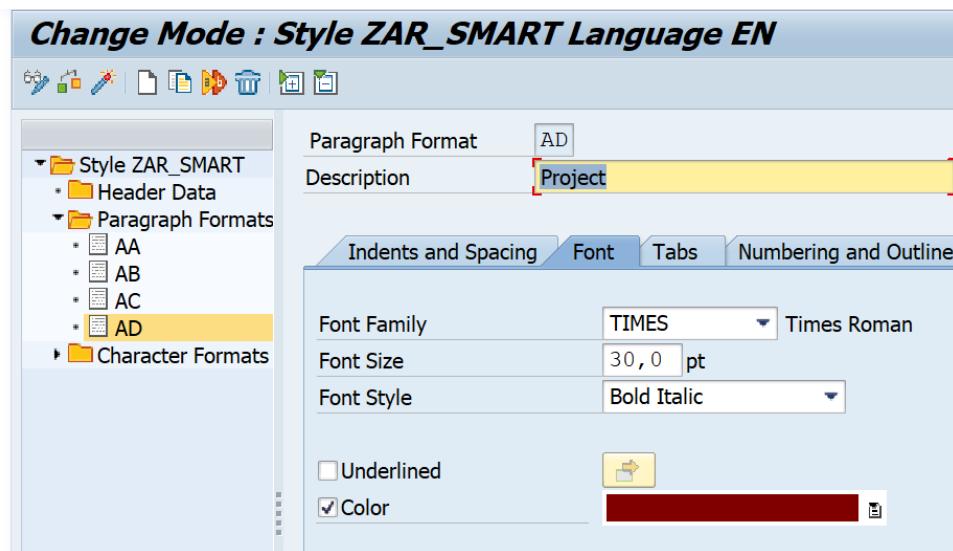
2. Black and Italic for Employee Id



3. Pink and Italic for Employee Heading



4. Brown and Bold and Italic for Project Details Heading



3. Output

Employee Details

Employee Id 101

Employee

Employee Id	101	Employee Name	SHIVAM SINGH
Manager	AMRIT RAJ	Salary	10000,00

Project Details



8

Creation of Bar Codes in Smartforms

- The Transaction code for Bar Code is SE73.

1. Creating Bar Code

- Step 1 :- Go to SE73 transaction code and Select System Bar Codes radio button.

SAPscript Font Maintenance: Initial Screen

Font Families System Fonts Printer Fonts System Bar Codes Printe

Selection

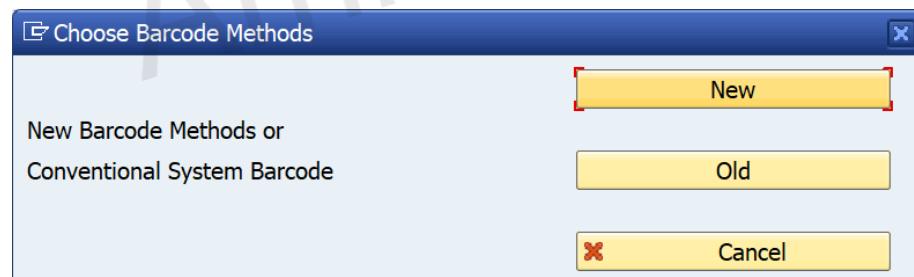
- Font Families
- System Fonts
- Printer Fonts
- System Bar Codes
- Printer Bar Codes

Display

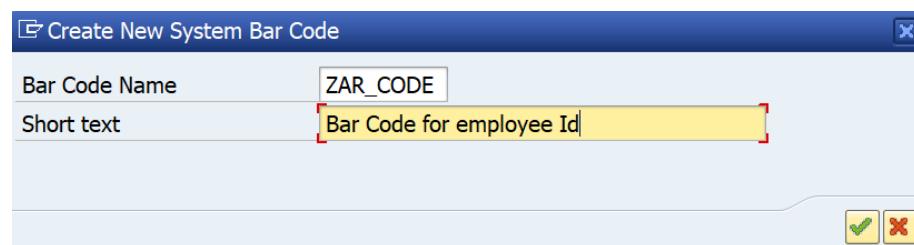
Change

Install TrueType Font

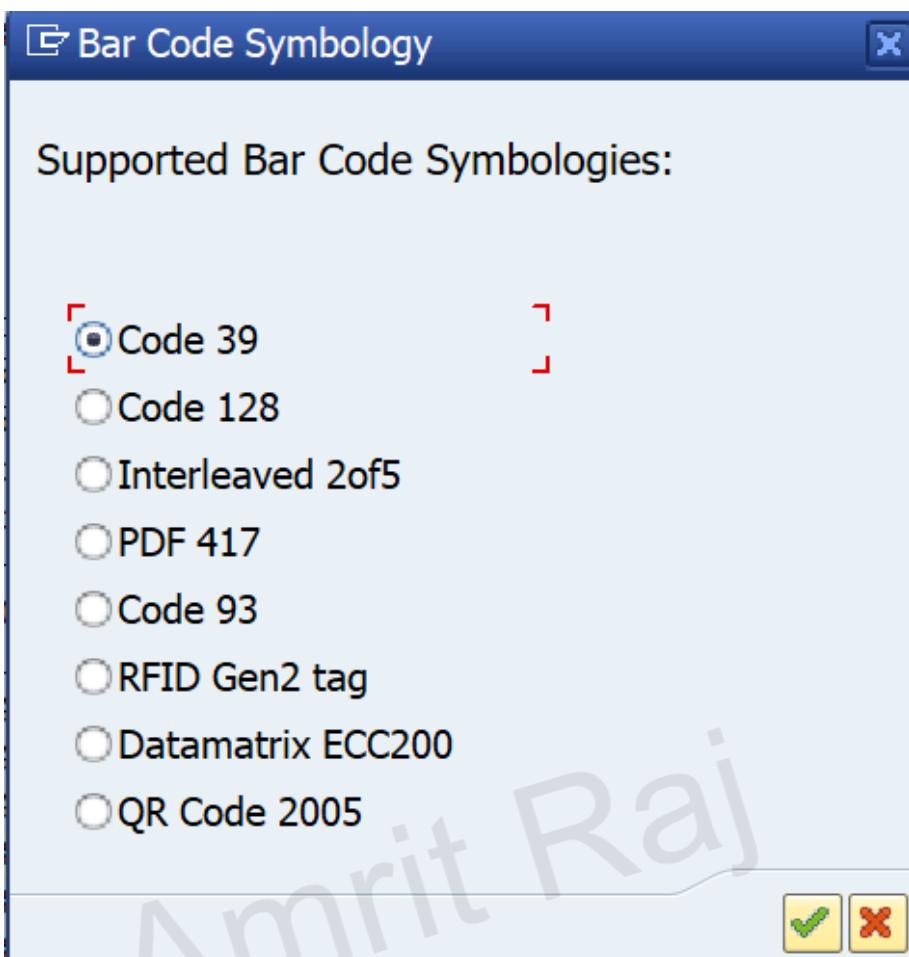
- Step 2 :- Click on Change and Click on create button.



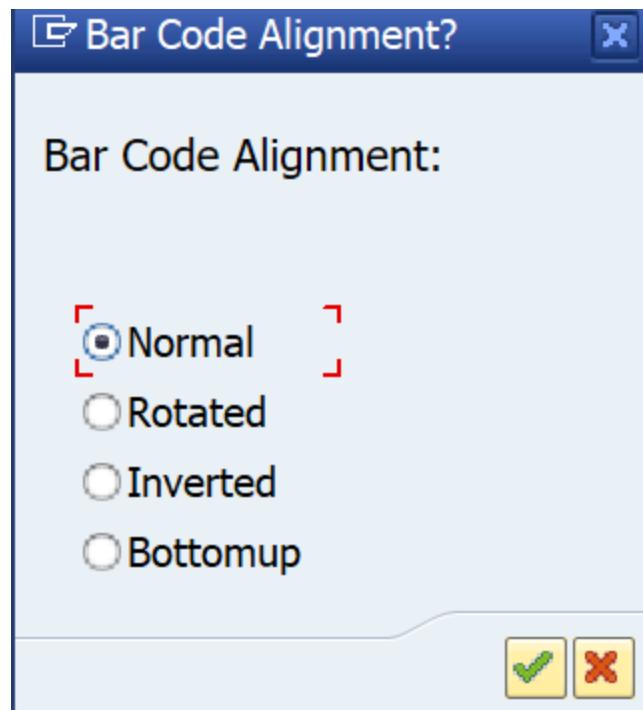
- Step 3 :- Click on New and give the name of Bar code and short description.



- Step 4 :- click on Okay button and select any format for bar code.



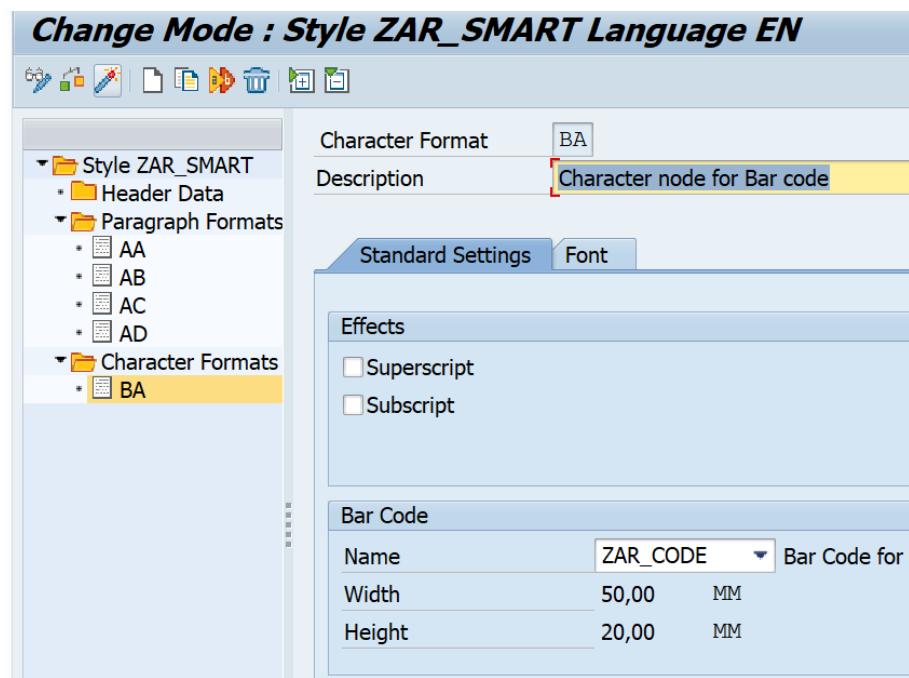
- Click on Okay button and select bar code alignment.



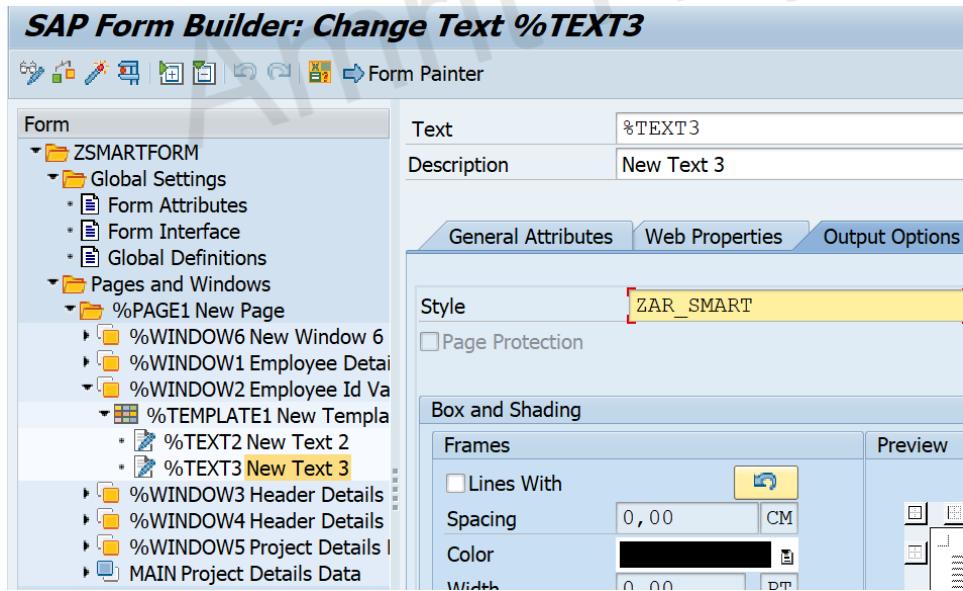
- Click on Okay button and save the bar code and assign the transport request.

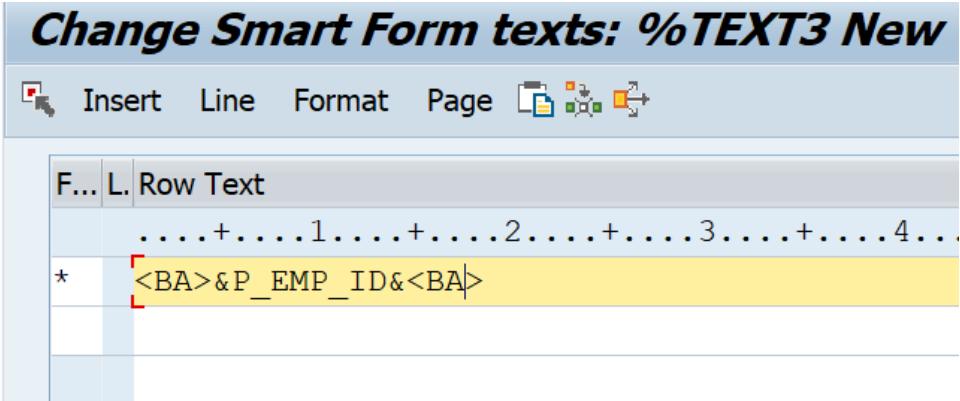
2. Creating Character Format for Bar Code

- For using bar codes in smartforms.
- We need to create character format in smartstyles.



3. Passing the Character format for Employee Id





Output

Employee Details

Employee Id

Employee

Employee Id	102	Employee Name	SHREYASHI TRIPATHI
Manager	AMRIT RAJ	Salary	10000,00

Project Details



A large watermark reading 'Amrit Raj' is overlaid on the entire output area.



9

Converting Smartform Output into PDF Format

1. Introduction

- Nowadays PDF are most widely accepted format.
- So, Customers always wants smartforms output into PDF format.
- So, to convert our smartform output into PDF format, we will follow the below steps.

2. Steps to Convert Smartform into PDF Format ?

- Step 1 :- Create a parameter for local file.

```
PARAMETERS : P_FILE TYPE LOCALFILE.  
PARAMETERS : P_EMP_ID TYPE ZAR_EMP_ID.
```

- Step 2 :- Call the F4_FILENAME function module to select the file in which you want to download the PDF.

```

AT SELECTION-SCREEN On VALUE-REQUEST FOR P_FILE.
CALL FUNCTION 'F4_FILENAME'
EXPORTING
  PROGRAM_NAME      = SYST-CPROG
  DYNPRO_NUMBER     = SYST-DYNNR
  FIELD_NAME        = ''
IMPORTING
  FILE_NAME         = P_FILE
.
```

- Step 3 :- Call the function module of the smartform in Start-Of-Selection and get the OTF data of the smartform.

```

START-OF-SELECTION.

DATA : LS_CONTROL TYPE SSFCTRLOP.
LS_CONTROL-GETOTF = 'X'.
LS_CONTROL-NO_DIALOG = 'X'.

DATA : LS_OUTPUT TYPE SSFCOMPOP.
LS_OUTPUT-TDDEST = 'LP01'.

DATA : LS_OUTPUT_INFO TYPE SSFCRESCL.
CALL FUNCTION '/1BCDWB/SF00000129'
EXPORTING
  * ARCHIVE_INDEX           =
  * ARCHIVE_INDEX_TAB       =
  * ARCHIVE_PARAMETERS      =
  CONTROL_PARAMETERS        = LS_CONTROL
  * MAIL_APPL_OBJ           =
  * MAIL_RECIPIENT          =
  * MAIL_SENDER              =
  OUTPUT_OPTIONS             = LS_OUTPUT
  USER_SETTINGS              = ''
  P_EMP_ID                  = P_EMP_ID
```

```

IMPORTING
*      DOCUMENT_OUTPUT_INFO      =
      JOB_OUTPUT_INFO          = LS_OUTPUT_INFO
*      JOB_OUTPUT_OPTIONS       =
EXCEPTIONS
      FORMATTING_ERROR        = 1
      INTERNAL_ERROR          = 2
      SEND_ERROR               = 3
      USER_CANCELED            = 4
      OTHERS                   = 5

```

- Step 4 :- Call CONVERT_OTF_2_PDF function to get the pdf data.

```

DATA : LT_LINES TYPE TABLE OF TLINE.
DATA : LT_DOCS TYPE TABLE OF DOCS.
CALL FUNCTION 'CONVERT_OTF_2_PDF'
* EXPORTING
*      USE_OTF_MC_CMD        = 'X'
*      ARCHIVE_INDEX          =
* IMPORTING
*      BIN_FILESIZE           =
TABLES
      OTF                      = LS_OUTPUT_INFO-OTFDATA
      DOCTAB_ARCHIVE            = LT_DOCS
      LINES                     = LT_LINES
EXCEPTIONS
      ERR_CONV_NOT_POSSIBLE    = 1
      ERR_OTF_MC_NOENDMARKER   = 2
      OTHERS                   = 3

```

- Step 5 :- We can use the GUI_DOWNLOAD function module to download the PDF.

```

DATA : LO_FILE TYPE STRING.
LO_FILE = P_FILE.

CALL FUNCTION 'GUI_DOWNLOAD'
  EXPORTING
    * BIN_FILESIZE           =
      FILENAME                = LO_FILE
      FILETYPE                 = 'BIN'
    * APPEND                  =
    * WRITE_FIELD_SEPARATOR   =
    * HEADER                  = '00'
    * TRUNC_TRAILING_BLANKS   =
    * WRITE_LF                 = 'X'
    * COL_SELECT               =
    * COL_SELECT_MASK          =
    * DAT_MODE                 =
    * CONFIRM_OVERWRITE        =
    * NO_AUTH_CHECK            =
    * CODEPAGE                 =
    * IGNORE_CERR               = ABAP_TRUE
    * REPLACEMENT              = '#'
    * WRITE_BOM                 =
    * TRUNC_TRAILING_BLANKS_EOL = 'X'
    * WK1_N_FORMAT              =
    * WK1_N_SIZE                 =
    * WK1_T_FORMAT               =
    * WK1_T_SIZE                 =
    * WRITE_LF_AFTER_LAST_LINE   = ABAP_TRUE
    * SHOW_TRANSFER_STATUS       = ABAP_TRUE
    * VIRUS_SCAN_PROFILE         = '/SCET/GUI_DOWNLOAD'
    * IMPORTING                 =
    * FILELENGTH                =
  TABLES
    DATA_TAB                  = LT_LINES
    * FIELDNAMES                =

```

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

Code

```
PARAMETERS : P_FILE TYPE LOCALFILE.  
PARAMETERS : P_EMP_ID TYPE ZAR_EMP_ID.  
  
AT SELECTION-SCREEN On VALUE-REQUEST FOR P_FILE.  
CALL FUNCTION 'F4_FILENAME'  
EXPORTING  
PROGRAM_NAME      = SYST-CPROG
```

```

DYNPRO_NUMBER      = SYST-DYNNR
FIELD_NAME        = ' '
IMPORTING
FILE_NAME         = P_FILE

.

START-OF-SELECTION.

DATA : LS_CONTROL TYPE SSFCTRLOP.
LS_CONTROL-GETOTF = 'X'.
LS_CONTROL-NO_DIALOG = 'X'.

DATA : LS_OUTPUT TYPE SSFCOMPOP.
LS_OUTPUT-TDDEST = 'LP01'.

DATA : LS_OUTPUT_INFO TYPE SSFCRESCL.
CALL FUNCTION '/1BCDWB/SF00000129'
EXPORTING
*   ARCHIVE_INDEX      =
*   ARCHIVE_INDEX_TAB   =
*   ARCHIVE_PARAMETERS   =
CONTROL_PARAMETERS      = LS_CONTROL
*   MAIL_APPL_OBJ       =
*   MAIL_RECIPIENT      =
*   MAIL_SENDER         =
OUTPUT_OPTIONS           = LS_OUTPUT
USER_SETTINGS             = ' '
P_EMP_ID                  = P_EMP_ID
IMPORTING
*   DOCUMENT_OUTPUT_INFO    =
JOB_OUTPUT_INFO           = LS_OUTPUT_INFO
*   JOB_OUTPUT_OPTIONS      =
EXCEPTIONS
FORMATTING_ERROR          = 1
INTERNAL_ERROR             = 2
SEND_ERROR                 = 3

```

```

USER_CANCELED          = 4
OTHERS                 = 5

DATA : LT_LINES TYPE TABLE OF TLINE.
DATA : LT_DOCS TYPE TABLE OF DOCS.
CALL FUNCTION 'CONVERT_OTF_2_PDF'
* EXPORTING
*   USE_OTF_MC_CMD      = 'X'
*   ARCHIVE_INDEX        =
* IMPORTING
*   BIN_FILESIZE         =
TABLES
OTF                    = LS_OUTPUT_INFO-OTFDATA
DOCTAB_ARCHIVE         = LT_DOCS
LINES                  = LT_LINES
EXCEPTIONS
ERR_CONV_NOT_POSSIBLE = 1
ERR_OTF_MC_NOENDMARKER = 2
OTHERS                 = 3

DATA : LO_FILE TYPE STRING.
LO_FILE = P_FILE.

CALL FUNCTION 'GUI_DOWNLOAD'
EXPORTING
*   BIN_FILESIZE         =
*   FILENAME              = LO_FILE
*   FILETYPE               = 'BIN'
*   APPEND                 = ' '
*   WRITE_FIELD_SEPARATOR  = ' '
*   HEADER                 = '00'
*   TRUNC_TRAILING_BLANKS = ' '
*   WRITE_LF                = 'X'

```

```

*   COL_SELECT           = ' '
*   COL_SELECT_MASK      = ' '
*   DAT_MODE             = ' '
*   CONFIRM_OVERWRITE    = ' '
*   NO_AUTH_CHECK        = ' '
*   CODEPAGE              = ' '
*   IGNORE_CERR          = ABAP_TRUE
*   REPLACEMENT          = '#'
*   WRITE_BOM             = ' '
*   TRUNC_TRAILING_BLANKS_EOL = 'X'
*   WK1_N_FORMAT          = ' '
*   WK1_N_SIZE             = ' '
*   WK1_T_FORMAT          = ' '
*   WK1_T_SIZE             = ' '
*   WRITE_LF_AFTER_LAST_LINE = ABAP_TRUE
*   SHOW_TRANSFER_STATUS  = ABAP_TRUE
*   VIRUS_SCAN_PROFILE    = '/SCET/GUI_DOWNLOAD'
*   IMPORTING
*   FILELENGTH            =
TABLES
  DATA_TAB                = LT_LINES
*   FIELDNAMES            =
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.
* Implement suitable error handling here
ENDIF.
```

Output

Employee Details*Employee Id***Employee**

Employee Id	103	Employee Name	ABHISHEK YADAV
Manager	AMRIT RAJ	Salary	10000,00

Project Details

Employee Id	Project Id	Project Name
103	100	SAP ABAP DEVELOPMENT
103	200	SAP ABAP DEVELOPMENT
103	300	SAP ABAP DEVELOPMENT



10

Email Triggering in Smartforms

1. Introduction

- To send mail into PDF, we will make use of the various standard classes related to BCS (Business Communication Services).
- We can use SOST transaction code while sending the mail to external system in order to check whether everything is perfect fine or not in our system.

2. Important Classes used for Email Purpose

- The various standard classes we will use for this purpose are as follows :-
 - CL_BCS :-
 - This class is used for creating the send request, adding the recipient, sending the document etc.
 - CL_DOCUMENT_BCS :-
 - This class is used for creating the document, adding the attachment etc.

- We will use this class for writing documents.
 - CL_CAM_ADDRESS_BCS :-
 - This class is used to create the external recipients.
 - This class is used to for sending the mail to outside address of SAP.
 - Out of these above classes, the first two are most important for Email purpose.
-

3. Steps Used for Email Writing Normally

- When ever send any mail, we use below steps.
 - We pass the recipient address (where we want to send the mail).
 - We pass the subject
 - We write the text for the email purpose (i.e. Description for mail) .
 - We attach the document for the same.
 - Then we have to click on send to send the mail.
 - This whole process is maintained by the first two classes.
 - CL_BCS
 - CL_DOCUMENT_BCS
-

4. Steps to Send Smartform PDF as an E-Mail Attachment

1. Get the BIN_FILE of the smartform.
2. Convert the BIN_FILE from XSTRING format to binary format.
3. Create the send request.
4. Create the SAP user/external user (recipient).
5. Add the recipient.
6. Create the document

7. Add the attachment.
 8. Set the document
 9. Activate/deactivate immediate sending.
 10. Send
 11. Commit
-

5. Implementation

- Step 1 :- Call the function module of the smartform to get the OTF data.

```

PARAMETERS : P_EMP_ID TYPE ZAR_EMP_ID.

START-OF-SELECTION.

DATA : LS_CONTROL TYPE SSFCTRLOP.
LS_CONTROL-GETOTF = 'X'.
LS_CONTROL-NO_DIALOG = 'X'.

DATA : LS_OUTPUT TYPE SSFCOMPPOP.
LS_OUTPUT-TDDEST = 'LP01'.

DATA : LS_OUTPUT_INFO TYPE SSFCRESCL.
CALL FUNCTION '/1BCDWB/SF00000129'
EXPORTING
*      ARCHIVE_INDEX          =
*      ARCHIVE_INDEX_TAB       =
*      ARCHIVE_PARAMETERS      =
      CONTROL_PARAMETERS      = LS_CONTROL
*      MAIL_APPL_OBJ          =
*      MAIL_RECIPIENT         =
*      MAIL_SENDER             =
      OUTPUT_OPTIONS           = LS_OUTPUT
      USER_SETTINGS            = ' '

```

```

    P_EMP_ID          = P_EMP_ID
    IMPORTING
    * DOCUMENT_OUTPUT_INFO      =
        JOB_OUTPUT_INFO           = LS_OUTPUT_INFO
    * JOB_OUTPUT_OPTIONS       =
    EXCEPTIONS
        FORMATTING_ERROR        = 1
        INTERNAL_ERROR           = 2
        SEND_ERROR                = 3
        USER_CANCELED             = 4
        OTHERS                     = 5

```

- Step 2 :- Now we have OTF data, so we will convert it to BIN_FILE using CONVERT_OTF function module, It will return me the BIN_FILE.

```

DATA : LT_LINES TYPE TABLE OF TLINE,
       LV_BIN_FILE TYPE XSTRING.

CALL FUNCTION 'CONVERT_OTF'
    EXPORTING
        FORMAT                  = 'PDF'
    *  MAX_LINEWIDTH          = 132
    *  ARCHIVE_INDEX           = ''
    *  COPYNUMBER               = 0
    *  ASCII_BIDI_VIS2LOG      = ''
    *  PDF_DELETE_OTFTAB       = ''
    *  PDF_USERNAME              = ''
    *  PDF_PREVIEW               = ''
    *  USE_CASCADING             = ''
    *  MODIFIED_PARAM_TABLE     =
    IMPORTING
    *  BIN_FILESIZE             =
        BIN_FILE                 = LV_BIN_FILE
    TABLES

```

```

OTF                      = LS_output_info-otfdata
LINES                    = LT_LINES
EXCEPTIONS
ERR_MAX_LINEWIDTH       = 1
ERR_FORMAT               = 2
ERR_CONV_NOT_POSSIBLE   = 3
ERR_BAD_OTF              = 4
OTHERS                   = 5

```

- Step 3 :- We have got our BIN_FILE in form of XSTRING format, now we need to convert it into binary format. So, we will use SCMS_XSTRING_TO_BINARY function module for the same purpose.

```

DATA : LT_BINARY_TAB type SOLIX_TAB.

CALL FUNCTION 'SCMS_XSTRING_TO_BINARY'
  EXPORTING
    BUFFER          = LV_BIN_FILE
  * APPEND_TO_TABLE = ''
  * IMPORTING
  * OUTPUT_LENGTH   =
  TABLES
    BINARY_TAB      = LT_BINARY_TAB

```

- Step 4 :- Now, we have the contents which I want to send a email, So now I will create a send request using CL_BCS class.
 - We have a CREATE_PERSISTENT method in CL_BCS class which we will use for create request purpose.
 - It will return me the object for CL_BCS class.

```
DATA : LO_BCS TYPE REF TO CL_BCS.
```

```
TRY.
```

```

CALL METHOD CL_BCS=>CREATE_PERSISTENT
      RECEIVING
          RESULT = LO_BCS
      .
      CATCH CX_SEND_REQ_BCS.
ENDTRY.

```

- Step 5 :- Now, to send the mail to the external user,I will have to create a recipient for external user.
 - For the same purpose, we have CL_CAM_ADDRESS_BCS class and we will call its method CREATE_INTERNET_ADDRESS, it will return the object of this class.

```

TRY.
CALL METHOD CL_CAM_ADDRESS_BCS=>CREATE_INTERNET_ADDRESS
      EXPORTING
          I_ADDRESS_STRING = 'atulsing2505@gmail.com'
          *     I_ADDRESS_NAME      =
          *     I_INCL_SAPUSER      =
      RECEIVING
          RESULT           = LO_EXTERNAL_USER
      .
      CATCH CX_ADDRESS_BCS.
ENDTRY.

```

- Step 6 :- Now, we have got the external address, then we will add this recipient to CL_BCS class using ADD_RECIPIENT method.

```

TRY.
CALL METHOD LO_BCS->ADD_RECIPIENT
      EXPORTING
          I_RECIPIENT   = LO_EXTERNAL_USER
          *     I_EXPRESS    =
          *     I_COPY       =
          *     I_BLIND_COPY =
          *     I_NO_FORWARD =

```

```
CATCH CX_SEND_REQ_BCS.  
ENDTRY.
```

- Step 7 :- Now, we have added the recipient, now we will create the document for the same purpose.
 - We will provide, subject, text and will attach the PDF for the same purpose in the document part.
 - We will use CREATE_DOCUMENT method of CL_DOCUMENT_BCS class to create the document.

```
DATA : LT_TEXT TYPE TABLE OF SOLI,  
       LS_TEXT TYPE SOLI.  
  
LS_TEXT-LINE = 'Dear Sir'.  
APPEND LS_TEXT to LT_TEXT.  
CLEAR LS_TEXT.  
  
LS_TEXT-LINE = 'Below is the attached Smartform'.  
APPEND LS_TEXT to LT_TEXT.  
CLEAR LS_TEXT.  
  
LS_TEXT-LINE = 'Thanks and Regards'.  
APPEND LS_TEXT to LT_TEXT.  
CLEAR LS_TEXT.  
  
LS_TEXT-LINE = 'Amrit Raj'.  
APPEND LS_TEXT to LT_TEXT.  
CLEAR LS_TEXT.  
  
DATA : LO_DOCUMENT type ref to CL_DOCUMENT_BCS.  
  
TRY.  
CALL METHOD CL_DOCUMENT_BCS=>CREATE_DOCUMENT  
      EXPORTING
```

```

        I_TYPE          = 'RAW'
        I SUBJECT       = 'Employee Details'
*
        I_LENGTH        =
*
        I_LANGUAGE      = SPACE
*
        I_IMPORTANCE    =
*
        I_SENSITIVITY   =
        I_TEXT          = LT_TEXT
*
        I_HEX           =
*
        I_HEADER         =
*
        I_SENDER         =
*
        IV_VSI_PROFILE  =

RECEIVING
        RESULT          = LO_DOCUMENT

.

CATCH CX_DOCUMENT_BCS.
ENDTRY.

```

- Step 8 :- Now we will add the smartform and a Subject for the Mail, for that purpose we will use ADD_ATTACHMENT method of CL_DOCUMENT_BCS class.

```

TRY.
CALL METHOD LO_DOCUMENT->ADD_ATTACHMENT
EXPORTING
        I_ATTACHMENT_TYPE     = 'PDF'
        I_ATTACHMENT_SUBJECT  = 'Employee Details Using Smartfor
*
        I_ATTACHMENT_SIZE     =
*
        I_ATTACHMENT_LANGUAGE = SPACE
*
        I_ATT_CONTENT_TEXT    =
        I_ATT_CONTENT_HEX     = LT_BINARY_TAB
*
        I_ATTACHMENT_HEADER   =
*
        IV_VSI_PROFILE        =
.

CATCH CX_DOCUMENT_BCS.
ENDTRY.

```

- Step 9 :- Now we have to set the document, so for that purpose we will use the SET_DOCUMENT method of CL_BCS class to set the document and we will pass the object of CL_DOCUMENT_BCS class into this method.

```

TRY.

CALL METHOD LO_BCS->SET_DOCUMENT
  EXPORTING
    I_DOCUMENT = LO_DOCUMENT
  .
  CATCH CX_SEND_REQ_BCS.
ENDTRY.

```

- Step 10 :- Last step is to prepare mail for sending so we will call the SET_SEND_IMMEDIATELY method of CL_BCS class and we will pass it to true.

```

TRY.

CALL METHOD LO_BCS->SET_SEND_IMMEDIATELY
  EXPORTING
    I_SEND_IMMEDIATELY = 'X'
  .
  CATCH CX_SEND_REQ_BCS.
ENDTRY.

```

- Step 11 :- Send the mail using send method of CL_BCS class.

```

DATA : LV_RESULT type OS_BOOLEAN.

TRY.

CALL METHOD LO_BCS->SEND
  *  EXPORTING
  *    I_WITH_ERROR_SCREEN = SPACE
  RECEIVING
    RESULT          = LV_RESULT
  .
  CATCH CX_SEND_REQ_BCS.
ENDTRY.

```

```

IF LV_RESULT is NOT INITIAL.
  COMMIT WORK.
  MESSAGE 'Mail is Sent' TYPE 'S'.
ELSE.
  MESSAGE 'Mail is not Sent' type 'E'.
ENDIF.

```

Code

```

PARAMETERS : P_EMP_ID TYPE ZAR_EMP_ID.

START-OF-SELECTION.

DATA : LS_CONTROL TYPE SSFCTRLOP.
LS_CONTROL-GETOTF = 'X'.
LS_CONTROL-NO_DIALOG = 'X'.

DATA : LS_OUTPUT TYPE SSFCOMPOP.
LS_OUTPUT-TDDEST = 'LP01'.

DATA : LS_OUTPUT_INFO TYPE SSFCRESCL.
CALL FUNCTION '/1BCDWB/SF00000129'
  EXPORTING
    * ARCHIVE_INDEX           =
    * ARCHIVE_INDEX_TAB       =
    * ARCHIVE_PARAMETERS      =
    CONTROL_PARAMETERS        = LS_CONTROL
    * MAIL_APPL_OBJ           =
    * MAIL_RECIPIENT          =
    * MAIL_SENDER              =
    OUTPUT_OPTIONS             = LS_OUTPUT
    USER_SETTINGS              = ' '
    P_EMP_ID                  = P_EMP_ID

```

```

IMPORTING
*  DOCUMENT_OUTPUT_INFO      =
  JOB_OUTPUT_INFO           = LS_OUTPUT_INFO
*  JOB_OUTPUT_OPTIONS       =
EXCEPTIONS
  FORMATTING_ERROR          = 1
  INTERNAL_ERROR             = 2
  SEND_ERROR                 = 3
  USER_CANCELED               = 4
  OTHERS                      = 5

DATA : LT_LINES TYPE TABLE OF TLINE,
       LV_BIN_FILE TYPE XSTRING.

CALL FUNCTION 'CONVERT_OTF'
EXPORTING
  FORMAT                  = 'PDF'
*  MAX_LINEWIDTH            = 132
*  ARCHIVE_INDEX             = ''
*  COPYNUMBER                = 0
*  ASCII_BIDI_VIS2LOG        = ''
*  PDF_DELETE_OTFTAB         = ''
*  PDF_USERNAME                = ''
*  PDF_PREVIEW                  = ''
*  USE_CASCADING                = ''
*  MODIFIED_PARAM_TABLE        =
IMPORTING
*  BIN_FILESIZE               =
  BIN_FILE                  = LV_BIN_FILE
TABLES
  OTF                       = LS_OUTPUT_INFO-OTFDATA
  LINES                      = LT_LINES
EXCEPTIONS
  ERR_MAX_LINEWIDTH          = 1
  ERR_FORMAT                  = 2

```

```
ERR_CONV_NOT_POSSIBLE      = 3
ERR_BAD_OTF                 = 4
OTHERS                      = 5
```

```
DATA : LT_BINARY_TAB type SOLIX_TAB.
```

```
CALL FUNCTION 'SCMS_XSTRING_TO_BINARY'
  EXPORTING
    BUFFER                  = LV_BIN_FILE
  * APPEND_TO_TABLE        = ' '
  * IMPORTING
  * OUTPUT_LENGTH          =
  TABLES
    BINARY_TAB             = LT_BINARY_TAB
```

```
DATA : LO_BCS TYPE REF TO CL_BCS.
```

```
TRY.
  CALL METHOD CL_BCS=>CREATE_PERSISTENT
    RECEIVING
      RESULT = LO_BCS

    CATCH CX_SEND_REQ_BCS.
  ENDTRY.
```

```
DATA : LO_EXTERNAL_USER TYPE REF TO CL_CAM_ADDRESS_BCS.
```

```
TRY.
  CALL METHOD CL_CAM_ADDRESS_BCS=>CREATE_INTERNET_ADDRESS
```

```

EXPORTING
  I_ADDRESS_STRING = 'atulsing2505@gmail.com'
*   I_ADDRESS_NAME    =
*   I_INCL_SAPUSER    =
RECEIVING
  RESULT          = LO_EXTERNAL_USER

.
CATCH CX_ADDRESS_BCS.

ENDTRY.

TRY.
CALL METHOD LO_BCS->ADD_RECIPIENT
  EXPORTING
    I_RECIPIENT = LO_EXTERNAL_USER
*    I_EXPRESS   =
*    I_COPY      =
*    I_BLIND_COPY =
*    I_NO_FORWARD =
.
CATCH CX_SEND_REQ_BCS.

ENDTRY.

DATA : LT_TEXT TYPE TABLE OF SOLI,
       LS_TEXT TYPE SOLI.

LS_TEXT-LINE = 'Dear Sir'.
APPEND LS_TEXT to LT_TEXT.
CLEAR LS_TEXT.

LS_TEXT-LINE = 'Below is the attached Smartform'.
APPEND LS_TEXT to LT_TEXT.
CLEAR LS_TEXT.

LS_TEXT-LINE = 'Thanks and Regards'.
APPEND LS_TEXT to LT_TEXT.

```

```

CLEAR LS_TEXT.

LS_TEXT-LINE = 'Amrit Raj'.
APPEND LS_TEXT to LT_TEXT.
CLEAR LS_TEXT.

DATA : LO_DOCUMENT type ref to CL_DOCUMENT_BCS.

TRY.
CALL METHOD CL_DOCUMENT_BCS=>CREATE_DOCUMENT
EXPORTING
  I_TYPE          = 'RAW'
  I SUBJECT       = 'Employee Details'
*   I_LENGTH        =
*   I_LANGUAGE      = SPACE
*   I_IMPORTANCE    =
*   I_SENSITIVITY   =
  I_TEXT          = LT_TEXT
*   I_HEX           =
*   I_HEADER        =
*   I_SENDER        =
*   IV_VSI_PROFILE =
RECEIVING
  RESULT          = LO_DOCUMENT
.

CATCH CX_DOCUMENT_BCS.

ENDTRY.

TRY.
CALL METHOD LO_DOCUMENT->ADD_ATTACHMENT
EXPORTING
  I_ATTACHMENT_TYPE     = 'PDF'
  I_ATTACHMENT SUBJECT  = 'Employee Details Using Smartform'
*   I_ATTACHMENT_SIZE    =
*   I_ATTACHMENT LANGUAGE = SPACE

```

```

*      I_ATT_CONTENT_TEXT      =
I_ATT_CONTENT_HEX      = LT_BINARY_TAB
*      I_ATTACHMENT_HEADER     =
*      IV_VSI_PROFILE          =

.

CATCH CX_DOCUMENT_BCS.

ENDTRY.

TRY.

CALL METHOD LO_BCS->SET_DOCUMENT
EXPORTING
I_DOCUMENT = LO_DOCUMENT

.

CATCH CX_SEND_REQ_BCS.

ENDTRY.

TRY.

CALL METHOD LO_BCS->SET_SEND_IMMEDIATELY
EXPORTING
I_SEND_IMMEDIATELY = 'X'

.

CATCH CX_SEND_REQ_BCS.

ENDTRY.

DATA : LV_RESULT type OS_BOOLEAN.

TRY.

CALL METHOD LO_BCS->SEND
*  EXPORTING
*  I_WITH_ERROR_SCREEN = SPACE
RECEIVING
RESULT                  = LV_RESULT

.

CATCH CX_SEND_REQ_BCS.

ENDTRY.

```

```
IF LV_RESULT is NOT INITIAL.  
    COMMIT WORK.  
    MESSAGE 'Mail is Sent' TYPE 'S'.  
ELSE.  
    MESSAGE 'Mail is not Sent' type 'E'.  
ENDIF.
```

Output

- We can go to SOST and check the contents of Mail.

Display Document: Employee Details

Doc. contents Attributes Recipient list Attachments

Employee Details

Created AMRIT01

Dear Sir
Below is the attached Smartform
Thanks and Regards
Amrit Raj

[Employee Details Using Smartform](#)

