Capgemini

ABAP Part III

Lesson 03: SmartForms

Lesson Objectives

After completing this lesson, participants will be able to -

- Know the Smartforms Architecture
- Work with SAP Form Builder
- Work with Texts, Addresses and Graphics
- Work with Data in Forms
- Work with Tables and Templates
- Flow Control
- Integrate SmartForms into Application Programs
- Work with Fonts and Bar Codes

Smart Forms-Introduction

Tool to create and maintain forms with minimal programming effort

Easy to create the form without much programming knowledge

Allows to execute simple modifications to the form and in the form logic by using graphical tools

Basic Features of SMARTFORM

Data retrieval and form logic are separated from each other.

Application program passes data to Smartform through Function module interface which is generated automatically on Smartform activation.

Reduces the implementation cost.

Key Benefits of SMARTFORM

Less Programming Efforts

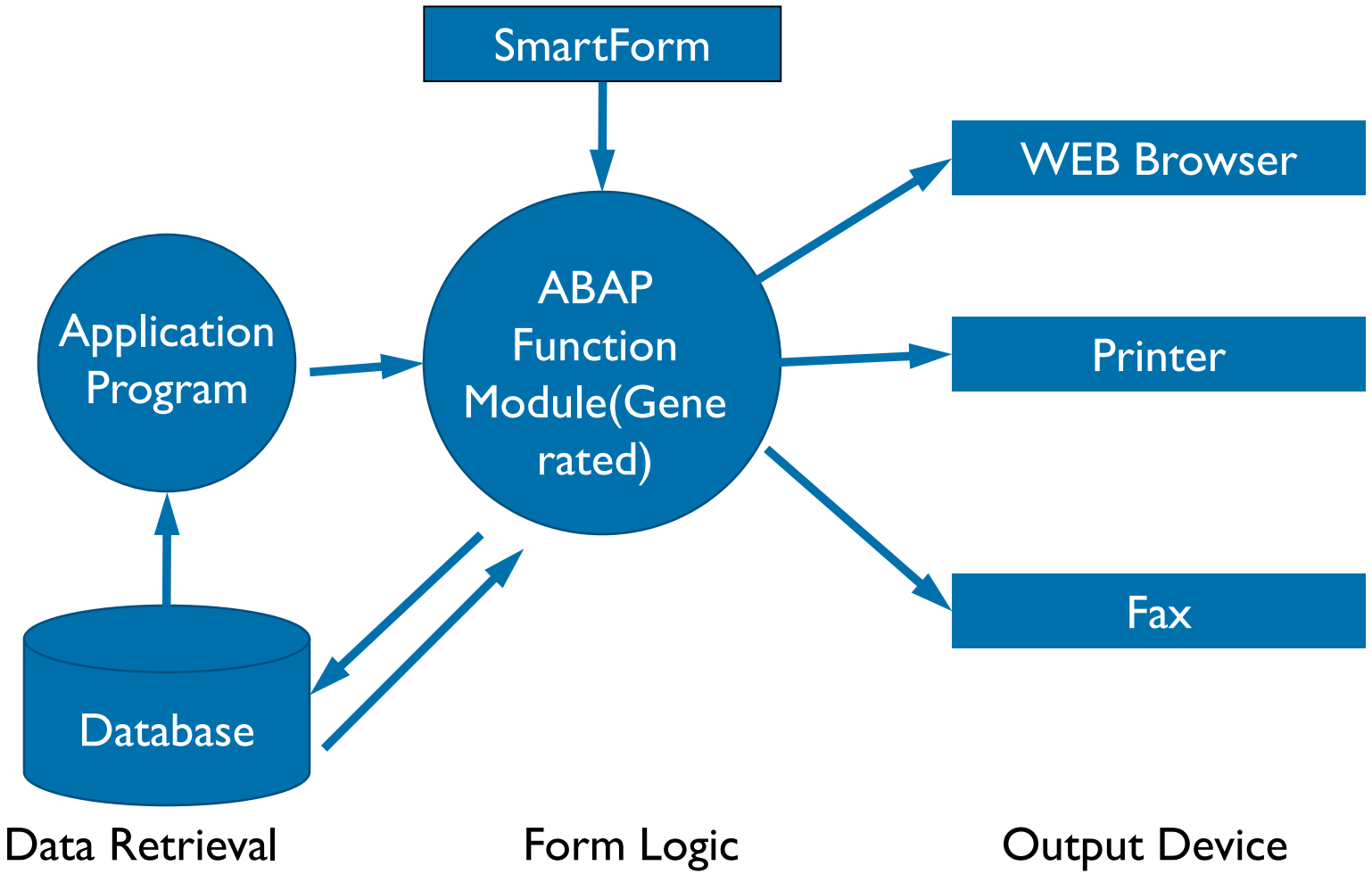
Output of background graphics, for form design

Colored output of texts

User-friendly and integrated Form Painter for the graphical design of forms.

Graphical Table Painter for drawing tables

SMARTFORM Architecture



Smartform Architecture - Structure

A Smart form has the following attributes:

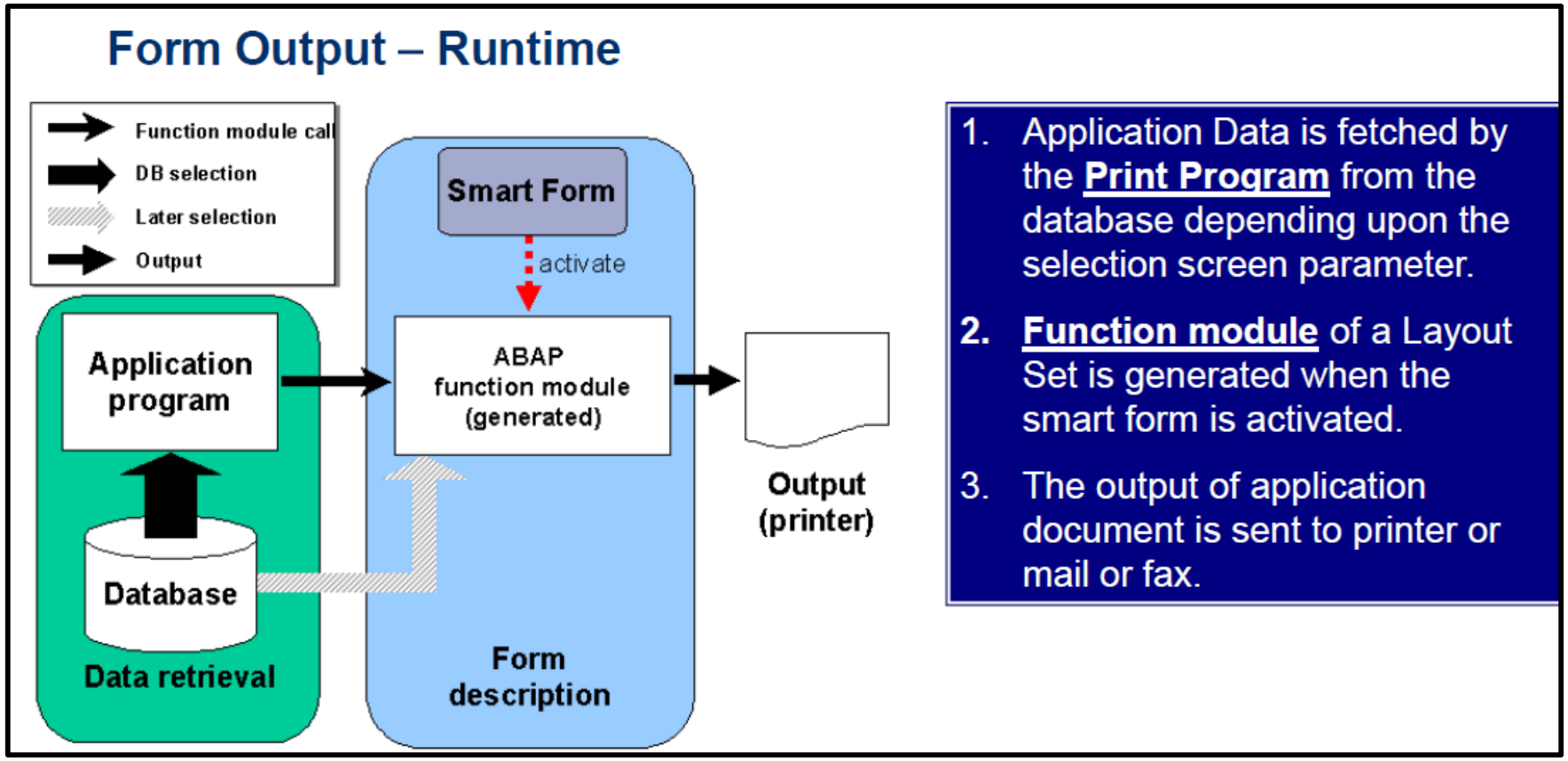
- 1.Layout: A layout. In the layout, you define how the output data is positioned, its appearance in graphics, and the design of the pages.
- 2.Form logic : control the flow of the form output.
- 3.Form interface to transfer application data to the form definition.

Transactions

SMARTFORMS	Create Smartform
SMARTSTYLES	Create style
SO10	Create standard Text
SE38	Create print program
SE78	Upload Logo

Architecture - Form Output Runtime

The following graphics show you the architecture that is implemented when you create and print a Smart form.



Transaction Codes in Smart Forms

Different Transaction codes used while working with Smartforms are

Transaction Code	Description
Smartforms	To Create forms
SE78	To create any graphic image which in turn can be used as logo or background picture in Smart forms

Structure of SmartForm

SMARTFORMS provides a graphical user interface which is divided into three different parts:

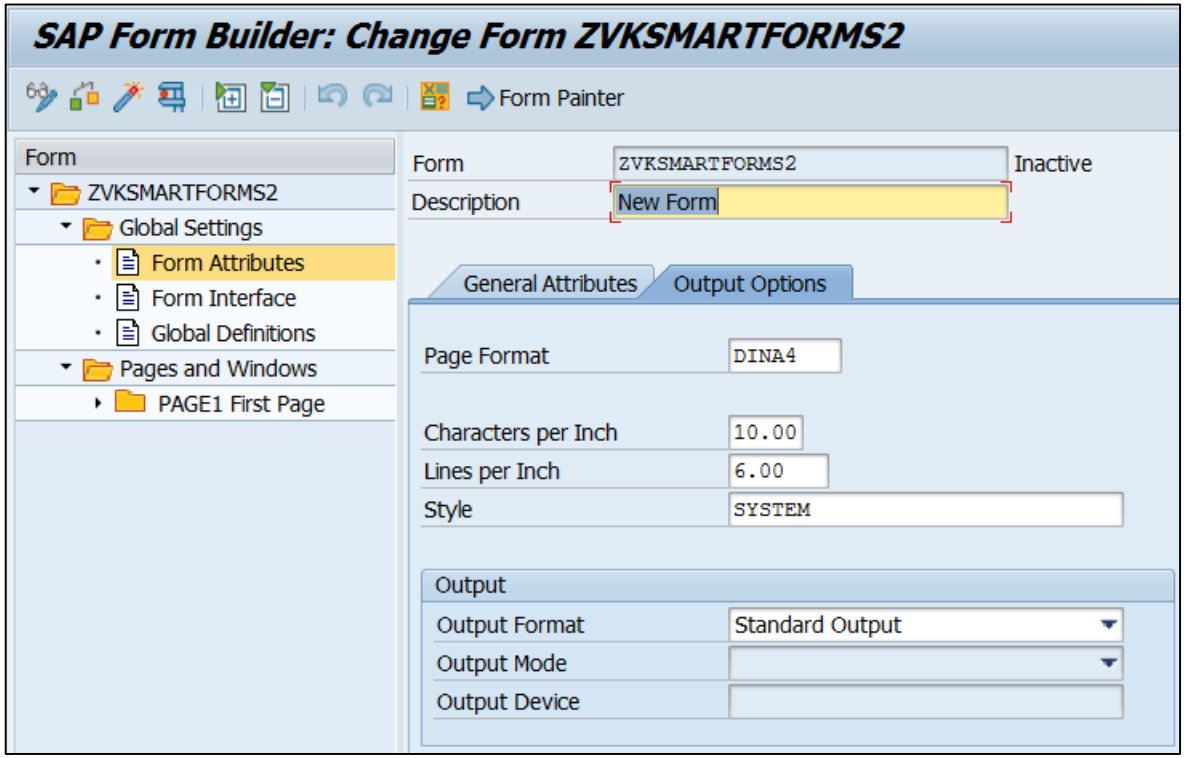
- Navigation Panel
- PC Editor
- Form Painter

Demo: View Smartform Interface

Smart Forms - Navigation Panel

Global Settings Has 3 sections

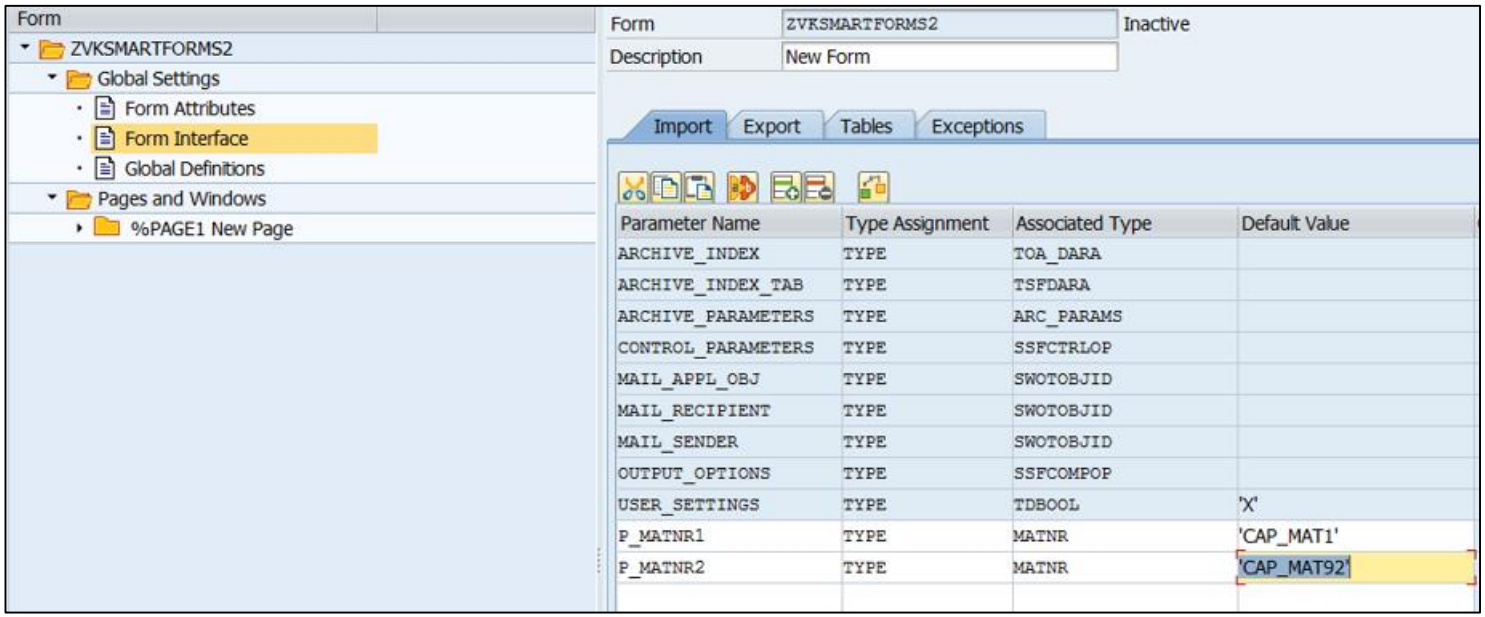
- Form Attributes
  - Define settings necessary for printing like page format, characters per inch, flag for XSF output etc



Navigation Panel - Global Settings

Form Interface

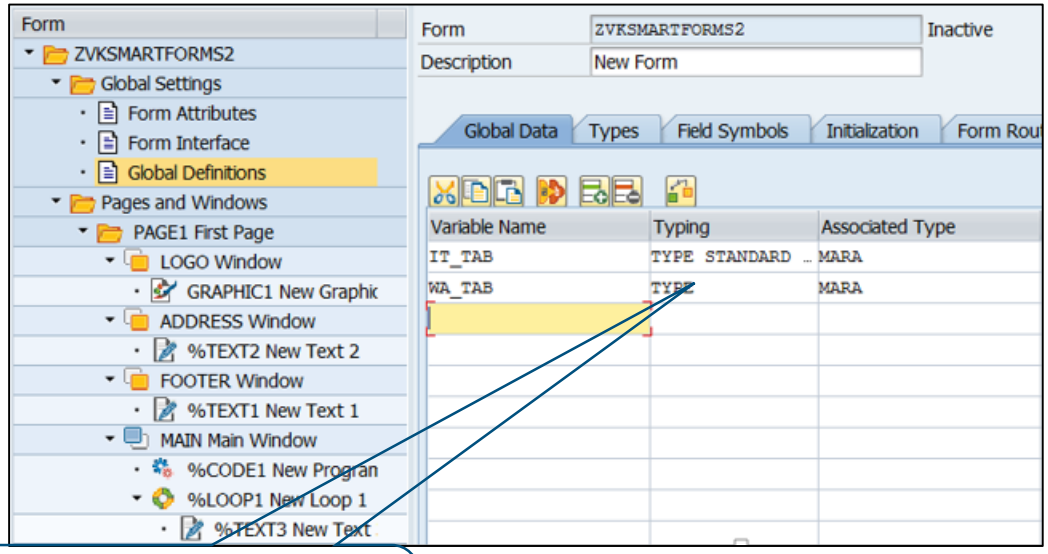
- To transfer application data to form definition
- The IMPORT EXPORT parameters and INTERNAL TABLES are declared
- Exception handling is also taken care.



Navigation Panel - Global Settings (Contd.).

Global Definition

- Allows user to declare variables which can be used on global scope



Specify the Data Objects used in Form

Pages and Windows

Provides list of all components of form

All the basic elements are maintained under this node

- Pages
- Windows
- Graphic
- Address

Pages

Each form consists of one or more pages

The first page in the tree structure is the start page

The page layout includes the page format and the position of windows on a page

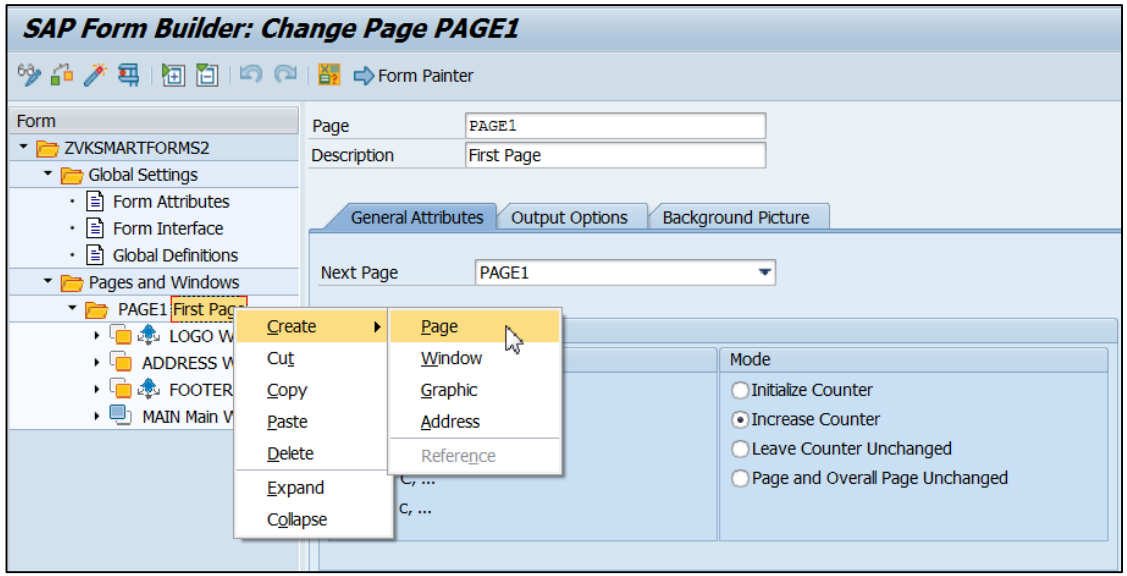
Procedure for Creating Page

Select an existing page node to position the new page node

Create a new page node in the navigation tree of the Form Builder

Name and description has to be specified

The format and mode of the page counter has to be specified on General Attributes tab



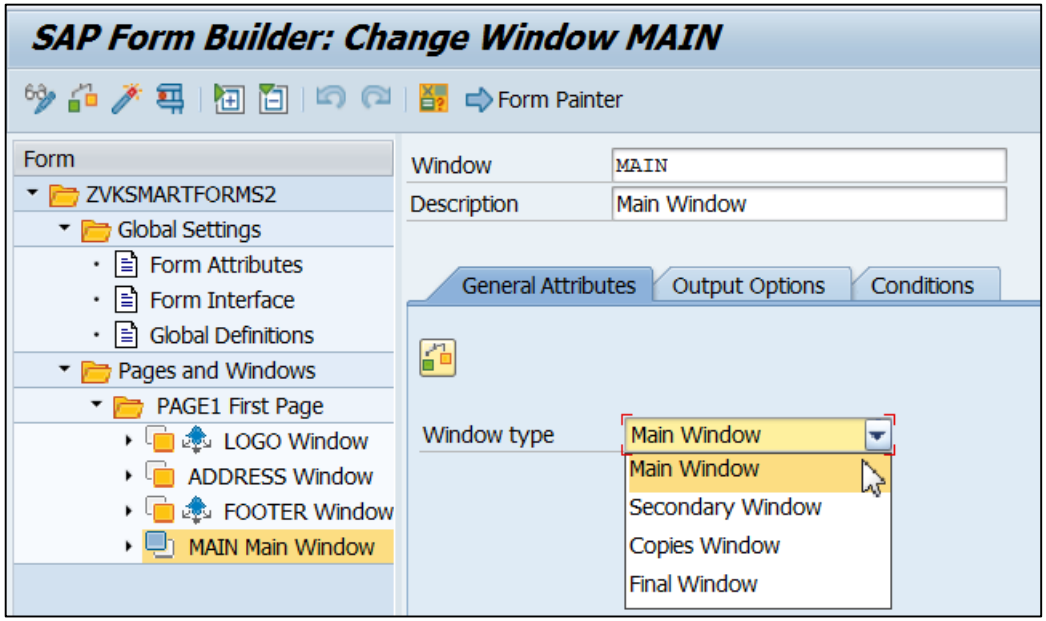
Windows

Output areas for all output data

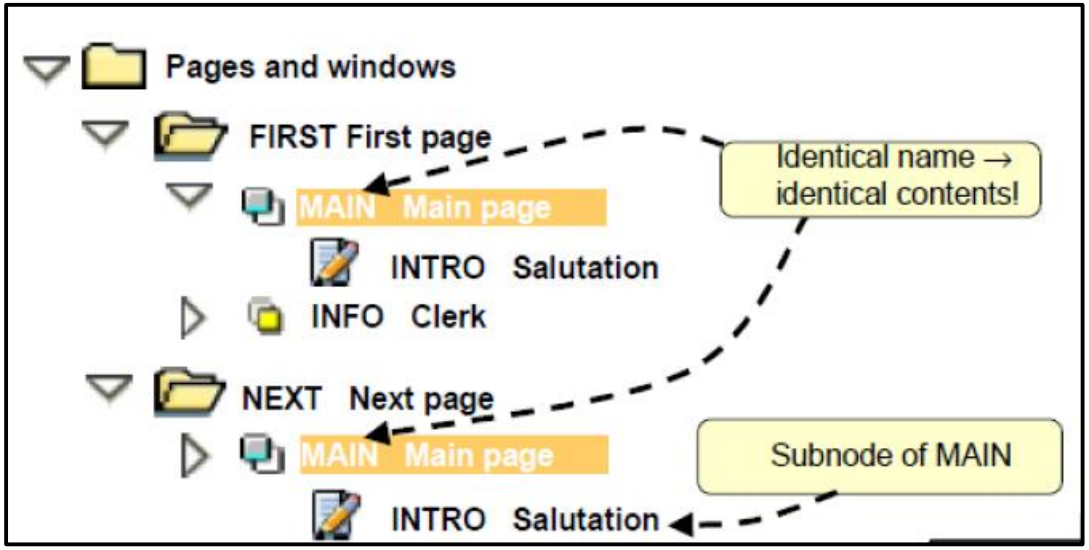
Size and position are set in the Form Painter

Following are the Window Types in Smartforms

- Main window
- Secondary window
- Copies Window
- Final Window



Form Builder - Structuring Pages



Main Window

Display the text and data that runs in to several pages

It automatically triggers the page break

Only one window in a form is main window

The main window must have the same width on each page

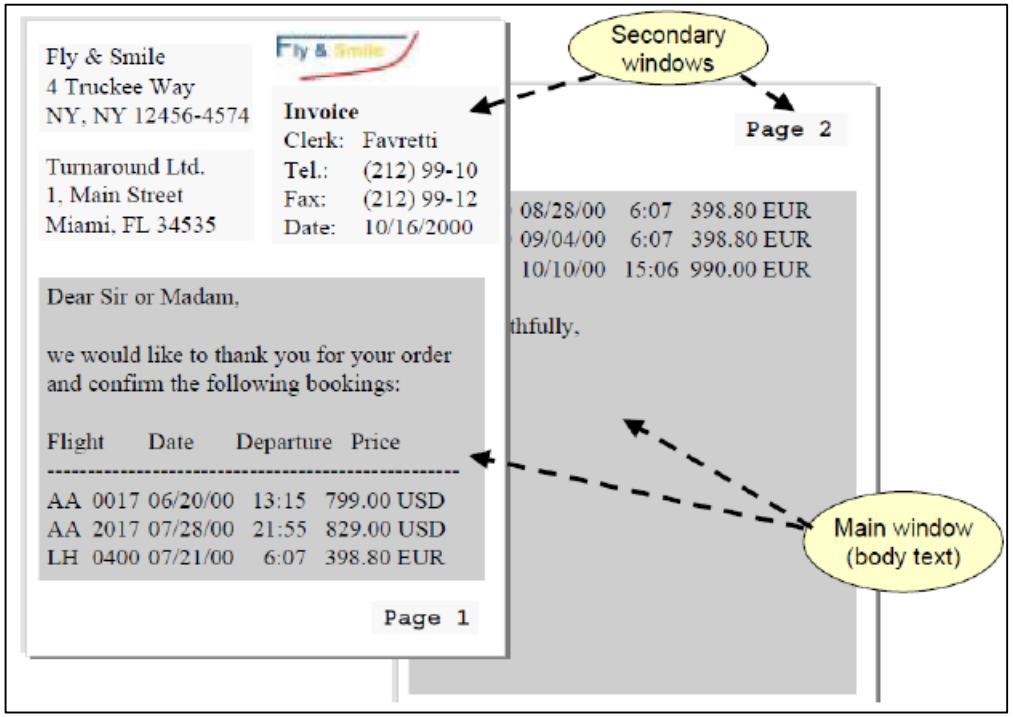
A page without main window must not call itself as next page, since this would trigger an endless loop

Secondary-Windows

Text and data displayed in a predetermined output area.

Text and data that do not fit into the secondary window are truncated and not displayed.

Form Builder - Main and Secondary Windows



Copies-Window

The content will appear either in the copy form or original form

This is used for printing the copies of the form

Final-Window

Final window is used to display values which are processed in the initial pages

Texts and Data in a Form:

Texts and data are entered using PC Editor

Various Operations performed in the PC Editor

Tables or templates can be used to display texts and data in table format

Positioning of Texts on the Form

All the texts in the form are displayed using text nodes

The only exception is addresses, which are displayed using their own node

Entering Texts in PC Editor

New texts are entered in PC Editor

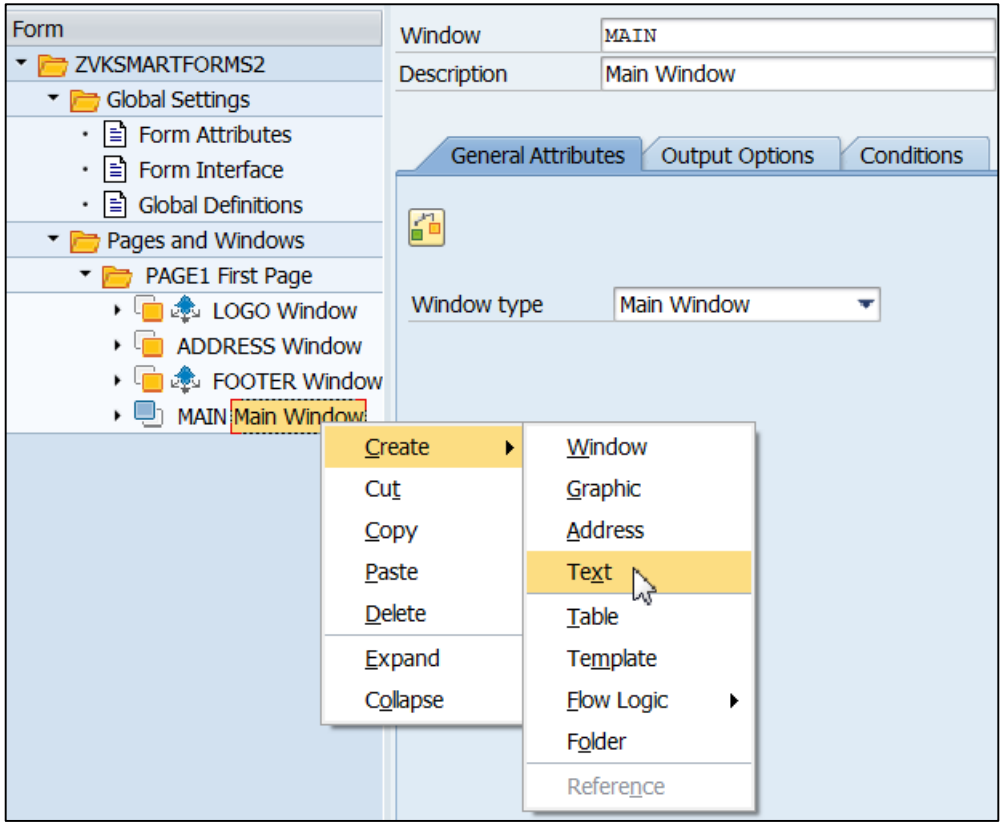
The system fields and the user-defined fields are used to include data from form interface

These fields are replaced with values when the form is processed

Creating Text

Procedure

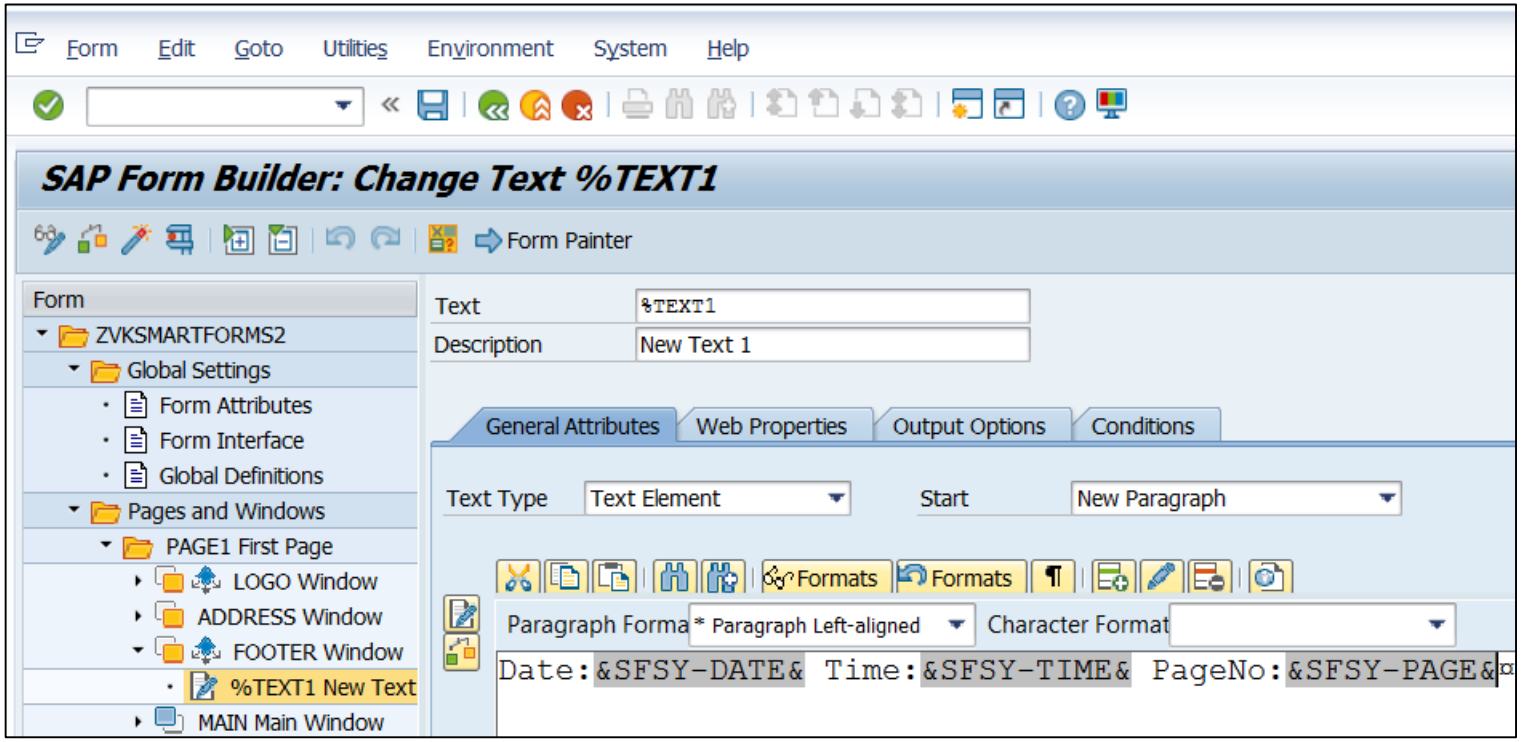
- 1. Create a text node in the navigation tree of the Form Builder.



Text Element in Form

Enter a unique name for the node and a node description.

On the General Attributes tab choose Text Element as text.



Demo: Create a smartform without table

Integrating the Smart Form into the Application

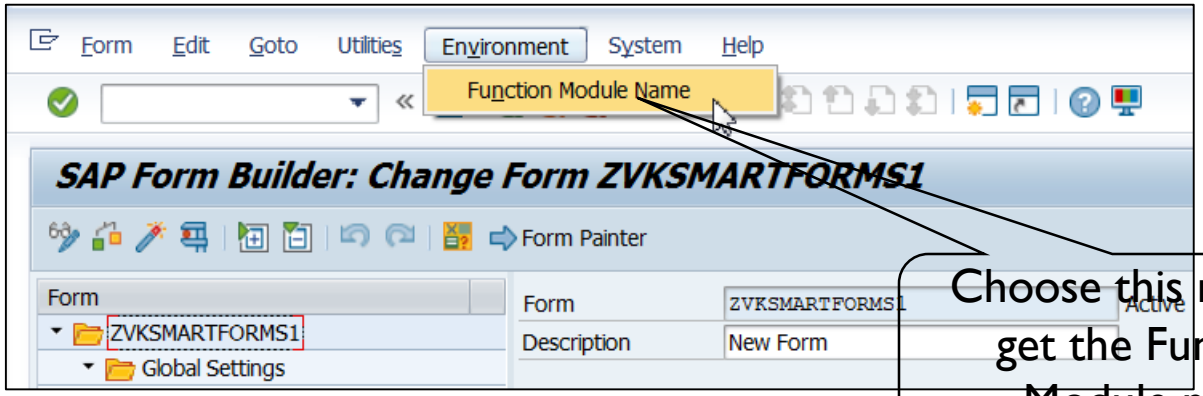
Form printing triggered by calling function modules

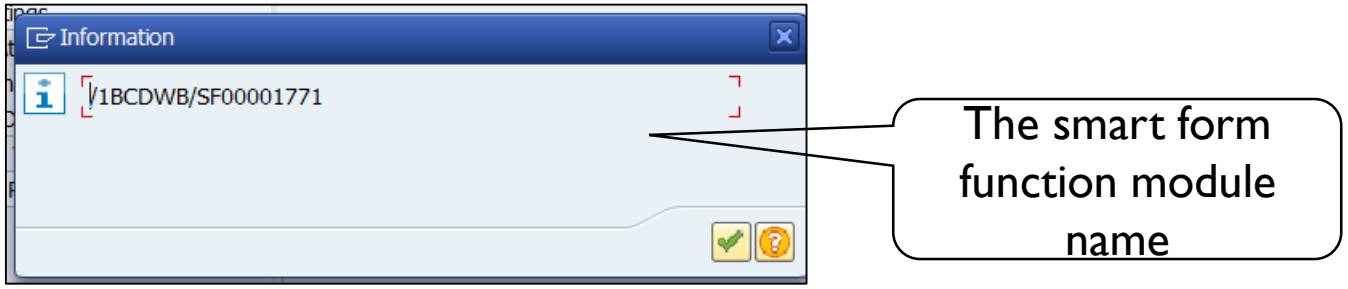
Name of the form determines the name of the generated function module.

The name of the generated function module is unique only within one system.

Procedure

In the Form Builder call the function Environment ☐Name of the function module and copy its name.





Procedure (Contd.).

In the application program define a variable of type RS38L_FNAM for the name of the generated function module:

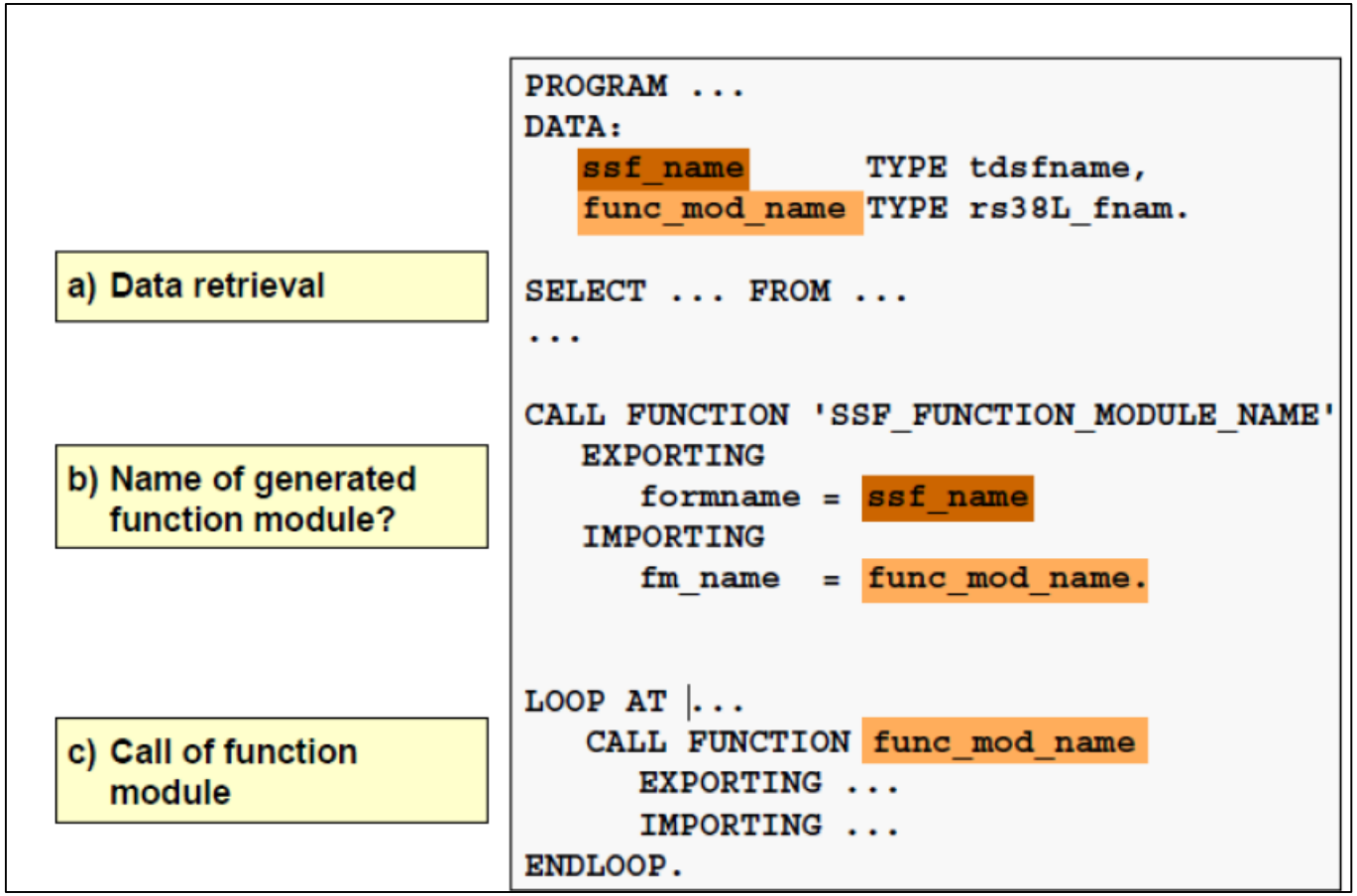
- Data fm_name type RS38L_FNAM.

The Smart Form can be called in other parts of the application program as well.

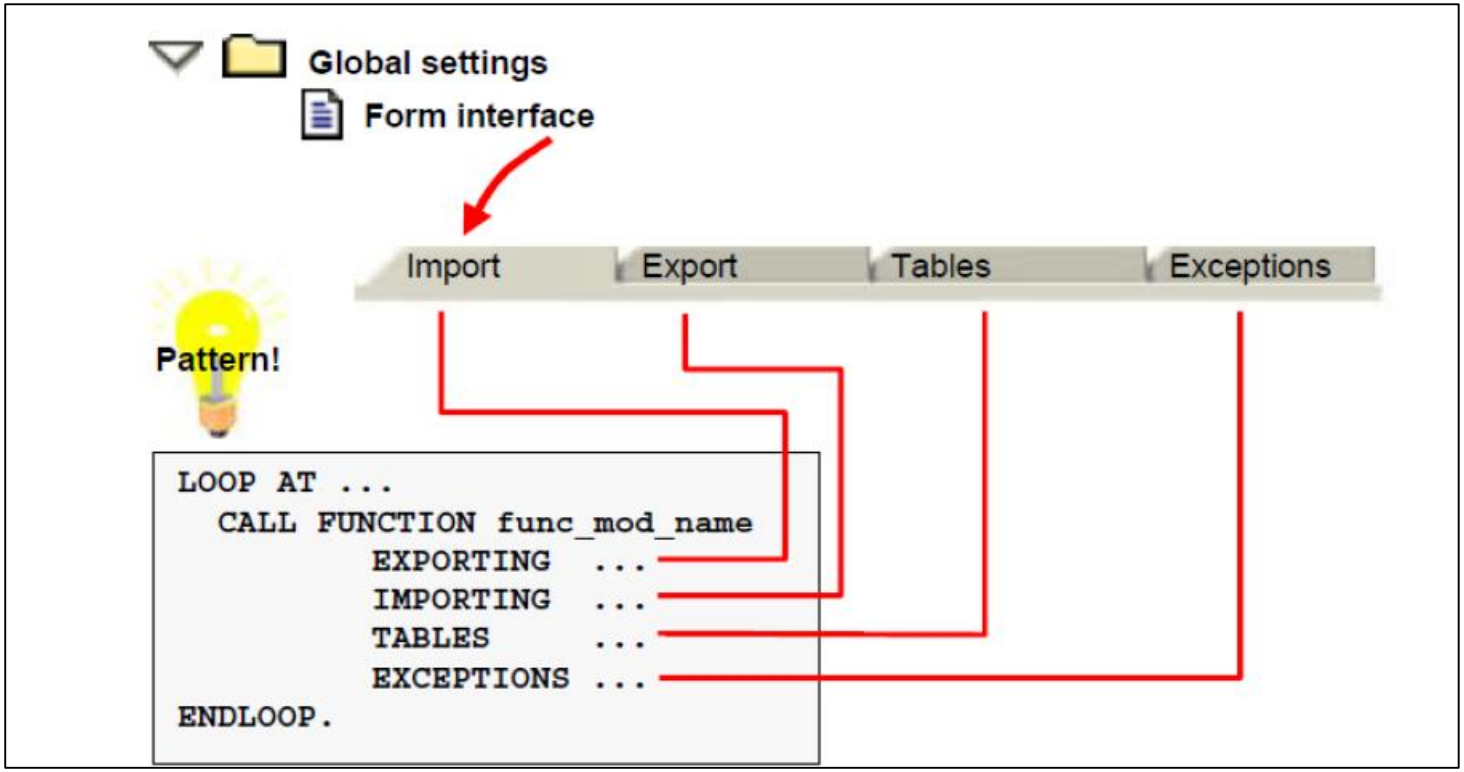
Function module SSF_FIELD_LIST - to list form parameters

Function module SSF_FUNCTION_MODULE_NAME - returns the name of generated function module

Integration in ABAP programs



Integration into Application Programs – Generated Function Module



Integration into Application Programs – Control Structure CONTROL_PARAMETERS

CONTROL_PARAMETERS
(Export parameters of the function module generated)

Type:	`ssfctrlop`
`no_open`	No new spool request
`no_close`	Do not close spool request
`device`	Output device (`'PRINTER'`, `'TELEFAX'`, `'MAIL'`)
`no_dialog`	No dialog box for output
`preview`	Print preview
`langu`	Language
`startpage`	Start page ≠ default

Demo: Integrate smartform in application

Text Modules

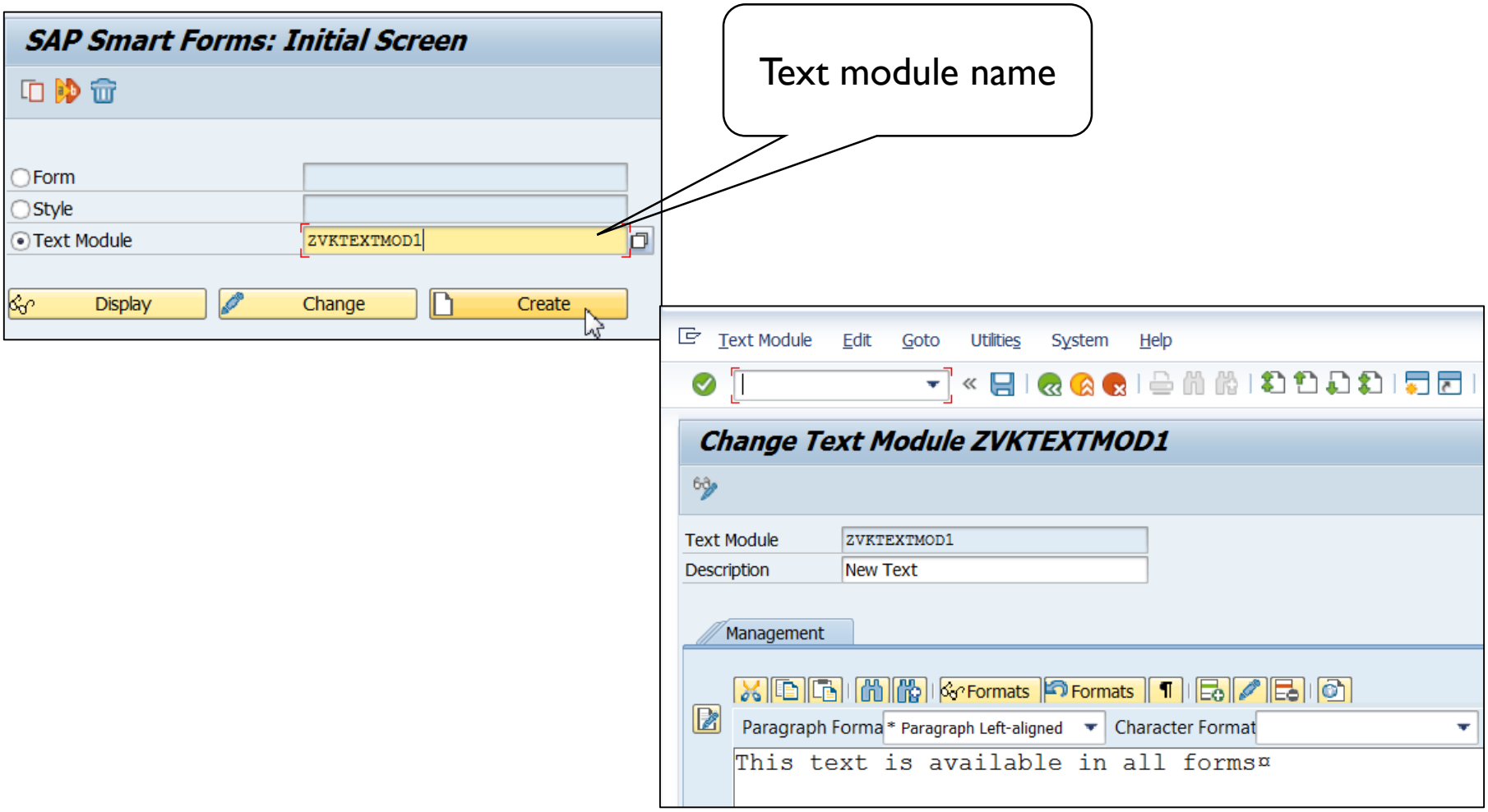
Text modules are used to centrally store texts that are used frequently in forms in the system.

Text modules are included in forms using texts nodes

Allows easy use of text from a text module in several forms

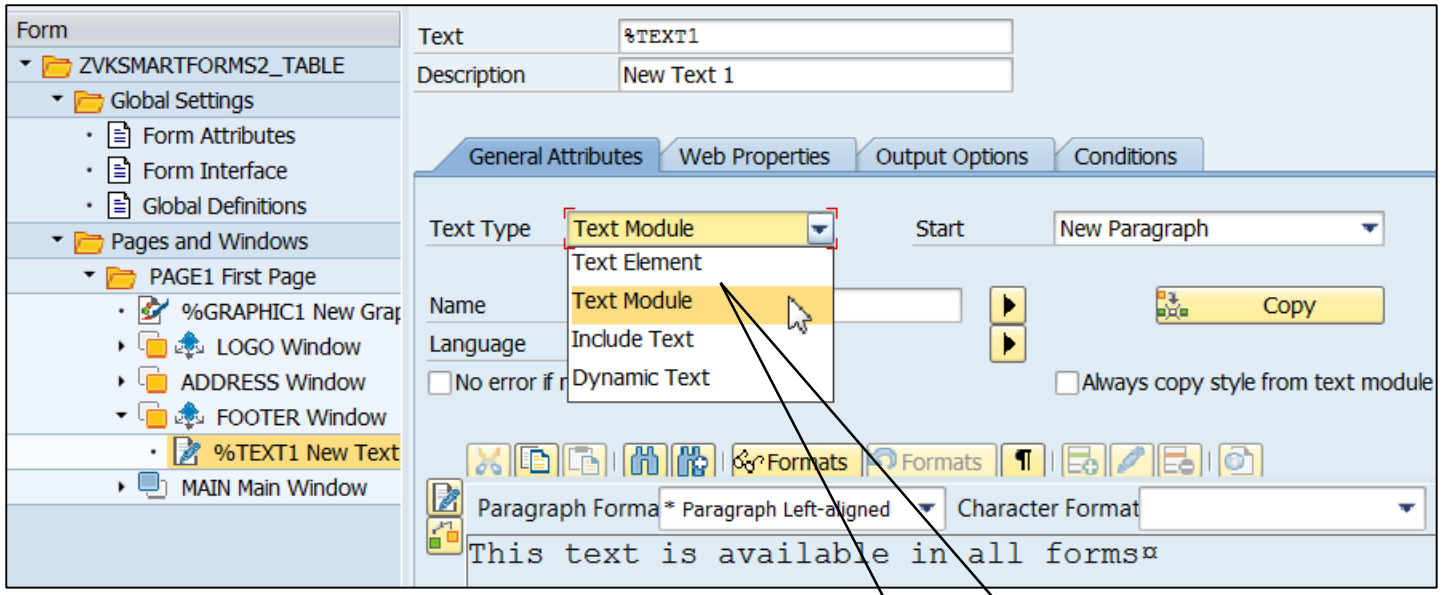
It Can be used across clients

Creating Text Module (Contd.).



Include Text Module in Form

- 1. Create a text node in the navigation tree
- 2. In General attributes tab change the type as 'TEXT MODULE'
- 3. Change text name to the name of the text module



Choose to insert the text created

Demo: Create and include text module in a form

Inserting Addresses:

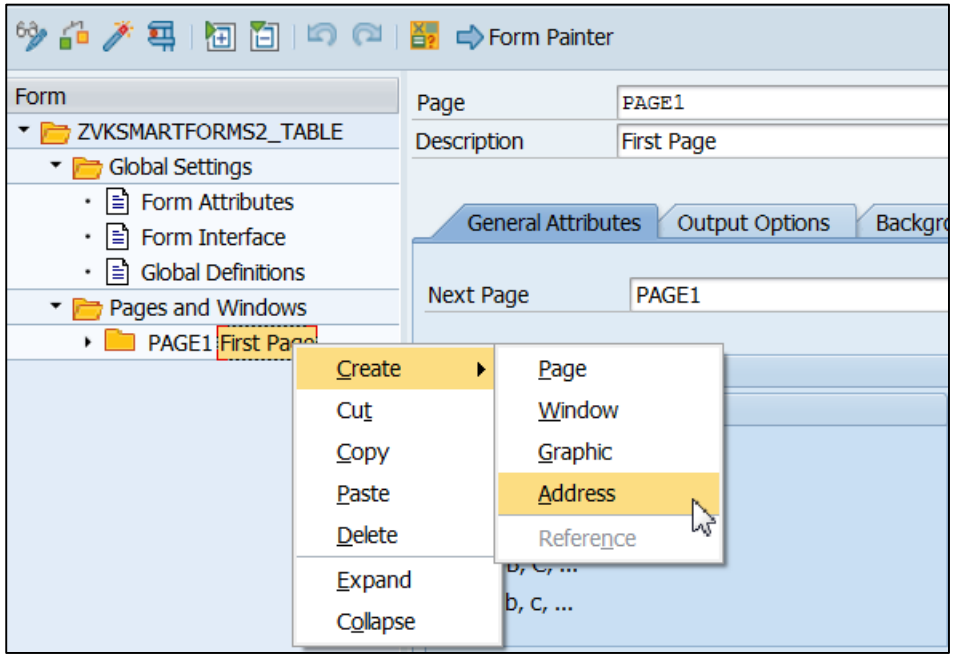
Administered using the Business Address Services (BAS)

According to the postal regulations of the sender country, the address is formatted

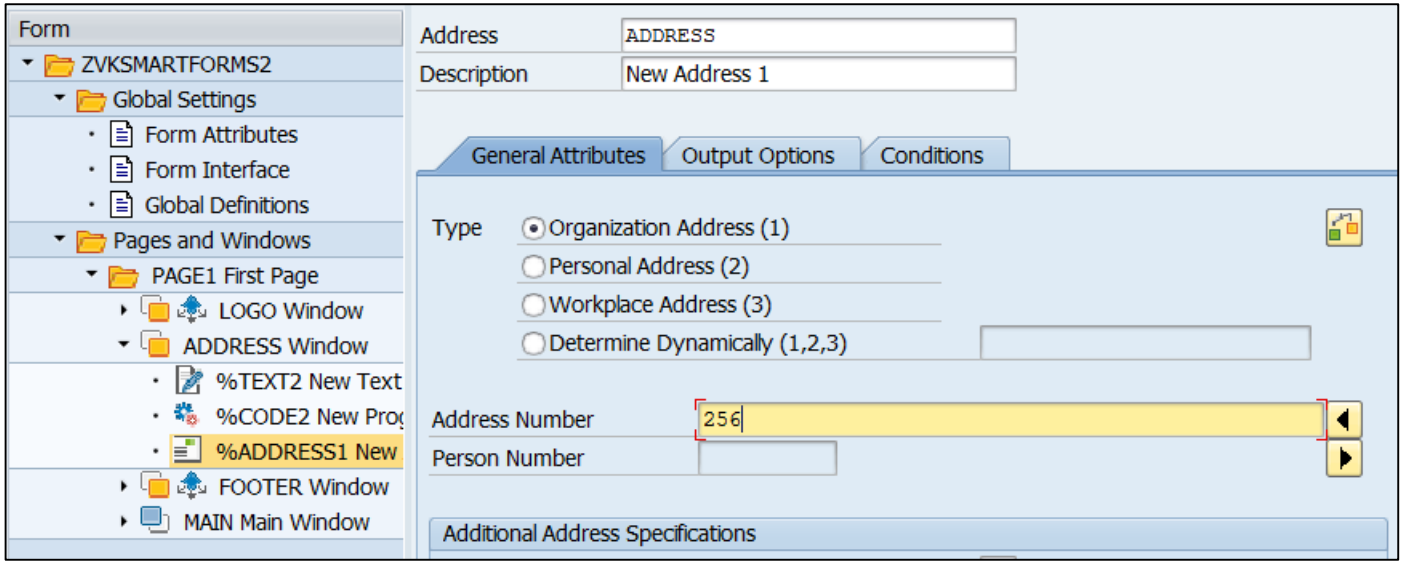
Three address types

- Company addresses(address type 1)
- Personal addresses(address type 2)
- Workplace addresses(address type 3)

Creating Address-type



Address:



Graphics In Smartforms

To import, administer and transport images or graphics

They can be incorporated statically into a form or include them dynamically using an appropriate field

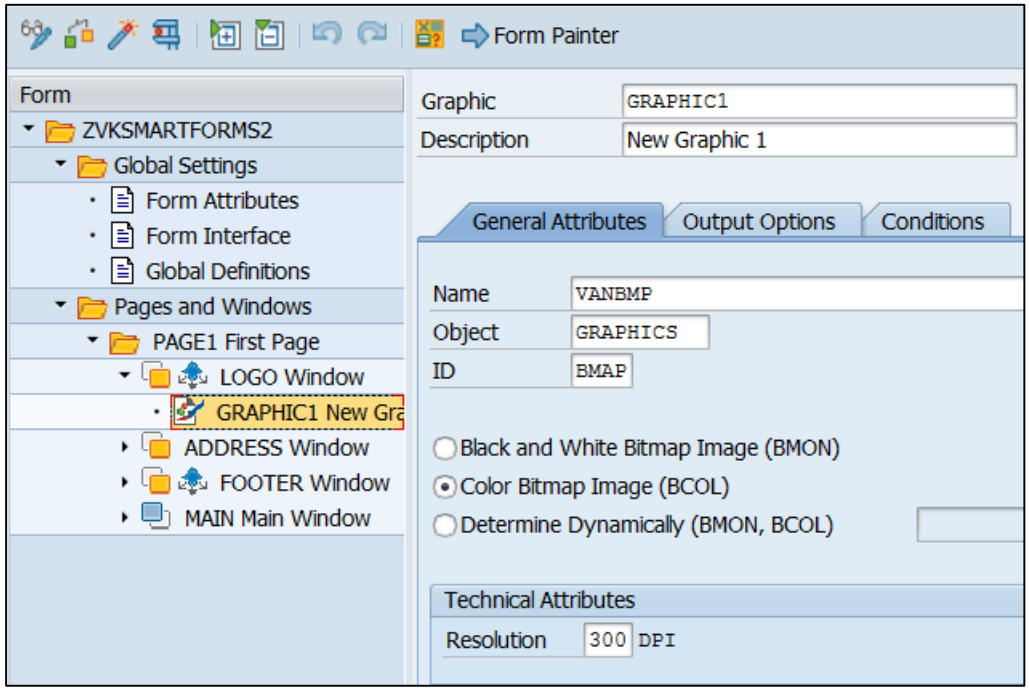
Images can be included in background as well

Demo: Create Address

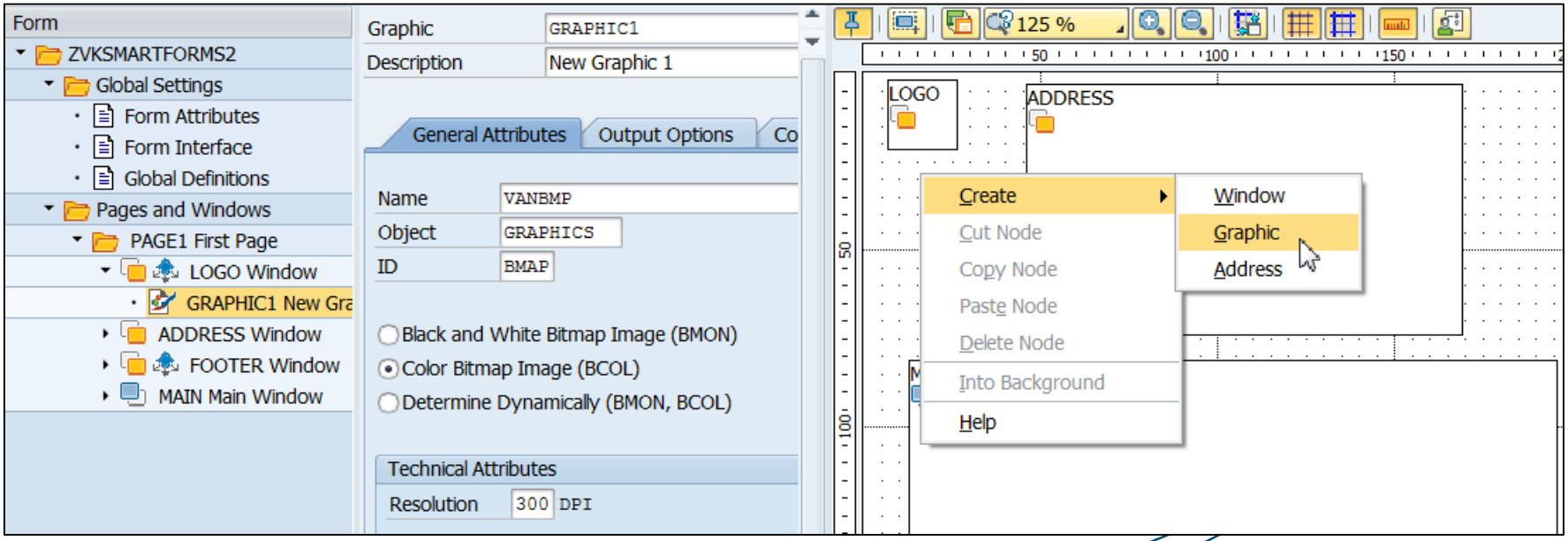
Graphics

*.BMP and *.TIF files can be imported and used in forms.

SE78 - Graphic administration.



Creating Graphics in Smart Forms



Create window to add Images

Demo: Create Graphics in Smartforms

Working with Tables

To display or print contents in tabular form

Node types

- Template node
  - Static - The number of columns and lines are determined before the actual output
- Table node
  - Dynamic - Table size depends on the amount of data selected at runtime

Line Types Specifies

- Width of table line
- The layout of both node types
- Also the width of the individual cells within the table line

Printing tables

The table can be designed independent of the number of lines

The size of the table depends on how much data the application program passes to the form at runtime

Tables cannot be nested.

Output of a table can be divided into

- header
- main area
- footer

Accessing Application Data

The application program reads the data to be displayed on the form as a table from the database and writes it into an internal table

When calling the Smart Form, this internal table is passed to the form interface to access it within the form description

Access the internal table to display it on the form line by line

To accessing several internal table for table output whose entries are interdependent combine loop and table nodes

Reading Internal-Tables

The table is printed in the form line by line as the number of selected entries in the internal table differs

The table node defines a table layout

The Data tab is used to access internal tables, which exists for loop nodes and for table nodes

Procedure to read Internal Tables

Create a work area for the internal table in the global definitions

Go to Data tab of the loop or table node and mark Internal Table else the loop is deactivated

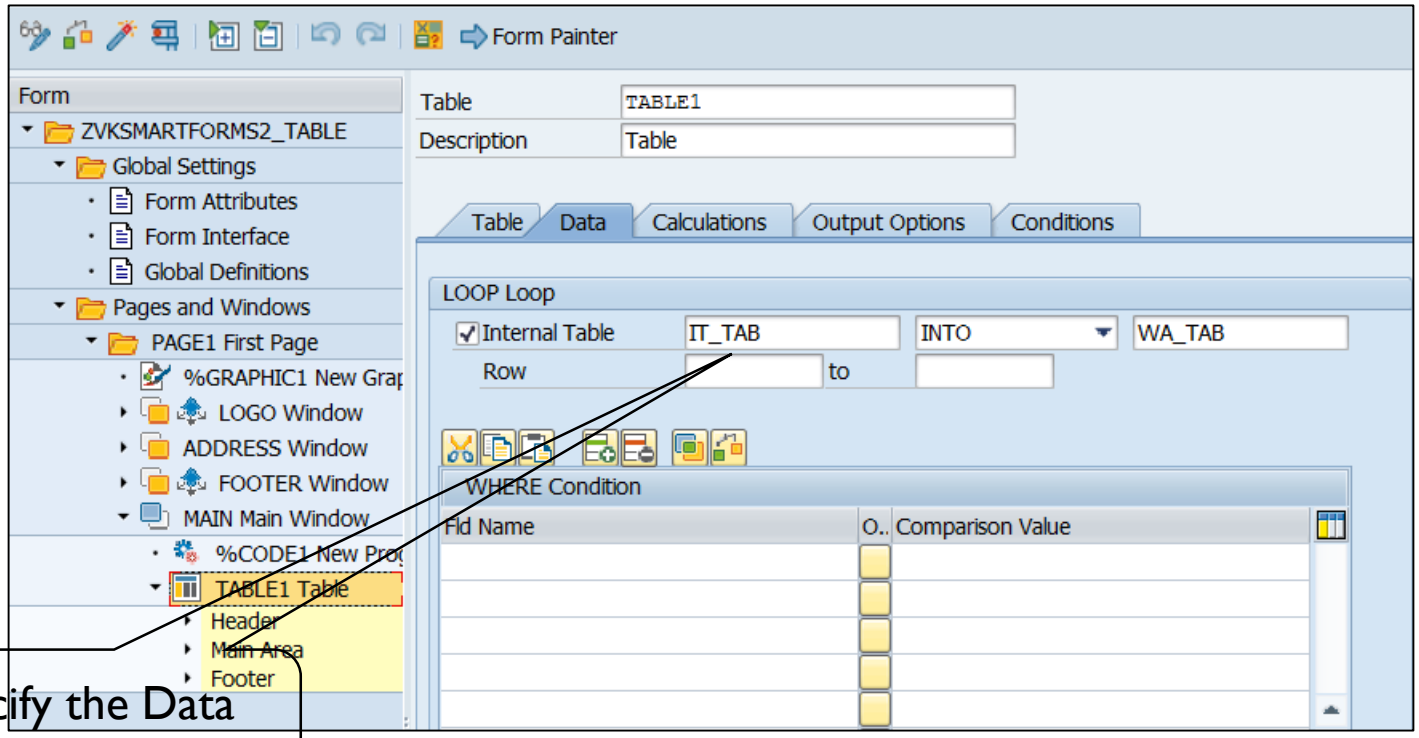
Enter the name of the internal table that is passed at the form interface

Enter the assignment type (INTO or ASSIGNING) and a work area (structure with the same type as the table line or the field symbol)

- If a table with header line is used as work area specify the internal table name again
- If desired, use the input fields Line and To to limit the lines of the internal table that is to be read

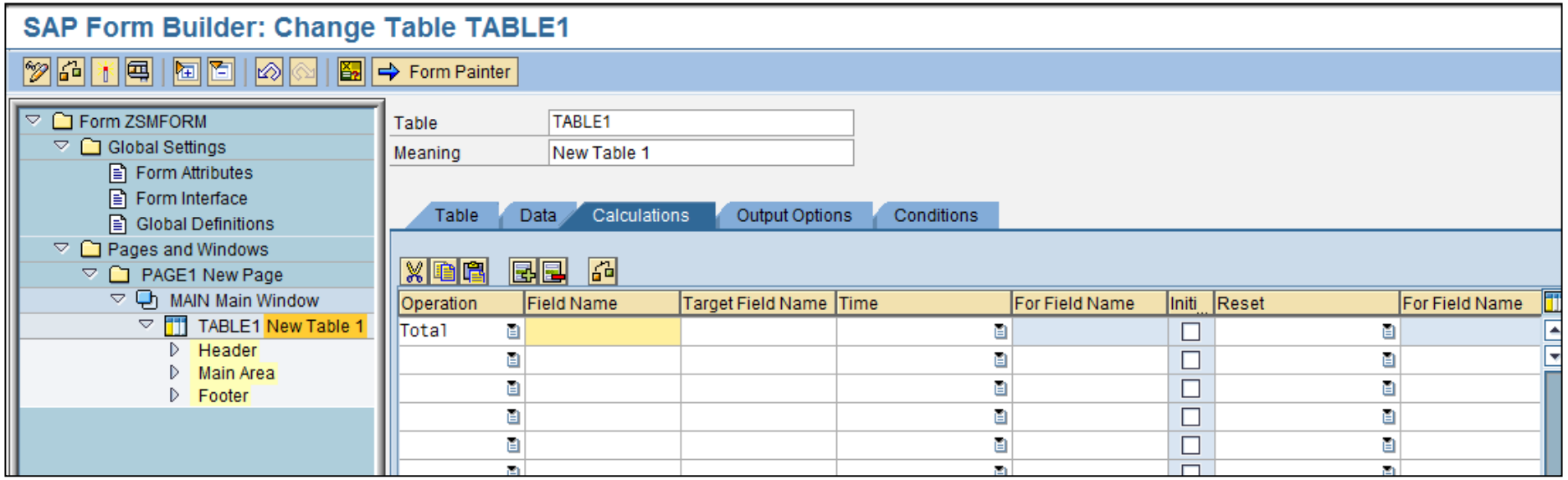
Use the group box WHERE Condition to select a particular part of the data in the internal table

Procedure



Specify the Data source for tables

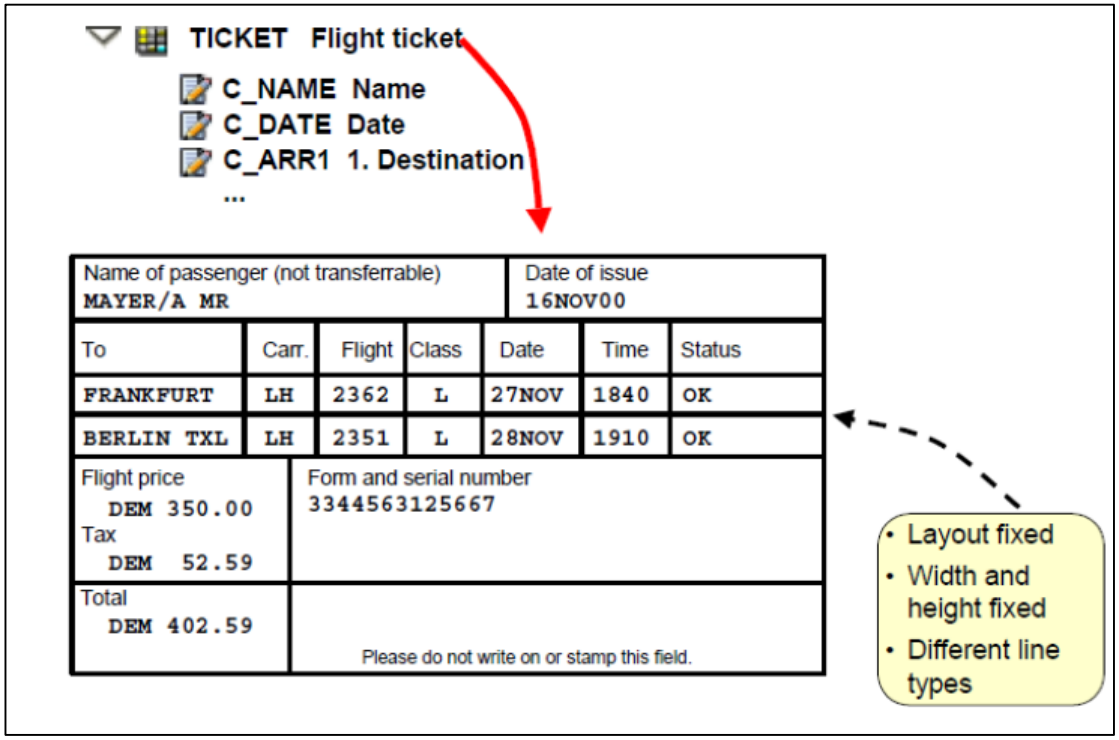
Table Calculations - Procedure



Demo: Create table in smartform

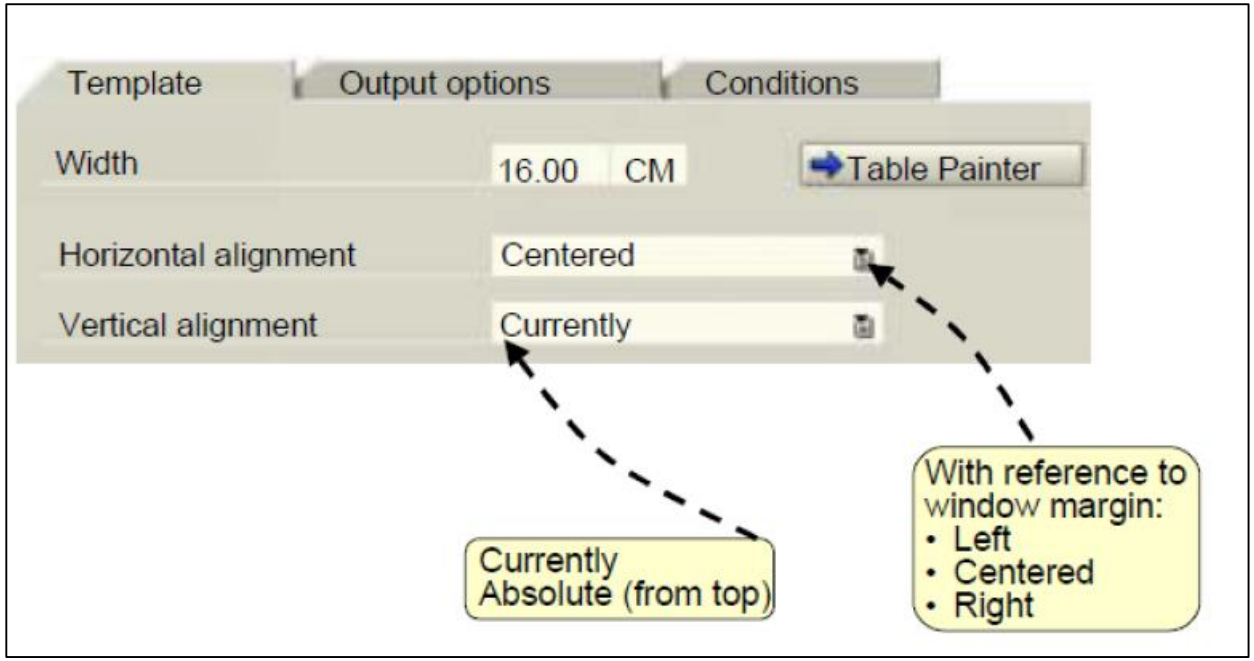
Template

- 1.You use the Template node type to output tables with a fixed layout and size.
- 2.Templates cannot be nested

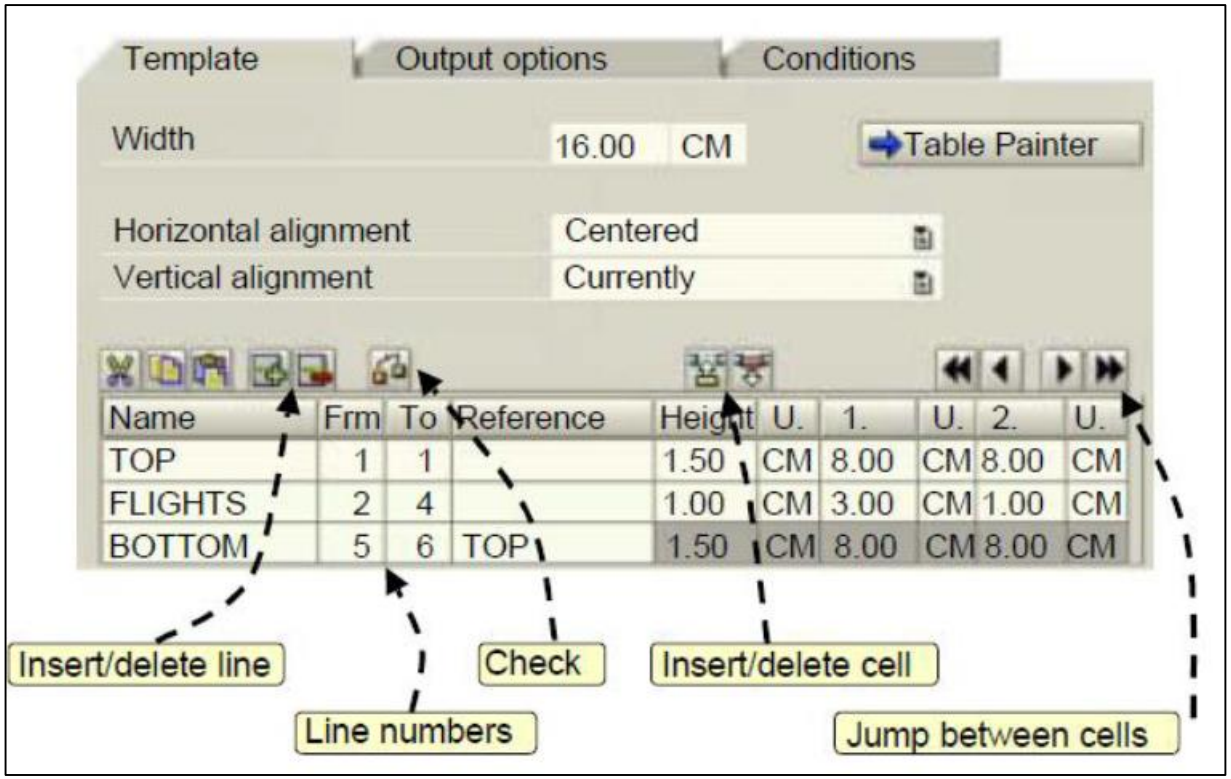


Capgemini Public

Template Layout



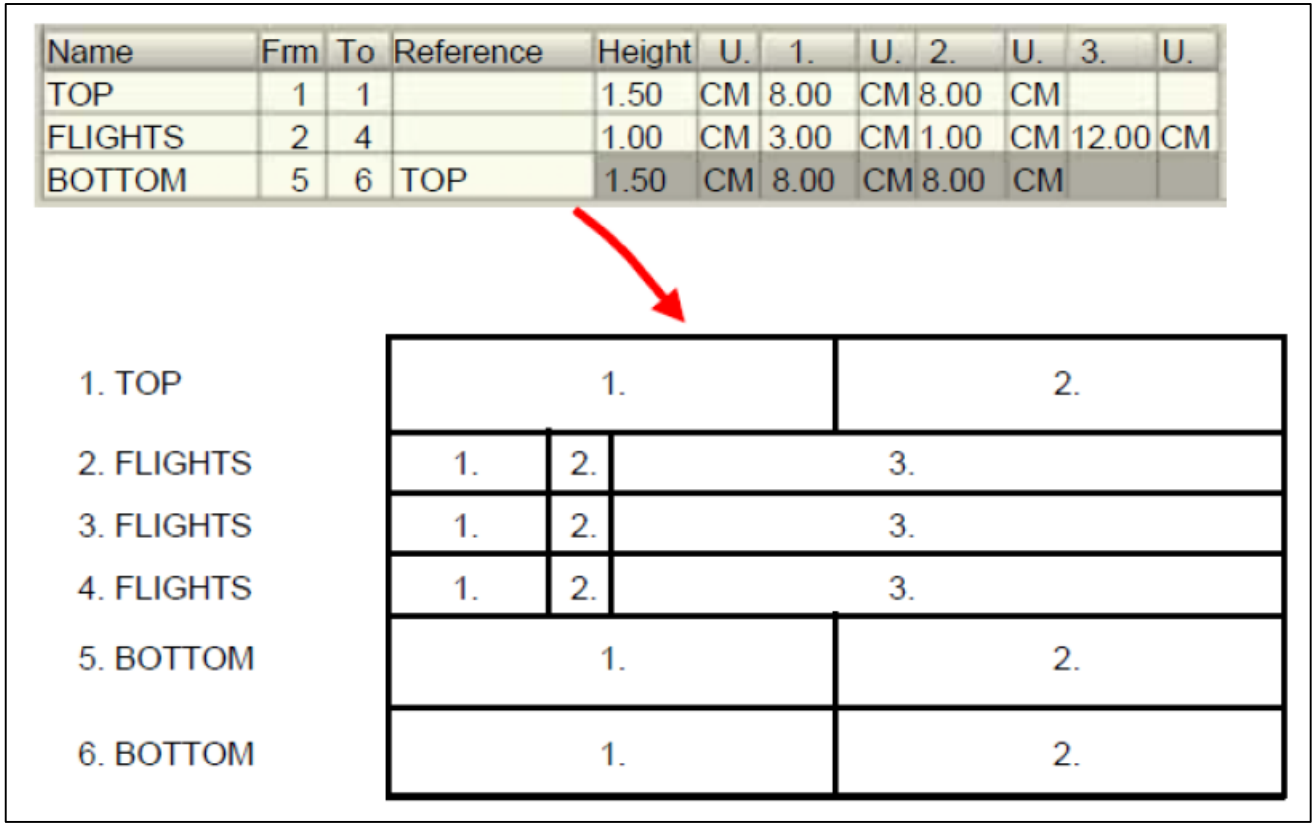
Template Layout



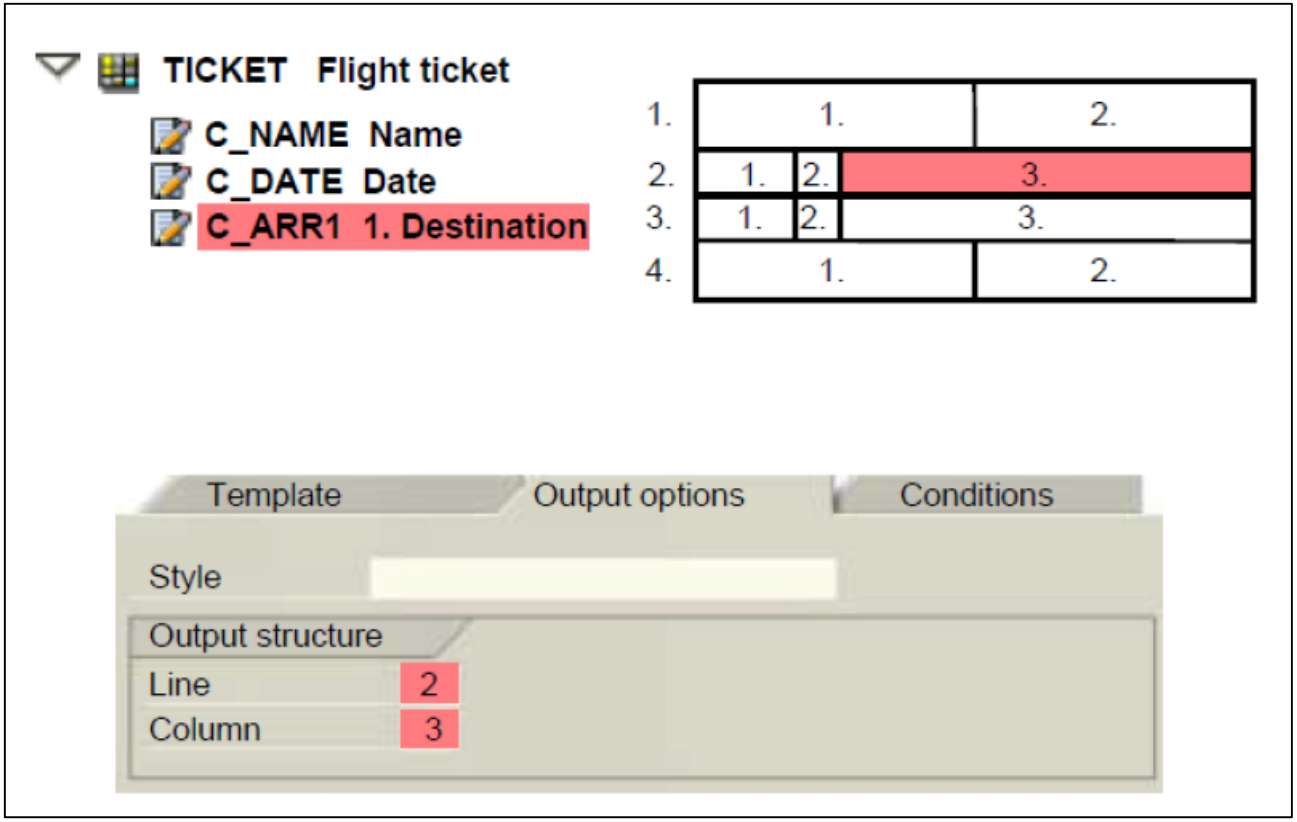
Capgemini Public

Template Layout

Name	Frm	To	Reference	Height	U.	1.	U.	2.	U.	3.	U.
TOP	1	1		1.50	CM	8.00	CM	8.00	CM		
FLIGHTS	2	4		1.00	CM	3.00	CM	1.00	CM	12.00	CM
BOTTOM	5	6	TOP	1.50	CM	8.00	CM	8.00	CM		



Template Layout



Demo: Create template in smartform

Dynamic Page-Break

Page break triggered when the main window of a page is full

Only the contents of main window can spread over several pages

Page-Numbering

&SFSY-PAGE&

- Specifies current page number

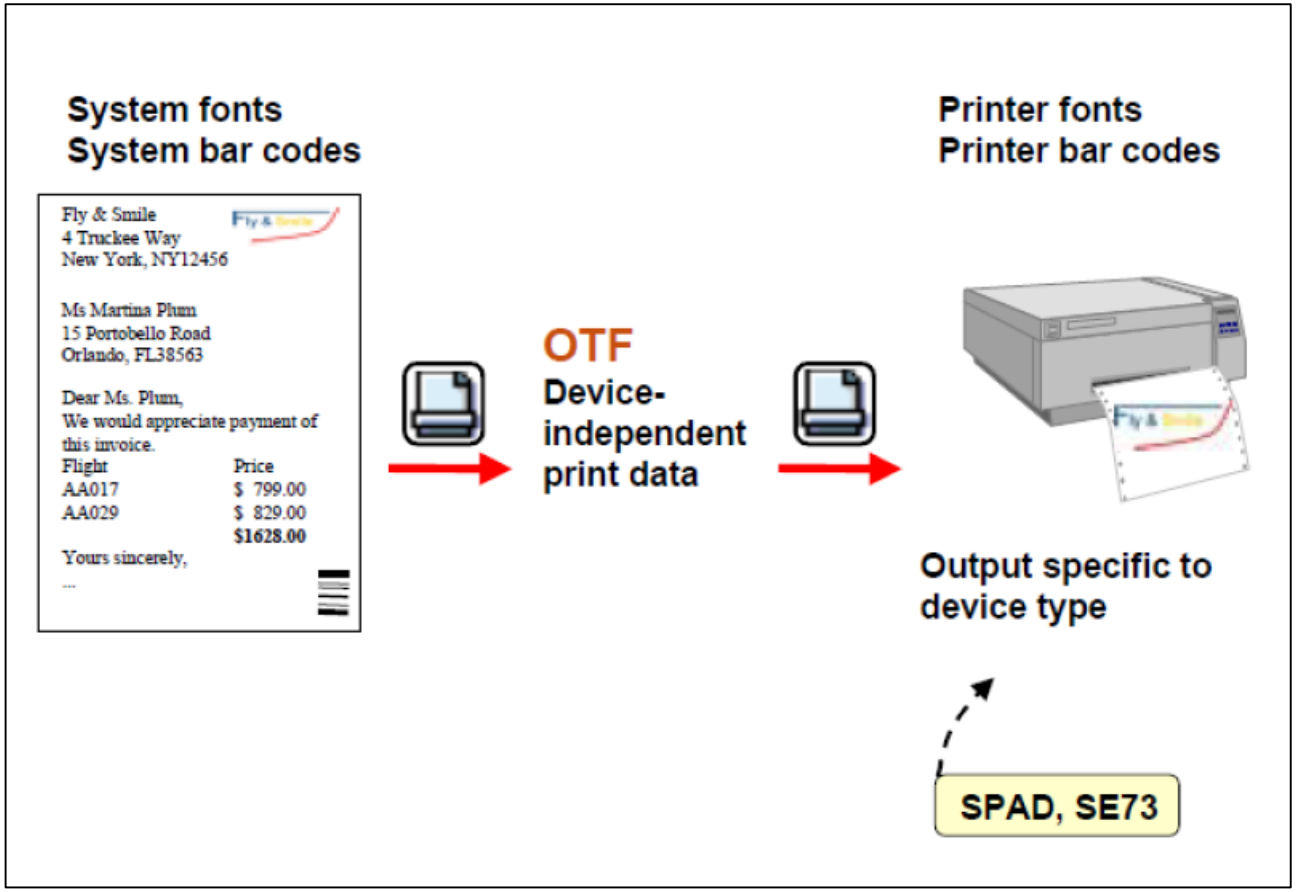
&SFSY-FORMPAGES&

- Specifies total number of pages in the form

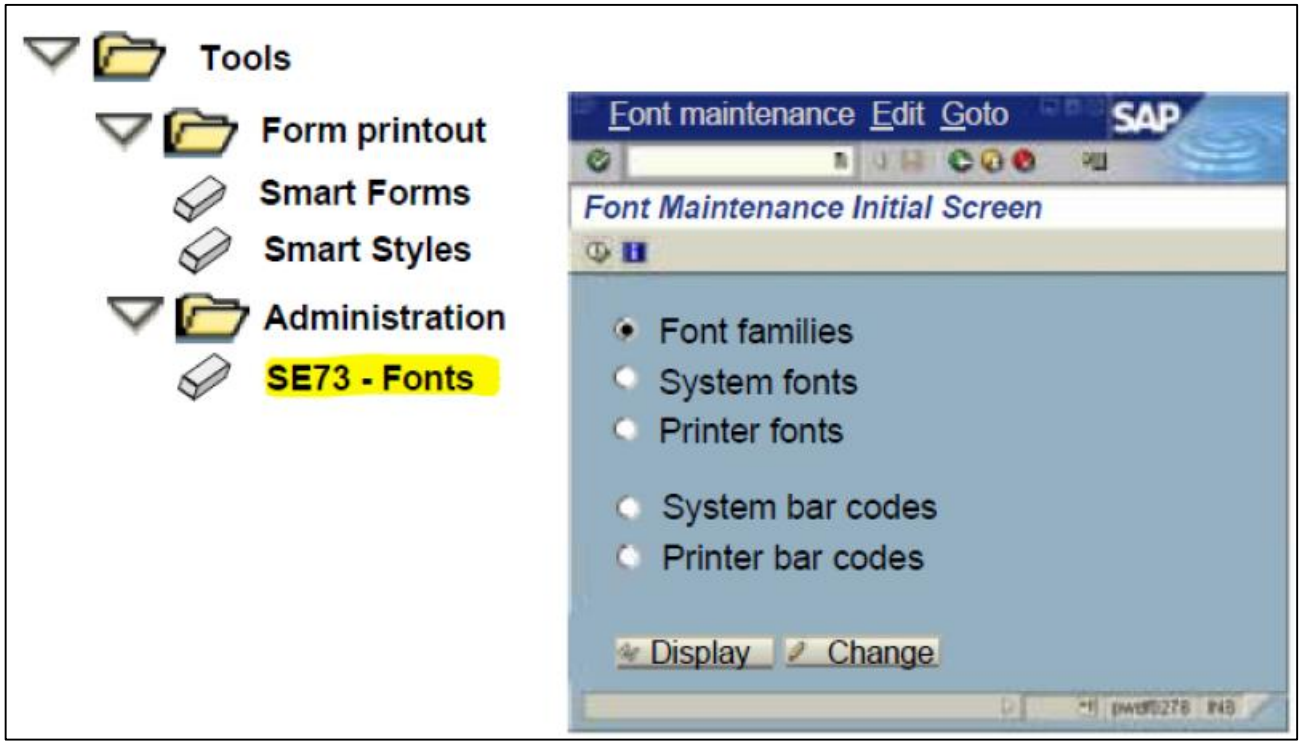
&SFSY-JOBPAGE&

- Specifies total number of pages in all forms in the print job

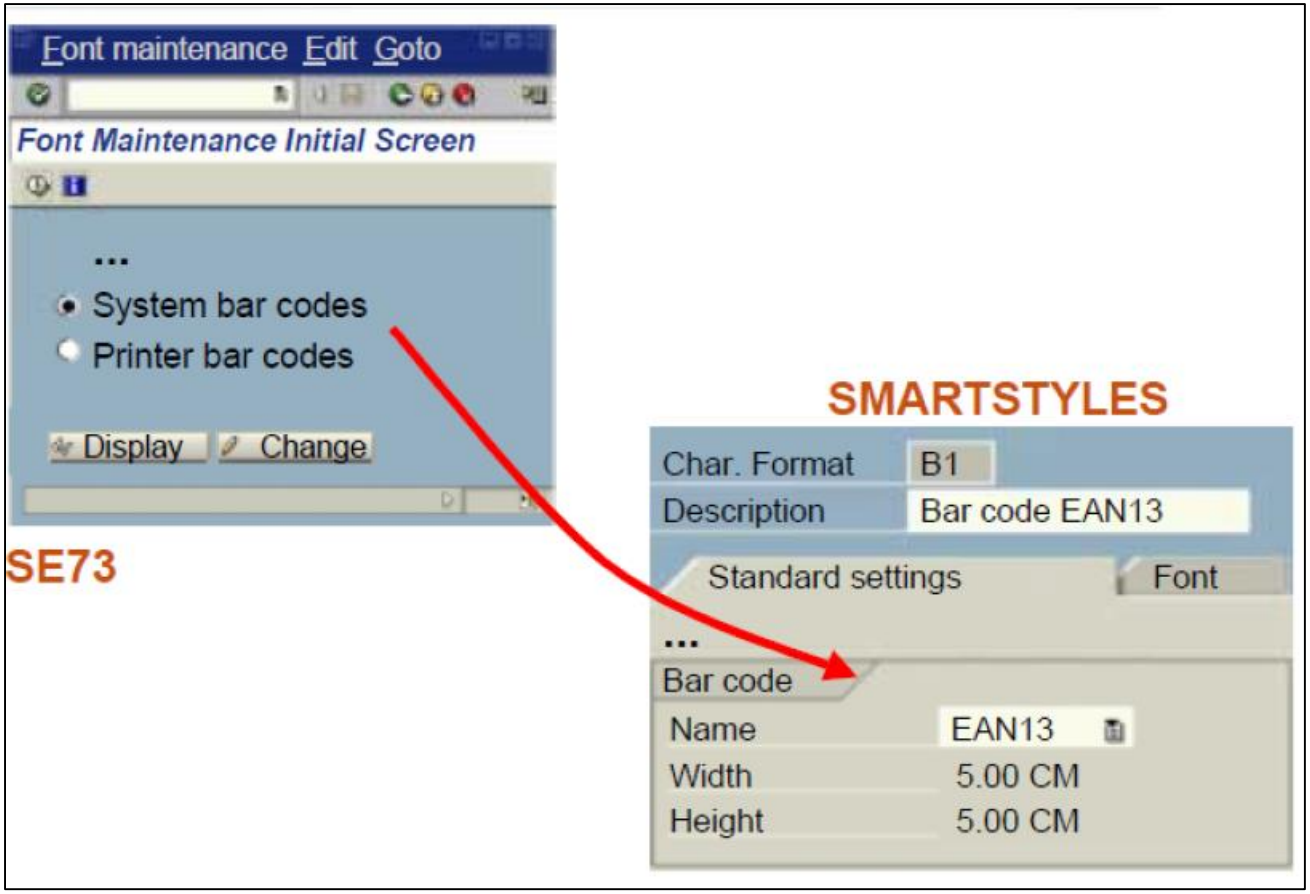
Font and Bar Code Maintenance



Accessing Font Maintenance



Maintaining and Using Bar Codes



OSS Notes for Fonts and Bar Codes

- 0008928 List of supported printers/device types
- 0005196 Printing bar codes with SAPscript
- 0017054 How to copy or change a device type
- 0012462 How can I define a new printer font?
- 0317851 Printing PDF files in 4.6C/4.6B/4.5B/4.0B
- 0201307 TrueType fonts for Smart Forms/SAPscript

Summary

In this lesson, you have learnt:

- The Smartforms Architecture
- SAP Form Builder
- Texts, Addresses and Graphics
- Tables and Templates
- Flow Control
- How to Integrate SmartForms into Application Programs
- Fonts and Bar Codes

Review Question

Question 1._____ is used to transfer application data to the form definition.

Question 2: Text modules are included in forms using _____.