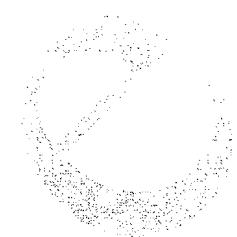


eMax Technologies

Smart Forms



compiled By:

Ganapati Adimulam

What is SmartForms ?

SAP Smart Forms is new print solution introduced in SAP Basis Release 4.6C as the tool for creating and maintaining forms to Print the Business Documents such as Experience Letters,Offer Letters,Purchase Orders,Sales Orders.

In One word , SmartForms is the Smartest way to work with Forms.

Note : All the SAP Script (Old Solution for Printing Forms) functionalities are still Possible in SMARTFORMS .

Note : Event SAP Scripts can be converted into Smart Forms Also.

Note : In Smartforms most of the functionalities can be achieved simply through Clickings(Double Click , Right Click ...).

Features Of SAP Smart Forms

OR

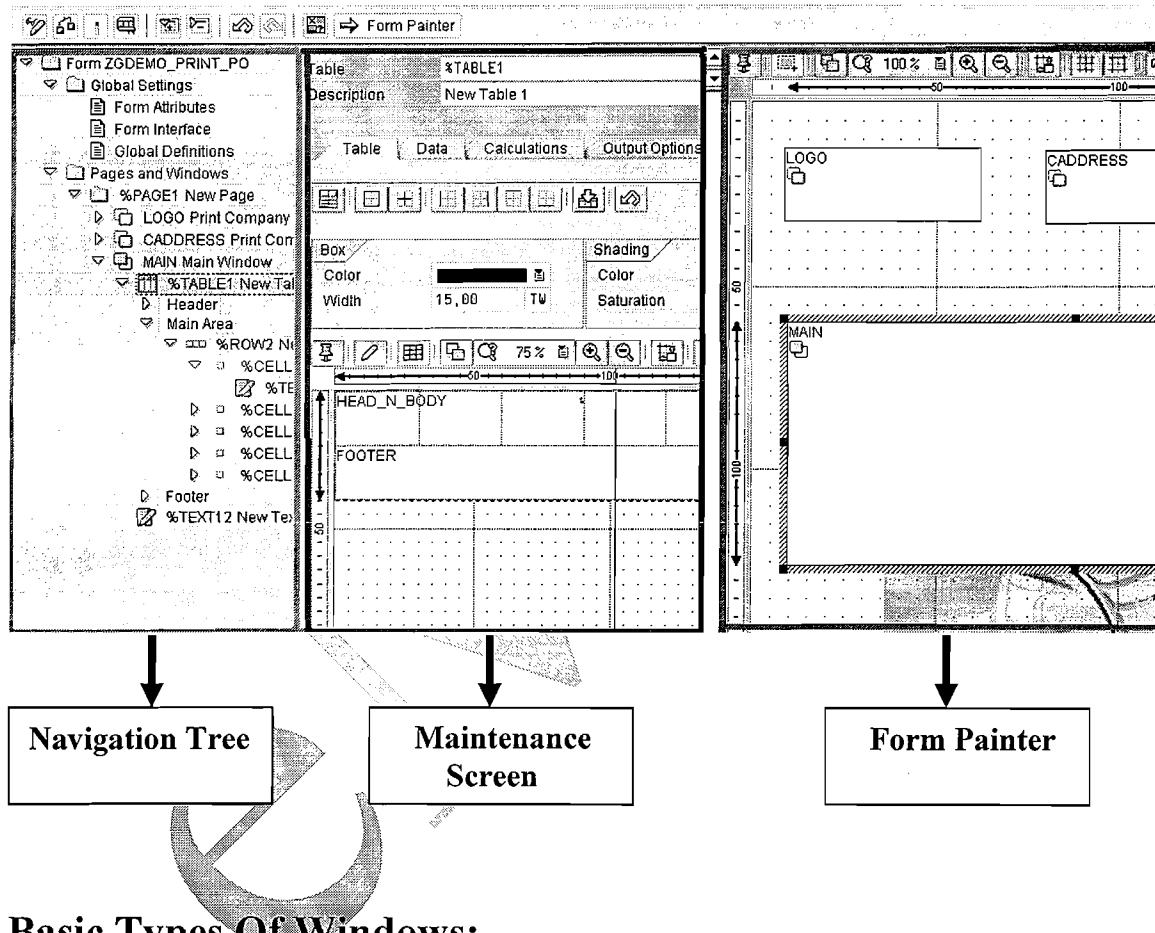
Differences Between SAP Script and SmartForms

SNO	SAP Script	SmartForms
1	Client Dependent	Client Independent
2	All the Pages of the FORM should have the same page format.	Each page can have it's own page Format
3	Pages cannot be Printed with Background Pictures.	Can be
4	Text Cannot be Printed with Colors	Can be
5	No Tool for Printing Data in Tabular Format	Tool Table Painter is available for printing data in Tabular Format.
6	Data Transfer from Program to Layout is Window by Window through Function Module <code>WRITE_FORM</code>	Data Transfer from Program to Layout is at once through the Special and Unique Function Module generated automatically at the time of Activating the Smartform.
7	Paragraph and Character formats are not re usable.	Paragraph and Character formats are re usable via SMARTSTYLES.

Structure of a Form

A form consists of pages, output areas, addresses, graphics (such as company logo, Scanned Signature Of HR Manager), and data or text contents. Within an output area, you can use static or dynamic tables to display data or texts in lines, columns, or cells. To further structure and format text and data, use paragraphs with the paragraph and character formats. In SAP we call output areas as Windows.

A Form Consists Of :



Basic Types Of Windows:

- i) Main Window
- ii) Secondary Window
- iii) Copies Window
- iv) Copies Window

You can position windows anywhere on a page, even overlapping. You can position the same window (determined by the technical name on several pages of a form, so that the same contents are displayed on all these pages. You can choose a different size for the window on each page, **except for the main window (Height Should be same in all the Pages).**

Main 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 , the system continues displaying the text in the main window of the Same Page/next page. It automatically triggers the page break.

You can define only one window in a form as main window. AND the Same MAIN Window can be Placed up to 99 times on the same page which ranges from MAIN00 to MAIN98.The Data on MAIN Window always starts printing from MAIN00 and after Completely filled the text will be extended to MAIN01, MANIN02 etc.

Note : The main window must have the same width on each page and with in page also when MAIN01,MAIN02 exists, but the height can differ.

Note : A page without main window must not call itself as next page, since this would trigger an endless loop. In such a case, the system automatically terminates after three pages.

Note : MAIN Window is used to Print the Dynamic Length Data such as Purchase Order Items, Sales Order Items etc

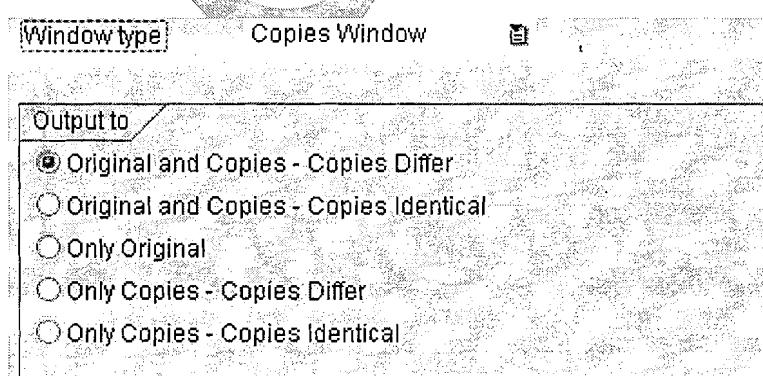
Secondary Windows : In a secondary window you display text and data in a predetermined output area. There is no flow text display with page break. If you position a secondary window with the same name on several pages, the system displays the contents of this secondary window on each page.

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

Note : Secondary Window is used to Print the Fixed Length Data such as Company Address, Vendor Address etc.

Copies Windows : Copies Window is to define an output area for the print output, whose content you want to appear either **only on the copy or only on the original**. This allows you to flag copies as copies when the form is printed.

Note : Copies Window helps to determine where to print the inferior nodes of a copies window:



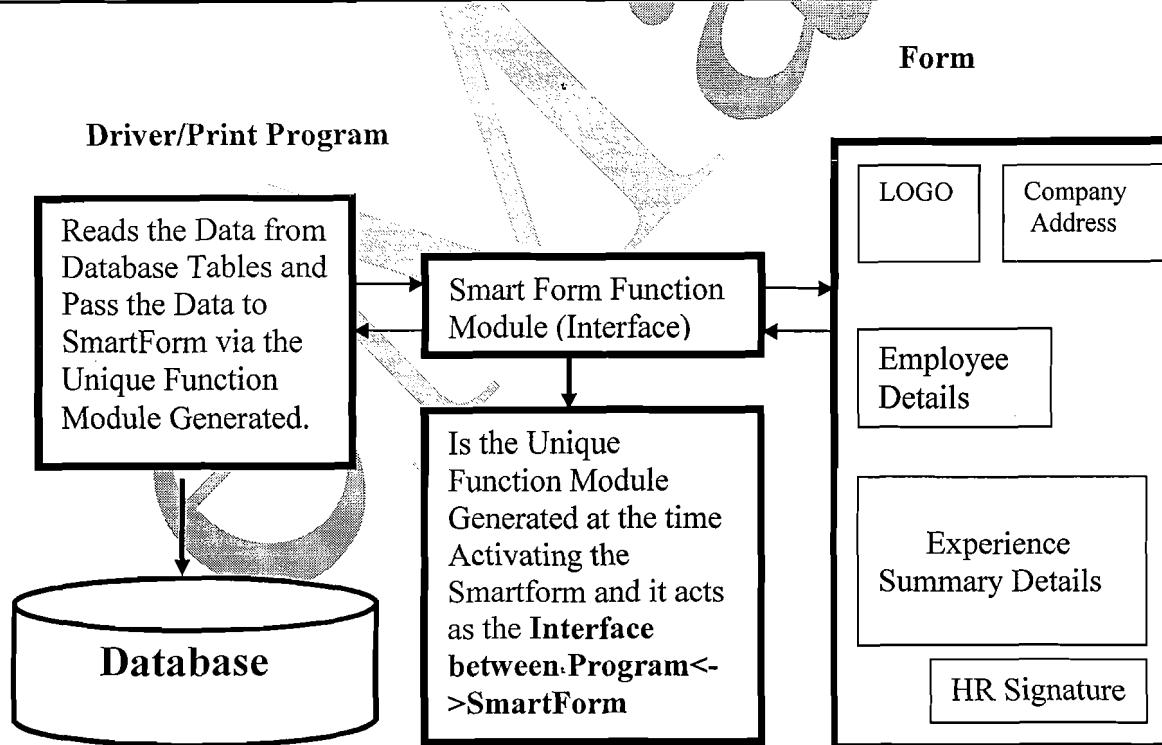
Note : We can use the system fields **SFSY-COPYCOUNT** or **SFSY-COPYCOUNT0** to query whether the current output is the original or, respectively, which number the copy has.

- If you mark **Identical Copies** SAP Smart Forms buffers the output to increase performance. Within such a window, any queries of the system fields **SFSY-COPYCOUNT** or **SFSY-COPYCOUNT0** have no effects.
- If you mark **Different Copies** SAP Smart Forms generates the contents again for each copy. You use this option if, for example, you want to number your copies consecutively (Copy 1, Copy 2, Copy 3, and so on).

Final Window : You may want to display or query values on the first page that are determined only during processing. For example, you may want to name the grand total in the letter text of an invoice. However, this amount is determined only after listing all individual items. Or you may want to query on the first page within a condition the total number of pages, which the system calculates only after processing all pages.

In such a case, you use the final window: Processing first skips all windows of this type in the tree and works its way to the end of the tree. Only after the actual processing is finished, the final windows are processed in the order in which they appear in the tree (from top to bottom). Now any information is available that is known only at the end of the form processing.

Communication Between Driver Program ↔ Smart form :



Note : When you activate the Smart Form, the system generates a unique function module that encapsulates all attributes of the Smart Form. As soon as the application program calls the function module, the Smart Form uses the module's interface (which corresponds to the form interface) to transfer data from Program to Layout.

Note : We Can still write all the data retrieval statements to select further data during form processing. However, we should not use this method of data retrieval. Especially When We print mass forms, performance will be degraded considerably.

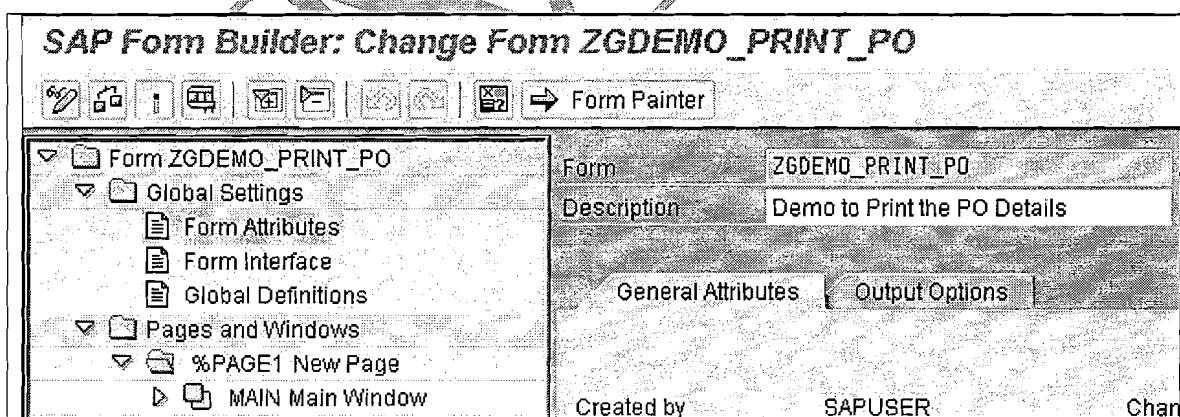
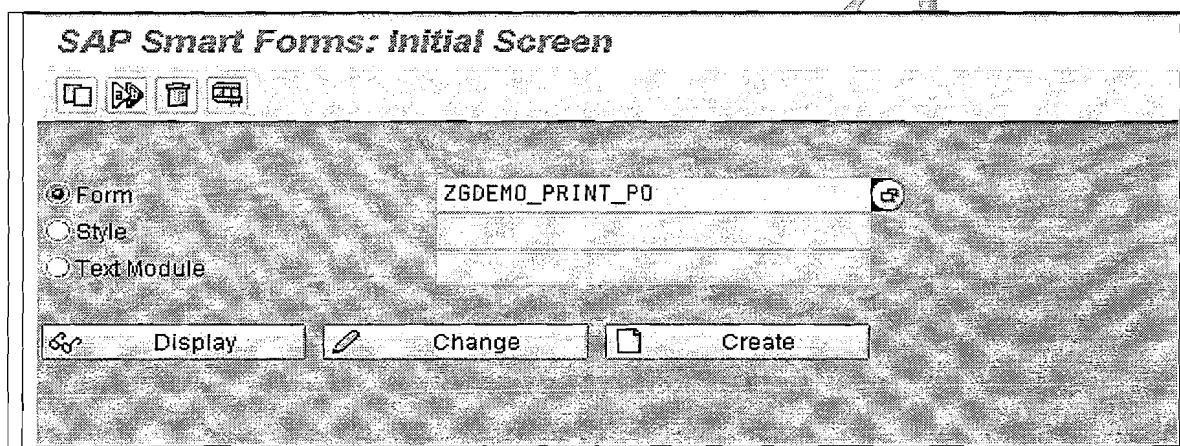
Note : Use the Driver Program to Retrieve the data and Form to Print the Data Always.

Creating Forms and Printing Data :

Steps to Create Smart Form:

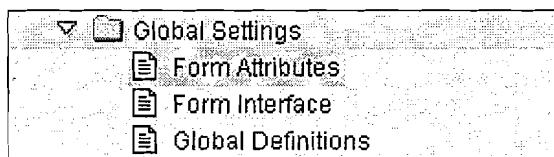
Execute Transaction SMARTFORMS

Select Form and Enter the Name for the form and Click on Create



Enter the Description

Notice the Global Settings



Form Attributes in Detail : Header Data of the form.

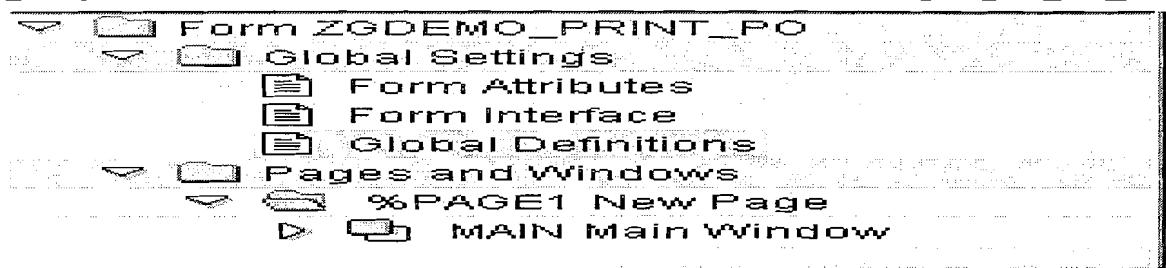
Form	ZGDEMO_PRINT_PO	Active					
Description	Demo to Print the PO Details						
General Attributes		Output Options					
Created by	SAPUSER	Changed by	SAPUSER				
Date	12.04.2008	Date	12.04.2008				
Time	15:49:55	Time	16:56:17				
Package	\$TMP						
Language Attributes <table border="1"> <tr> <td>Language</td> <td>EN</td> </tr> <tr> <td>Translate</td> <td> <input checked="" type="radio"/> To All Languages <input type="radio"/> To Selected Languages <input type="radio"/> Do Not Translate </td> </tr> </table>				Language	EN	Translate	<input checked="" type="radio"/> To All Languages <input type="radio"/> To Selected Languages <input type="radio"/> Do Not Translate
Language	EN						
Translate	<input checked="" type="radio"/> To All Languages <input type="radio"/> To Selected Languages <input type="radio"/> Do Not Translate						
Access <table border="1"> <tr> <td><input type="checkbox"/> Restricted Language Control</td> </tr> <tr> <td><input type="checkbox"/> Only Process Installed Languages</td> </tr> </table>				<input type="checkbox"/> Restricted Language Control	<input type="checkbox"/> Only Process Installed Languages		
<input type="checkbox"/> Restricted Language Control							
<input type="checkbox"/> Only Process Installed Languages							

Form Interface in Detail : Smart Form Function Module Interface.

Form	ZGDEMO_PRINT_PO	Active			
Description	Demo to Print the PO Details				
Import Export Tables Exceptions					
     					
Parameter Name	Type Assignment	Associated Type	Default Value	Optional	Pass
ARCHIVE_INDEX	TYPE	T0A_DARA		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ARCHIVE_INDEX_TAB	TYPE	TSFDARA		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ARCHIVE_PARAMETERS	TYPE	ARC_PARAMS		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CONTROL_PARAMETERS	TYPE	SSFCTRLDP		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MAIL_APPL_OBJ	TYPE	SWTOBJID		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MAIL_RECIPIENT	TYPE	SWTOBJID		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MAIL_SENDER	TYPE	SWTOBJID		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
OUTPUT_OPTIONS	TYPE	SSFCOMPDP		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
USER_SETTINGS	TYPE	TDBBOOL	X	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Global Definitions in Detail: Form Variables, Which Can be accessed from any window of the Form.

Notice that by default , PAGE1 and Window MAIN is Created .



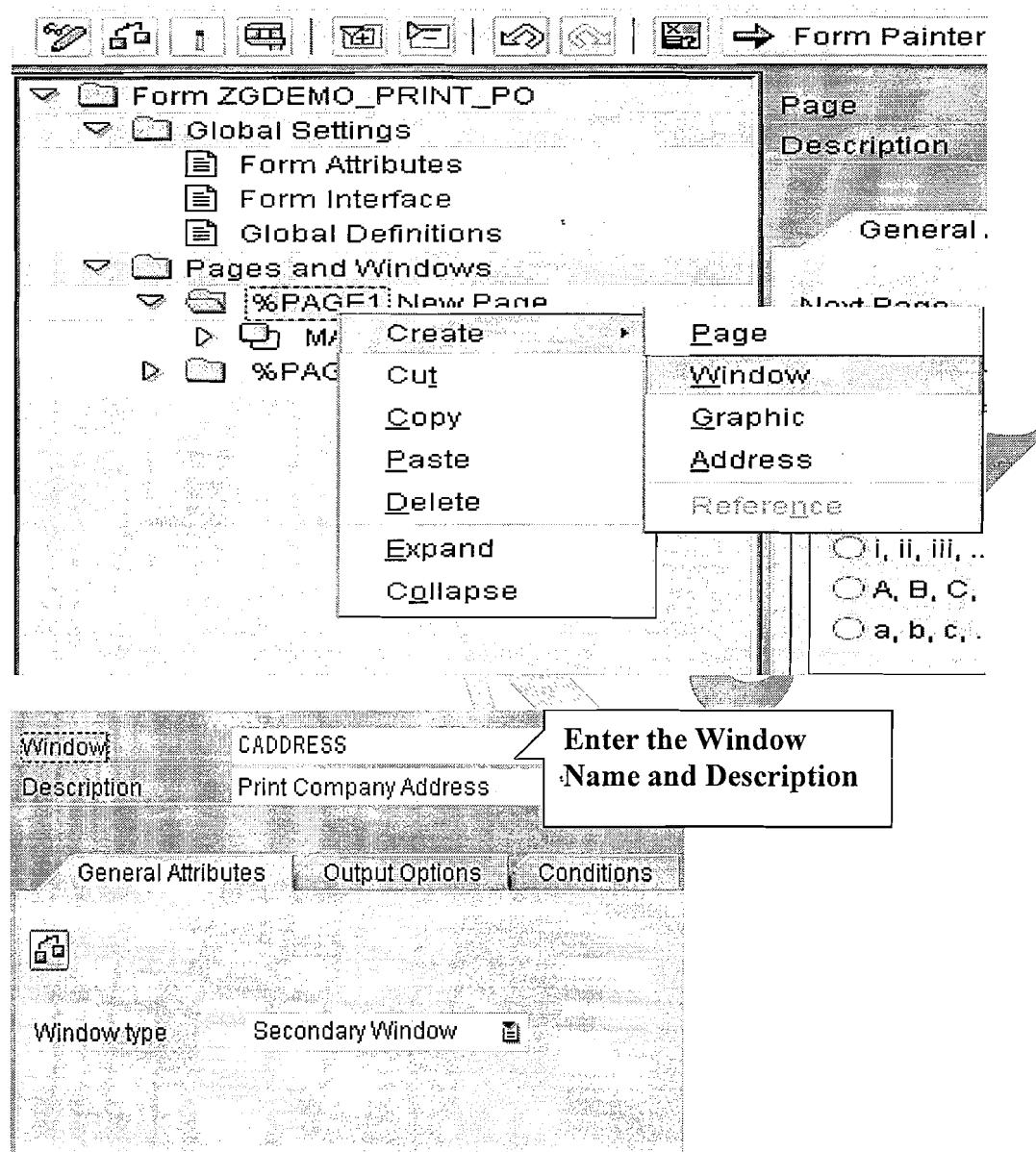
To Create Further Pages/Windows

For Pages : Right Click On the Page ->Create->Page

This screenshot shows the 'Pages and Windows' section of the application. A context menu is open over a page named '%PAGE1 New Page'. The menu options include Create, Cut, Copy, Paste, Delete, Expand, and Collapse. A sub-menu for 'Page' is displayed with options: Page, Window, Graphic, Address, and Reference. Below the main window, there's a form with fields for 'Page' (set to '%PAGE2'), 'Description' (set to 'New Page 2'), and tabs for 'General Attributes' and 'Output Option'. A callout bubble points to the 'Page' field with the instruction: 'Change the PAGE name , Description and Next Page if required Else Continue with Default things.' At the bottom, there's a 'Page Counter' panel with sections for 'Format' (radio buttons for 1, 2, 3, ..., I, II, III, ..., I, ii, iii, ..., A, B, C, ..., a, b, c, ...) and 'Mode' (radio buttons for Initialize Counter, Increase Counter, Leave Counter Unchanged, and Page and Overall Page Unchanged).

For Windows : Right Click On the Page ->Create->Windows

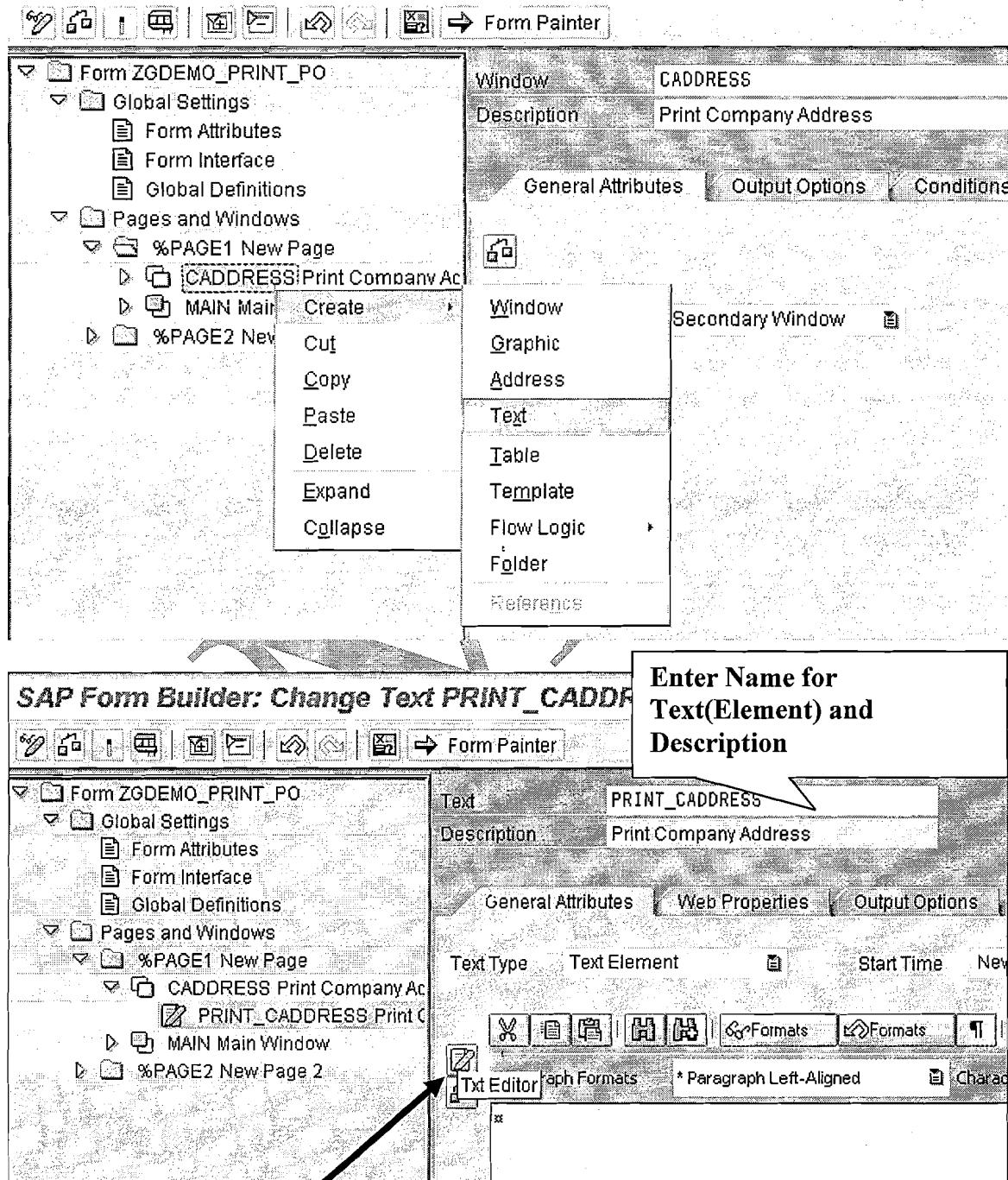
SAP Form Builder: Change Page %PAGE:



Note : Repeat the above Procedure for each New Page/Window.

To Print the Text On the WINDOW :

Displaying of all texts in the form using text nodes. The only exception is addresses, which are displayed using their own node.

Right Click On the respective Window and Create -> Text

Click On Text Editor

Change Smart Form texts: PRINT

		1	2	3	
*	CCODE	:	&WA_T001-BUKRS&		
*	NAME	:	&WA_T001-BUTXT&		
*	CITY	:	&WA_T001-ORT01&		
*	COUNTRY	:	&WA_T001-LAND1&		
*	Web	:	SAP@eMAX.COM		

Enter the Data to be Printed .

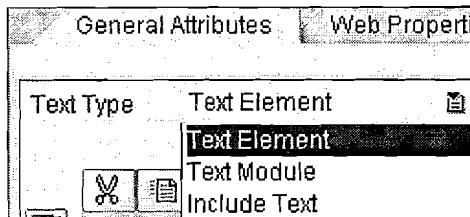
Positioning the Text

The position of the text depends on the superior node (for example, you can assign text to a table cell) as well as on the processing sequence in the tree structure. In addition, you determine the output of a text node in relation to the preceding node on the same level. You can append the text directly to the output of this node or start it in a new paragraph or a new line (by choosing the *Beginning* radio button). **For two subsequent text nodes, this has the following effects:**

Linked Text Nodes

Setting for the Beginning of the Second Paragraph	Description
New Paragraph	Text output starts with a new paragraph. Formatting of the text depends solely on the paragraph and character formats of the second text node.
New Line	Text output starts with a new line in the same paragraph of the preceding text node. SAP Smart Forms keeps the paragraph format of the preceding text node.
Append Directly	Text output starts immediately after the last output of the preceding text node. SAP Smart Forms keeps the paragraph format of the preceding text node.

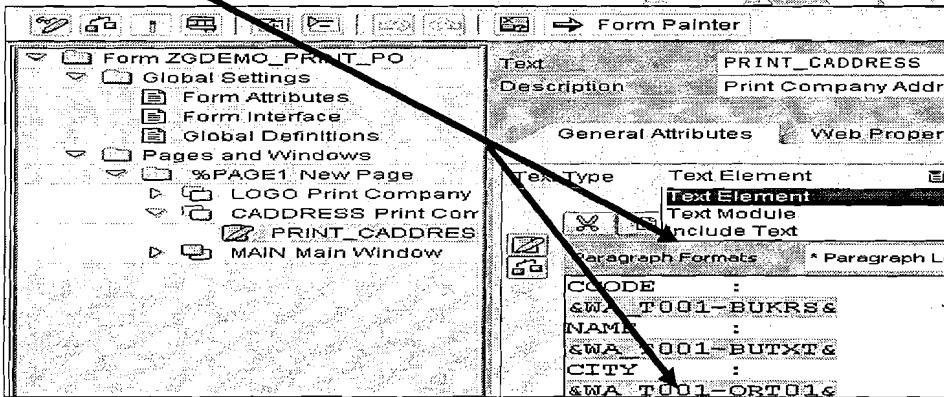
Types of Texts:



- Text element** : is the Default TEXT Type .
- Text module** : To include a text module
- Include text** : To include an existing SAPscript text

Text Element in Detail : To include data from the form interface (data from application tables) or system data (date, time) into the text, use system fields or user-defined fields in the text. When processing the form, SAP Smart Forms replaces these fields with the corresponding values.

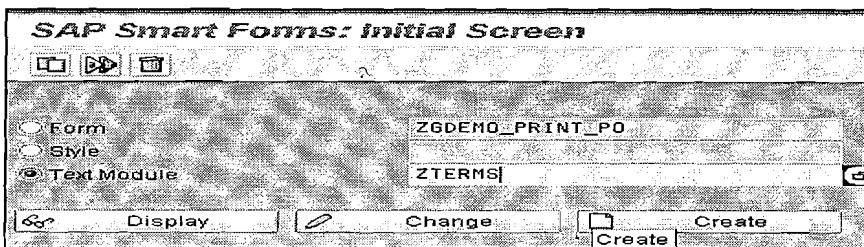
Example :

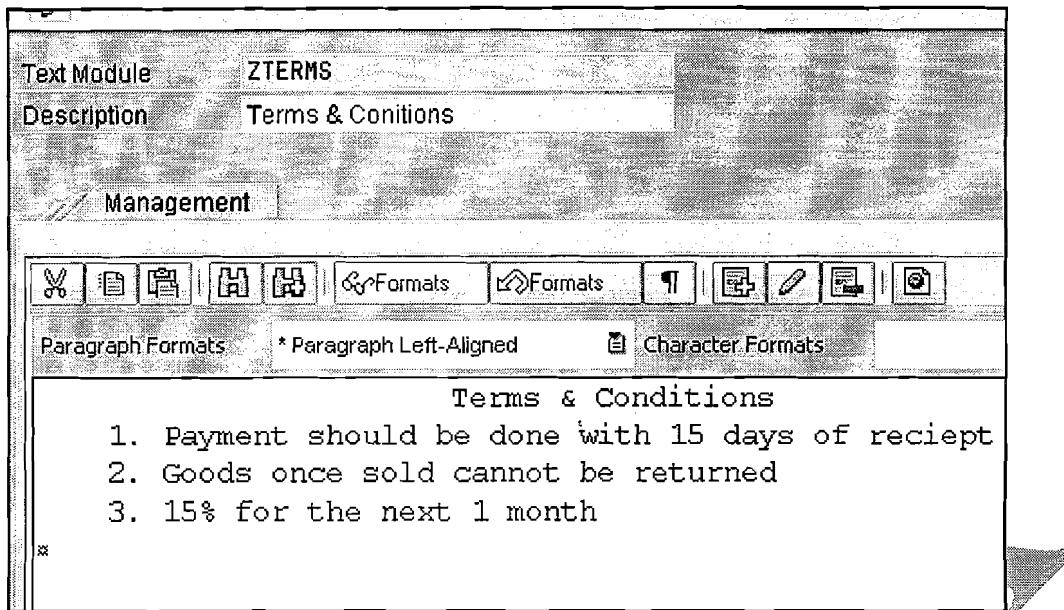


Text Modules You use the text type *Text Module* to refer to an existing **text module** in the system. This allows you to easily use texts from one text module in several forms.

Note: Text modules can be used across clients and are connected to the transport and translation systems.

Steps to Create Text Module: Execute Transaction SMARTFORMS and Select Text Module , Provide the name ->Create





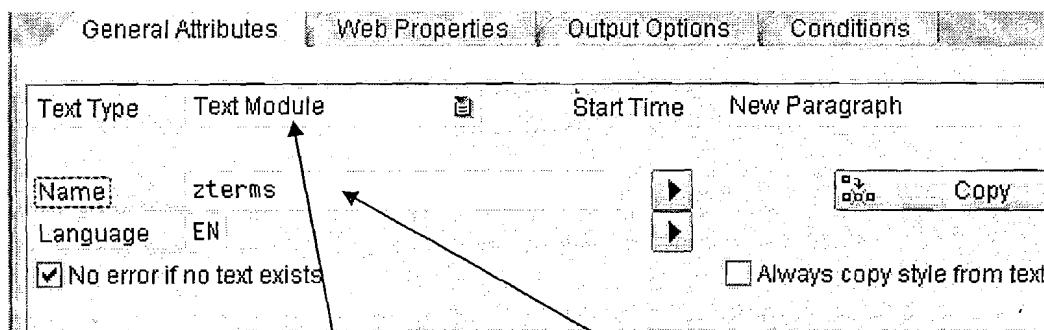
Enter the Description and the Text to be included.

Click On Management Management to maintain the Language Specific Text.

SYSTEM	LAISO	TEXT
	AF	Afrikaans
	AR	Arabic
	BG	Bulgarian
	CA	Catalan
	CS	Czech
	DA	Danish
	DE	German
	EL	Greek
	ES	Spanish
	ET	Estonian
	FI	Finnish
	FR	French
	HE	Hebrew
	HR	Croatian
	HU	Hungarian
	ID	Indonesian
	IS	Icelandic
	IT	Italian
	IN	Japanese

Now this Module Can be used in any no of Forms.

To Include it in the Windows, Create the Node TEXT and Select the type as Text Module



Select the Text type as Text Module , Enter the Text Module Name, Language and

Select No error if no text exists (to Print the form even the Text doesn't exist).

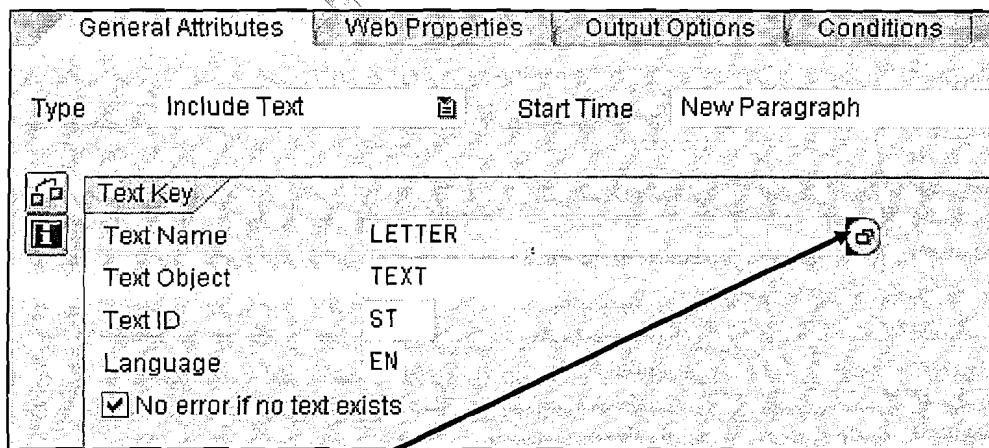
Including SAPscript Texts :

You use the text type **Include Text** of the text node to refer to a **SAPscript Standard text** that already exists in the system. To identify the text, you need the *Text Name*, the *Text Object*, the *Text ID*, and the *Language*. This allows you to easily use a text in several forms

Note : This text type corresponds to the SAPscript statement INCLUDE. To maintain them, use the standard text editor (transaction SO10).

Note : If there is no need for you to use old SAPscript texts, use **text modules** instead. They can be used across clients and are connected to the transport and translation systems.

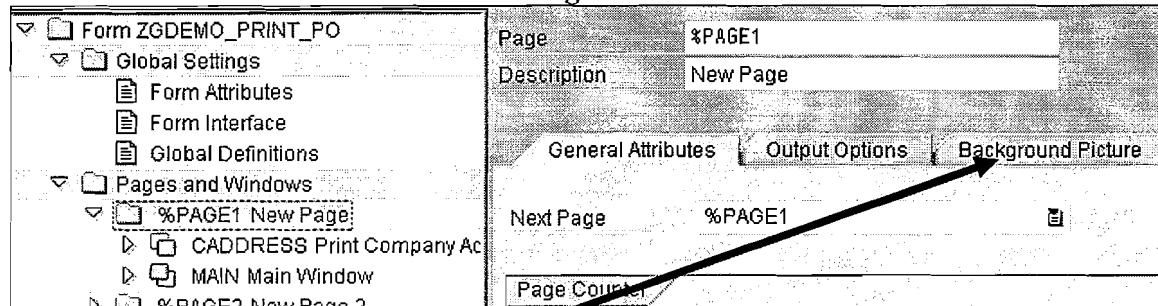
To Include -> Create the Node Text and Select Text Type as Include Text



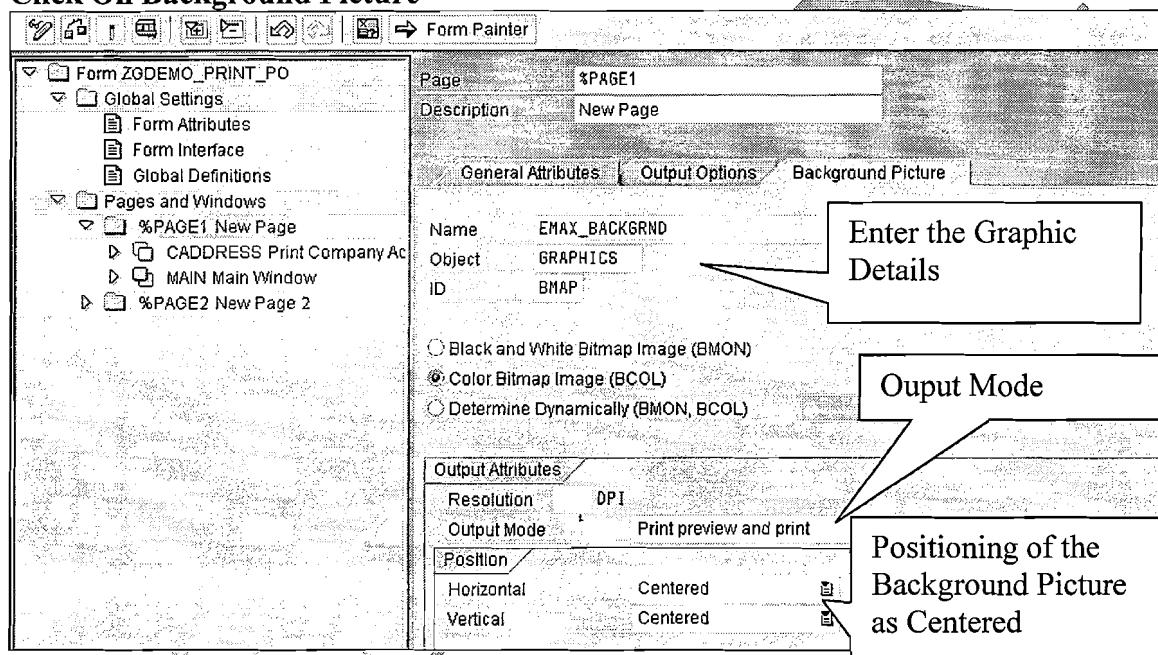
Enter the Text Name (Select From the list).

Printing Background Pictures on the Page:

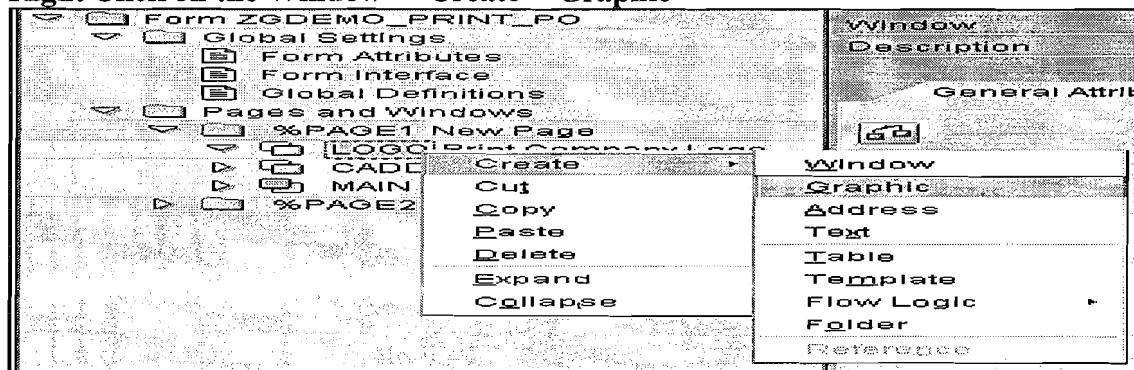
Double Click On the PAGE on the background is to be Printed.

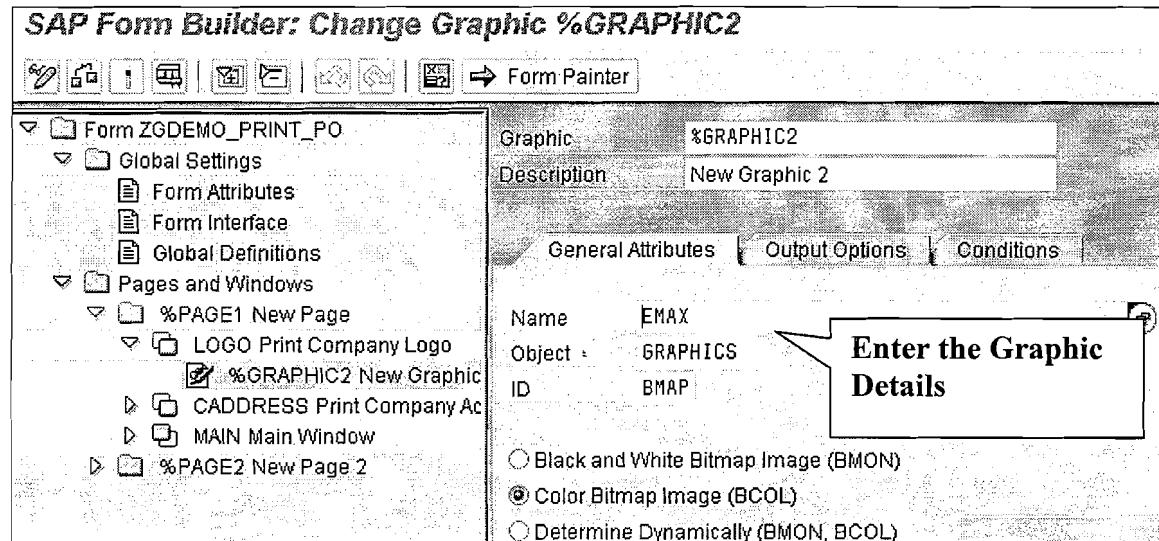


Click On Background Picture

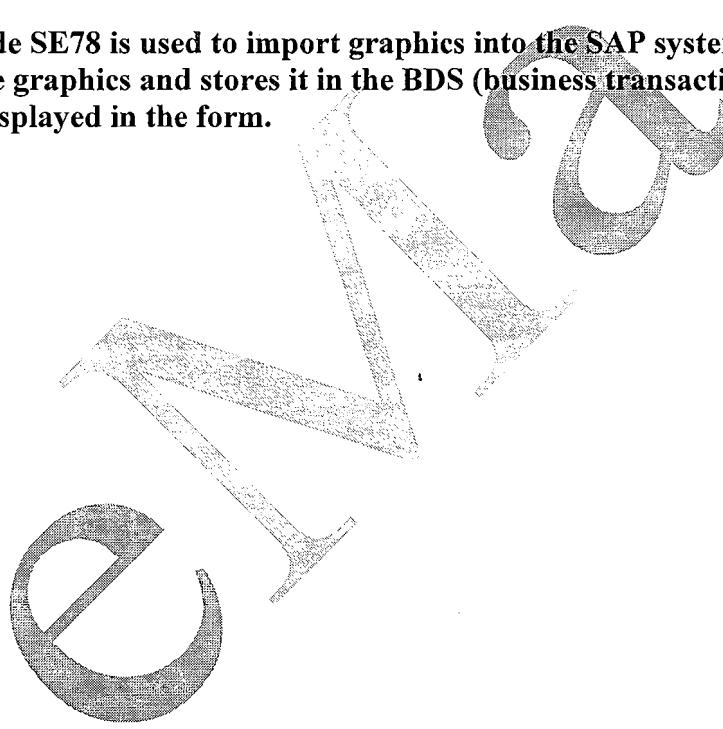
**Printing Picture On Window :**

Right Click on the Window -> Create ->Graphic





Note : Tcode SE78 is used to import graphics into the SAP system. The transaction imports the graphics and stores it in the BDS (business transaction server) and then it can be displayed in the form.



Printing Data in Tabular Format:

Note : There are TWO Types Of NODEs available to print the data in Tabular Format.

- The **table node** : Tables are *dynamic* because the table size depends on the amount of data selected at runtime.
- The **template node** : Templates are *static* because the number of columns and lines are determined before the actual output.

Note : Line Types are used to Specify the Layout of both node types .

Line Types : Line types are used to construct the layout of templates and tables.

Line Type describes the following attributes of a table line:

- The width of the table line
- The number and width of individual cells within the table line

Note : A major difference between line types of templates and line types of tables lies in the height: line types of templates all have the same height, while SAP Smart Forms determines the height of table lines dynamically (depending on the output in the table).

Line Types for Tables :

The height of a table depends on the number of table lines and on the height of all table lines included in the output. The height of each table line depends on the highest cell, which again depends on the contents to be displayed in this cell. When filling table cells with contents, nothing is truncated, instead, the cell height is increased.

Define Line Types

L1		
L2		
L3		

L1 - For Header

L2 - For Body (Items)

L3 - For Footer

Table Output Using Line Types

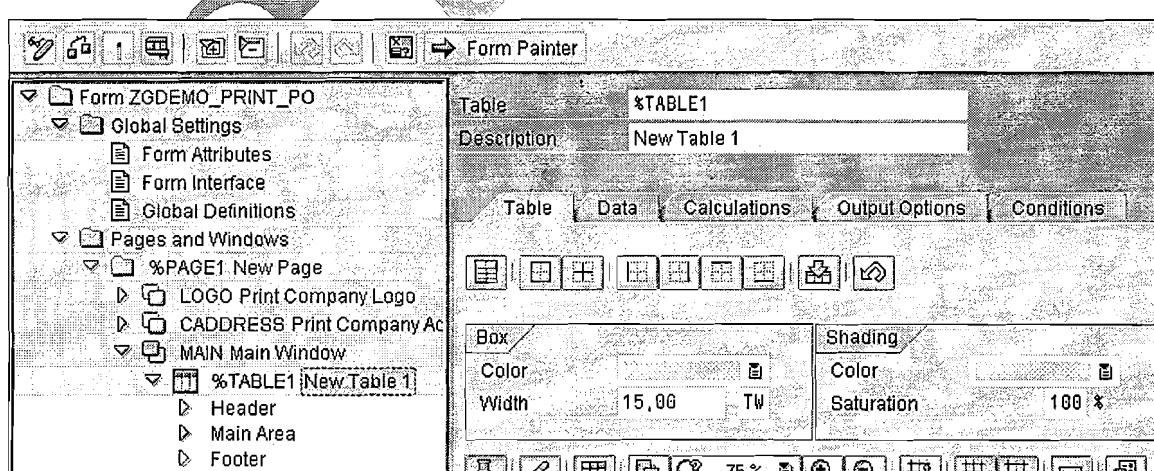
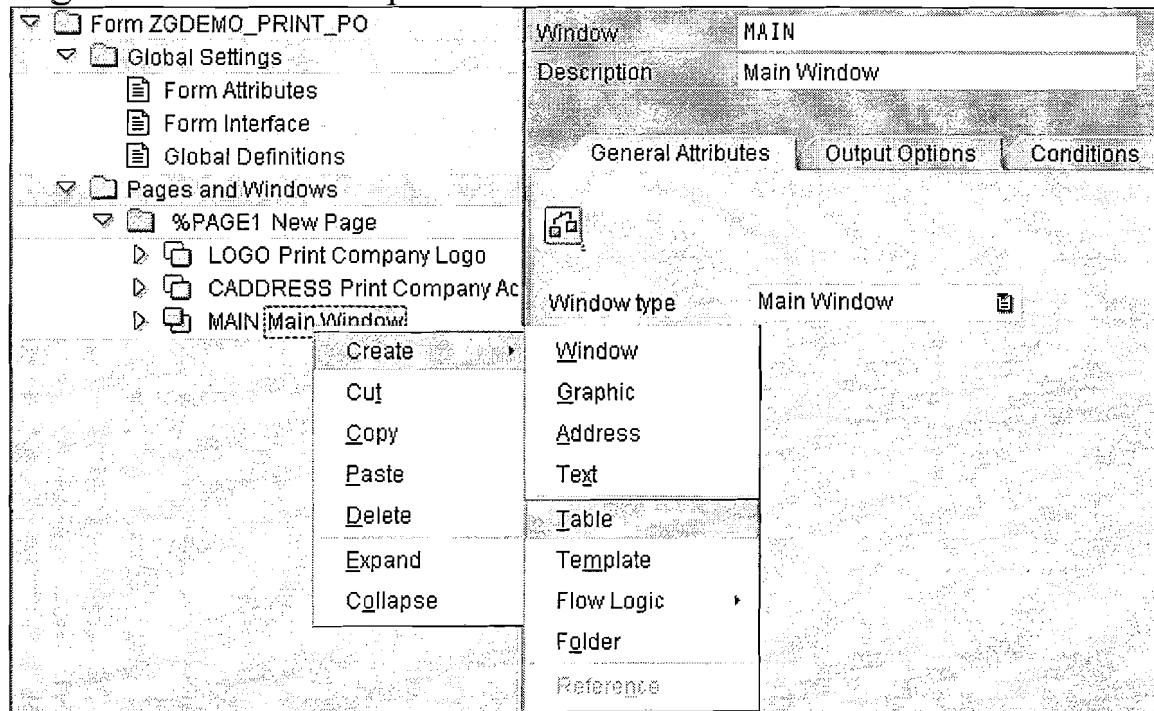
L1		
L2		
L3		

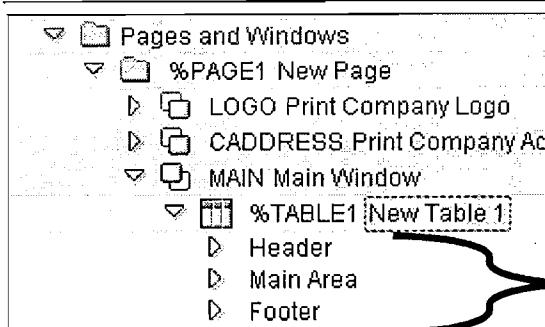
Printing Tables:

Note: The table can cover one or more pages, depending on the number of lines. This is why you display tables in the **main window Always**.

SAP Smart Forms divides a table into a **Header**, a **Main area**, and a **Footer**. These output areas of the table automatically appear as directly inferior nodes of the table node in the **navigation tree**.

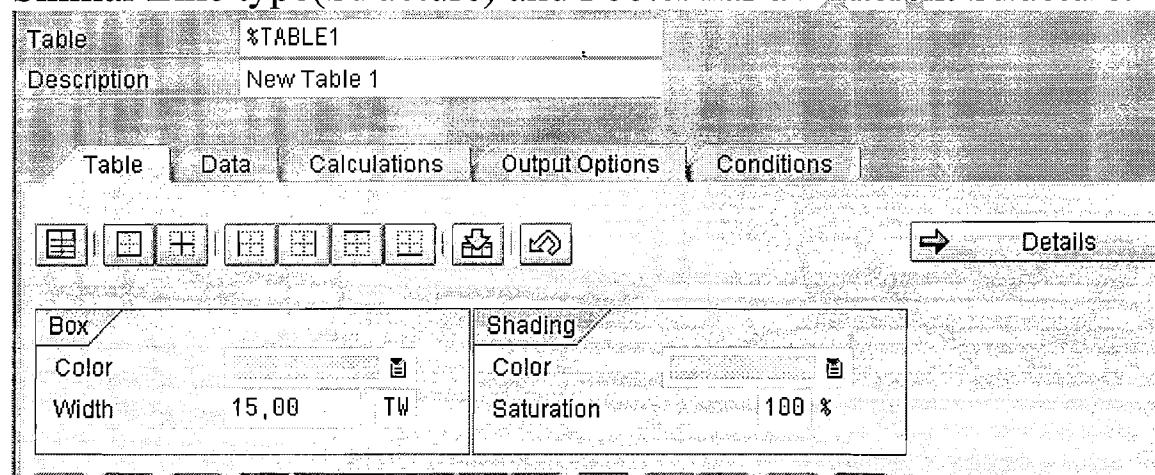
Right Click on the respective Window->Create->Table



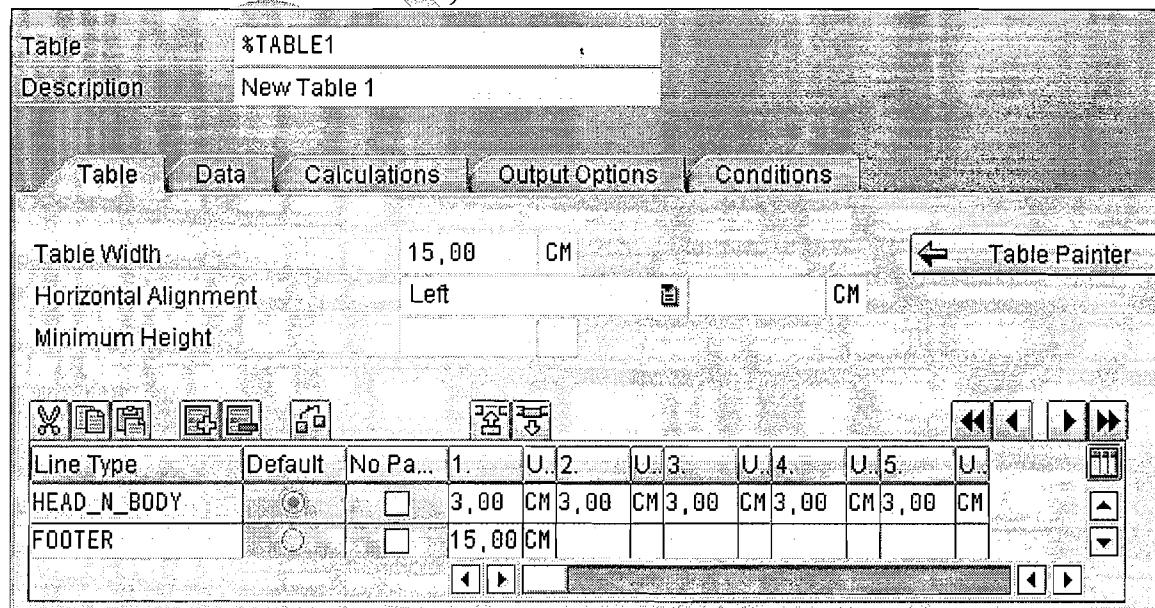


Notice that a Table and Header(Called Only Once), Main Area(Called For Each Record/Line) and Footer(at the End of the Page/Table)

Each Area can has it's own Line Structure and Generally Header (For Column Headings) and Main Area(Contents) contains Similar Line type(Structure) and Footer has a Different Structure.



Click On **Select Pattern ...**,
Click On **Details**,



Define the Line Types : HEAD_N_BODY with 5 Columns

HEAD_N_BODY	<input type="radio"/>	<input type="checkbox"/>	3,00 CM				
-------------	-----------------------	--------------------------	---------	---------	---------	---------	---------

This Line type Can be used for both Header and MAIN Area.

Line FOOTER

FOOTER	<input type="radio"/>	<input type="checkbox"/>	15,00 CM								
--------	-----------------------	--------------------------	----------	--	--	--	--	--	--	--	--

Make Sure that the Table

Table Width	15,00	CM
-------------	-------	----

Width Equals to the Sum of all the Columns of the Line Type.

HEAD_N_BODY	<input type="radio"/>	<input type="checkbox"/>	3,00 CM					
-------------	-----------------------	--------------------------	---------	---------	---------	---------	---------	---------

Similarly FOOTER

FOOTER	<input type="radio"/>	<input type="checkbox"/>	15,00 CM							
--------	-----------------------	--------------------------	----------	--	--	--	--	--	--	--

Table		Data		Calculations		Output Options		Conditions																															
Table Width	15,00	CM																																					
Horizontal Alignment	Left								CM																														
Minimum Height																																							
<table border="1"> <tr> <td>Line Type</td> <td>Default</td> <td>No Pa.</td> <td>1.</td> <td>U.2</td> <td>U.3</td> <td>U.4</td> <td>U.5</td> <td>U.6</td> <td>U.7</td> </tr> <tr> <td>HEAD_N_BODY</td> <td><input type="radio"/></td> <td><input type="checkbox"/></td> <td>3,00 CM</td> </tr> <tr> <td>FOOTER</td> <td><input type="radio"/></td> <td><input type="checkbox"/></td> <td>15,00 CM</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>										Line Type	Default	No Pa.	1.	U.2	U.3	U.4	U.5	U.6	U.7	HEAD_N_BODY	<input type="radio"/>	<input type="checkbox"/>	3,00 CM	FOOTER	<input type="radio"/>	<input type="checkbox"/>	15,00 CM												
Line Type	Default	No Pa.	1.	U.2	U.3	U.4	U.5	U.6	U.7																														
HEAD_N_BODY	<input type="radio"/>	<input type="checkbox"/>	3,00 CM	3,00 CM	3,00 CM	3,00 CM	3,00 CM	3,00 CM	3,00 CM																														
FOOTER	<input type="radio"/>	<input type="checkbox"/>	15,00 CM																																				

Click On Data

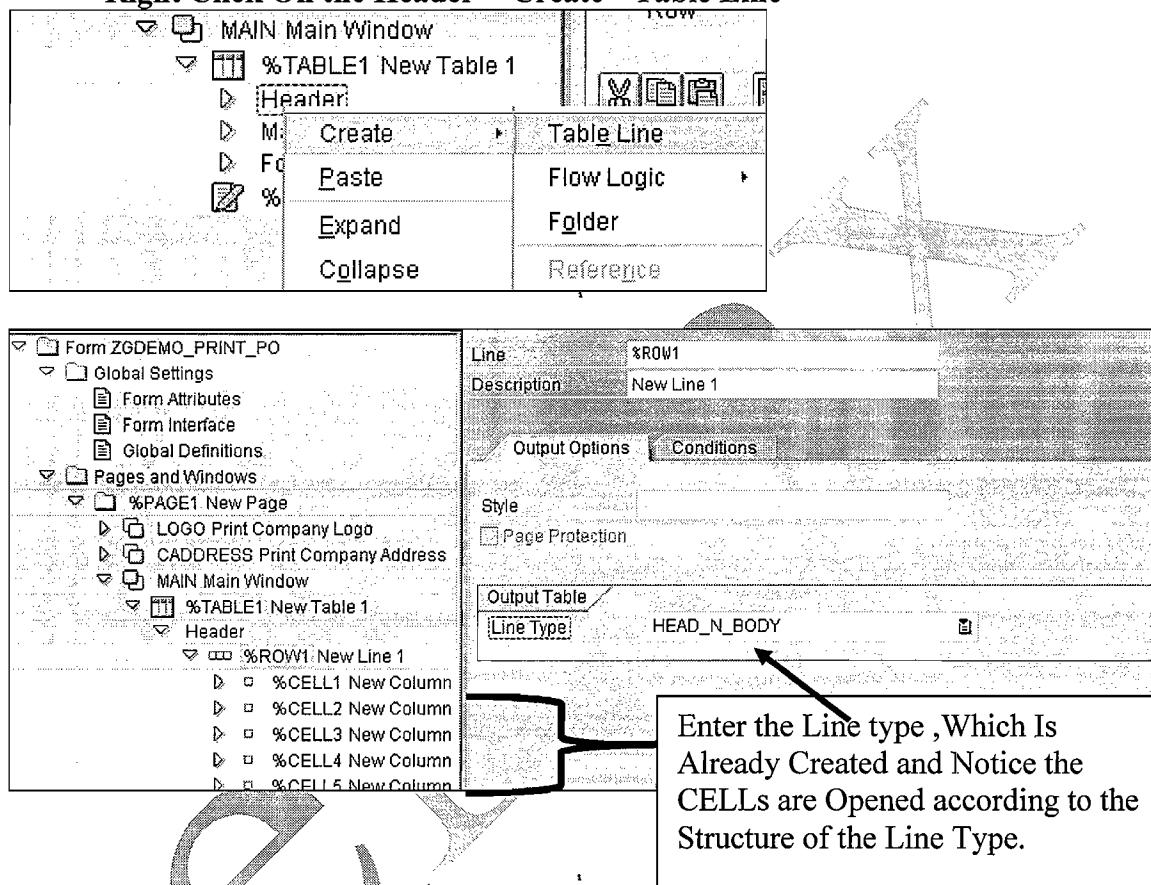
Table	TABLE1														
Description	New Table 1														
Table		Data		Calculations		Output Options		Conditions							
LOOP Loop <input checked="" type="checkbox"/> Operand IT_EKPO INTO WA_EKPO Row to WHERE Condition <table border="1"> <tr> <td>Fld Name</td> <td>O.</td> <td>Comparison Value</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table>										Fld Name	O.	Comparison Value			
Fld Name	O.	Comparison Value													

Provide the Internal Table to be looped and it's Work Area



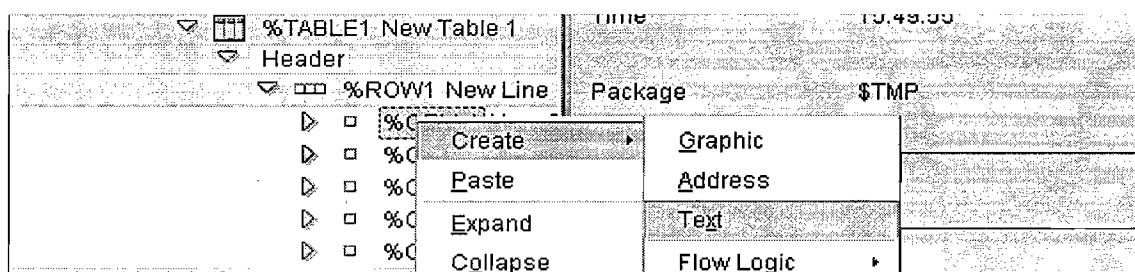
To Print Header Data(Column Headings)

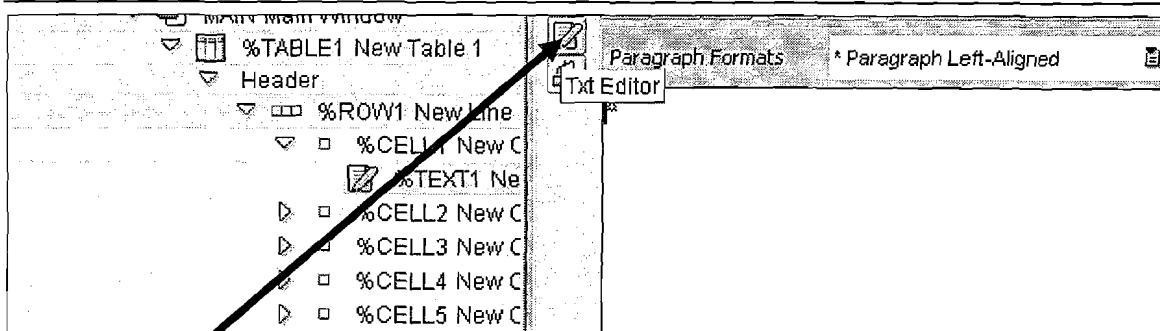
Right Click On the Header ->Create->Table Line



To Print Data in the Columns

Right Click On CELL1->Create->TEXT





Click On Txt Editor



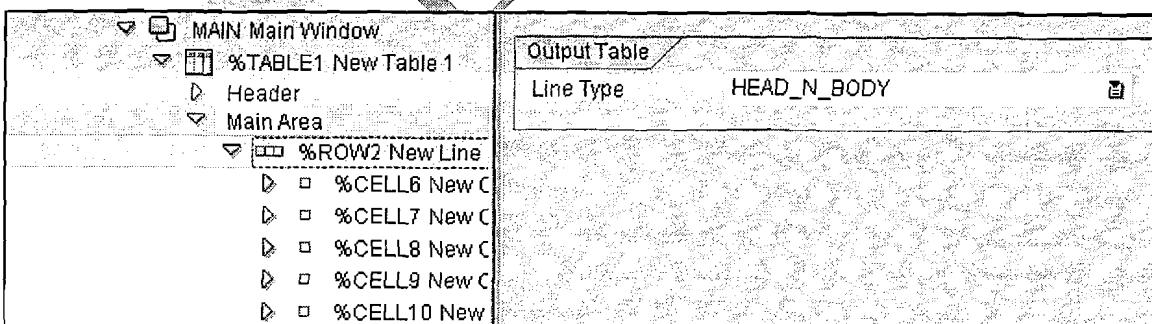
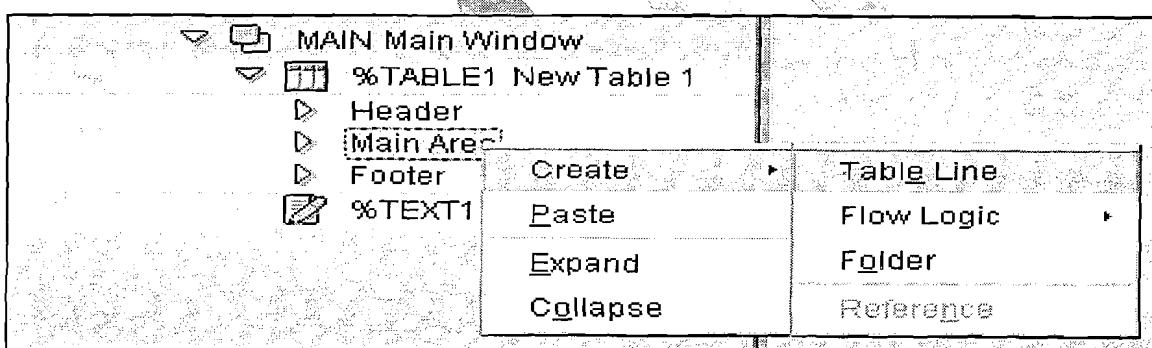
Enter the

Paragraph Format and Data to be Printed (Column Heading)

Note : Repeat the Same for all the other Cells(CELL2,3,4,5) with the respective Column Headings.

To Print the Body of the Table Control(Main Area) :

Right Click on the Main Area ->Create->Table Line

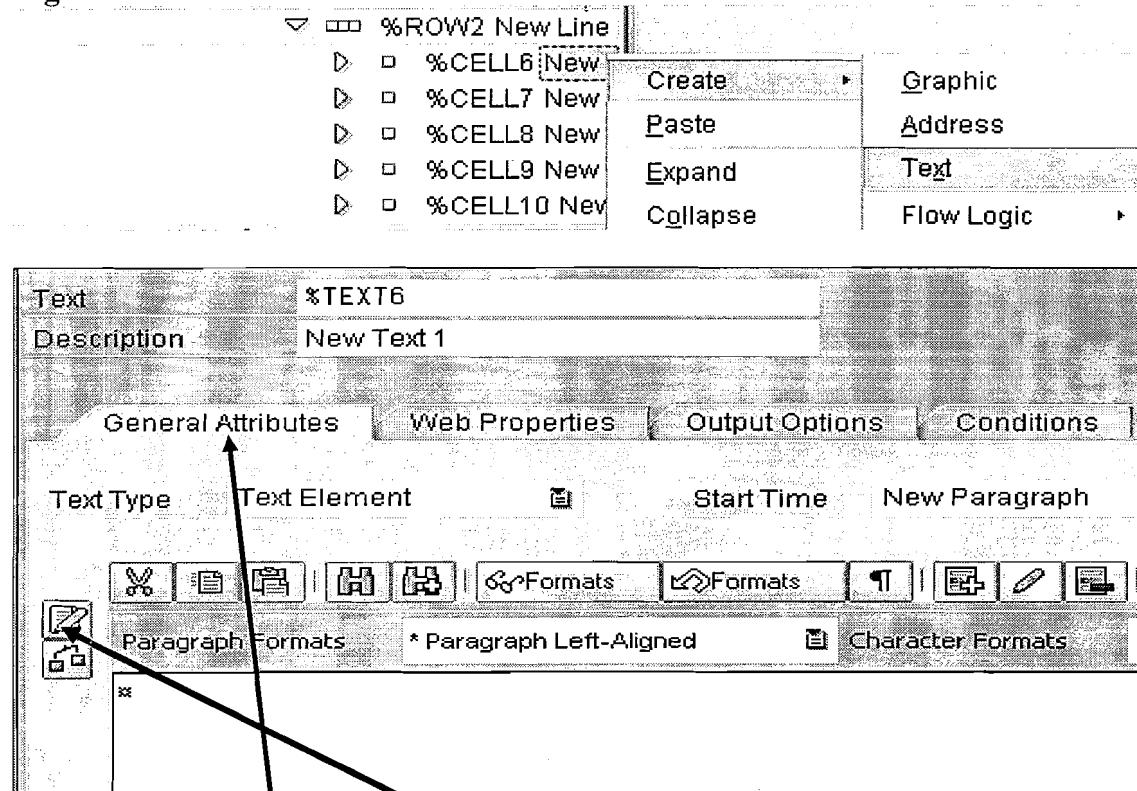


Enter the Line Type HEAD_N_BODY(Already used for both Header i.e for Column Headings).

Notice that CELLS (CELL6 to CELL10) are opened according to the line Structure.

To Print Data in the CELLS

Right Click on CELL6->Create->TEXT



Click on General Attributes then Txt Editor

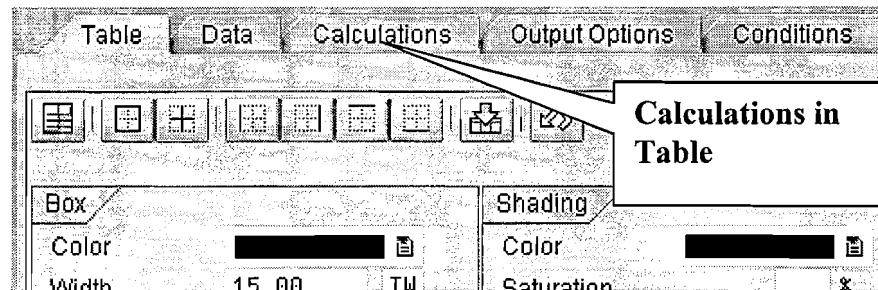
* &WA_EKPO-EBELN&

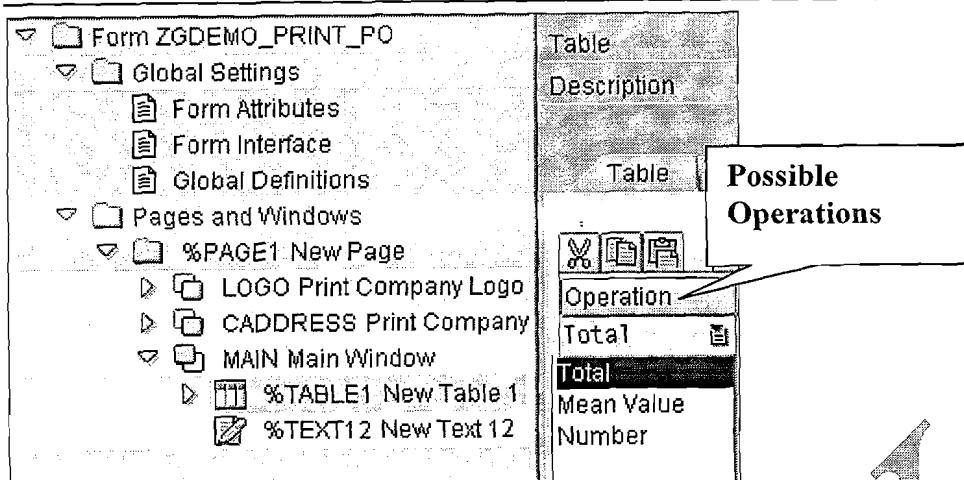
Enter the Paragraph format(* - Default Paragraph) and program Symbol(WA_EKPO-EBELN).

Note : Repeat the Same for all the other Cells(CELL7,8,9,10) with the respective content.

Calculations in Table :

Possible Operations in Calculations :





Operation is to determine the kind of calculation you want to execute within the table. Smart Forms assigns the result of the operation to the field you specified under **Target Field Name**. You can then use the target field name to display the result.

Possible Automatic Operations

Operation	Calculation
Total	< target field name > = < target field name > + < field name >
Average Value	< target field name > = <total of the last values of field name divided by their number >
Number	< target field name > = < target field name > + 1

Note: You can use the **Number** operation to find out and display the total number of items/Lines at the end.

Automatic calculations provided by the Calculations tab.

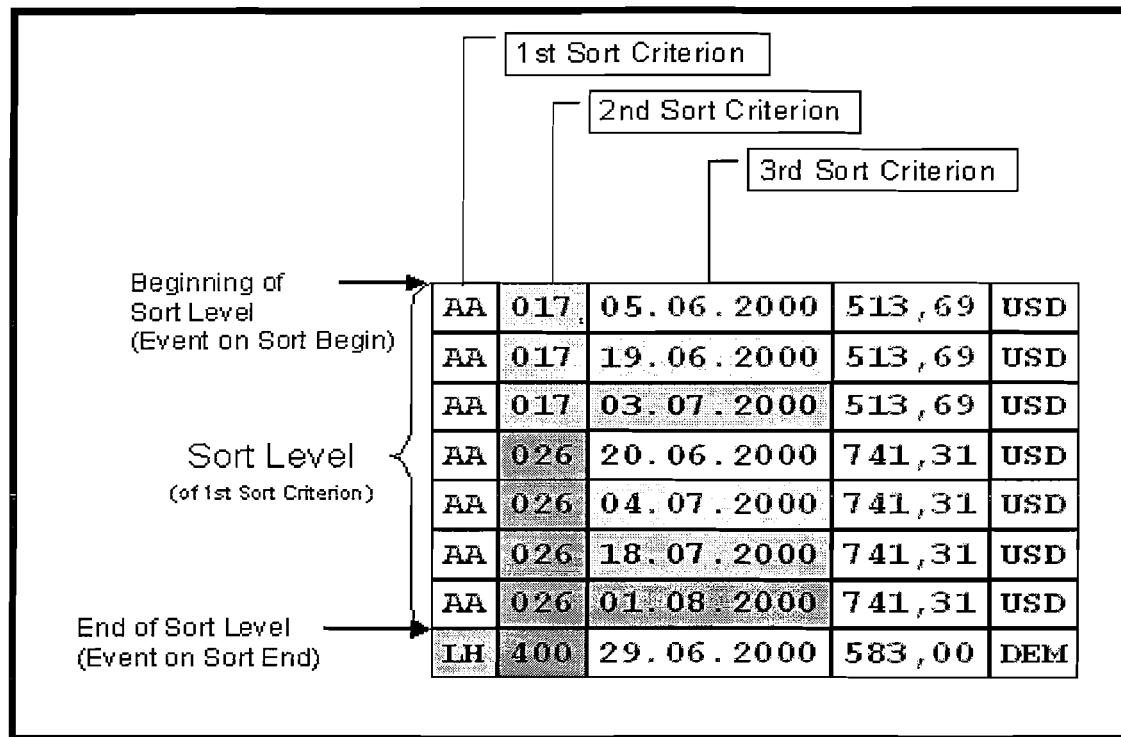
1. Define an additional global field, the **target field**, which you need to display the result of the operation.
2. On the Calculations tab of the table node select an **operation** first, then a field (that is, a column of the table), for which to execute the operation, and specify the calculation.
3. Depending on whether you execute a calculation **on all items of the table** or **on items of a sort level**, display your result either in the footer or at the beginning or end of a sort level.

Calculating Subtotals

You want to display subtotals after a sort level break of a **sort criterion**. So that Make Sure that the Internal Table is **Sorted** based on the required Fields.

Sort Criteria in Table:

Before you start reading an internal table, you can tell SAP Smart Forms to sort it. With the sorting, you can divide the table into different *sort levels*:

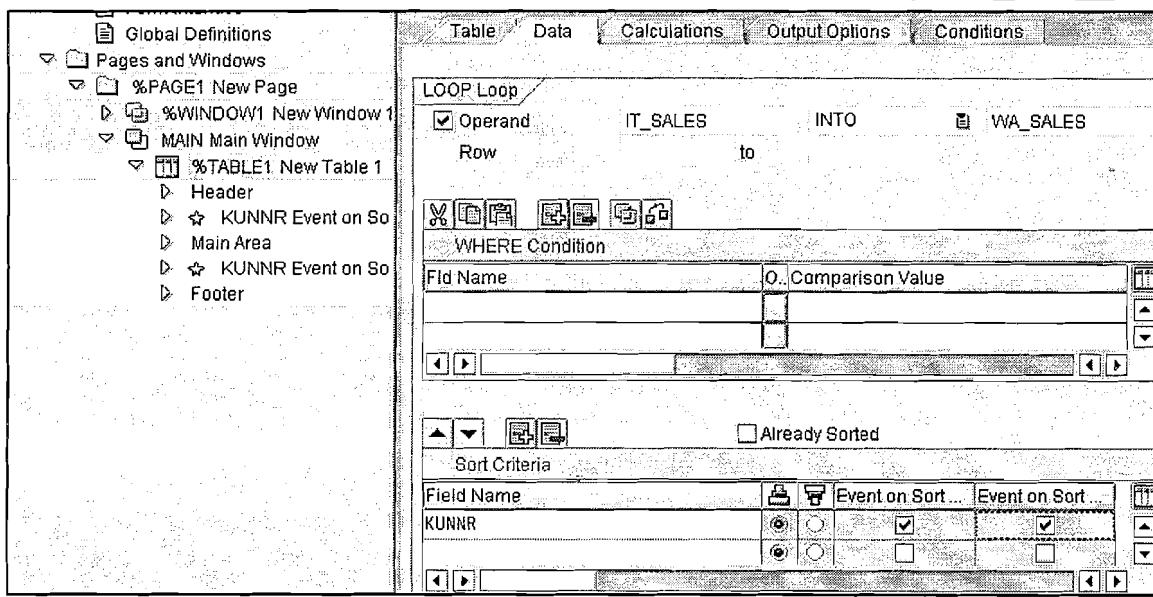


Note : This allows you to bundle data records and to display subtitles or subtotals at the beginning or end of a sort level.

To determine the sorting of your internal table, use the *Data* tab of the table or loop node. In the *Sort Criteria* group box you can enter the fields by which to sort the table.

The sequence in which you enter the field names into the table control determines the sort sequence. To change it at a later time, use the black arrows above the control. Use the radio buttons to choose whether to sort the field in ascending or in descending order.

Note : To display data(Sub Totals) before or after a sort level, mark Event on Sort Begin and/or Event on Sort End . In the navigation tree, an event node appears for which you can create inferior nodes. In these inferior nodes you specify the data you want to display at this event (for example, subtotals).



Note: SAP Smart Forms cannot recognize whether an internal table was sorted before (for example, before passing it at the form interface). If it was sorted and you want to use the output events, you must still enter the sorting into the table control and, in addition, mark **Already Sorted**.

Go to the **Calculations** tab of your table node and fill the following fields:

Column	Value
Operation	Total
Field Name	<WA-FieldName>
Target Field Name	<V_Total>
Event	After Loop
Initialization	(active)
Reset	Sort Criterion
Sort Criterion	<Field Name>

The subtotals are formatted according to the relevant <Field> and displayed after each sort level.

Calculating Grand Totals

On the Calculations tab fill the following fields:

Column	Value
Operation	Total
Field Name	<WA-Field>
Target Field Name	<V_Gtotal>
Event	After Loop
Initialization	(active)

1. To display the grand total, include a table line into the footer and include a text node into this table line.
2. You can specify whether to display the total at the end of each page (as subtotal) or as a grand total at the end of the table. To do this, activate or deactivate the option **At Page Break of the footer**.

Numbering Table Items Consecutively: You want to have a column in your table output in which the individual items of your table receive an ascending number.

Steps:

1. Define a global field of type I with the technical name **NUMBER** .
2. Enhance your **line type** for the table items and the table heading by one cell that displays the item number. For the output of this cell in the main area insert a text node. Use the text node to display the **NUMBER** field.

On the Calculations tab fill the following fields :

Column	Value
Operation	Number
Target Field Name	NUMBER
Event	Before Loop
Initialization	(active)

3. If desired, you can display **NUMBER** at the end of the table in the footer area.

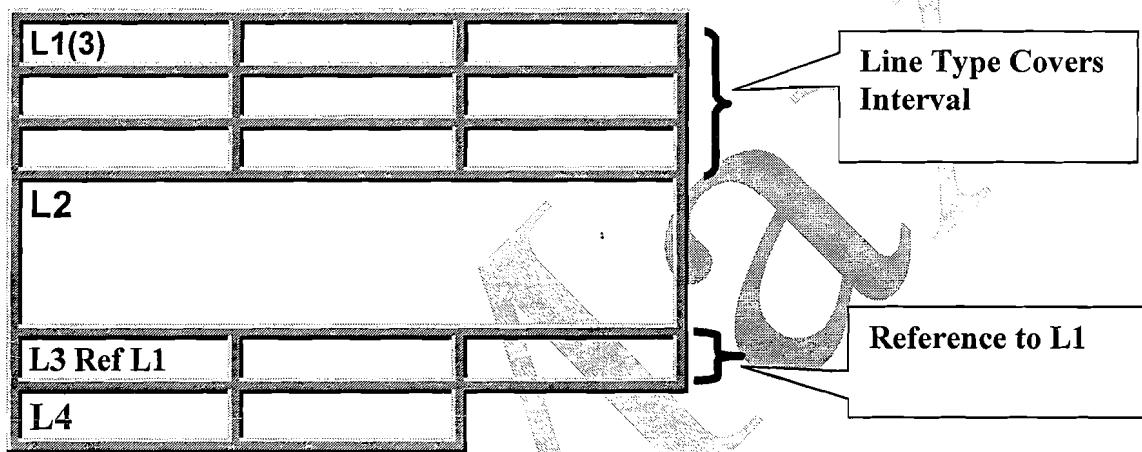
Note : The items of the table are numbered consecutively.

B) Templates in Detail(Static Table) :

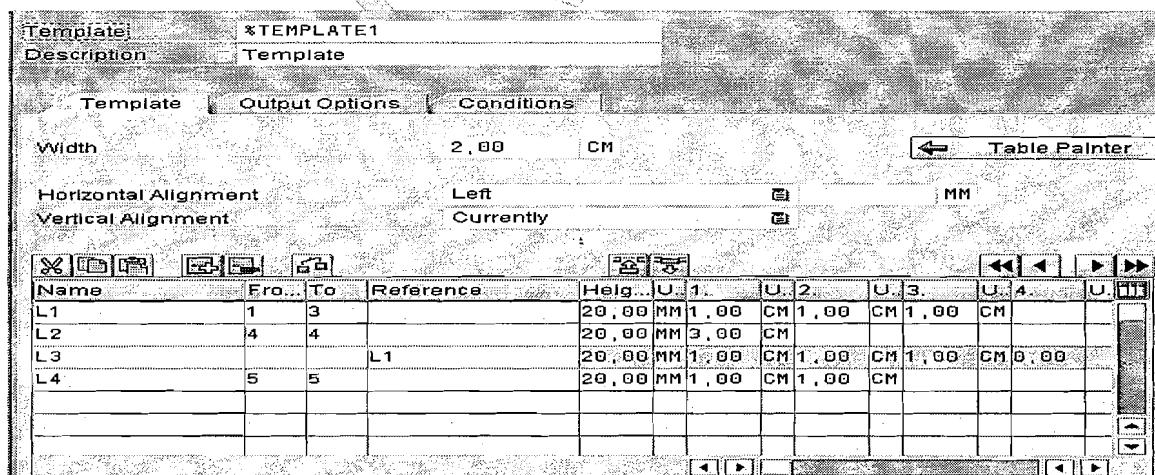
The layout of a template is **static**, which means that you determine it before the Smart Form is executed. To do this, you can define line types in the **Table Painter**, whose **sequences and appearances are the same as in the output**.

Line Types for Templates :

The layout of a template is static, which means that you determine it before the Smart Form is executed. To do this, you can define line types in the Table Painter, whose sequences and appearances are the same as in the output. Describing a template with line types could look like this:



The output of this template consists of five lines, for which only three different line types are used:



- Line type L1 for the first two line of the template. The Table Painter shows the interval in the bracket after the name of the line type.
- Line type L2 for the third line of the template. This line type has a different fixed line height.

- Line type L3, which is no line type **of its own**, but only a reference to L1. The Table Painter shows the name of the reference line type below the name of the line type.
- Line type L4 for the fifth line.

Note: By specifying an *interval* you can use one line type for **several subsequent lines** of the template. By specifying a *reference* you can reuse **any existing line type at any position of the template**. You can make changes only to the 'master' line type.

Note: Unlike with **line types for tables**, for templates you specify a fixed line height for the line type. Any output that does not fit into a cell is truncated.

Printing Templates: Use node type *Template* to display a table whose layout and size (number of lines and columns) is determined before the runtime of the application program. So that a template is also called a **static table**.

Creation Of a template, needs to define a table layout to determine the cell structure for each line. The cells are used to display the contents of the inferior nodes of the template node. This allows you to position text and a graphic side by side .

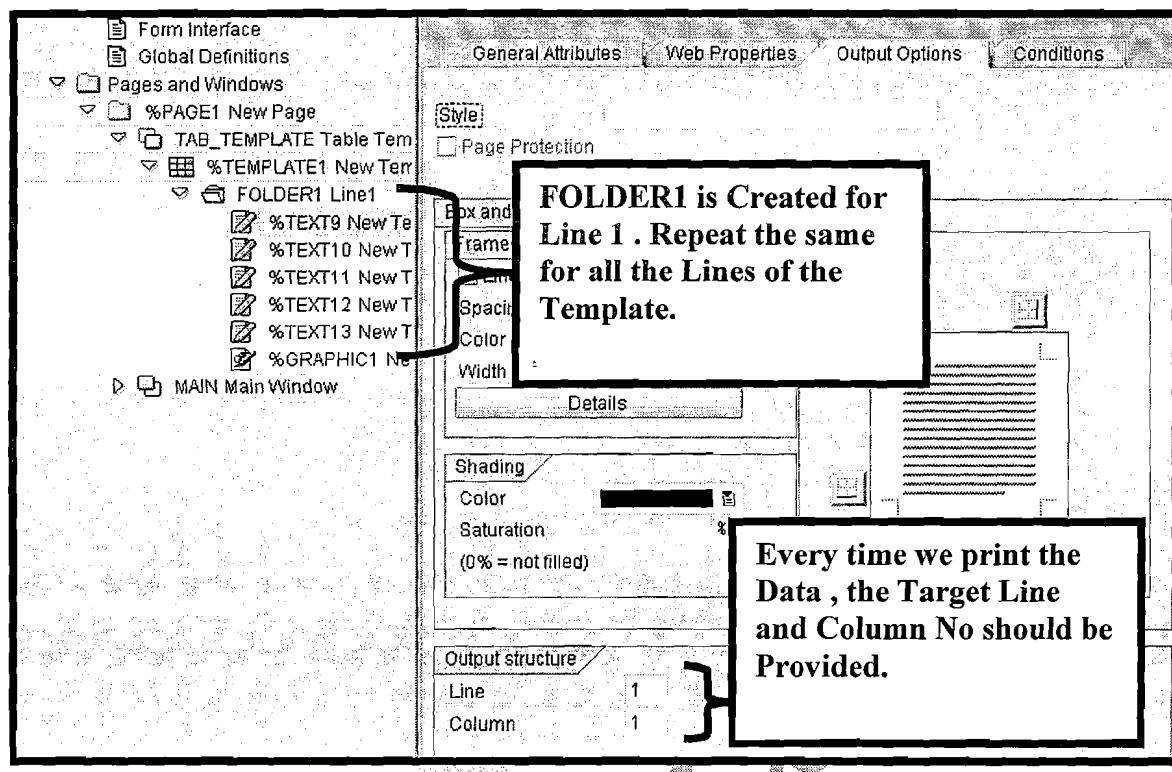
Note : You can create a template node only as an inferior node of a window node.

Displaying Contents in Cells

1. Use the context menu of the template node to create inferior nodes.

For clarity reasons create one **folder node** for each line. The folder node is an outline node that you can use to combine related nodes. For folder nodes, you need not fill in the input fields in the *Output structure* box (see next step).

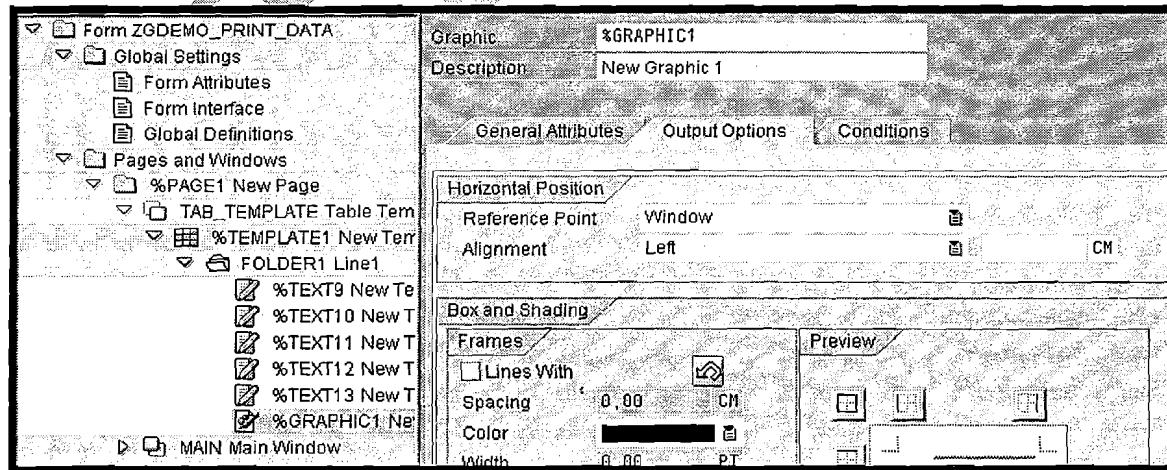
2. The *Output options* tabs of the inserted nodes now contain **an additional box Output structure with the fields Line and Column**. Assign each node to the cell in which you want the output to appear.



Displaying Graphics in Templates

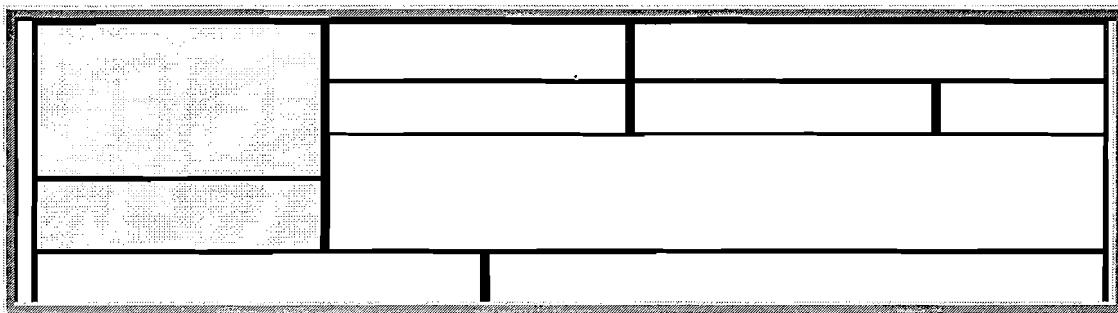
If you insert graphic nodes as inferior nodes of a template node, the *Output options* tab contains the *Horizontal position* box. The horizontal position usually refers to the output window.

Horizontal Position : You can determine the horizontal position of a graphic within a cell only in relation to the left cell margin. On the **Output options** tab of the graphic node you find the *Horizontal position* box. Set the *Reference point* to *Window* and the *Alignment* to *Left*. In the printout, the graphic appears at the left cell margin.



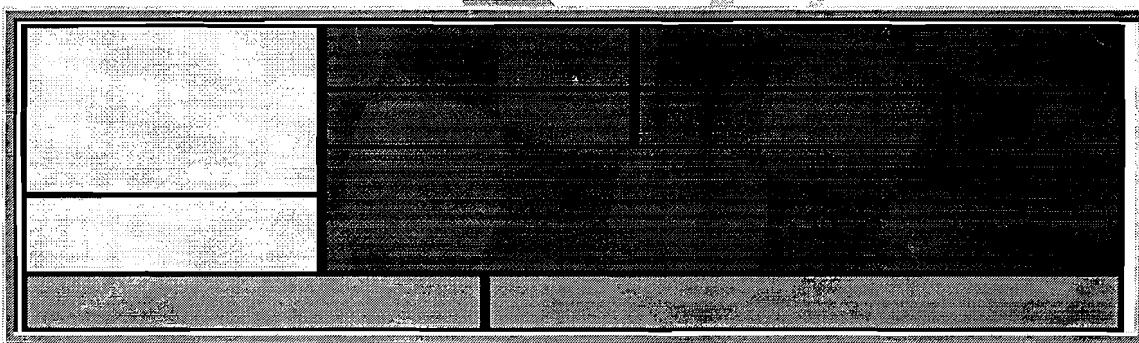
Combining Templates and Windows

You use line types to define the layout of templates. However, to create a template like the one below, you **cannot** use the template concept of Smart Forms:



The cells of the colored part cover several lines. However, SAP Smart Forms does not allow you to specify for a cell of a line type that you want this cell to cover several lines.

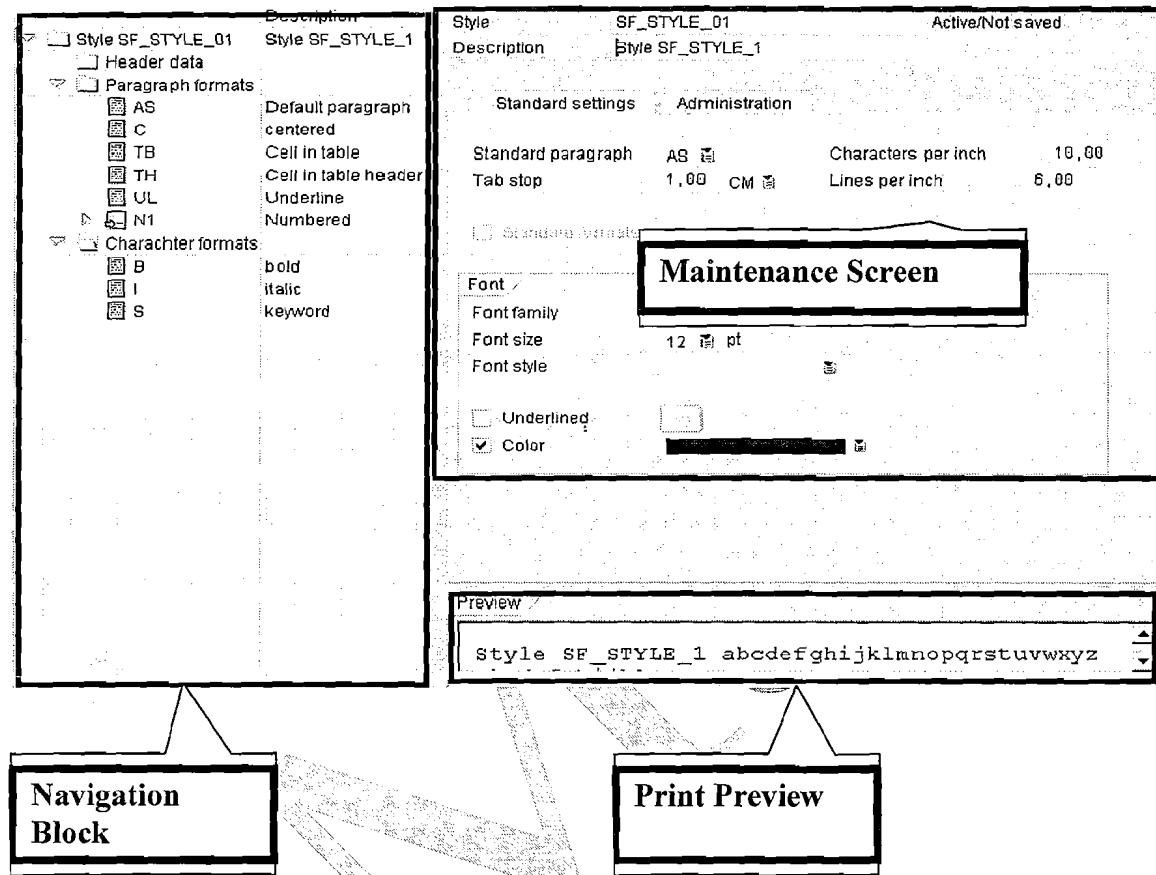
To print such a layout in a form, you must divide the template layout into several windows and create one template per window. To print the template above, you would need three templates in three windows in SAP Smart Forms (see differently colored sections):



This combination technique enables you to print templates of any complexity in a form.

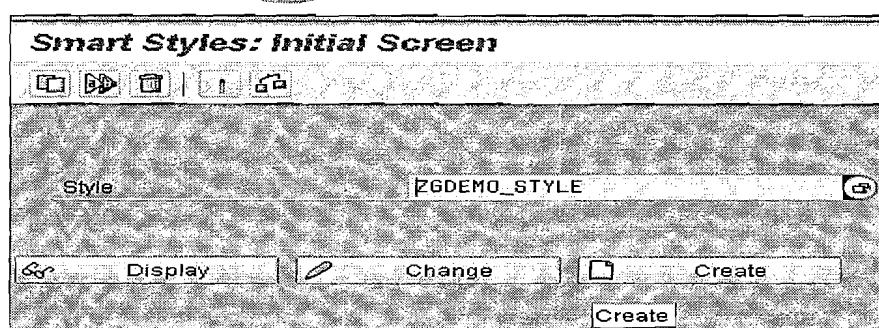
Smart Style in Detail :

Smart Style is the collection of re-usable Paragraph and Character Formats.



To Create Smart Style:

1. Choose transaction **SMARTSTYLES..**
The Smart Styles initial window appears.
2. Enter the style name
3. To create a Smart Style, choose *Create*.



Note : A Smart Style must be assigned to each Smart Form. Can be done globally for the entire Smart Form in the form attributes. In addition, you can assign a Smart Style locally to a node, for example, a text node. This assignment then applies for the entire subtree and overrules the global settings.

A Smart Style contains :

- Header data containing the default values(Paragraph) of a Smart Style
- Paragraph formats including indents and spacing, font attributes, tabs, and outline and numbering
- Character formats including effects (superscript, subscript), barcode and font attributes
- Colors and underlines for a paragraph or character format
- Preview

Paragraph Formats : A paragraph format contains information on indents, spacing, font settings, text color, tabs, numbering and outline. Each paragraph format must have a unique name.

To Create :

1. In change mode of the Smart Style, select the node *Paragraph Format* and choose *Create*.
2. In the *Paragraph Format* field enter a two-character paragraph key.
3. Select the desired attributes on the individual tabs.
4. Choose *Activate*.

Character Formats : Character formats are to assign special output attributes to **sections of texts or character strings within a paragraph**.

Attributes Of Character Formats :

- Font attributes and superscripting/subscripting
- Bar code

To Create Character Format:

1. In change mode of the Smart Style, choose the *Character Formats* node and then *Create*.
2. In the *Character Format* field, enter a two-character character key.
3. Select the desired attributes on the individual tab pages.
4. Choose *Activate*.

Miscellaneous :

Events : You specify an event to determine when to print a particular node.

You can specify events for the following node types :

- For each node type that offers the **Conditions tab**
 - For **headers and footers** of table and folder nodes
 - For **control breaks** before or after which you want to make a certain output (example, subtotals).

Note: Events control the output of the specified node and of all existing inferior nodes. If you specify several events, the system links them with a logical OR.

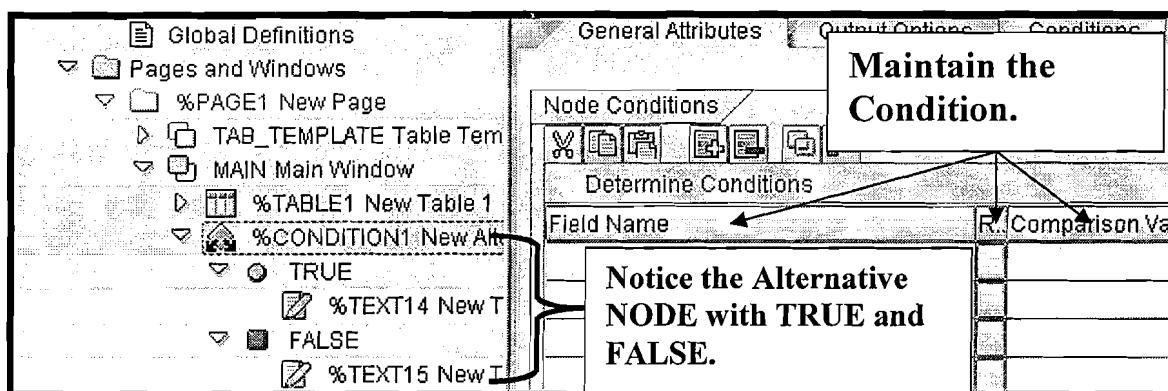
Events on the *Conditions* tab

Event	Description
only on first page	Output appears only on the first print page.
not on first page	Output appears on all print pages except the first.
only after end of main window	Output starts only after all nodes within the main window are processed. It makes no sense to activate this event for a node within the main window, since it would never be printed.
only before end of main window	Output appears as long as the system still processes nodes of the main window. This event is of interest for nodes that appear before or after the main window definition in the navigation tree of the Form Builder.
only on page <S>	Output appears only on print pages that are defined using draft page <S>.

Branching Within the Form : Via Alternative.

Helps to process one of two nodes, alternatively (including inferior nodes).

1. Create a node of type **Alternative**.
2. On the *General Attributes* tab determine an unstructured condition in the *Node Conditions* box. The system offers the same features as in the *Output Conditions* box on the **Conditions** tab.
3. The alternative node has two directly inferior nodes: **TRUE** and **FALSE**. Insert your inferior nodes there.



Note : If the condition is true, the system processes the directly inferior node **TRUE**, otherwise **FALSE**.

Processing Output Repeatedly:

Note : **LOOP** node is to read data from an internal table line by line.

You pass an internal table containing customer data and an internal table containing customer orders to the **form interface**. Use the loop to read the customer data and display it. Within the loop, you create a table and use the customer number to display the orders of this customer. This allows you to display the orders of all customers on one form.

Steps :

1. Create a node of type *Loop*.
2. Enter a unique name for the node and a description (for example, loop on customer table).
3. Read the data from an internal table into a work area.
4. Create inferior nodes in which you display the fields of the read table lines.

Migrating SAPscript Forms: SAP delivers Smart Forms for important business processes. If for your needs no such standard forms exist or if you have your own SAPscript forms, you can use two tools to migrate them.

Individual Migration

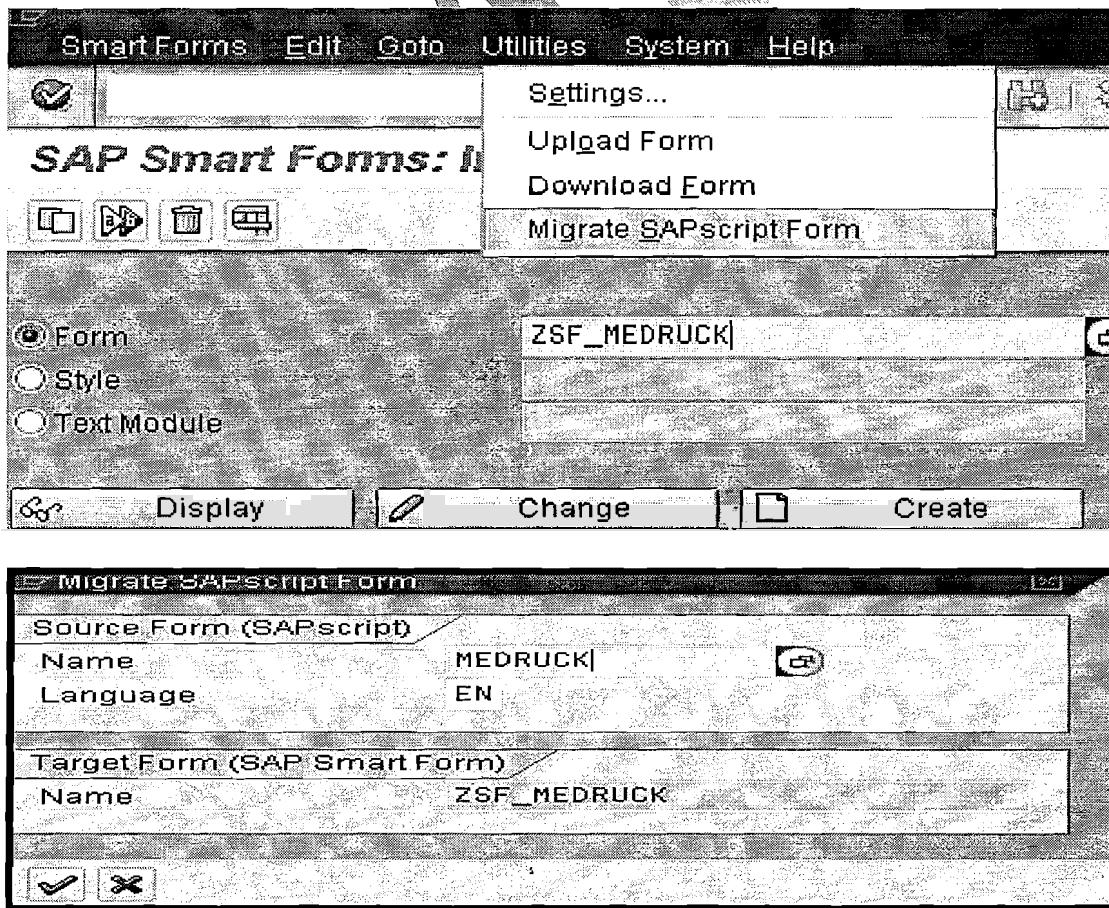
When converting a SAPscript style into a Smart Style, the system converts all paragraph and character formats with all their properties and attributes without any changes. Thus you can use the converted Smart Style without making any adaptations.

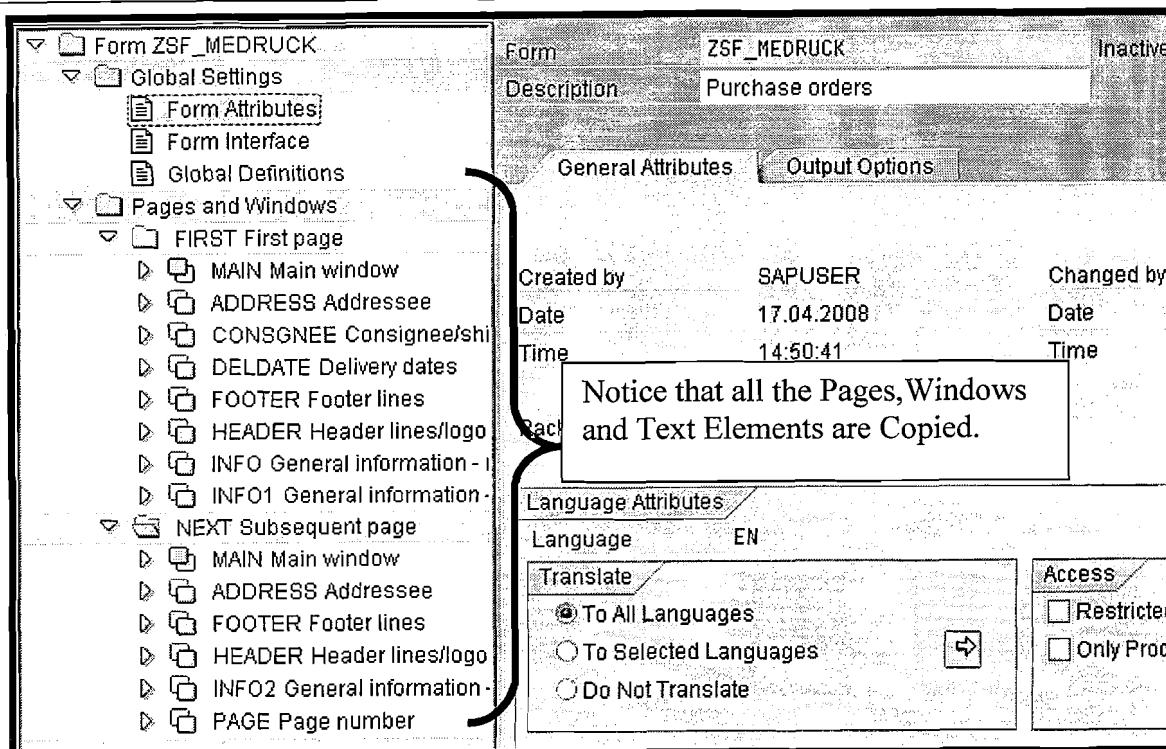
Note : The system does not migrate the print program (data retrieval) or the form logic contained in the print program. You must make the required changes to the respective print program yourself.

Migrating a SAPscript form

1. Go to the SAP Smart Forms initial screen (transaction SMARTFORMS).
2. In the *Form* field enter the name of the Smart Form you want to create.
3. Choose *Utilities* → *Migrate SAPscript form*.
The dialog window *Migrate SAPscript Form* appears.
4. Enter the name and the language of the source form (SAPscript).
5. Choose *Enter*.

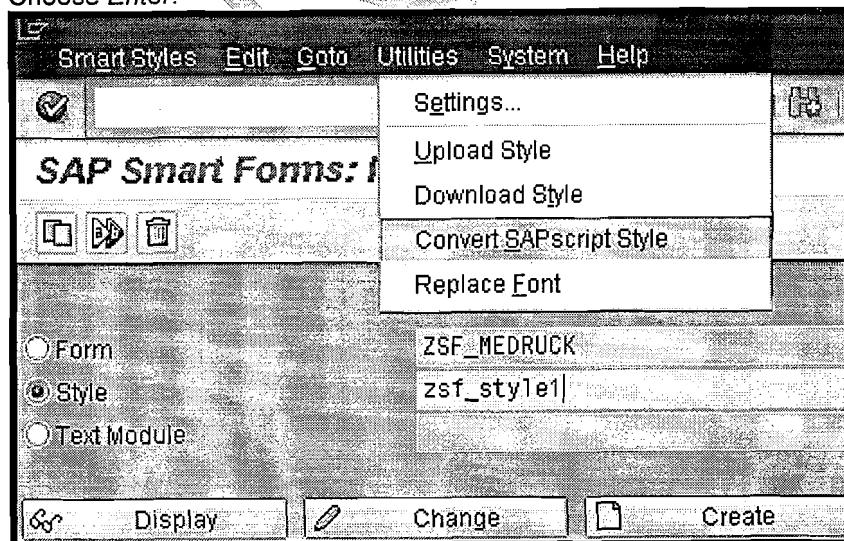
This takes you to the change mode of the SAP Form Builder.





Converting a SAPscript style

1. Go to the Smart Styles initial screen (transaction SMARTSTYLES).
2. In the Style name field enter the name of the Smart Style you want to create.
3. Choose Smart Styles → Convert SAPscript style.
4. Enter the name of the SAPscript style you want to convert.
5. Choose Enter.

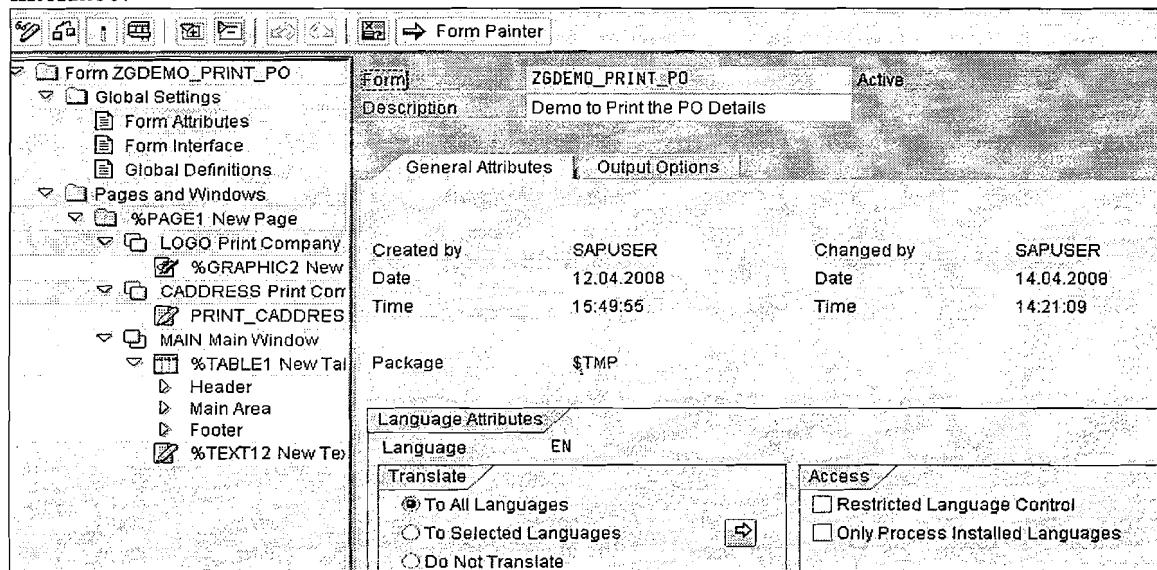


Mass Migration of SAPscript Forms

1. Execute the Standard Program **SF_MIGRATE**.
2. Select the names and the language of the SAPscript forms and choose *Execute*.
The system creates the Smart Forms under the names of the SAPscript forms plus the extension _SF and It also displays a list of the migrated forms.

Form Logic: In the Form Builder you 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:



This graphic already contains some nodes. The node **Global Settings** as well as its three inferior nodes **Form Attributes**, **Form Interface**, and **Global Definitions** always exists for any newly created forms. To describe the **Form Logic**, create a hierarchy under the node **Pages and Windows**. This hierarchy determines the rules used to process the nodes of the tree. **Depending on the node type, this could include:**

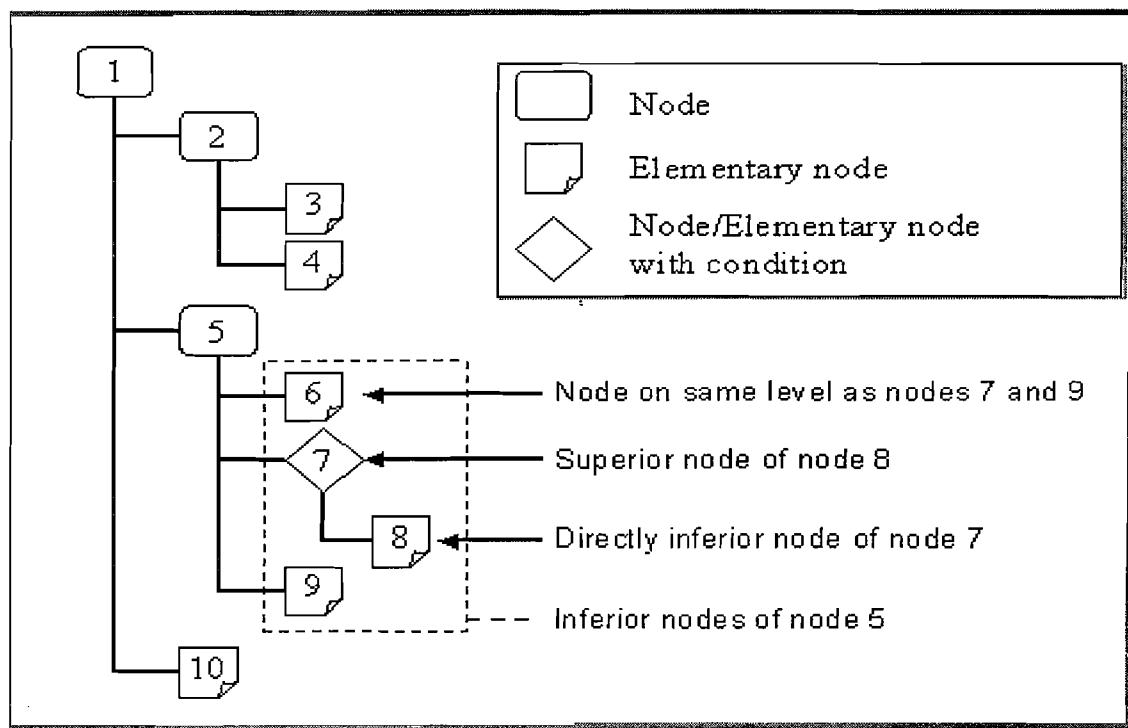
- Printing the node contents (text, addresses, graphics).
- executing the node statements.
- executing the inferior nodes of a node according to other rules (for example, in a *loop*).

You use the form logic to control the flow of the form output. The following rules apply throughout:

1. The nodes in the tree structure are processed from top to bottom. This is easier to understand.
2. For each node there is a tab page, which you can use to link the node to a **condition**. If the condition is true, the system processes the node. If not, it skips the node and all its **inferior nodes**.
3. You must define a next page for each page. However, you can also go to other pages dynamically .

Example for Form Logic

The diagram below shows the tree structure of the root node *Pages and Windows*:



The numbering of the nodes also shows the sequence in which they are processed (from top to bottom). The first node is the *root node*. The *inferior nodes* of a node are all nodes included directly under this node in the hierarchy. For node 5 these are the nodes 6 to 9 (directly inferior nodes are only the nodes of the next hierarchy level, that is 6, 7, and 9).

Note : A node without an inferior node is also called *elementary node*. Elementary nodes can never have inferior nodes. If the condition of node 7 is false, neither node 7 nor node 8 are processed. Processing resumes directly with node 9.

Overview Of Different Node Types:

When a Smart Form is Created, the tree structure of the Form Painter already contains two root nodes: i.e i) **Global Settings** ii) **Pages and Windows**.

- You use the inferior nodes of the *Global Settings* node to maintain *Form Attributes*, the *Form Interface*, and *Global Definitions*.
- You use the inferior nodes of the *Pages and Windows* node to create the pages of your form, position elements on these pages, and determine the sequence in which you want to process these elements.

Inferior Nodes of Node - Pages and Windows

Nodes and the other inferior nodes:

Icon	Node Type	Description	Possible Directly Inferior Nodes
	Page	A page of the form. Directly inferior nodes of this node can be placed directly on the page.	Window, graphic, address
	Window	Output area on a page. There are these window types: <ul style="list-style-type: none"> • Main window • Secondary window • Copies window • Final window 	All except page nodes

Elementary Nodes (Without Inferior Nodes)

Icon	Node Type	Description
	Text	You use this node to print any texts (and table contents), except addresses.
	Graphic	You use this node to position graphics in the form. To include background graphics, use node type Page.
	Address	You use this node to include an address. The system reads the address data directly from the database tables and formats it for print output.

Table Output

Icon	Node Type	Description	Possible Directly Inferior Nodes
	Template	You use this node to print a table with fixed layout	All except window, page, table, and template nodes
	Table	You use this node to print a table with dynamic layout	Directly inferior nodes are automatically the three nodes Header , Main Area and Footer . As their directly inferior nodes, loop nodes, folder nodes, or table nodes are allowed.

	Table line	You use this node to print a table line in a table or template	Table cell
	Table cell	You use this node to print a table cell of a table line according to the line type	As with the template node

Flow Control

Icon	Node Type	Description	Possible Directly Inferior Nodes
	Command	You use this node to execute special commands (next page, paragraph numbering, printer control)	(no inferior nodes)
	Loop	You use this node to process inferior nodes repeatedly.	All except window and page nodes
	Alternative	You use this node to branch depending on a condition.	Directly inferior nodes are automatically the nodes TRUE and FALSE. The directly inferior nodes of these can be all nodes except window and page nodes.

Other Nodes

Icon	Node Type	Description	Possible Directly Inferior Nodes
	Folder	You use this node to combine inferior nodes to logical groups	All except window and page nodes
	Complex section	You use this node to combine all attributes of the node types template, table, loop, folder	All except window and page nodes
	Program lines	You use this node to execute ABAP program code (for example, conversion routines).	(no inferior nodes)

Printing System Fields :

The following rules apply for system fields:

- You can display the values of any system fields in the form
- You are not allowed to assign values to system fields (in program lines nodes)
- You can query all system fields (for example, in conditions) except **SFSY-FORMPAGES** and **SFSY-JOBPAGES**. You can query **SFSY-FORMPAGES** only within final windows.

Note: All System Fields of Smart Forms are available in Structure SFSY.

Field Name	Description
&SFSY-DATE&	Displays the date.
&SFSY-TIME&	Displays the time of day in the form HH:MM:SS.
&SFSY-PAGE&	Inserts the number of the current print page into the text. You determine the format of the page number (for example, Arabic, numeric) in the page node.
&SFSY-FORMPAGES&	Displays the total number of pages for the currently processed form. This allows you to include texts such as 'Page x of y' into your output.
&SFSY-JOBPAGES&	Contains the total page number of all forms in the currently processed print request.
&SFSY-WINDOWNAME&	Contains the name of the current window (string in the <i>Window</i> field)
&SFSY-PAGENAME&	Contains the name of the current page (string in the <i>Page</i> field)
&SFSY-XSF&	Smart Forms sets this flag (<i>SFSY-XSF</i> = 'x'), if you want to print the form in XSF format or HTML format. On the <i>Conditions</i> tab of a node, you can use this field to suppress output that is intended only for these output formats (for example, pushbuttons).
&SFSY-COPYCOUNT&	Queries whether the original is printed or which number the copy has. <i>COPYCOUNT</i> = 1 : orginal, <i>COPYCOUNT</i> = 2 : first copy; <i>COPYCOUNT</i> = 3: second copy, and so on.
&SFSY-COPYCOUNT0&	Queries whether the original is printed or which number the copy has. <i>COPYCOUNT</i> = 0 : orginal, <i>COPYCOUNT</i> = 1 : first copy; <i>COPYCOUNT</i> = 2 : second copy, and so on.
SFSY-SUBRC	Return value that you can query in program lines nodes. This enables

	<p>you to react dynamically to error situations during output. Up to now, this field can be used only for include texts and text modules:</p> <ul style="list-style-type: none"> • SFSY-SUBRC = 0 :Text module or include text found and printed • SFSY-SUBRC = 4 :Text module or include text not found
SFSY-USERNAME	Logon name of the user who prints the form.

When using the fields **&SFSY-FORMPAGES&** or **&SFSY-JOBPAGES&**, you must keep all output pages in the main memory till the end of the form or the print job to allow these fields to be replaced with their respective values. For large forms or print jobs, this may require a huge amount of memory space.

Output/Formatting Options :

Output Options for Field Contents

You can use the *Formatting options* to adapt the value of a field before printing it. You can enter the relevant parameters directly behind the field name. Write the abbreviations of the different options in uppercase letters.

Numeric Fields

- The system first evaluates the length (<length>), if specified.
- If no length is specified, the system displays the value in its overall length.
- The trailing blank indicates a positive sign. To suppress it, use the formatting option S.
- Any offset (<offset>) specified is ignored.

Sequence of evaluation: (<length>), sign to the left(<), suppress blanks (C), right-justified display (R), insert fillers (F).

Character Fields

By default, the system displays the value of a field in its overall length, but truncates trailing blanks.

Sequence of evaluation: suppress blanks (C), <offset> and (<length>), right-justified display (R), insert fillers (F).

List Of Formatting Options for Fields

Syntax	Description
&field+<offset>&	Skips offsets (<offset>) in the field value (character fields only). If the offset is greater than the length of the value, nothing is displayed.
&field(<length>)&	Sets the output length to <length>.
&field(*)&	If the field is defined by a Data Dictionary type, Smart Forms sets the output length to the value specified there.
&field(S)&	Suppresses the sign
&field(<)&	Displays the sign to the left of the number
&field(.<nat.number>)&	Limits output of decimal places to <nat.number>
&field(E<nat.number>)&	Displays the field value with the fixed exponent <nat.number>. The mantissa is adapted to this exponent by shifting the decimal character and inserting zeros.
&field(T)&	Suppresses thousand indicators when displaying fields of types DEC, CURR, INT, and QUAN.
&field(Z)&	Suppresses leading zeros in numbers
&field(I)&	Suppresses display of initial values
&field(K)&	Deactivates a conversion routine specified in the Data Dictionary
&field(R)&	Right-justified display. Use this option only when specifying an output length as well.
&field(F<filler>)&	Replaces left-justified blanks in the value by the fill character <filler>.
&field(L)&	Converts a date field to a local date and displays it. The date is then formatted using the edit mask JPDAT. Since this representation uses Japanese characters, it is only to be used in the Japanese version of the SAP System.
&field(C)&	The system takes the field value as a sequence of words separated by blanks. Option C shifts these words to the left and leaves one blank in-between as a separator. Any leading blanks are omitted. This effect corresponds to that of the ABAP statement CONDENSE .

Steps to Convert SmartForms Output to PDF :

1. Print the smartform to the spool.
2. Note the spool number.
3. Download a PDF file (Acrobat Reader) version of the spool by running Program RSTXPDFT4 and entering the noted spool number.

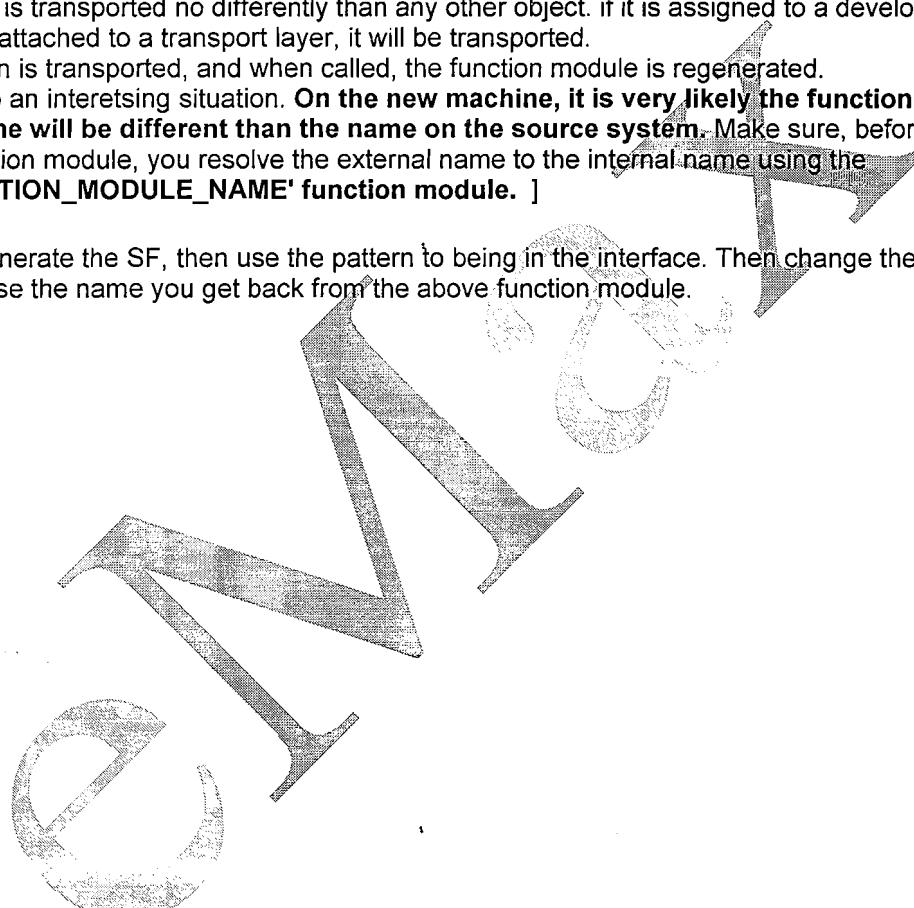
How do you make sure that both, the SMARTFORM & it's function module gets transported? Or does the FM with same name gets generated automatically in the transported client?

A smartform is transported no differently than any other object. if it is assigned to a development class that is attached to a transport layer, it will be transported.

The definition is transported, and when called, the function module is regenerated.

This leads to an interesting situation. **On the new machine, it is very likely the function module name will be different than the name on the source system.** Make sure, before you call the function module, you resolve the external name to the internal name using the '**'SSF_FUNCTION_MODULE_NAME'** function module.]

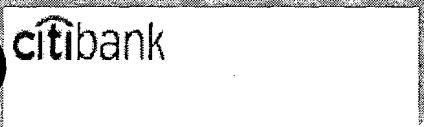
Typically, generate the SF, then use the pattern to bring in the interface. Then change the call function to use the name you get back from the above function module.



Example Program and Layout , Which Covers All the SmartForm Features:

Requirement : Develop a Form to Issue the Credit Card Statements to the Customers for City Bank.

Note : Sample Output of the form .

A					B	ORIGINAL																																				
C	<table border="1"> <tr><td colspan="2">Card NO : 018301004245</td></tr> <tr><td>Customer No</td><td>77777</td></tr> <tr><td>Customer Name</td><td>Adimulam</td></tr> <tr><td>Street/Hno</td><td>Sanath Nagar</td></tr> <tr><td>City</td><td>Sanath Nagar</td></tr> <tr><td colspan="2">ganapati.adimulam@yahoo.com</td></tr> </table>		Card NO : 018301004245		Customer No	77777	Customer Name	Adimulam	Street/Hno	Sanath Nagar	City	Sanath Nagar	ganapati.adimulam@yahoo.com		D	PAGE 1 Of 2 <table border="1"> <tr><td>Bank Key : 88991122</td></tr> <tr><td>Bank Name : City Bank INDIA Ltd</td></tr> <tr><td>City : Mumbai</td></tr> <tr><td>Street : Andheri</td></tr> <tr><td>Web : www.citibank.com</td></tr> </table>			Bank Key : 88991122	Bank Name : City Bank INDIA Ltd	City : Mumbai	Street : Andheri	Web : www.citibank.com																			
Card NO : 018301004245																																										
Customer No	77777																																									
Customer Name	Adimulam																																									
Street/Hno	Sanath Nagar																																									
City	Sanath Nagar																																									
ganapati.adimulam@yahoo.com																																										
Bank Key : 88991122																																										
Bank Name : City Bank INDIA Ltd																																										
City : Mumbai																																										
Street : Andheri																																										
Web : www.citibank.com																																										
F	<table border="1"> <thead> <tr><th>SNO</th><th>Trasaction Date</th><th>Trasact ion ID</th><th>Trasaction Amount</th><th>Trasaction Summary</th></tr> </thead> <tbody> <tr><td>1</td><td>07.01.2008</td><td>T1</td><td>1,000.00</td><td>SHOP @ RS BROS</td></tr> <tr><td>2</td><td>07.01.2008</td><td>T2</td><td>3,000.00</td><td>SHOP @ CHERMAS</td></tr> <tr><td>3</td><td>07.01.2008</td><td>T3</td><td>200.00</td><td>PETROL @ INDIAN OIL, ERRA GADDA</td></tr> <tr><td>4</td><td>10.01.2008</td><td>T4</td><td>200.00</td><td>PETROL @ INDIAN OIL, ERRA GADDA</td></tr> <tr><td>5</td><td>11.01.2008</td><td>T5</td><td>800.00</td><td>SWEETS @ ABHIRUCHI , AMEERPET</td></tr> <tr><td>6</td><td>12.01.2008</td><td>T6</td><td>5,000.00</td><td>SAREES @ KALA NIKETAN , AMEERPET</td></tr> </tbody> </table>				SNO	Trasaction Date	Trasact ion ID	Trasaction Amount	Trasaction Summary	1	07.01.2008	T1	1,000.00	SHOP @ RS BROS	2	07.01.2008	T2	3,000.00	SHOP @ CHERMAS	3	07.01.2008	T3	200.00	PETROL @ INDIAN OIL, ERRA GADDA	4	10.01.2008	T4	200.00	PETROL @ INDIAN OIL, ERRA GADDA	5	11.01.2008	T5	800.00	SWEETS @ ABHIRUCHI , AMEERPET	6	12.01.2008	T6	5,000.00	SAREES @ KALA NIKETAN , AMEERPET	G	Total Is : 10200.00	
SNO	Trasaction Date	Trasact ion ID	Trasaction Amount	Trasaction Summary																																						
1	07.01.2008	T1	1,000.00	SHOP @ RS BROS																																						
2	07.01.2008	T2	3,000.00	SHOP @ CHERMAS																																						
3	07.01.2008	T3	200.00	PETROL @ INDIAN OIL, ERRA GADDA																																						
4	10.01.2008	T4	200.00	PETROL @ INDIAN OIL, ERRA GADDA																																						
5	11.01.2008	T5	800.00	SWEETS @ ABHIRUCHI , AMEERPET																																						
6	12.01.2008	T6	5,000.00	SAREES @ KALA NIKETAN , AMEERPET																																						
H	Grand Total : 12200.00																																									

Signature Of Manager

I



Input & Outputs for Each Window :

A : Print the Logo Of the City Bank. In Real time , Either Logo or GRAPHIC is provided. If GRAPHIC is not provided, Convert it into GRAPHIC through SE78.

B : Print as ORIGINAL for the Original and for Copies as COPIES . This is of Window Copies. Use the System Variable &SFSY-COPYCOUNT& , Check it and Print it.

C : Address of the Customer from KNA1 and Print the Credit Card No from the Custom Table.

D : Print the Bank Address , Here in this example, We are printing the fixed Address i.e given in the Form Output. In Real time, Access table BNKA(Bank Master Data).

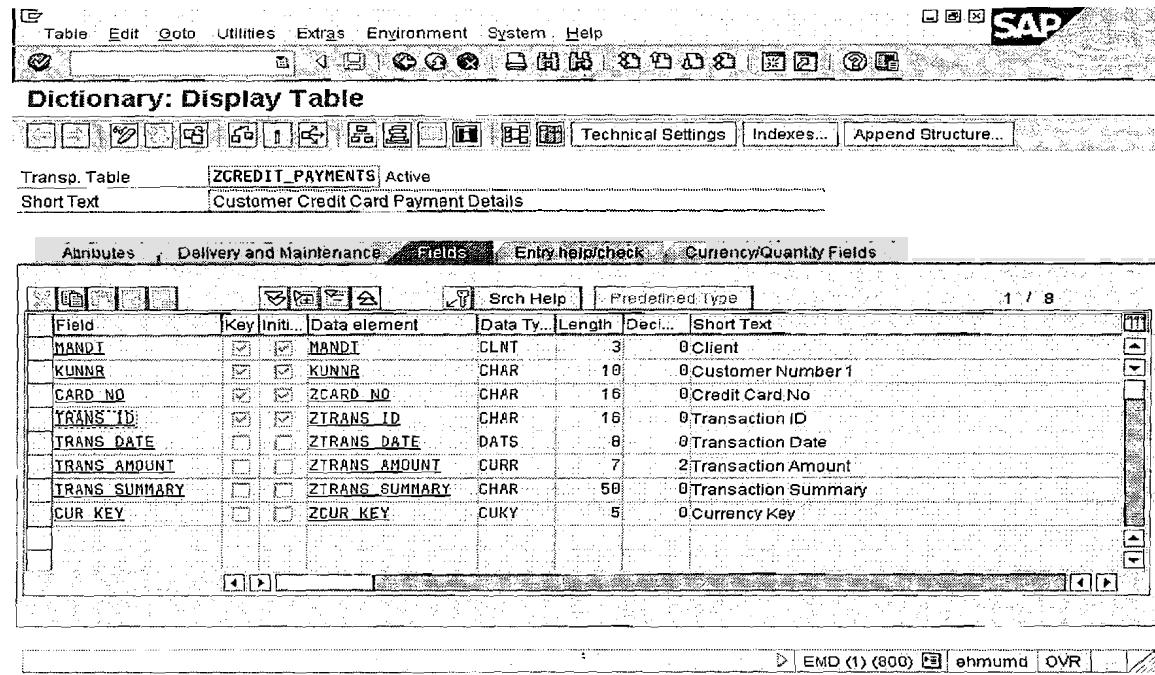
E: Print the Page numbers using the system variable

F,G,H: Print the Credit Transaction Details in the Tabular Format according to the Output format.

Note: Print the Total at Page Level (at end of each page) and also Grand Totals in the Last Page(at the End of the Main Window).

I: Print the Signature of the HR Manager as a scanned image (GRAPHIC) and Only in the last page (End Of Main Window).

Create a Custom Table with the below details to Maintain the Customer Credit Expenses .

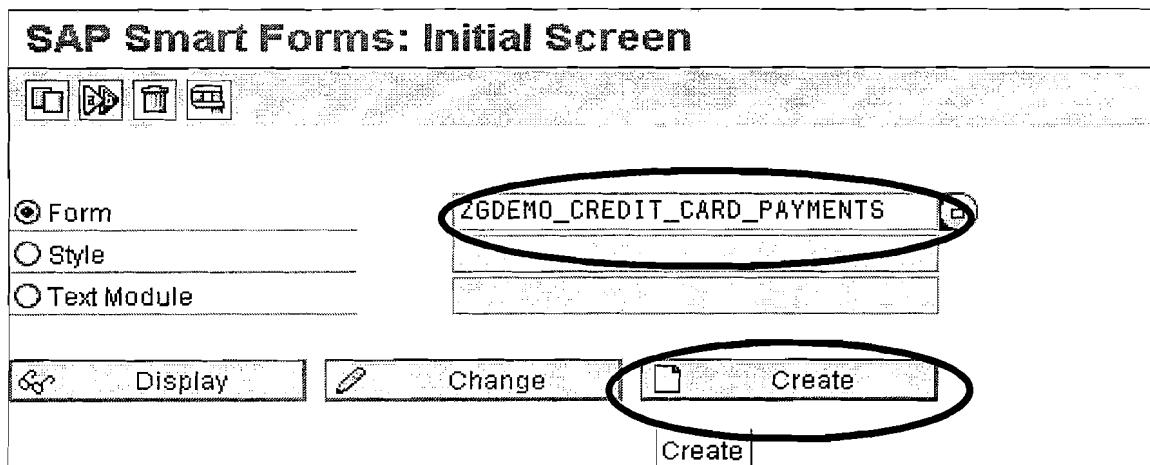


Maintain Data Records for some Customers i.e Utilities -> Table
Contents -> Create Entries.

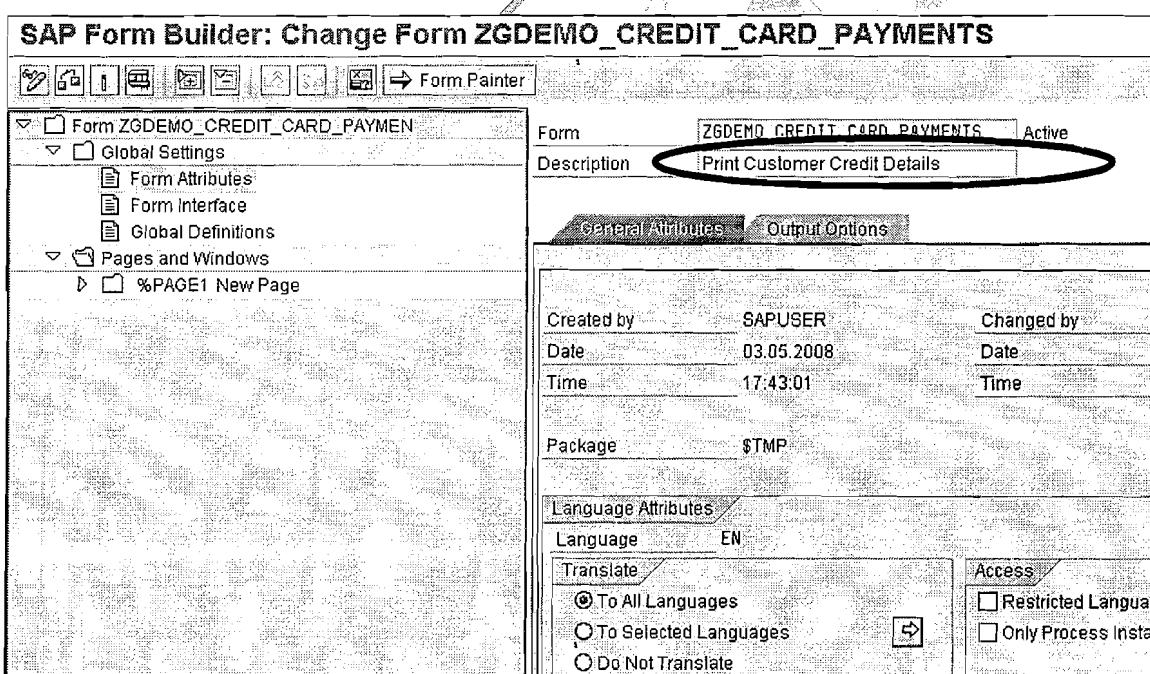
The screenshot shows the SAP Data Browser displaying the ZCREDIT_PAYMENTS table with the following data:

Cl.	Customer	CARD_NO	TRANS_ID	TRANS_DATE	TRANS_AMOU	TRANS_SUMMARY	CUR_KEY
800	0000077777	018301004245-T1		07.01.2008	1,000.00	SHOP @ RS BROS	INR
800	0000077777	018301004245-T2		07.01.2008	3,000.00	SHOP @ CHERMAS	INR
800	0000077777	018301004245-T3		07.01.2008	200.00	PETROL @ INDIAN OIL,ERRA GADDA	INR
800	0000077777	018301004245-T4		10.01.2008	200.00	PETROL @ INDIAN OIL,ERRA GADDA	INR
800	0000077777	018301004245-T5		11.01.2008	800.00	SWEETS @ ASHIRUCHI ,AMEERPET	INR
800	0000077777	018301004245-T6		12.01.2008	5,000.00	SAREES @ KALA NIKETAN ,AMEERPET	INR
800	0000077777	018301004245-T7		13.01.2008	1,000.00	BAGS @ PALEM LEATHERS ,PANJA GUTTA	INR
800	0000077777	018301004245-T8		13.01.2008	1,000.00	TICKETS @ KESINENI TRAVELS - ONLINE	INR

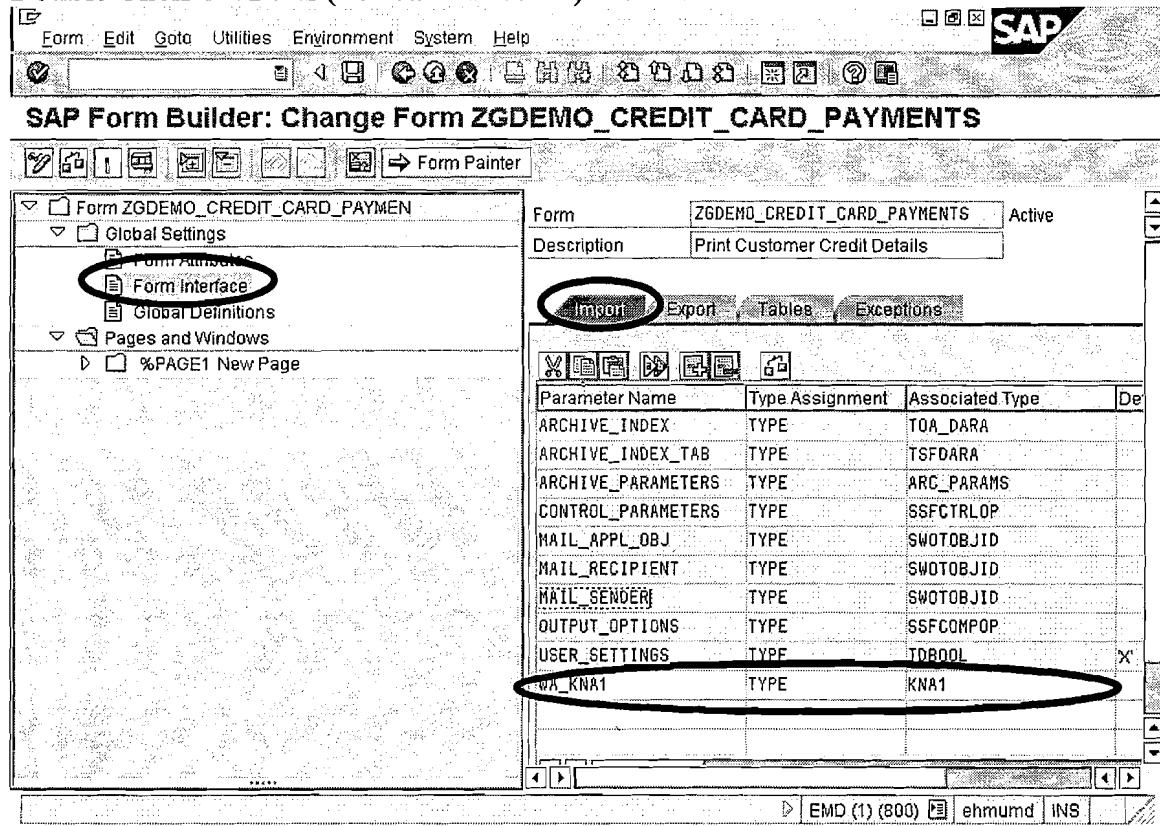
Step1 : Execute Transaction SMARTFORMS -> Select Form Option and Provide Name -> Click On Create.



Enter the Description

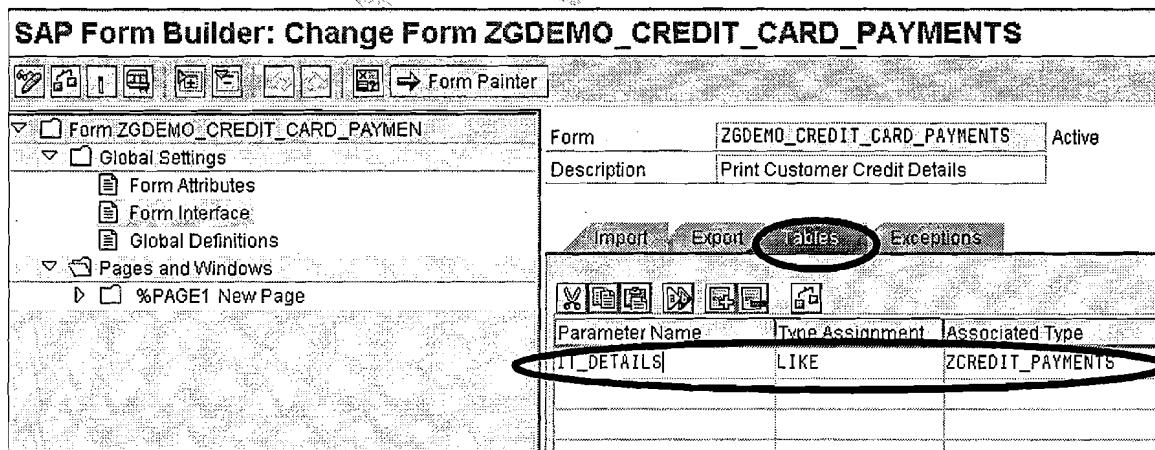


Double Click On Form(Function Module) Interface



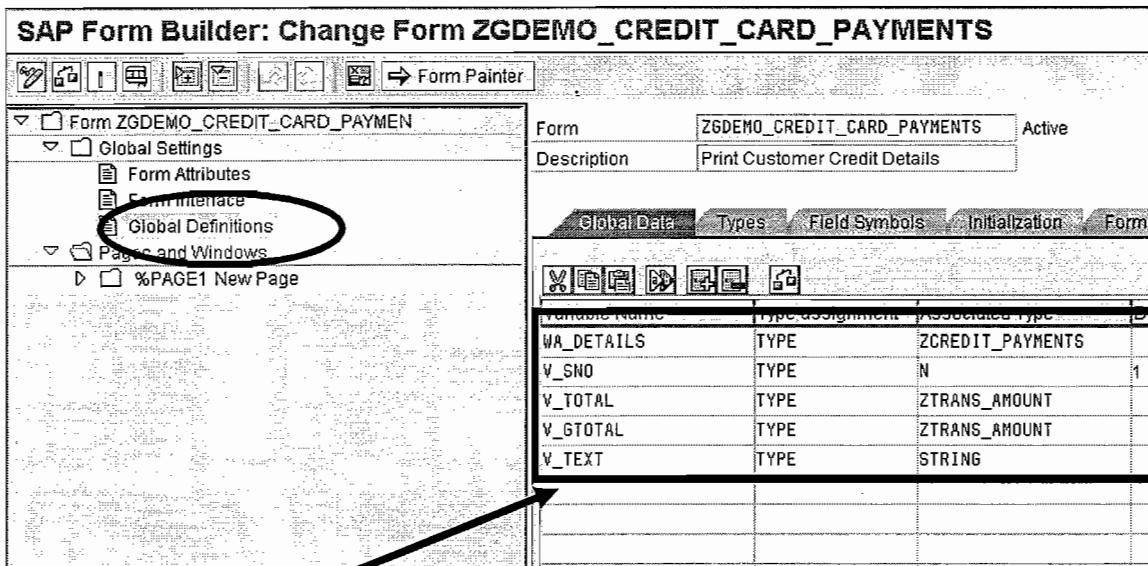
Click On Import and Provide all the Importing Parameters

Enter WA_KNA1 TYPE KNA1 . (Customer Master Data to Print Customer Address).



Click on Tables to Declare the Internals passed from the Driver Program
Enter IT_DETAILS LIKE ZCREDIT_PAYMENTS.

Double Click On Global Definitions



Declare the Variables

Till now Global Settings i.e Form Attributes, Form Interface and Global Settings are Completed.

Let us Create and Complete the Respective Functionality Window by Window

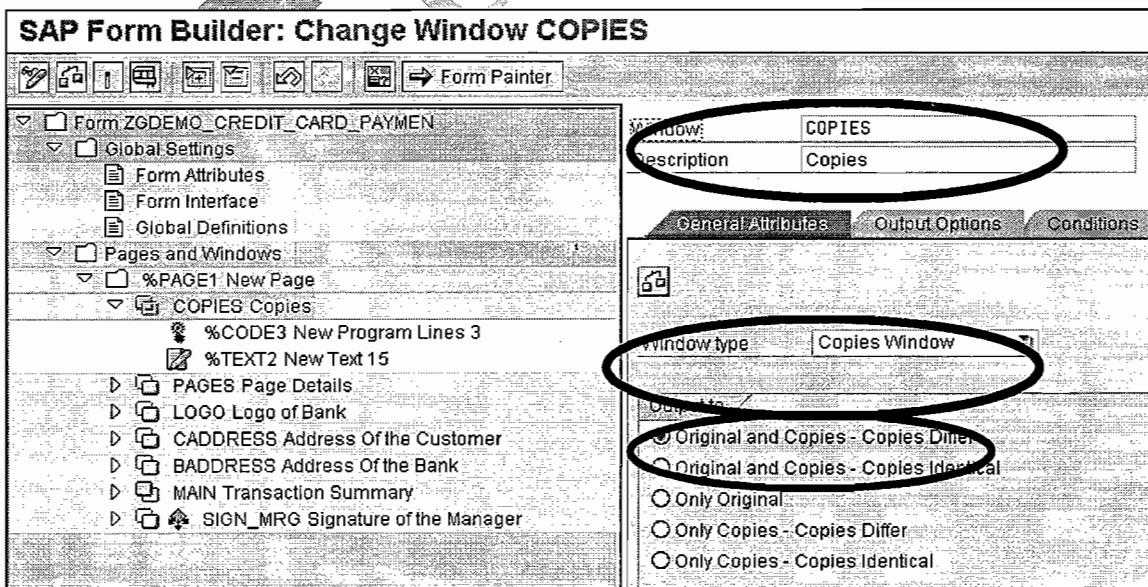
1) Window COPIES : **B**

Right Click On PAGE1 from Navigation Tree -> Create Window

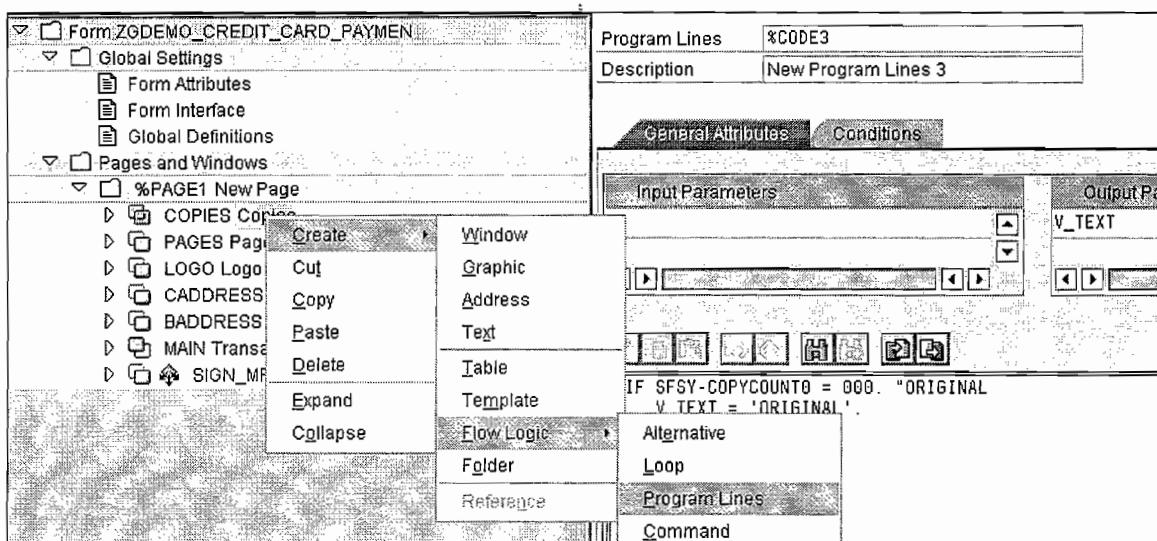
Window Name as COPIES and Description.

Select Window types Copies Window

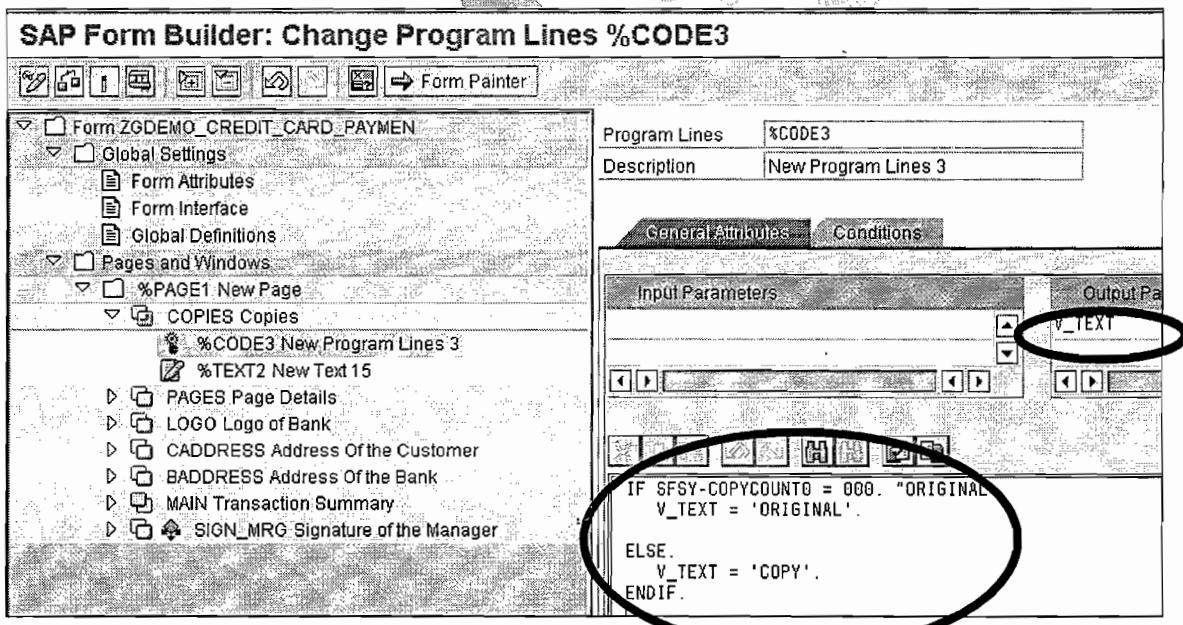
Select - Original and Copies



Right Click On COPIES Window ->Create FlowLogic -> Program Lines

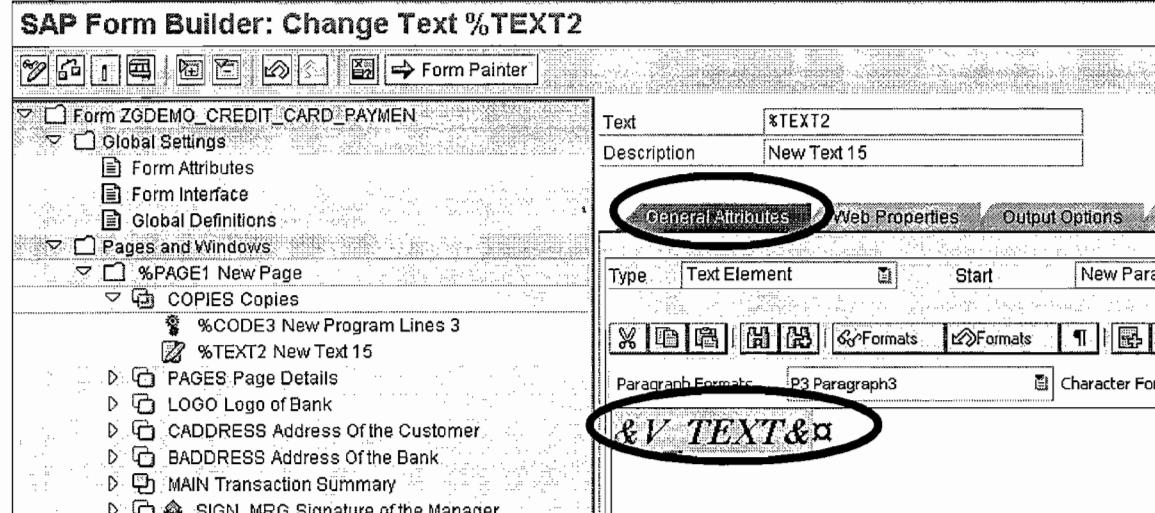


Enter Output Parameter V_TEXT



Check the SYSF-COPYCOUNT and Assign the text accordingly into V_TEXT.

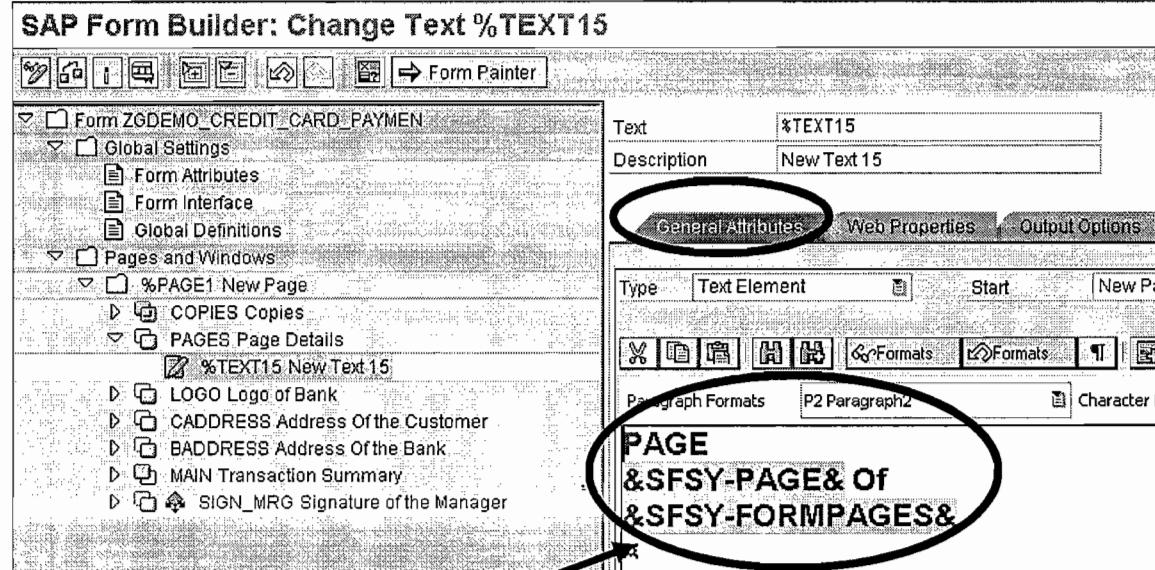
Right Click on COPIES WINDOW and Create TEXT(Text Element)



Click On General Attributes and Enter &V_TEXT& in the Text Editor.

2. Window PAGES: E

- A) Right Click on the COPIES Window -> Create->Window.
Window Name as PAGES
Description Page Details
- B) Right Click On Window PAGES -> Create -> TEXT(Text Element)
Click On General Attributes



Enter the Page Details From SFSY .

A

3. Window LOGO:

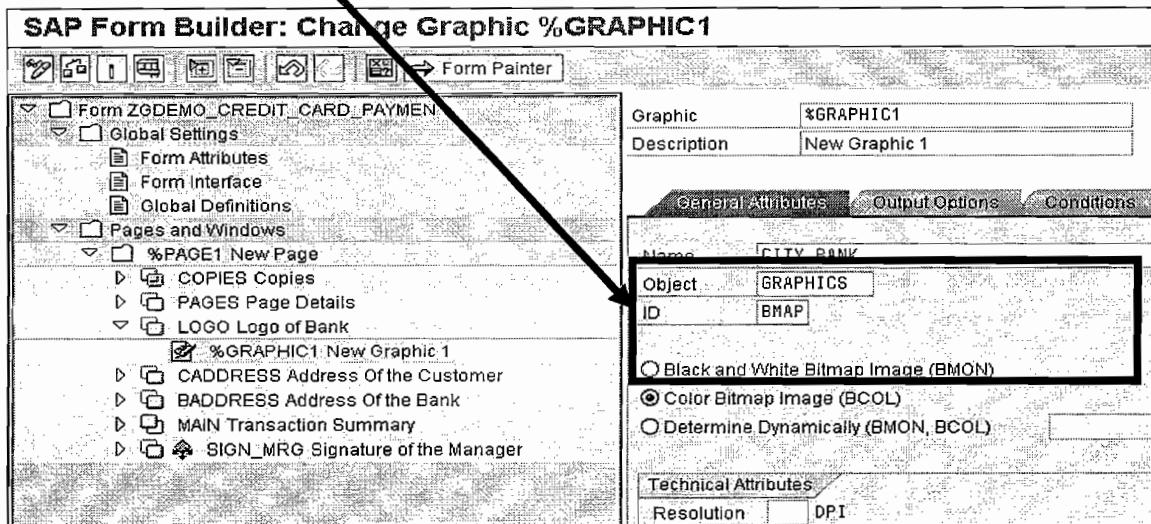
Right Click On PAGES Window -> Create -> Window

Window Name LOGO

Description Logo Of Bank.

Right Click On LOGO Window -> Create -> GRAPHIC

Provide the GRAPHIC Details.

**4. Window CADDRESS : Address Of Customer**

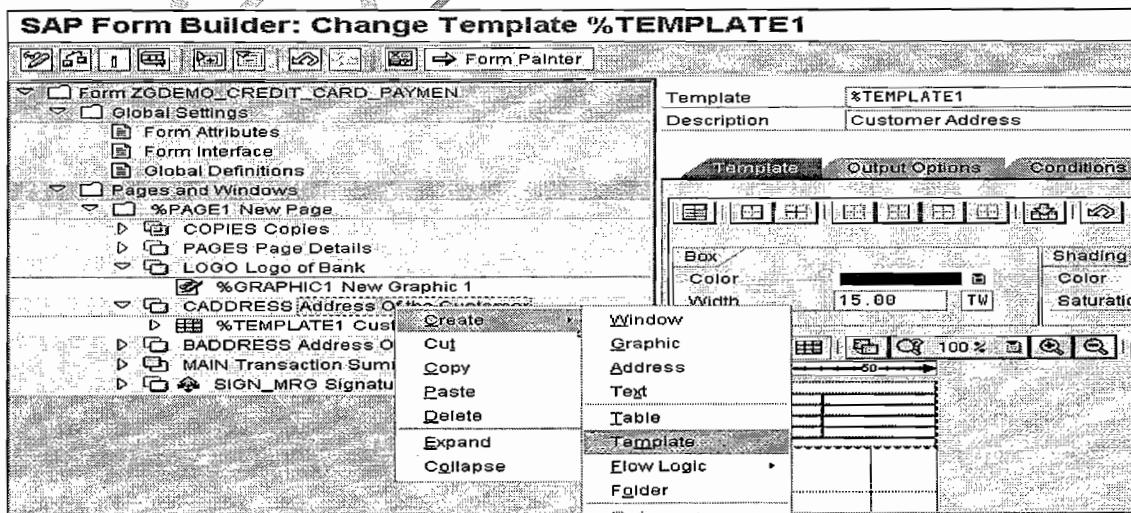
Right Click On Window LOGO -> Create -> Window

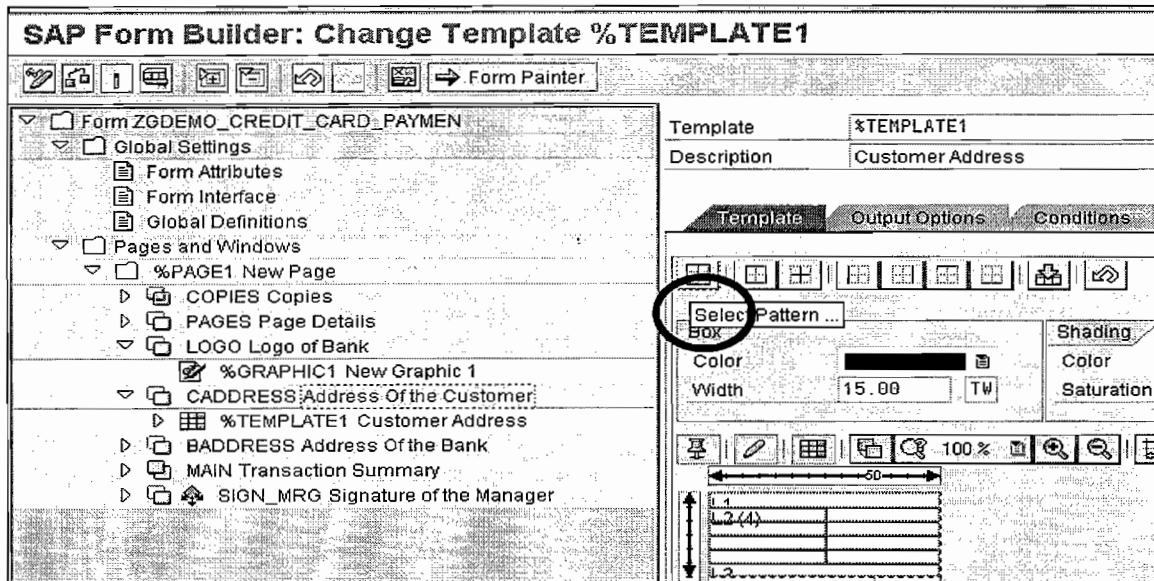
Window : CADDRESS

Description : Address Of Customer

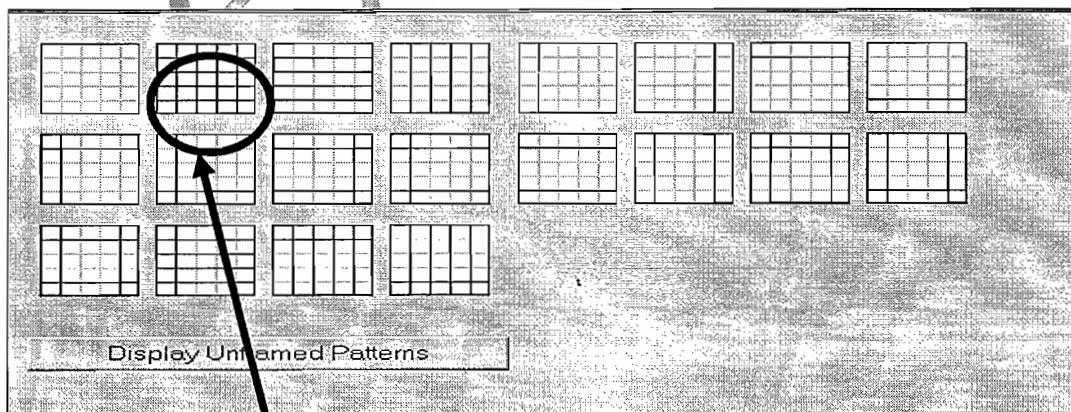
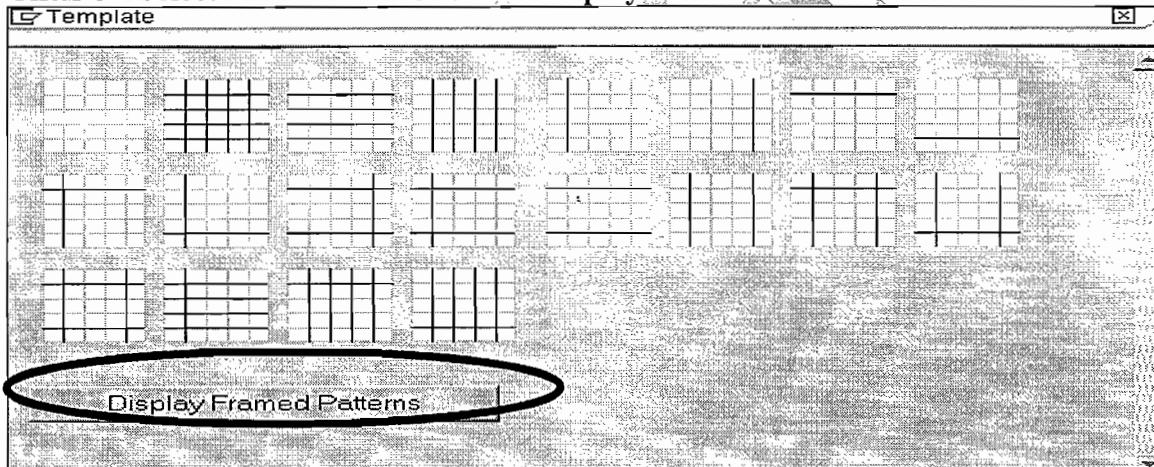
C

Now Right Click On Window CADDRESS -> Create -> Template



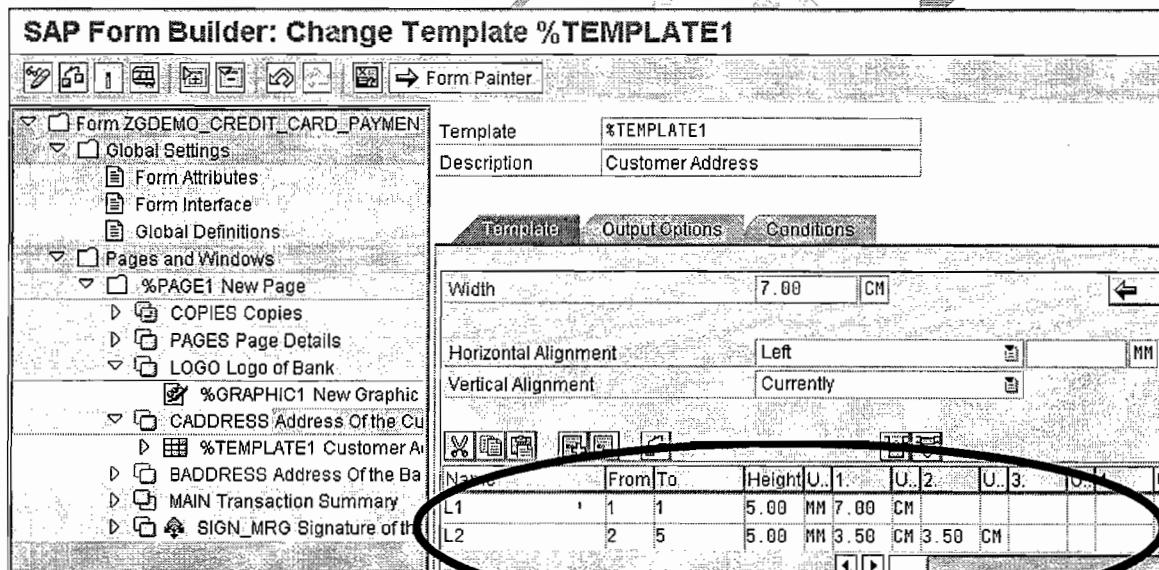
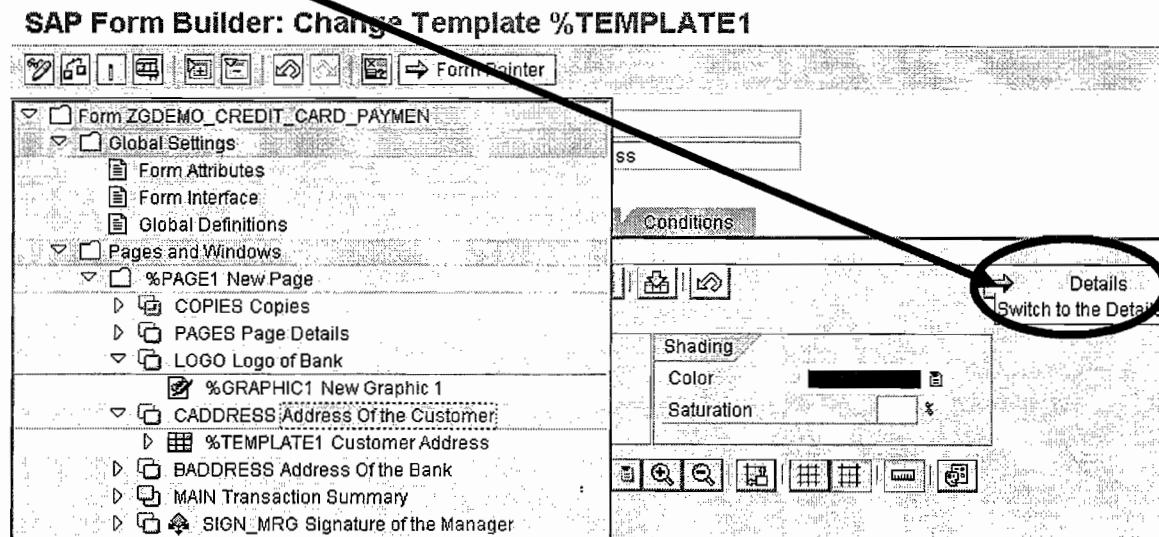


Click On Select Pattern and Click On Display Framed Pattern



Select One of the Pattern , which is used for Display purpose(Rows, Columns..)

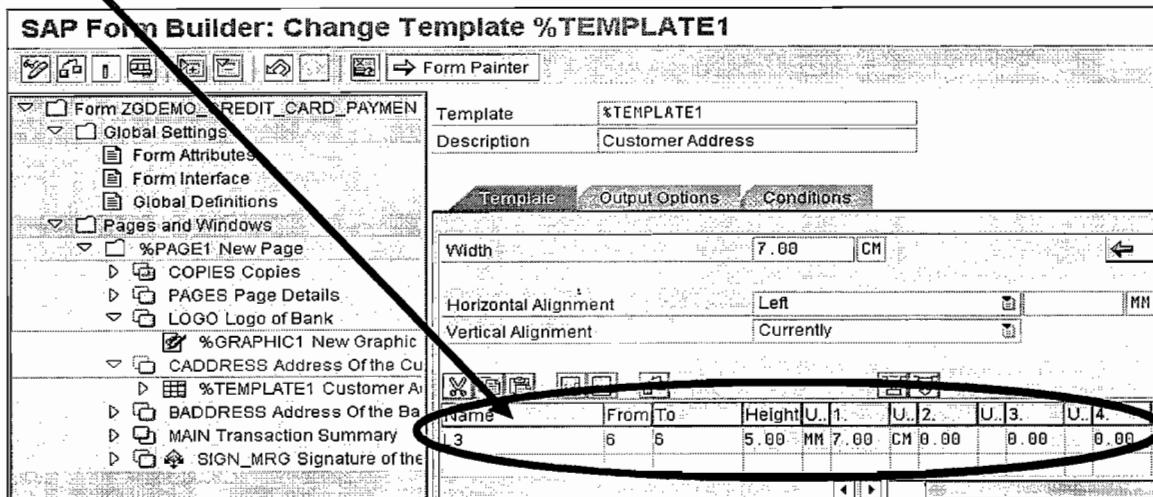
Click On Details to Define the Line Types for the Rows Of the Template like below



Define the Line Types

L1 for Row1

L2 for Row2 to Row5

L3 for Row6

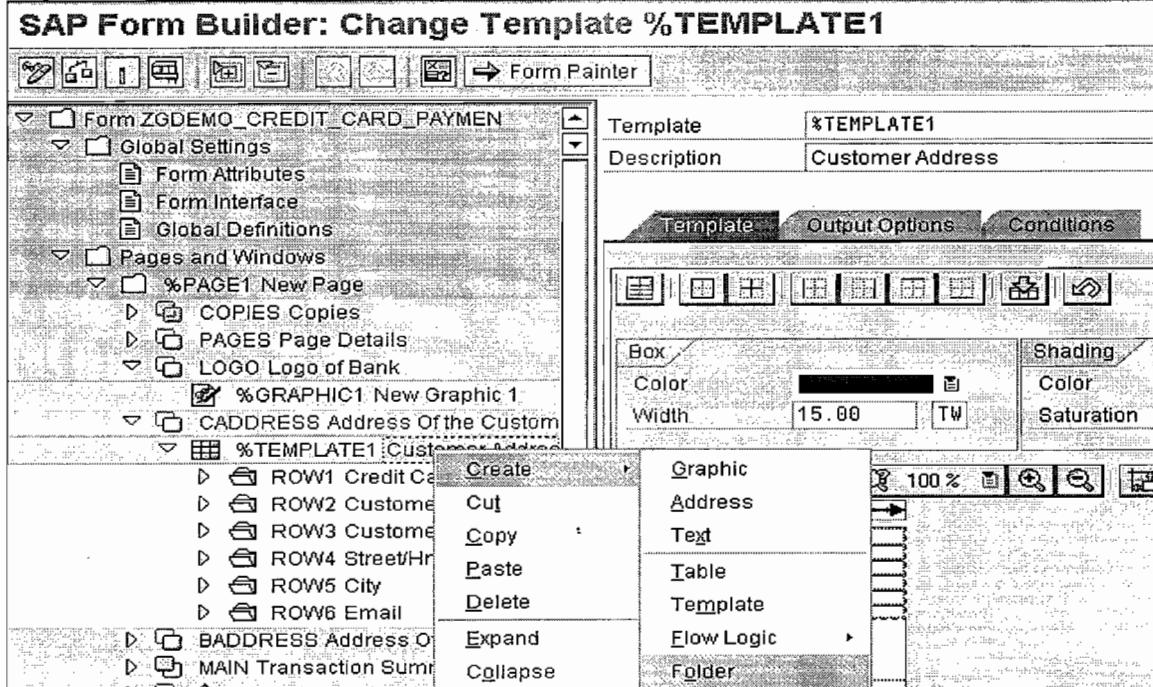
Now Design of Template is Completed.

Printing Data in the Rows Of Template

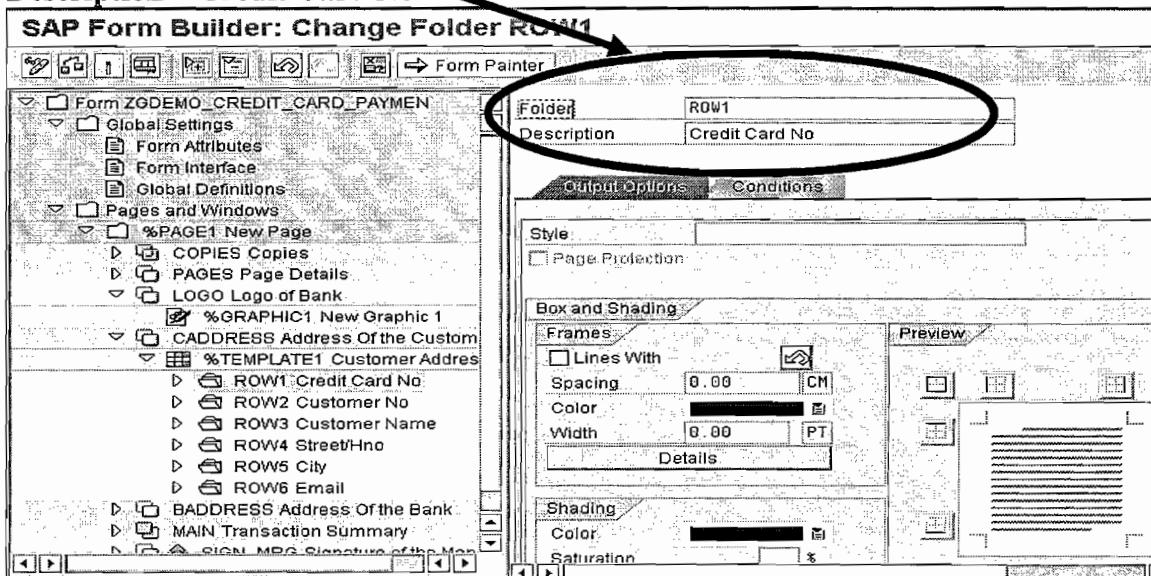
Note : Data should be printed on the Each Cell of the Template and Each Cell Requires one TEXT Element. So that it is better to group all the TEXT Elements of Each ROW into One FOLDER, this helps to understand the flow clearly.

Printing ROW1 :

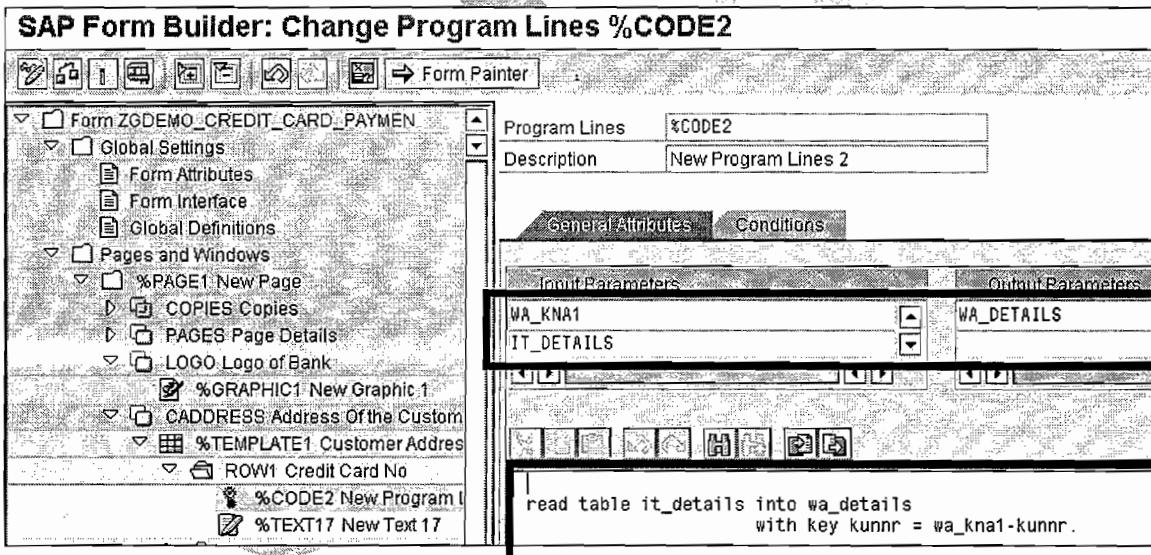
Right Click On TEMPLATE -> Create -> Folder



Folder ROW1
Description Credit Card No

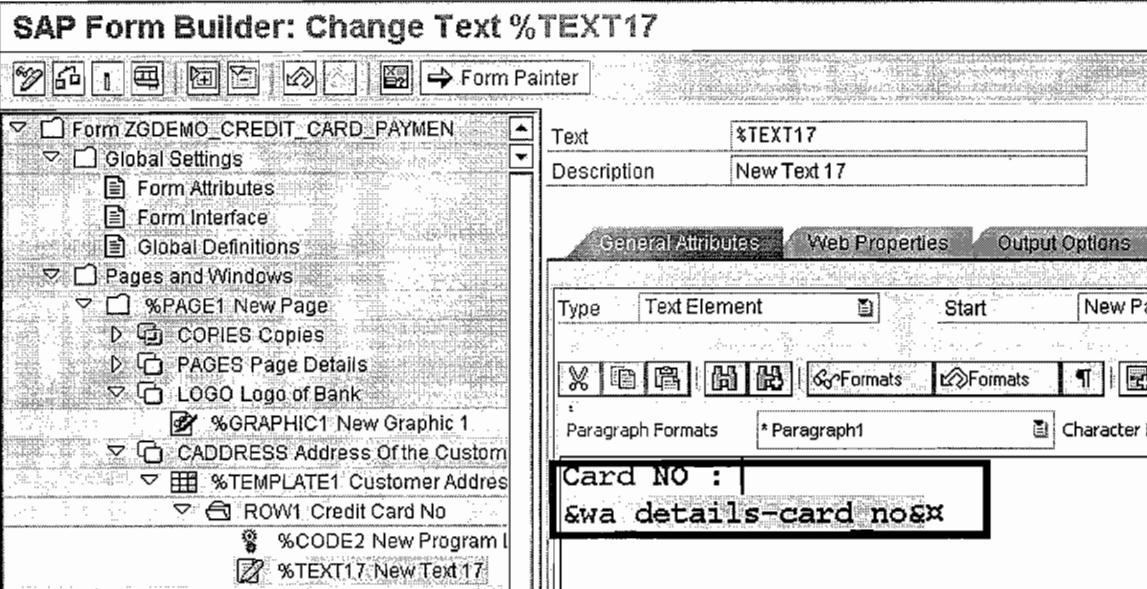


Right Click On ROW1(Folder) -> Create -> Program Lines



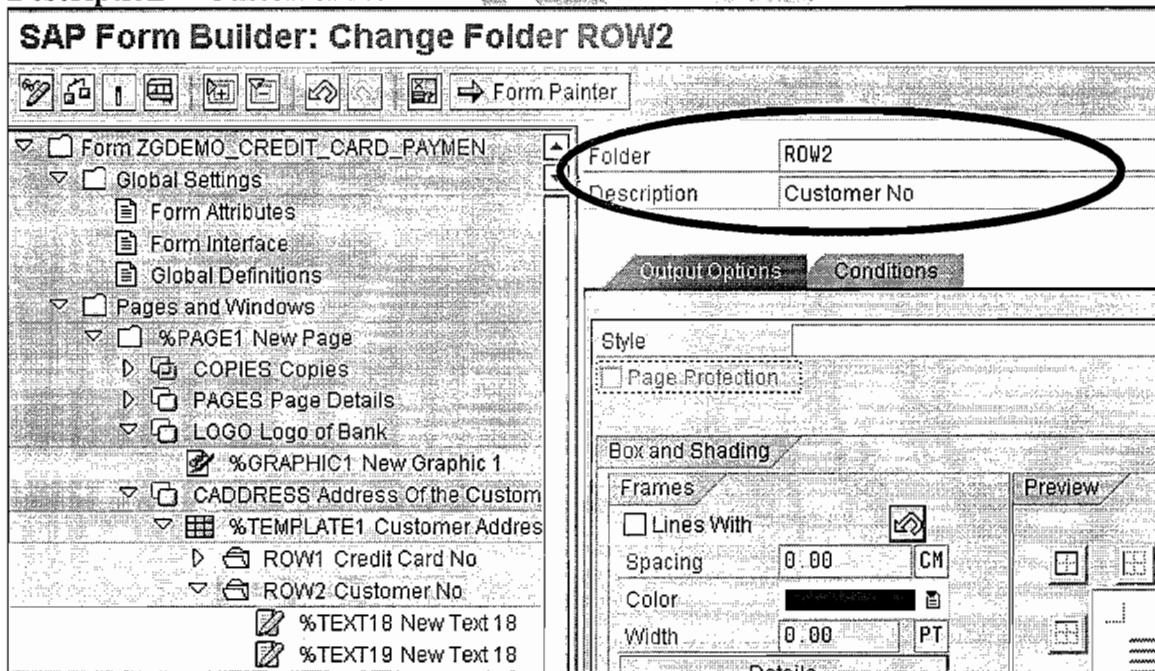
Provide Inputs and Outputs and Read the Card Details from Itab IT_DETAILS
For the Input Customer No.

To Print it, Right Click ROW1 -> Create TEXT after Program Lines .
Click On General Attribute -> write Card No Details.

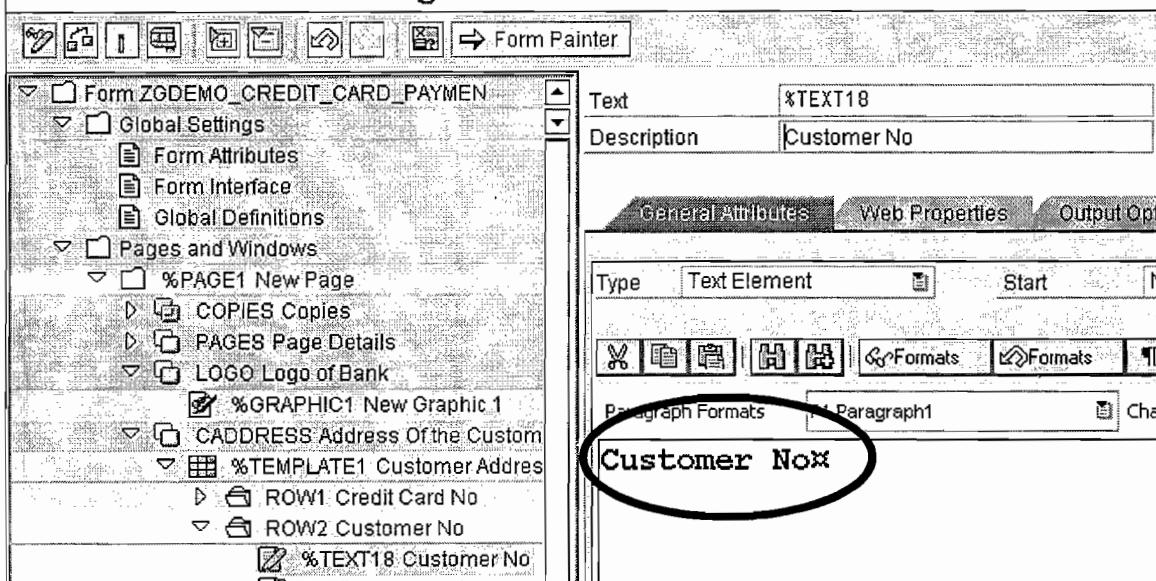
**ROW 2 :**

Right Click On ROW1(Folder) and Create FOLDER for ROW2 for Customer No.

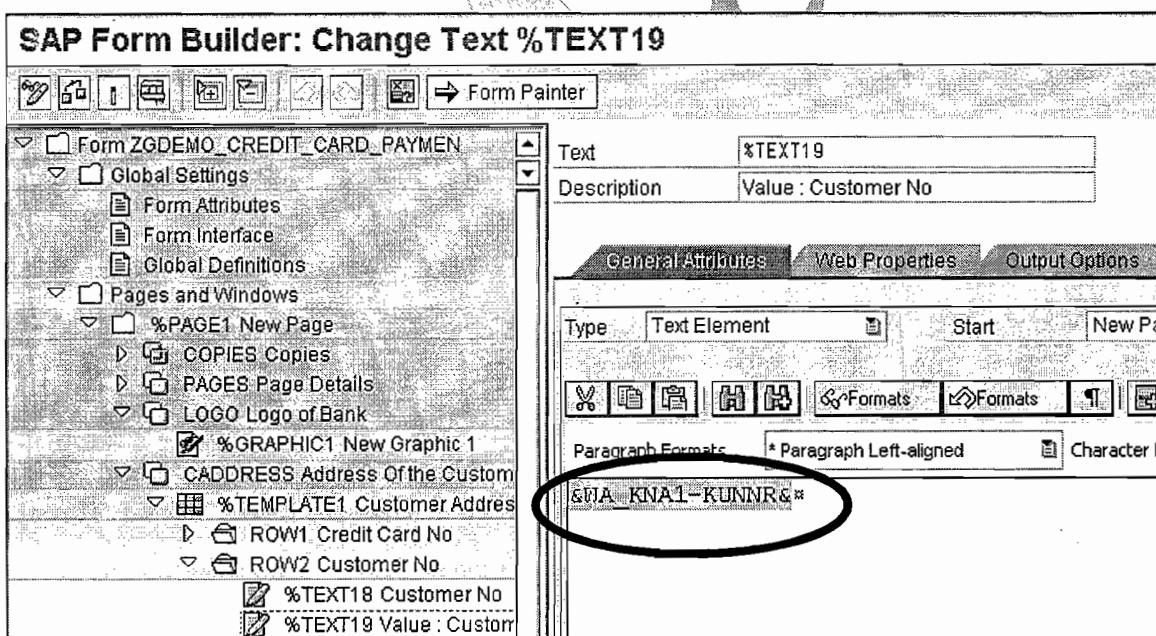
Folder ROW2
Description Customer No



Right Click On ROW2-> Create TEXT -> General Attribute -> Customer No

SAP Form Builder: Change Text %TEXT18

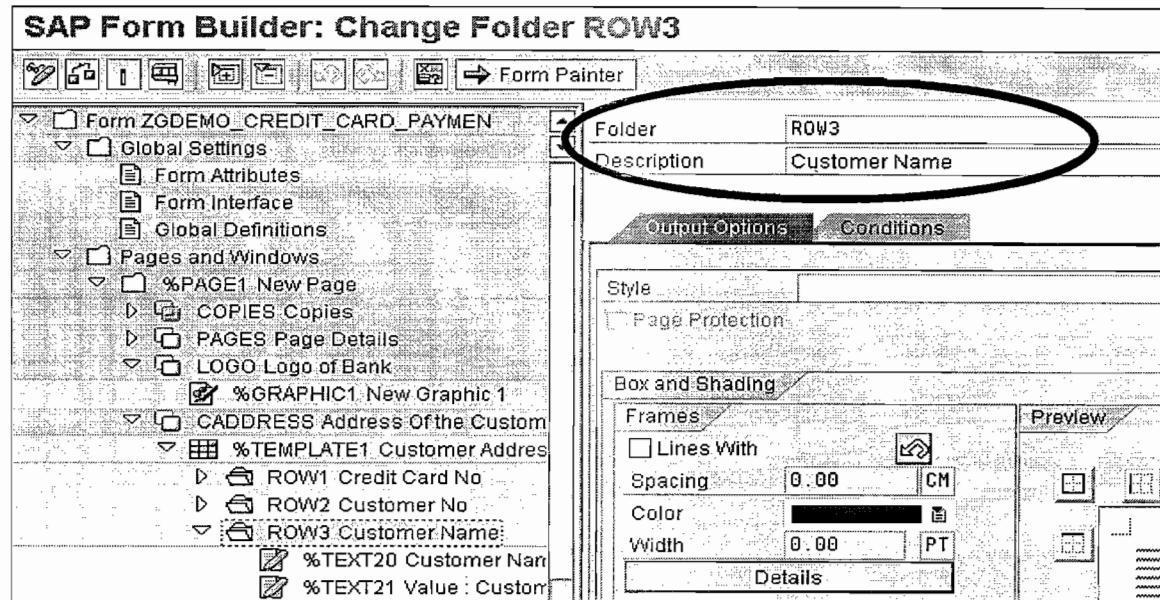
Right Click On ROW2-> Create TEXT -> General Attribute -> &WA_KNA1-KUNNR&



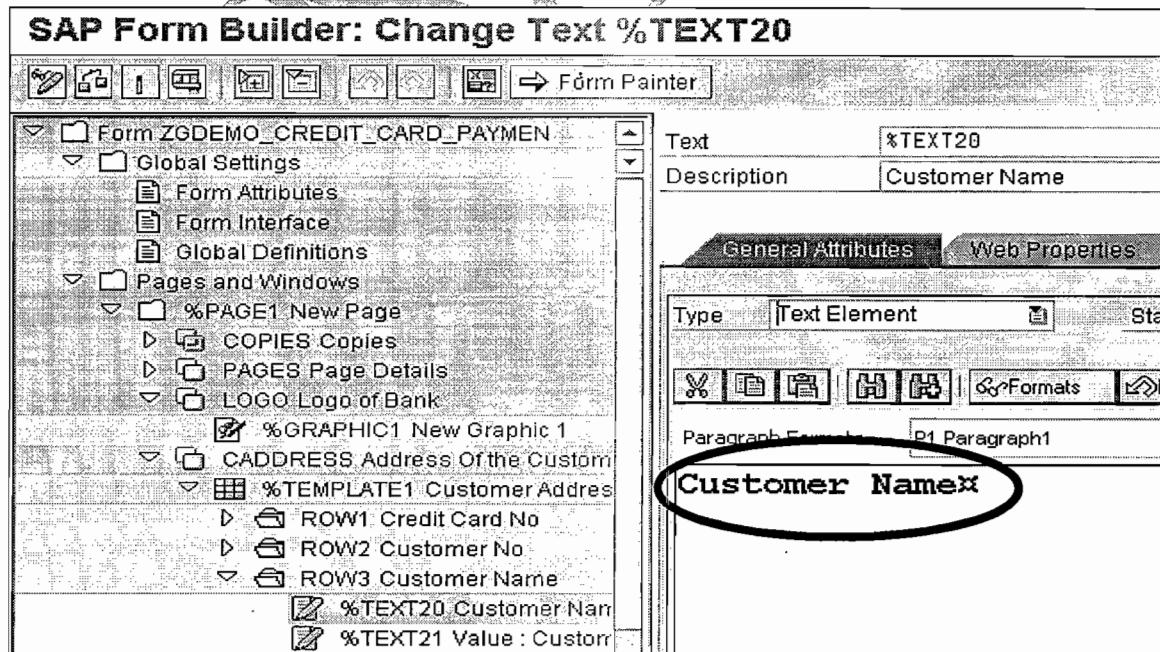
ROW 3 :

Right Click On ROW2(Folder) and Create FOLDER for ROW3 for Customer Name.

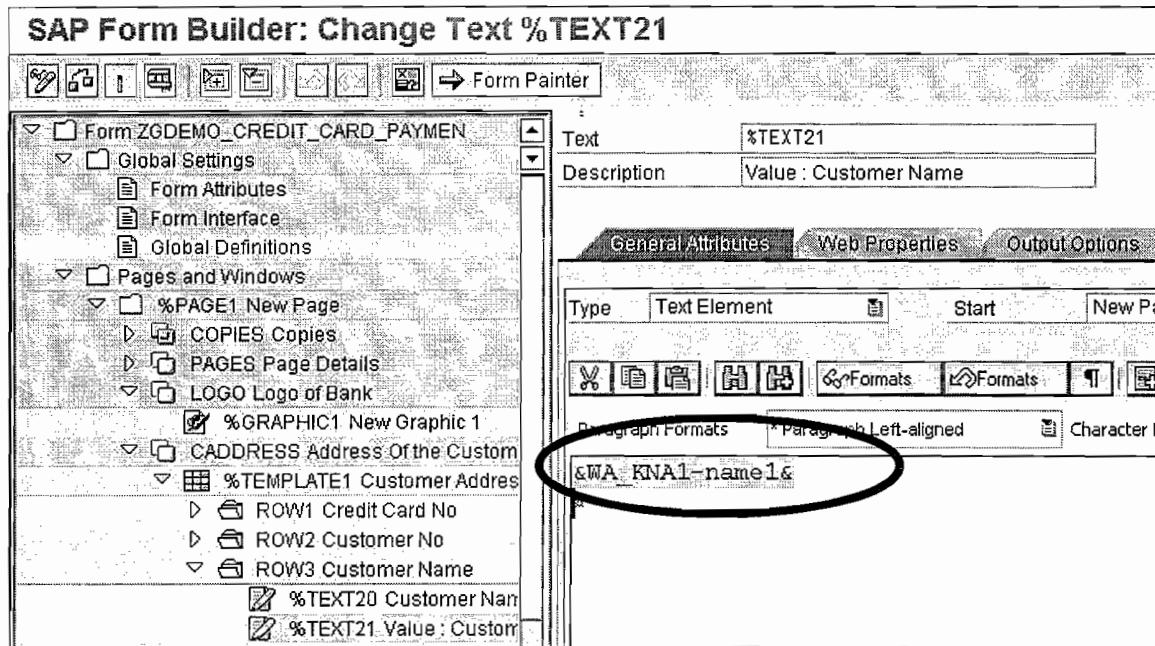
Folder ROW3
Description Customer Name



Right Click On ROW3-> Create TEXT -> General Attribute -> Customer Name



Right Click On ROW3-> Create TEXT -> General Attribute -> &WA_KNA1-NAME1&



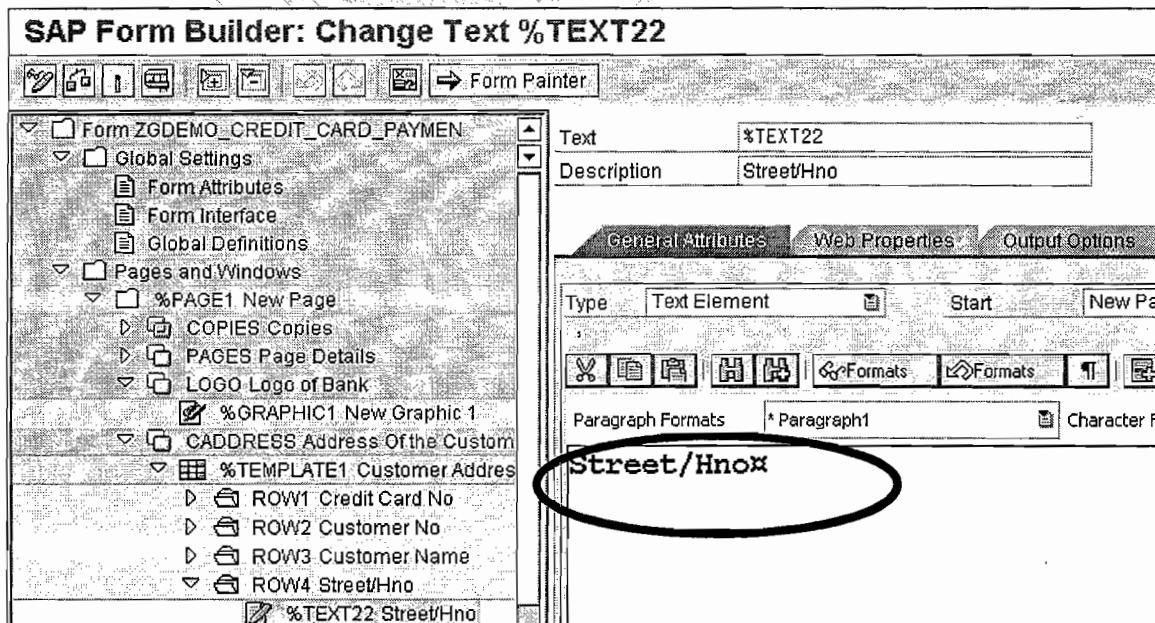
ROW4 :

Right Click On ROW3(Folder) and Create FOLDER for ROW4 for Street/Hno.

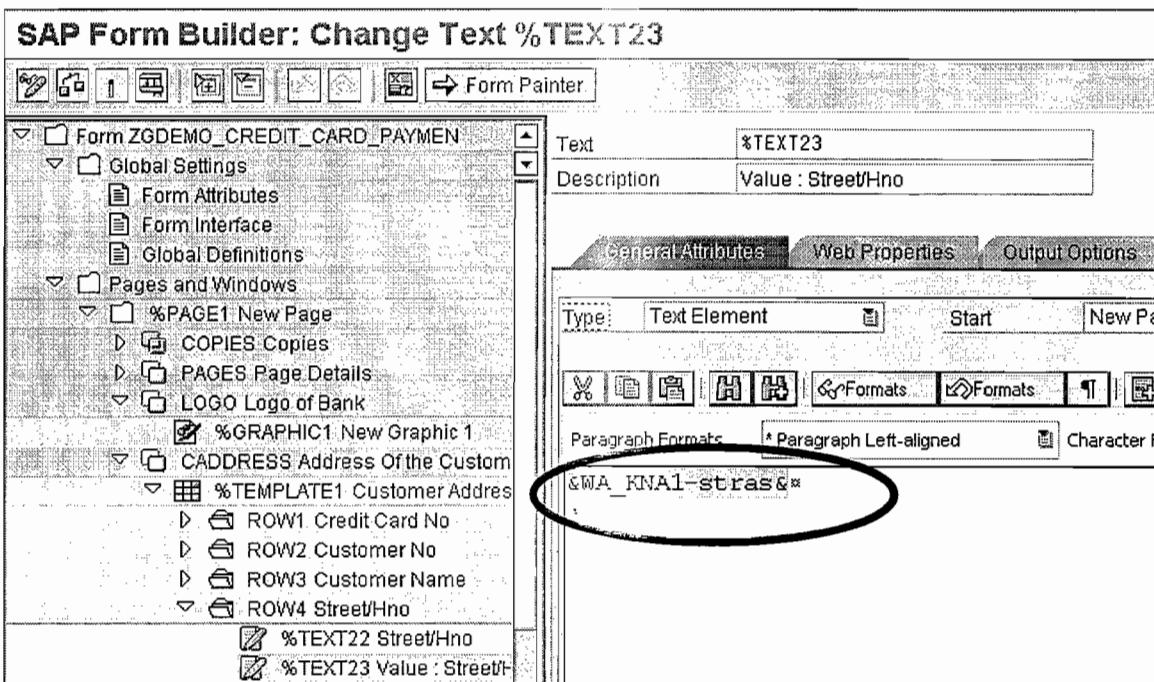
Folder ROW4

Description Street/Hno

Right Click On ROW4-> Create TEXT -> General Attribute -> Street/Hno

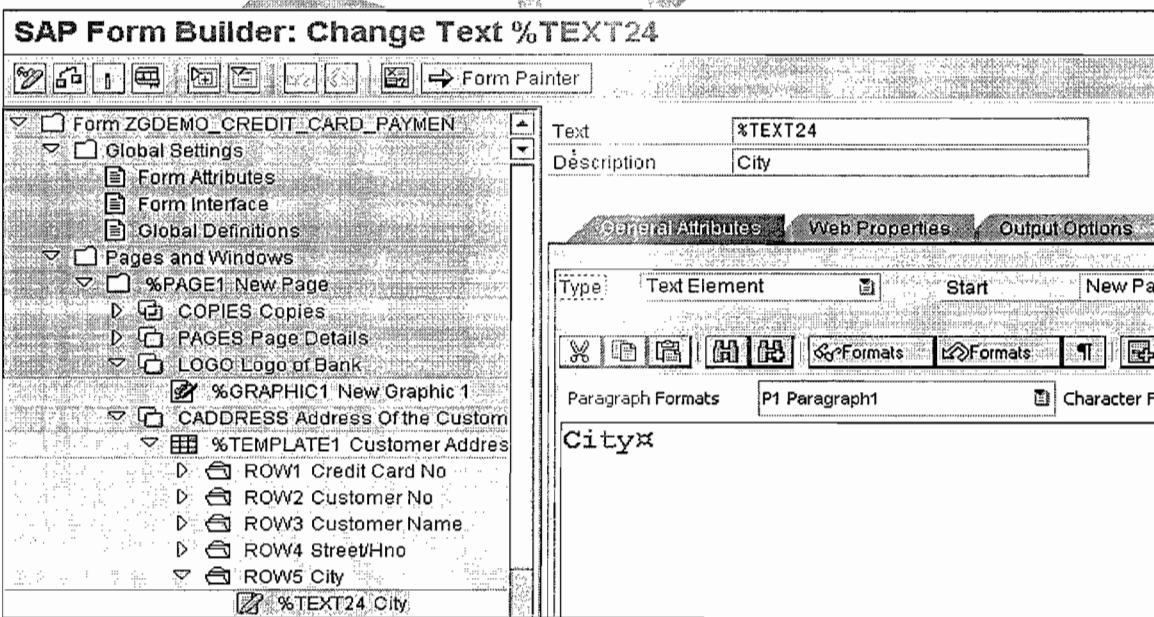


Right Click On ROW4-> Create TEXT -> General Attribute -> &WA_KNA1-STRAS&

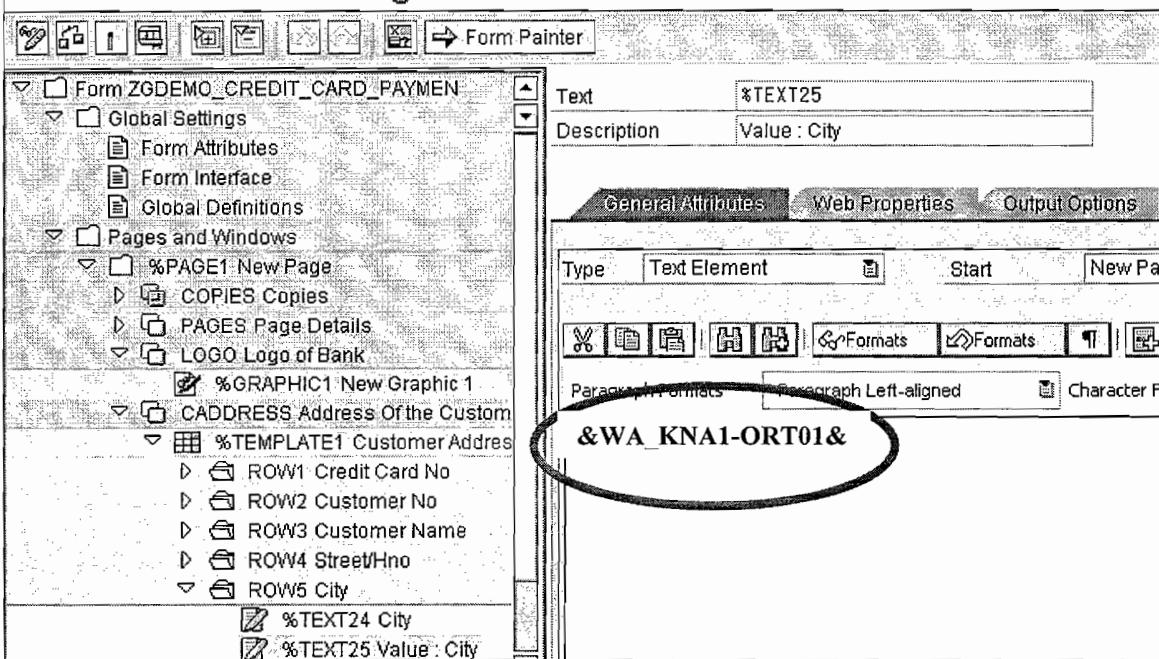
**ROW 5 :**

Right Click On ROW4(Folder) and Create FOLDER for ROW5 for City
Folder ROW5
Description City

Right Click On ROW5-> Create TEXT -> General Attribute -> City



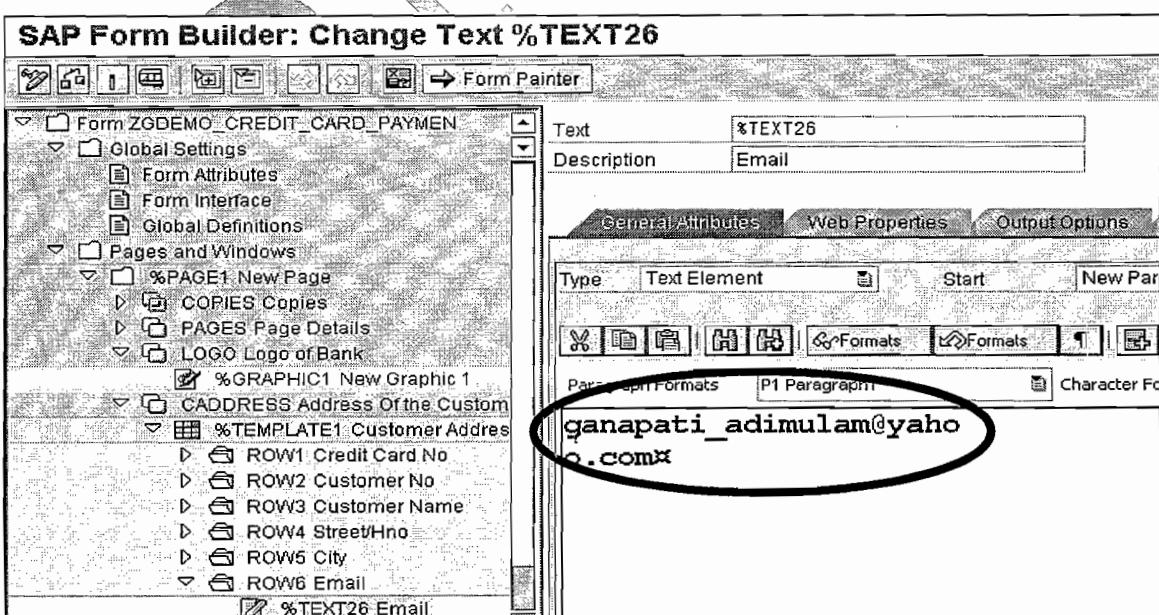
Right Click On ROW5-> Create TEXT -> General Attribute -> &WA_KNA1-
 ORT01&

SAP Form Builder: Change Text %TEXT25**ROW 6 :**

Right Click On ROW5(Folder) and Create FOLDER for ROW6 for Customer Email.

Folder ROW6
Description Email

**Right Click On ROW6-> Create TEXT -> General Attribute ->
Ganapati Adimulam@yahoo.co.in(Here we are printing some Mail and in Real time it should be from the KNA1/ from the Custom Table Credit Payments.**



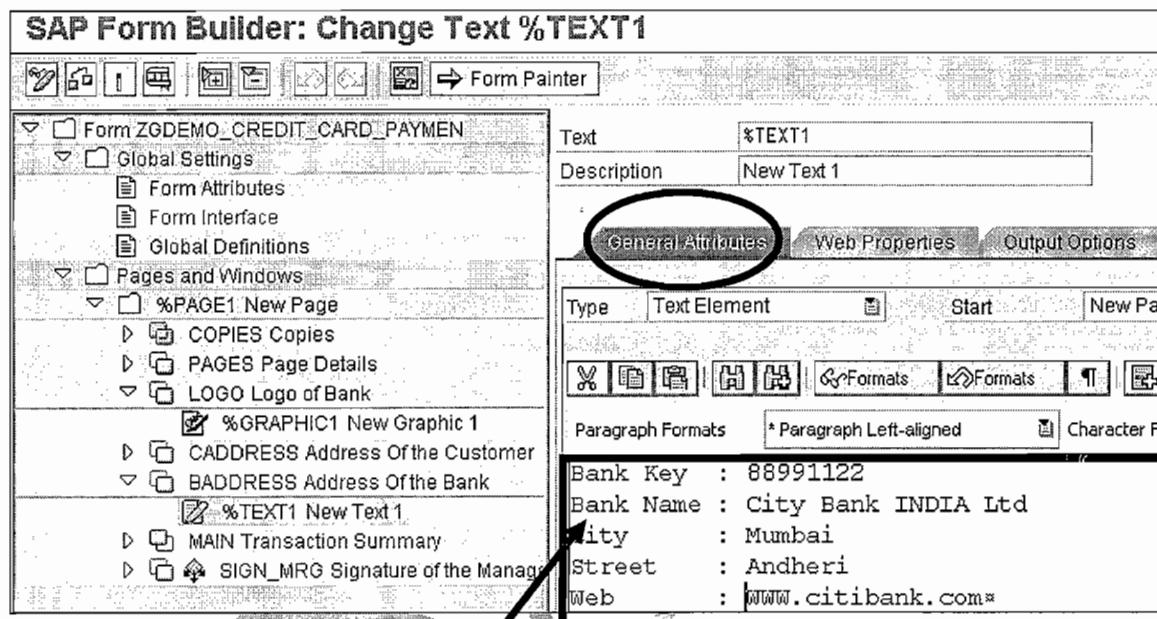
Window : BADDRESS**D**

Right Click On Window CADDRES -> Create -> Window

Window : BADDRESS

Description : Address of the Bank

Right Click On Window BADDRESS -> Create -> TEXT-> General Attributes



Here we Printing Some Fixed text as Bank ADDRESS and in real time, We get the Bank Details from Table BNKA.



F

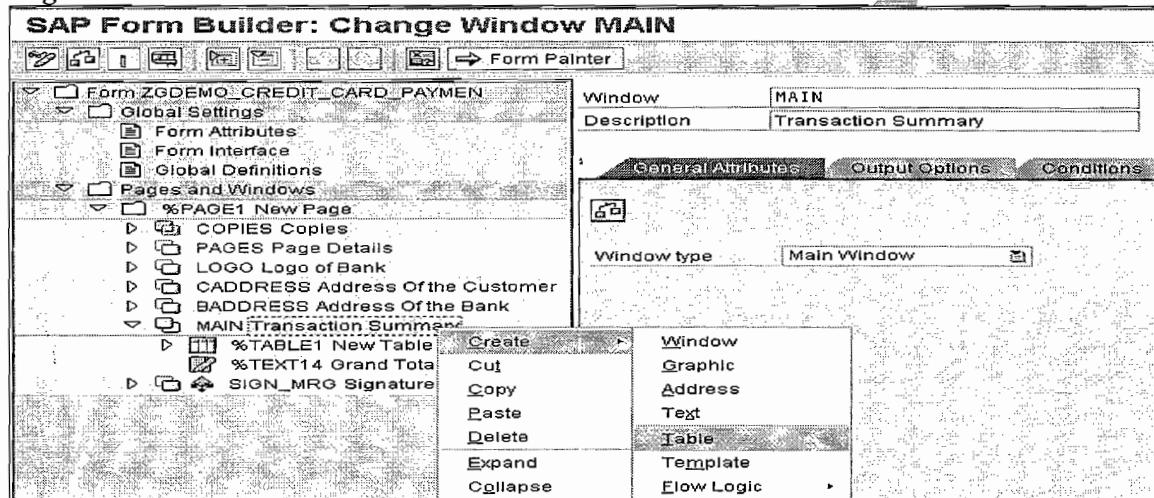
Window MAIN :

Right Click On Window BADDRESS -> Create -> Window

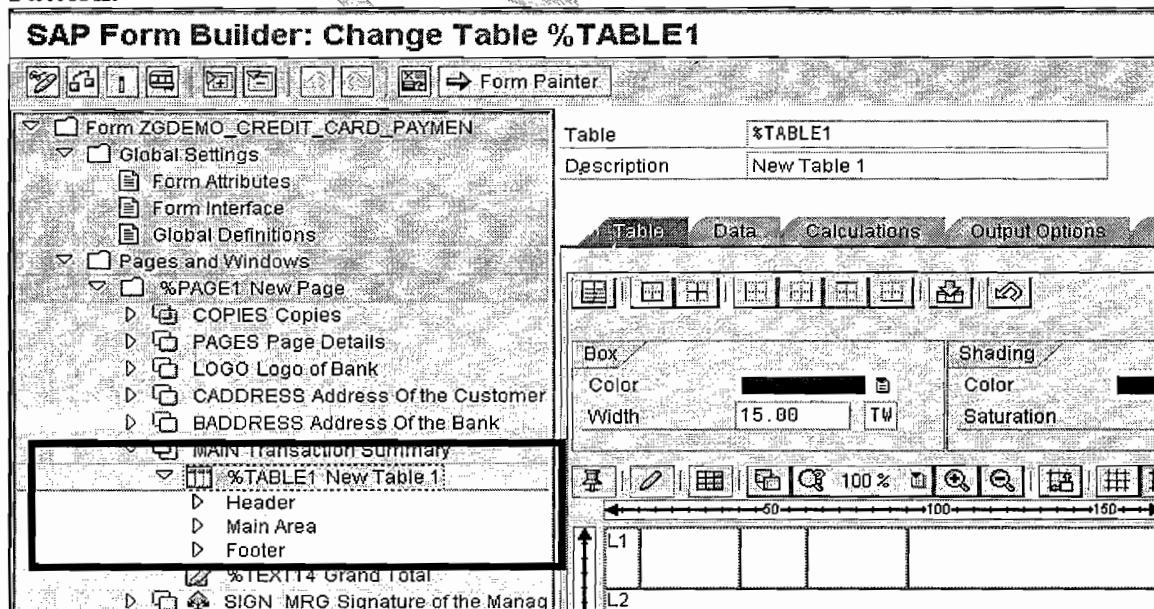
Window	MAIN
Description	Transaction Summary
Window Type	Main Window

Working with Table :

Right Click On MAIN Window -> Create -> Table

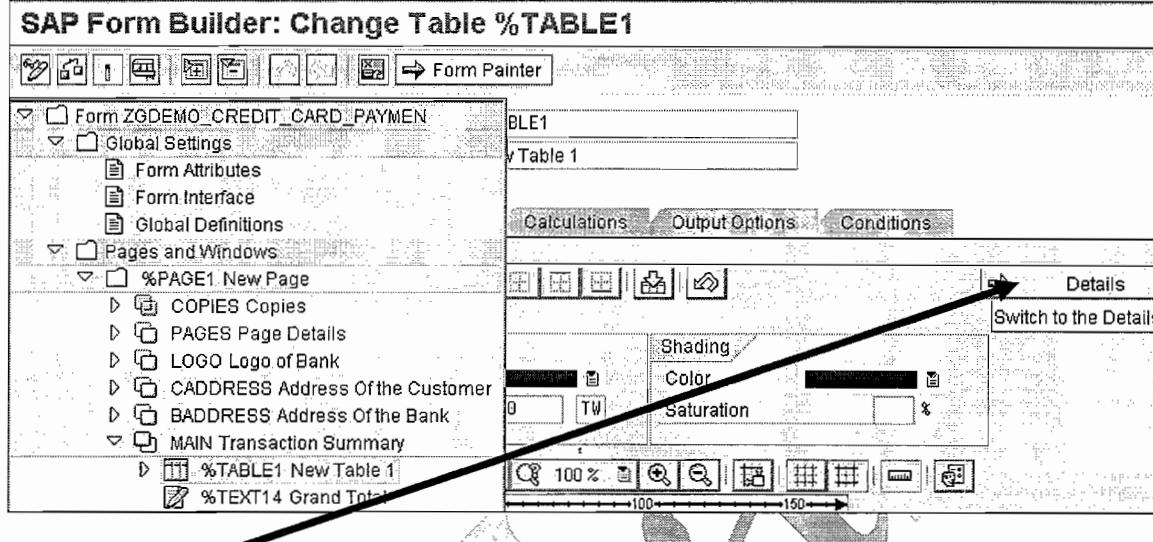


Select Pattern and select one of the framed Pattern , Which is similar to Template Pattern.

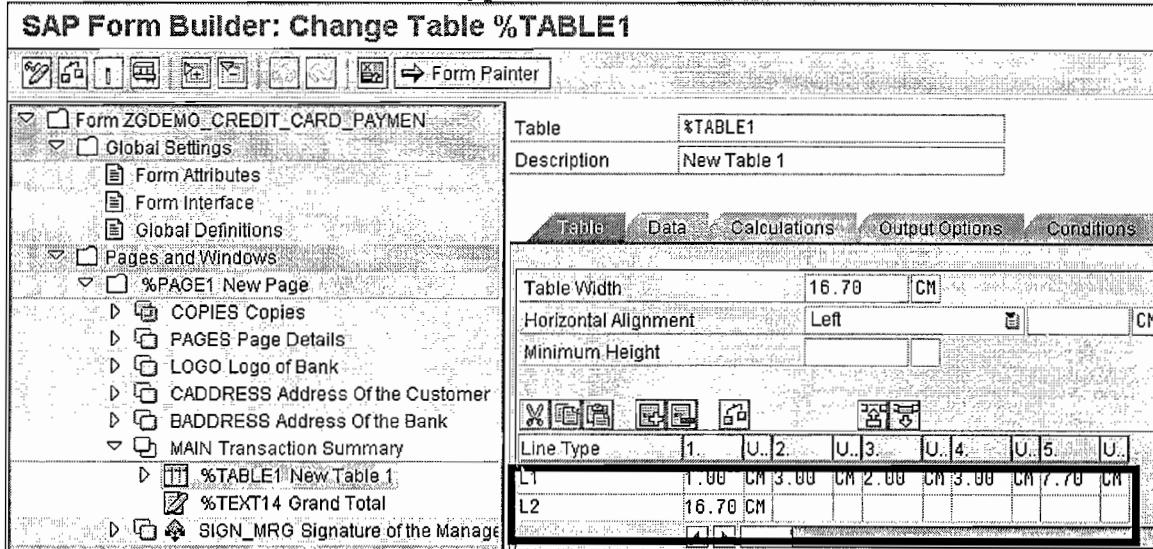


Notice that 3 Areas are Opened by Default: Header, main Area and Footer After Creating Table.

Note: Header and Footer is Called Only Once and MAIN Area is Called for each record of the Internal Table which is to be displayed in the Table.

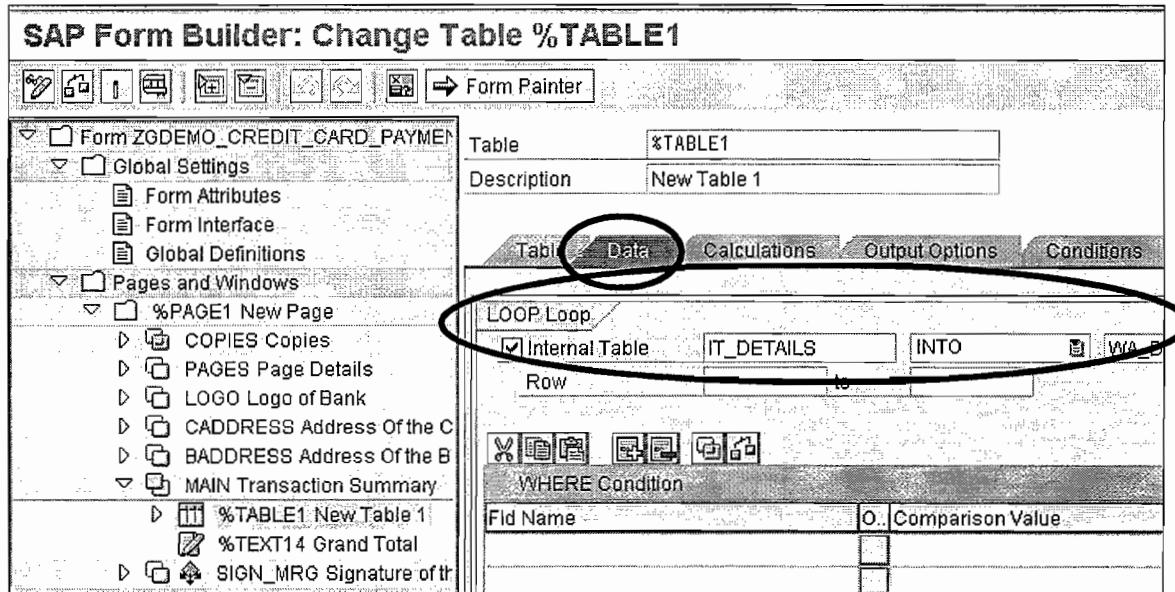


Click On Details to Define Line Types:

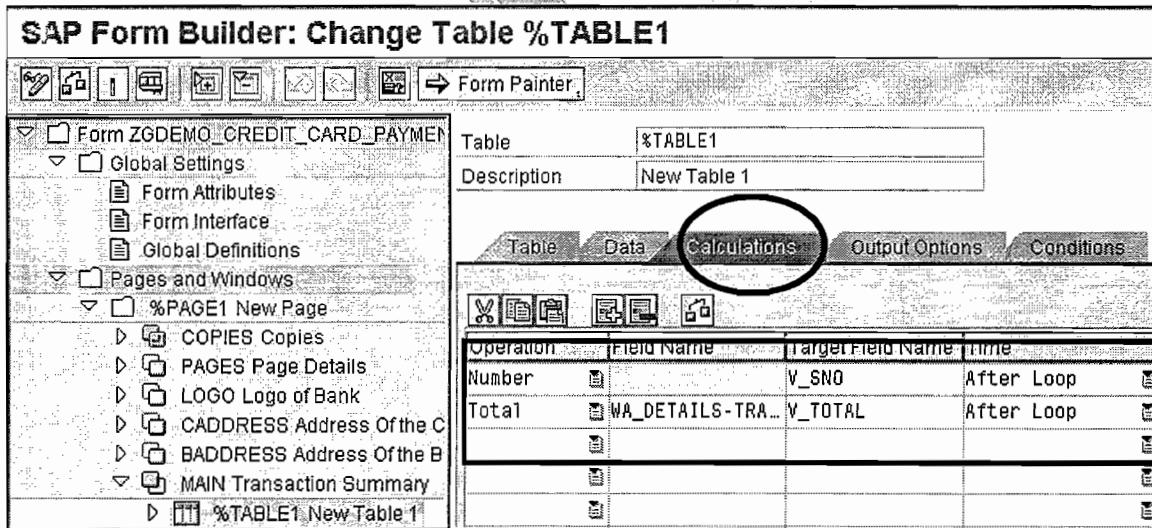


Line Type L1 with Cell1, Cell2, Cell3 and Cell4.

Line Type L2 with Only one Cell.

Click On Data

Provide the Internal Table to be looped , and the respective WorkArea.

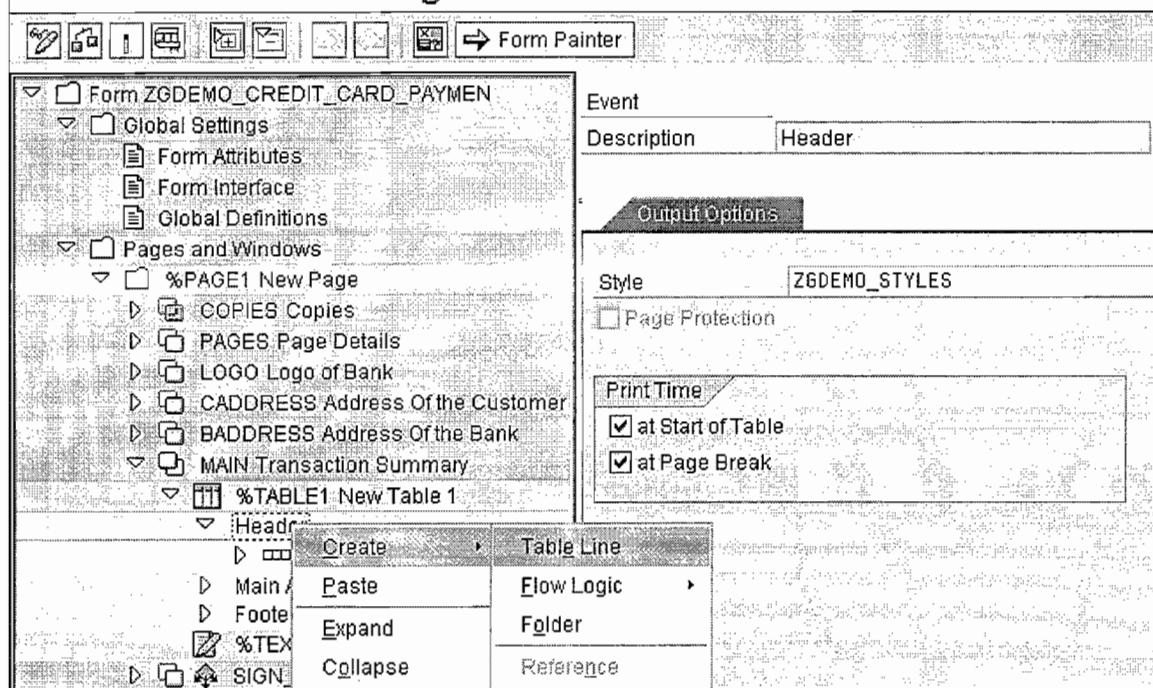
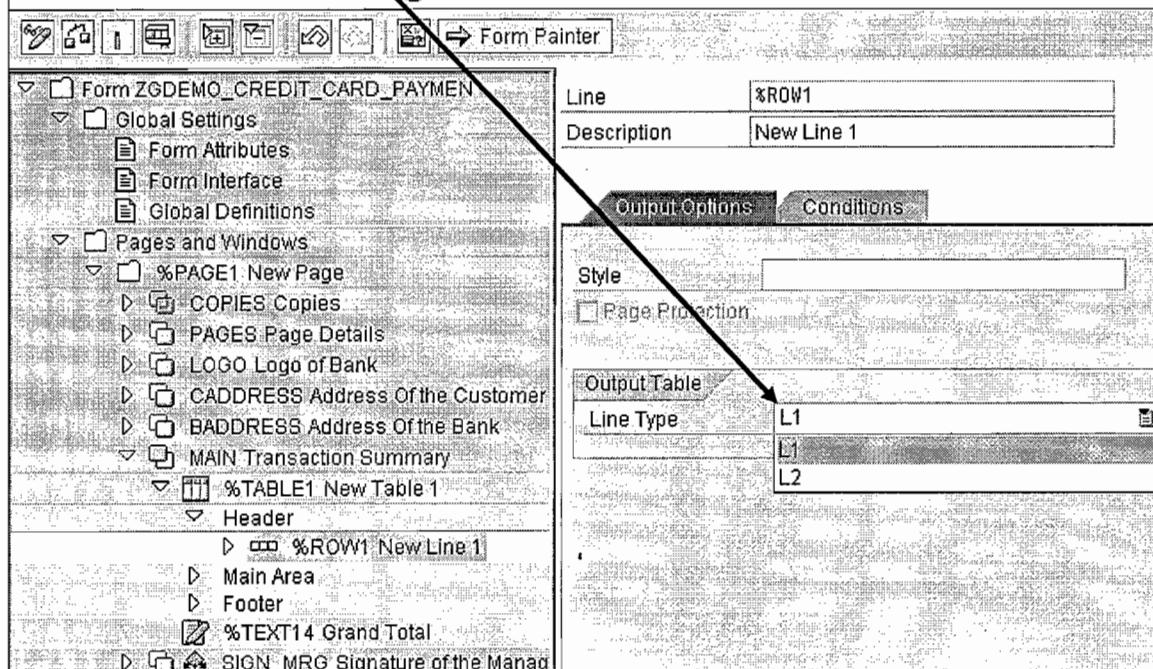
Click On Calculations Tab

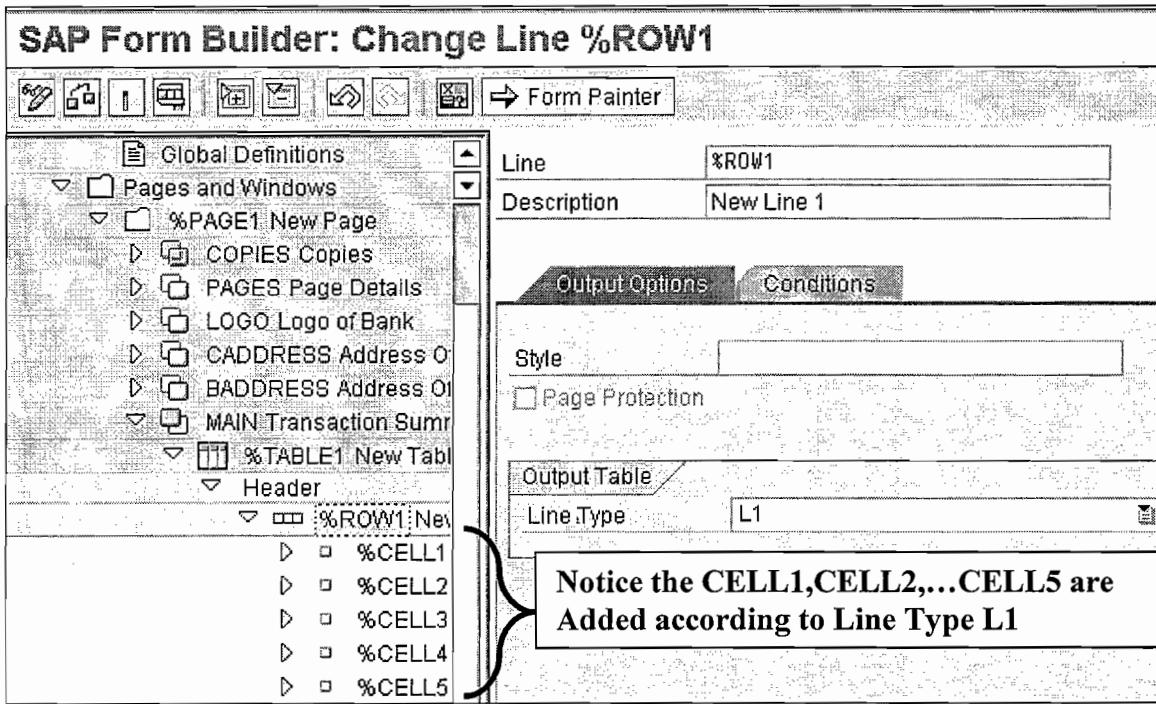
Select the Operation Number , Target Field Name V_SNO , Which is incremented by 1 for each record in the Internal Table , Initialize V_SNO with 1 , which was already done while declaring in the Global Data itself.

Select next Operation Total, to calculate the totals of all the records of internal table for the given field WA_DETAILS-TRANS_AMOUNT and the result into V_TOTAL.

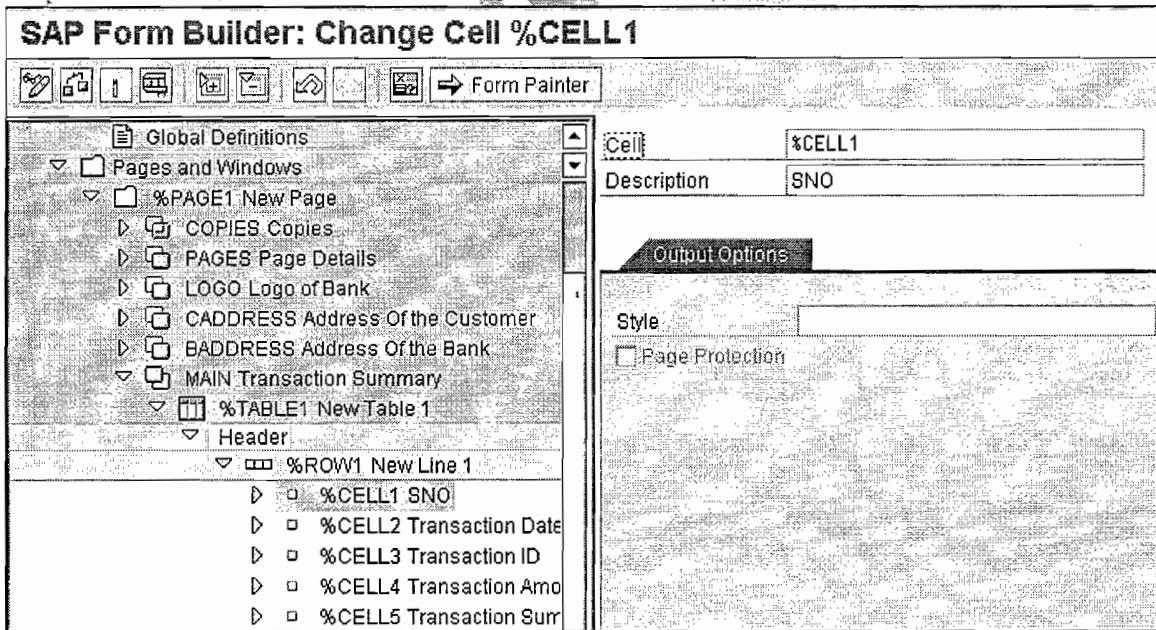
Printing Column Headings:

Right Click On Header -> Create -> Table Line

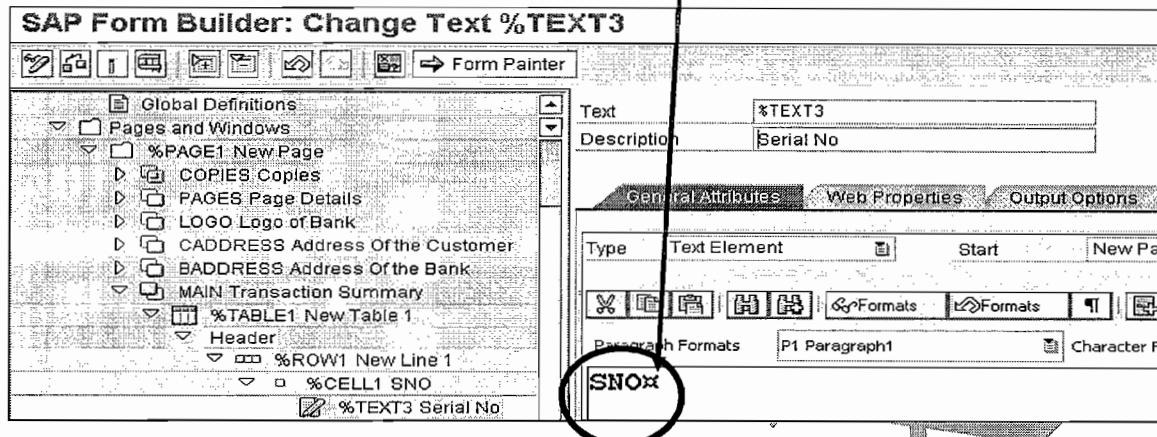
SAP Form Builder: Change Event %EVENT1**Provide the Line Type L1.****SAP Form Builder: Change Line %ROW1**



CELL1 : Double Click on Cell1 and Change the Description
CELL1 for SNO



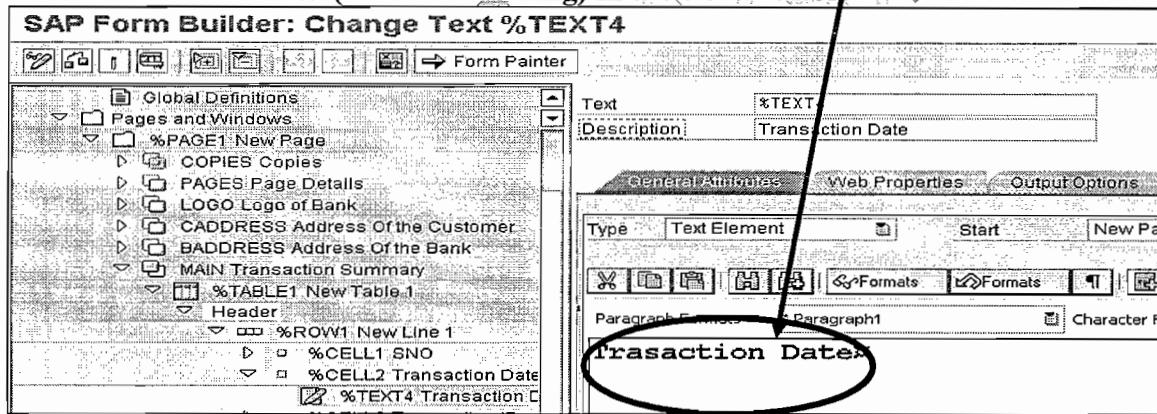
Right Click On CELL1 -> Create -> TEXT -> General Attributes-> Enter SNO in the Text Editor



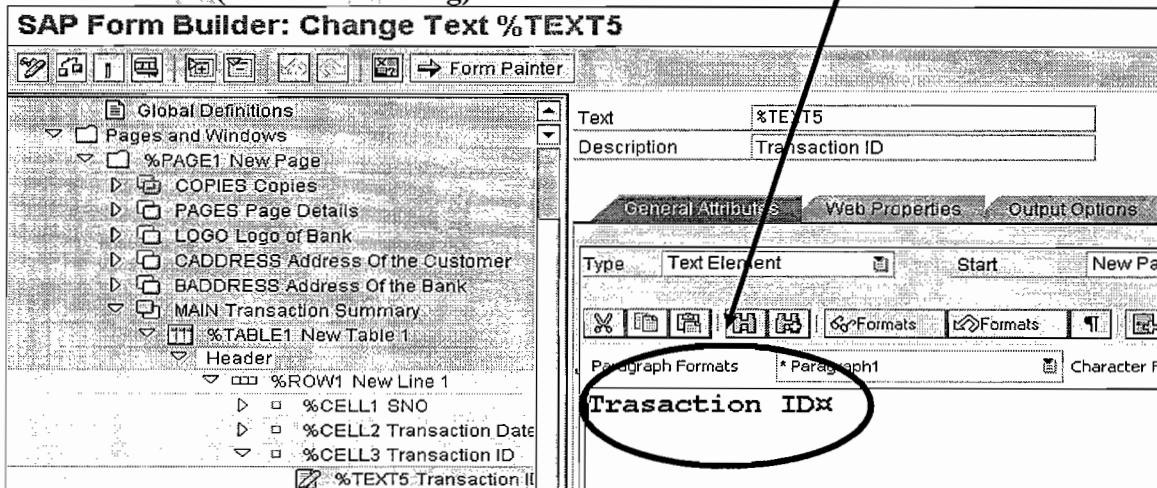
Note: Repeat the Same for all the CELLS with the respective TEXTs.

CELL2 : Right Click On CELL2 -> Create -> TEXT -> General Attributes

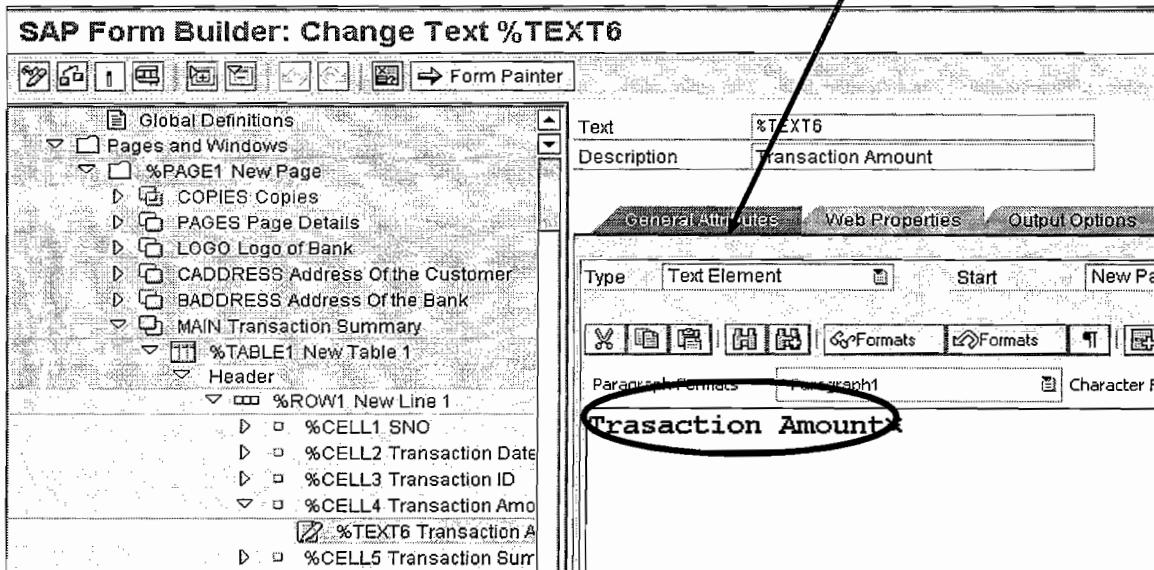
Enter Transaction Date(Column Heading) in the Text Editor.



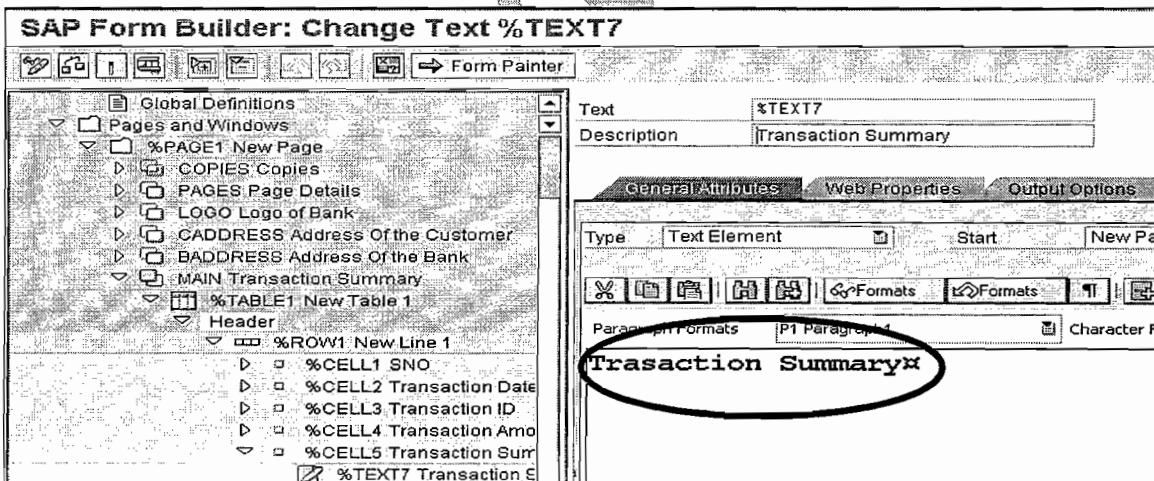
CELL3 : Right Click On CELL3 -> Create -> TEXT -> General Attributes-> Enter Transaction ID(Column Heading)



CELL4 : Right Click On CELL4 -> Create -> TEXT -> General Attributes->Transaction Amount(Column Heading)



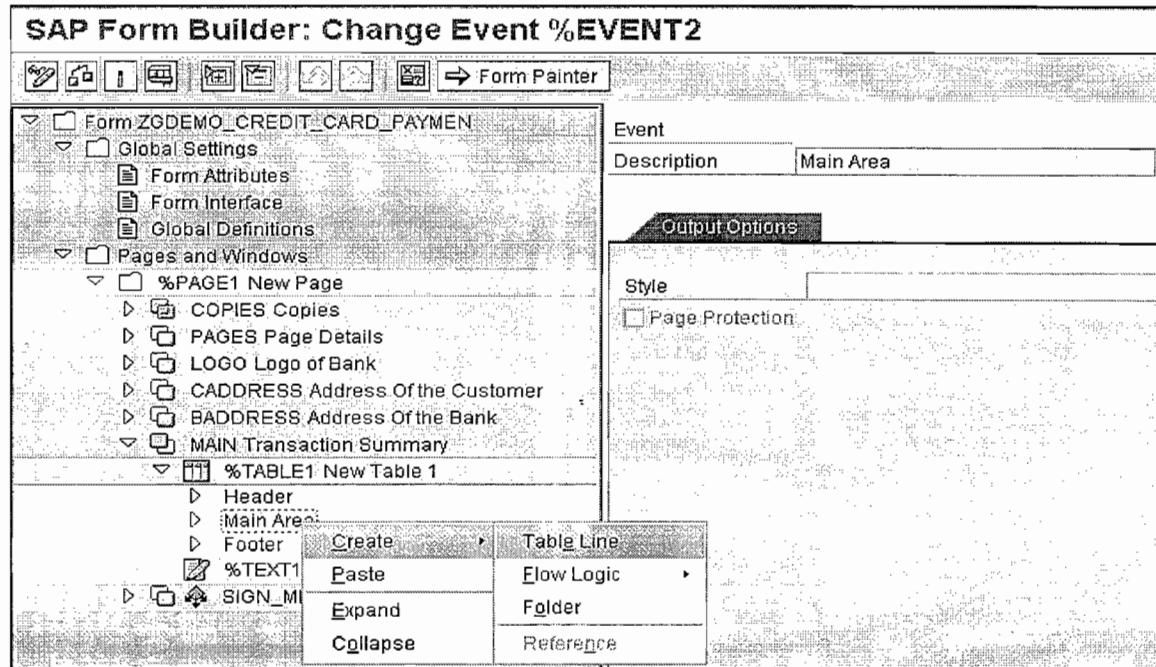
CELL5 : Right Click On CELL5-> Create -> TEXT -> General Attributes->Transaction Summary(Column Heading)



CELL	Description Provided
CELL1	SNO
CELL2	Transaction Date
CELL3	Transaction ID
CELL4	Transaction Amount
CELL5	Transaction Summary

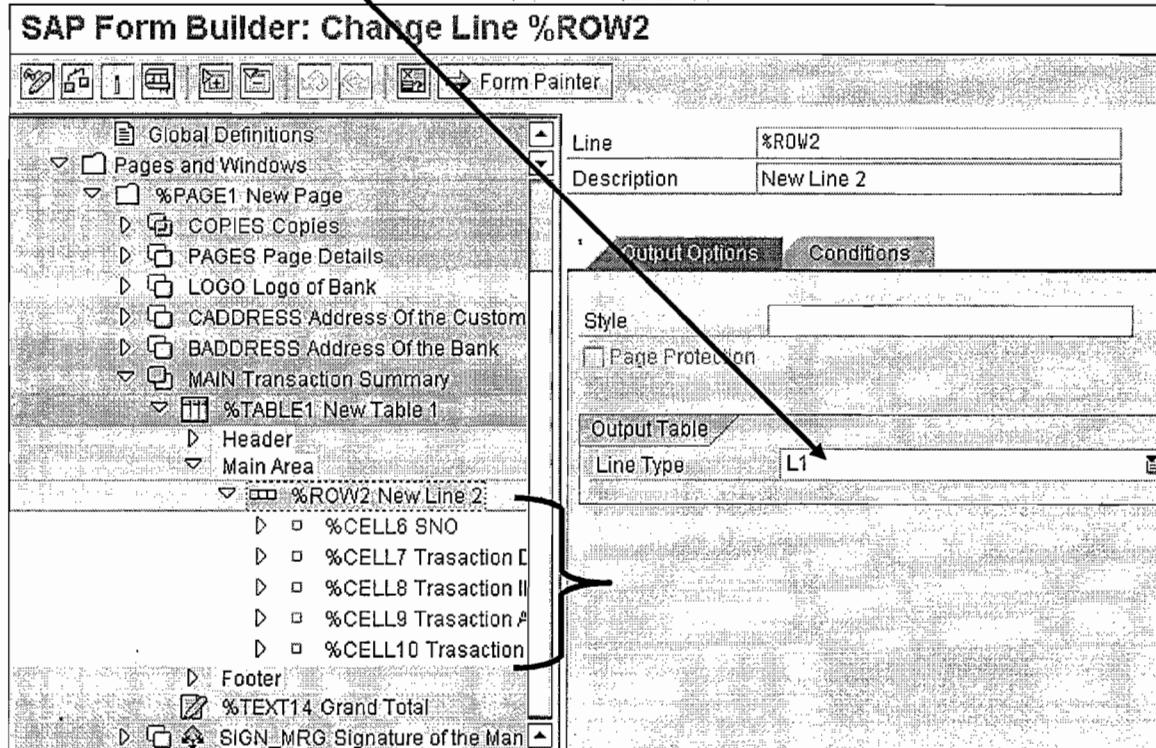
MAIN Area: is Used to Print the Data Records from the Internal Table, Record by Record.

Right Click Main Area -> Create -> Table Line



Provide the Line Type L1

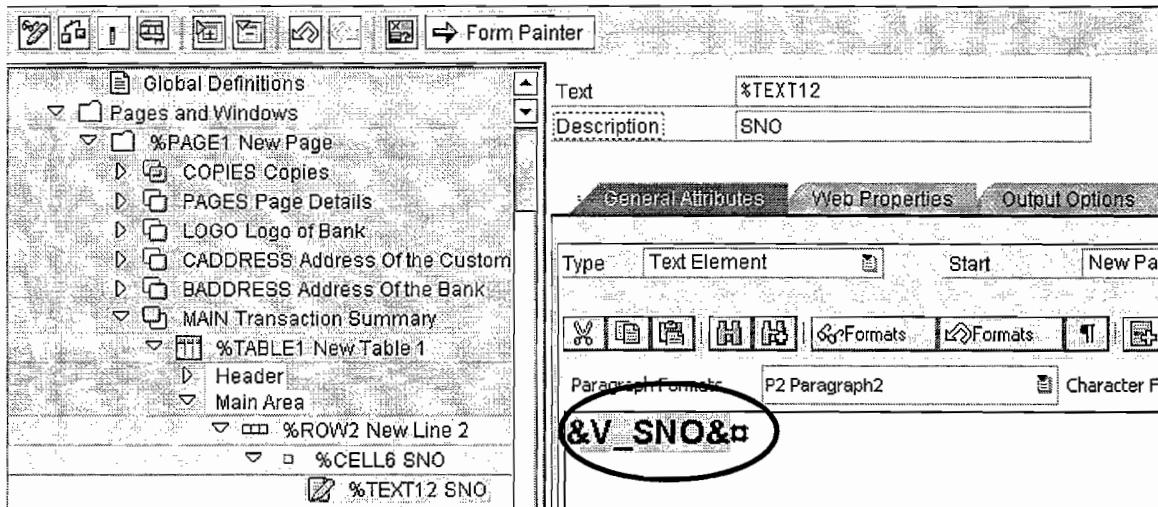
Notice the Cells



Notice that the CELLS CELL6 to CELL10 are added according to the Line type L1.

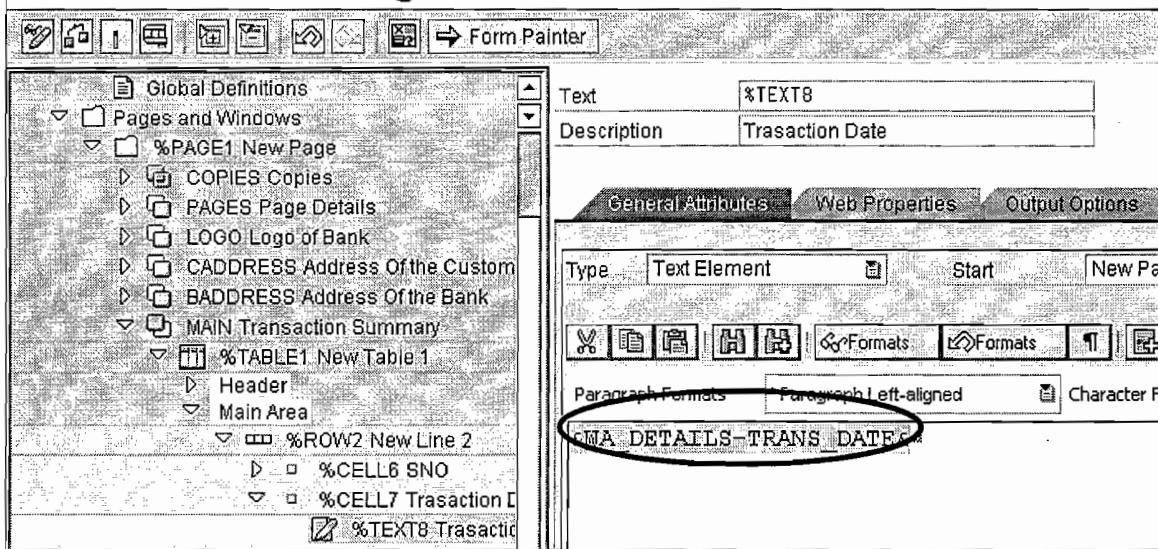
Printing Data in CELL6 : Right Click On CELL6 -> Create -> TEXT
 Click On General Attributes -> Text Editor and Enter &V_SNO& in the Text Editor

SAP Form Builder: Change Text %TEXT12

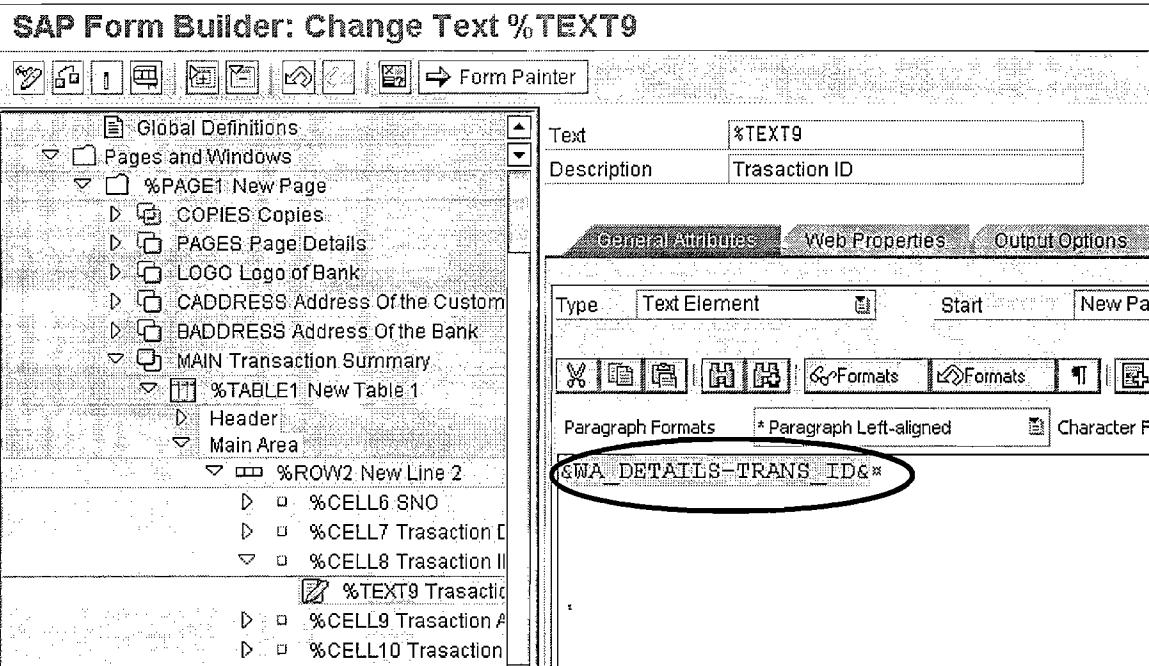


Printing Data in CELL7 : Right Click On CELL7 -> Create -> TEXT
 Click On General Attributes -> Text Editor and Enter &WA_DETAILS-TRANS_DATE&

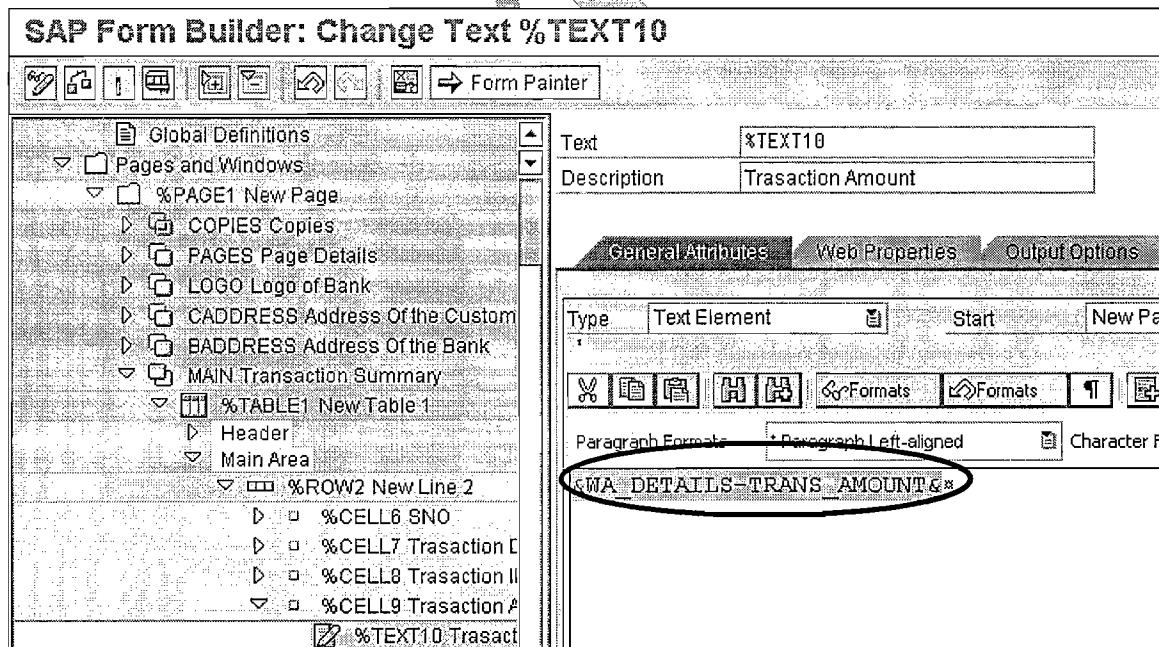
SAP Form Builder: Change Text %TEXT8



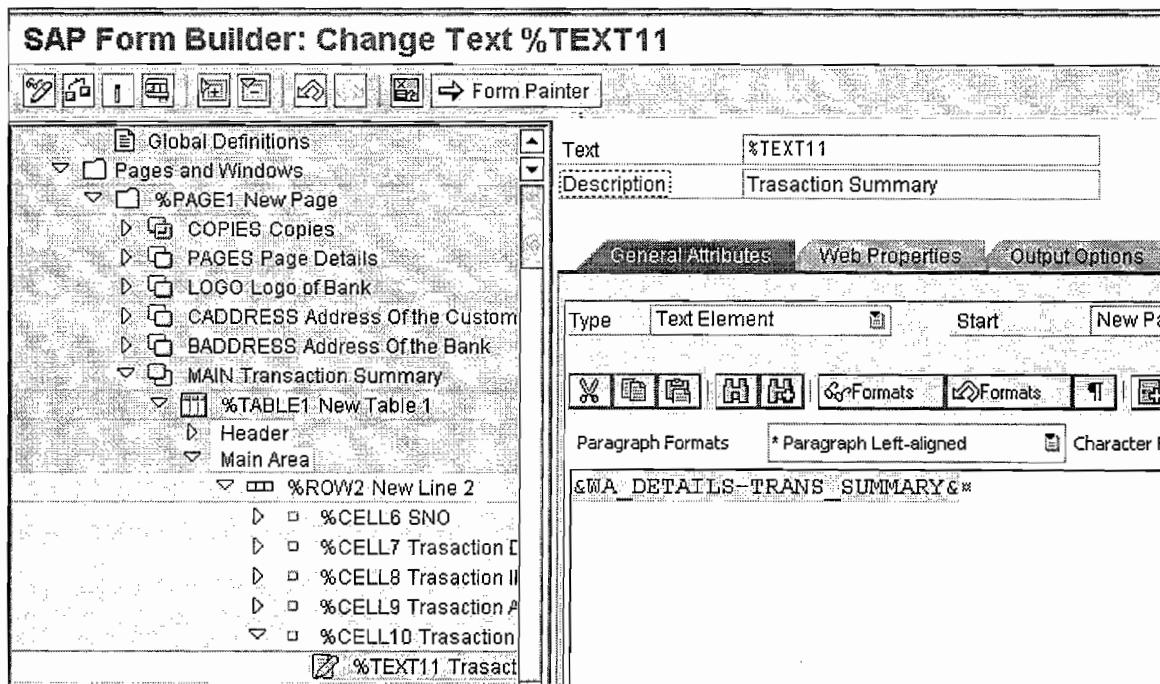
Printing Data in CELL8: Right Click On CELL8 -> Create -> TEXT
 Click On General Attributes -> Text Editor and Enter &WA_DETAILS-TRANS_TRANS_ID&



Printing Data in CELL9: Right Click On CELL9 -> Create -> TEXT
 Click On General Attributes -> Text Editor and Enter &WA_DETAILS-
 TRANS_TRANS_AMOUNT&

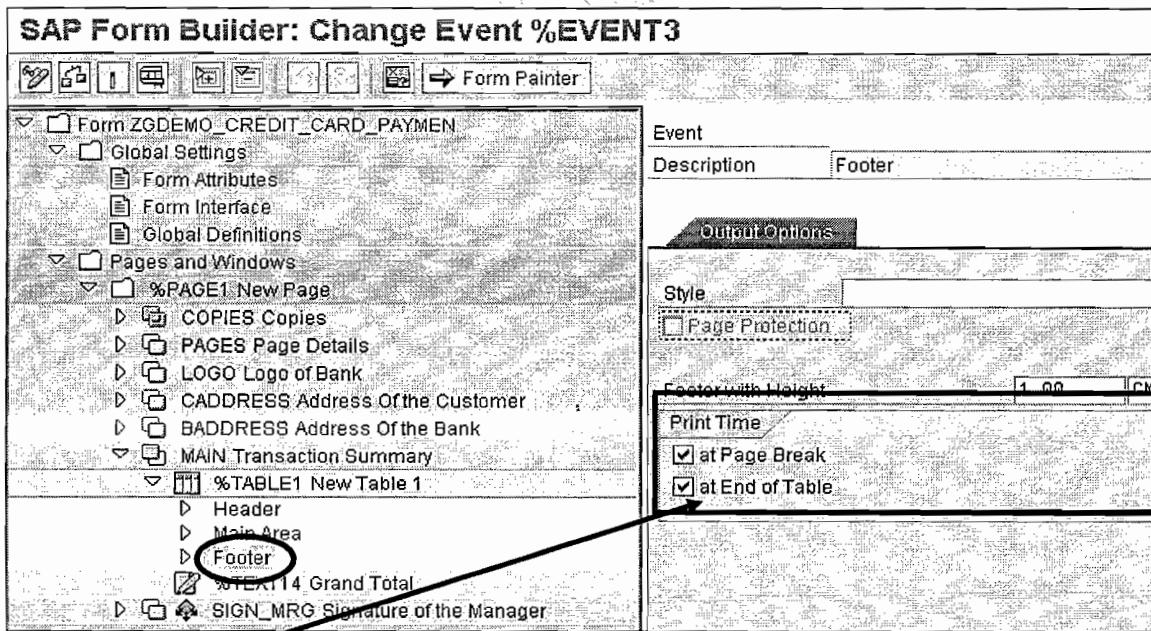


Printing Data in CELL10: Right Click On CELL10 -> Create -> TEXT
 Click On General Attributes -> Text Editor and Enter &WA_DETAILS-
 TRANS_SUMMARY&

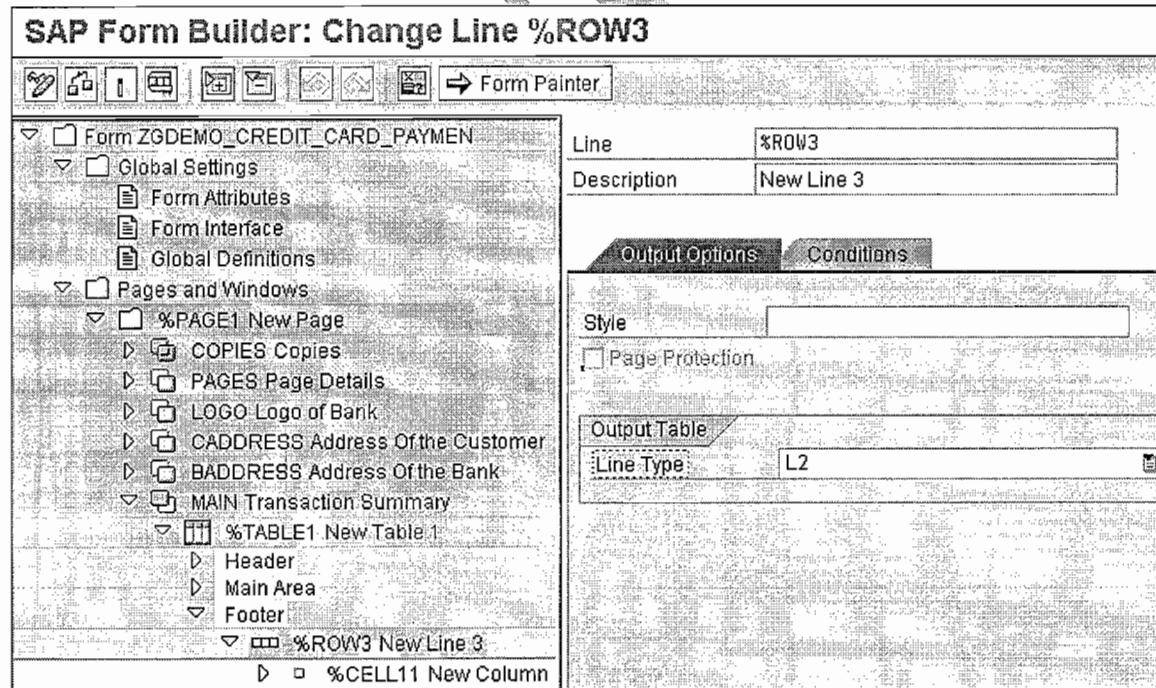
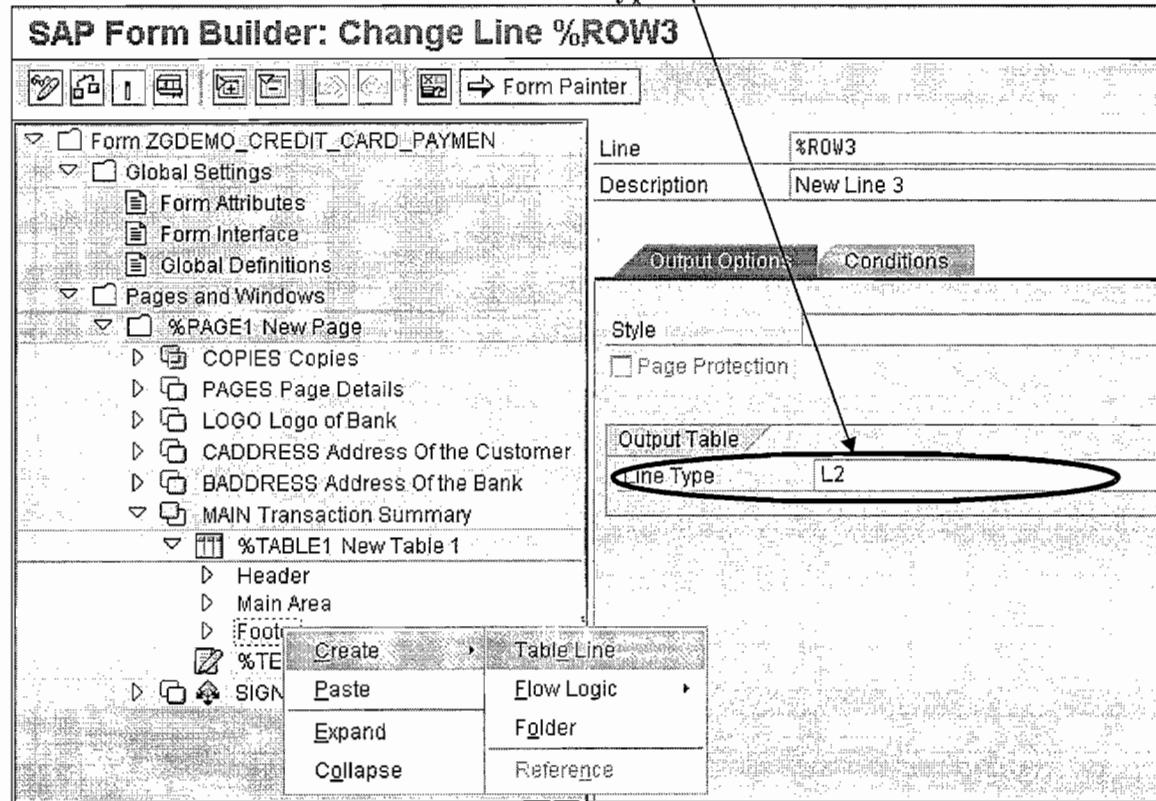


Printing Data in FOOTER:

Note: We Use FOOTER to Print the Totals, Which is Calculated through the Calculations TAB Of Table.

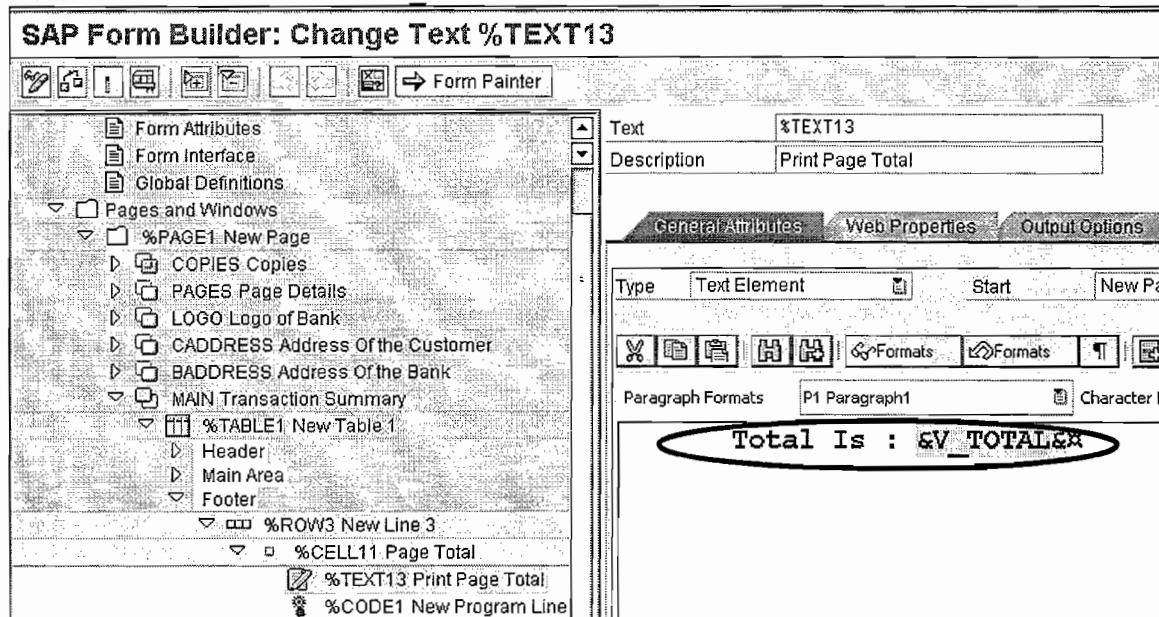


Notice the Print time, Which is Printed at the end of each Page(at Page Break) and at End of Table also.

FOOTER -> Create -> Table Line -> Line Type L2

Notice that CELL 11 is Added because Only One CELL is available in Line Type L2.

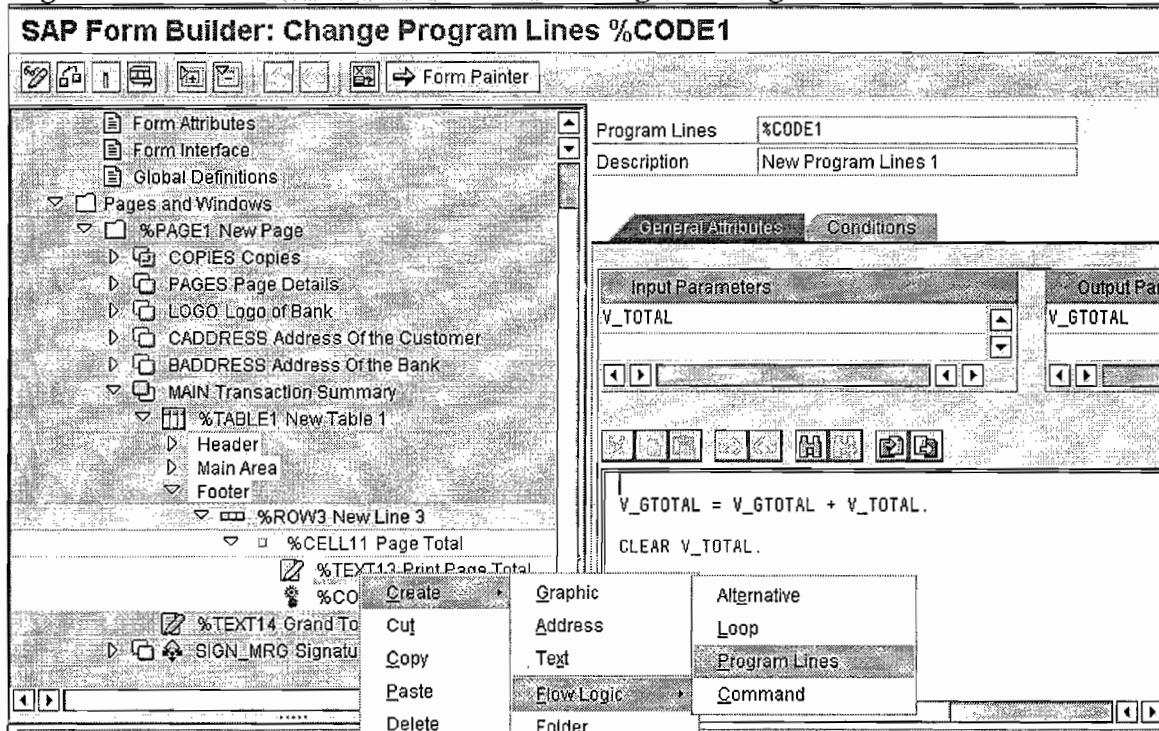
Right Click On CELL 11 -> Create -> TEXT -> General Attributes -> Open Text Editor -> Enter Total is : &V_TOTAL&

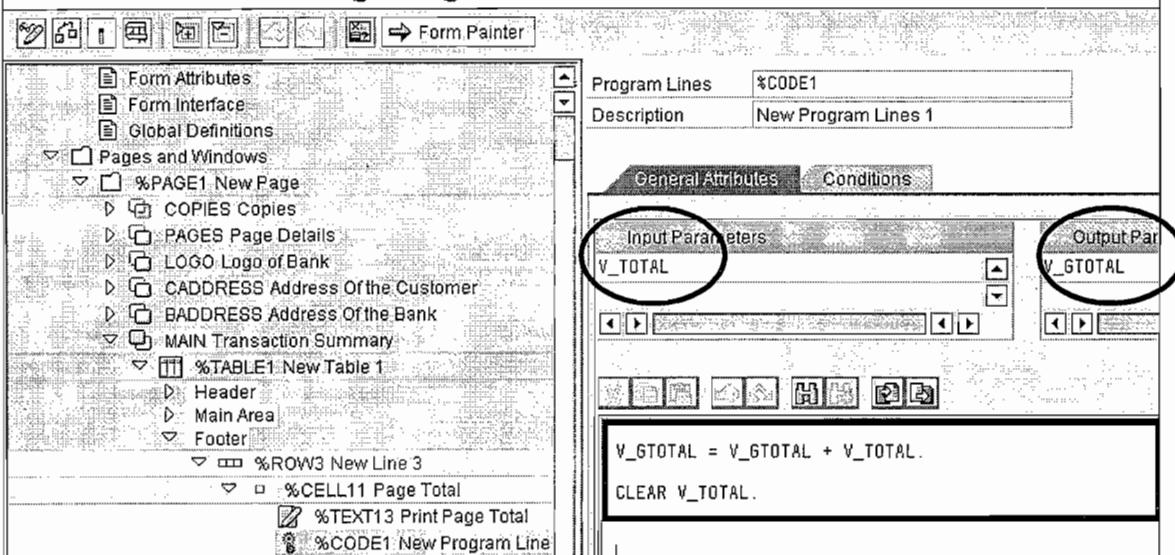


Since We Need to Print the Totals at the END of each Page, and the V_TOTAL gives the Cumulative SUM so that we Need to print it at for each page and Calculate the Grand Total and Print the Grand Total Only Once i.e. after Completing the MAIN Window (in Last Page).

To Calculate Grand Totals

Right Click On the Text -> Create -> Flow Logic -> Program Lines



SAP Form Builder: Change Program Lines %CODE1

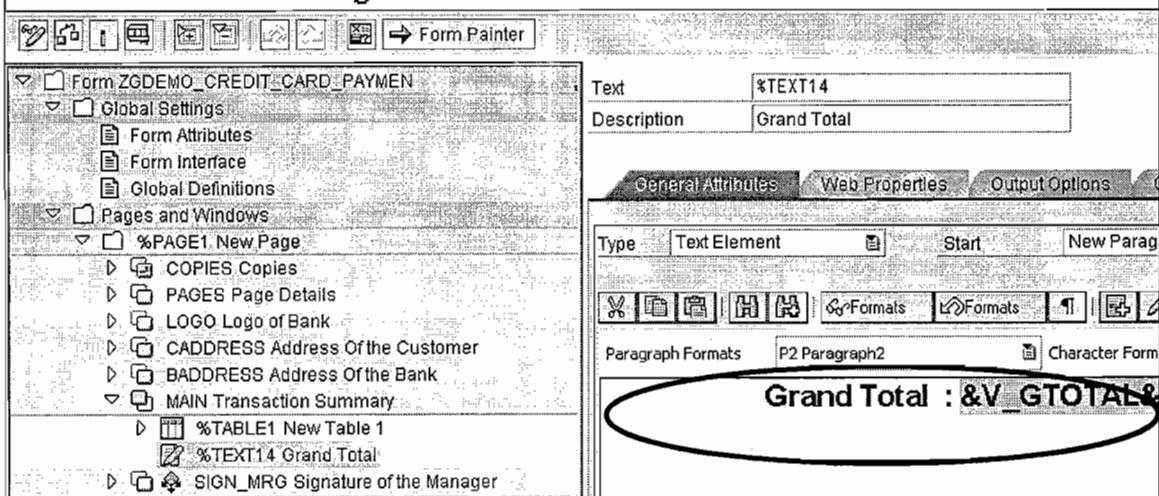
Enter the Input and Output Parameters and the V_TOTAL to V_GTOTAL and Clear V_TOTAL(Page Total).

Note : This Flow Logic is Executed for each output page and this (Grand Total) should be Printed at the END Of Table.

i.e ADD the Total of each page and Print it at the END of MAIN Windows.

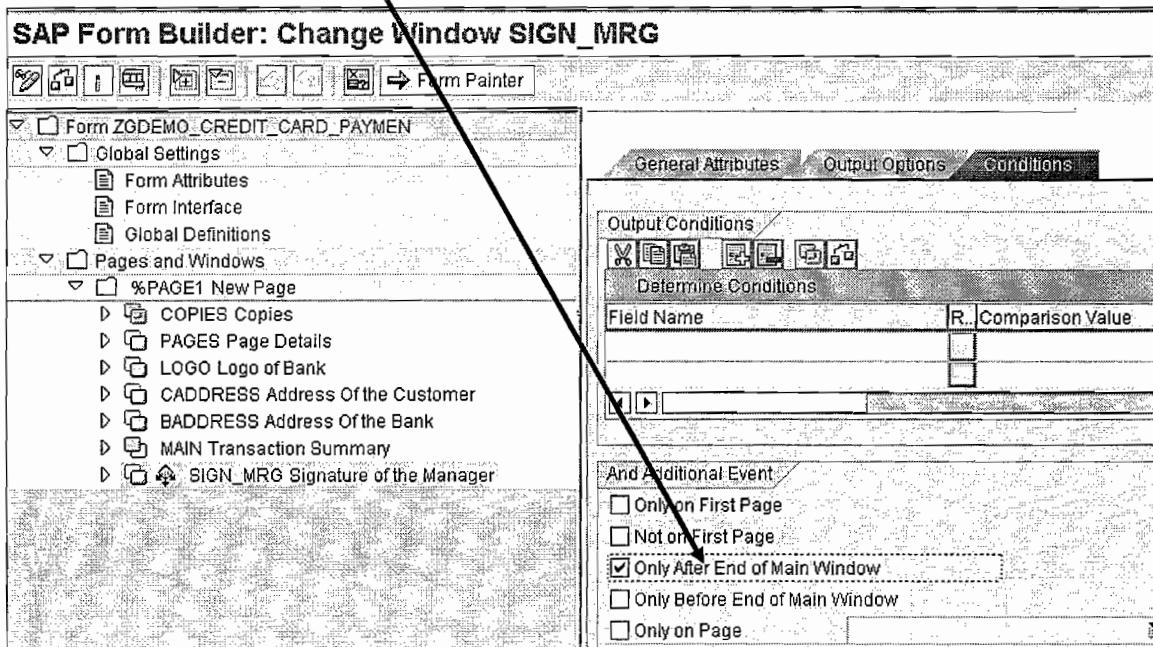
Right Click On TABLE -> Create -> TEXT (Element) -> General Attributes -> Open Text Editor and Enter
Grand Total : &V_GTOTAL&

Note : Notice that V_GTOTAL is Printed through text Element TEXT14 and which is Added after TABLE i.e after Printing all the data from the TABLE.

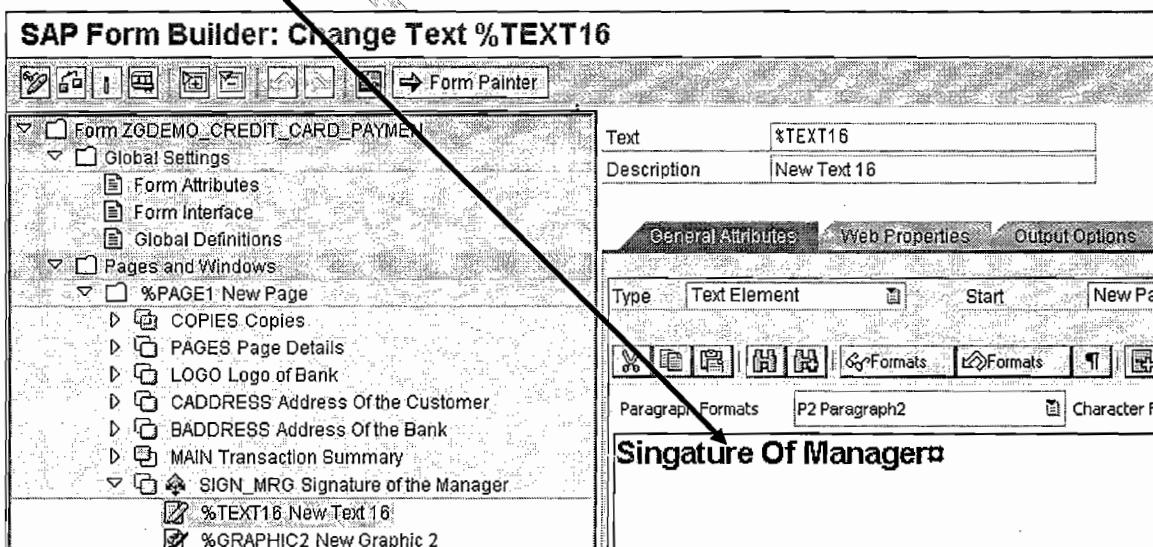
SAP Form Builder: Change Text %TEXT14

Window SIGN_MGR : Should be Printed Only in the last page(After Printing the Data from MAIN Window Completely.

Notice the Output Options, Select Only After End of Main Window, to make sure that the Window is Printed at the END of MAIN Window.



Right Click On Window HR_SIGN -> Create Text -> General Attributes -> Enter text : Signature Of Manager.



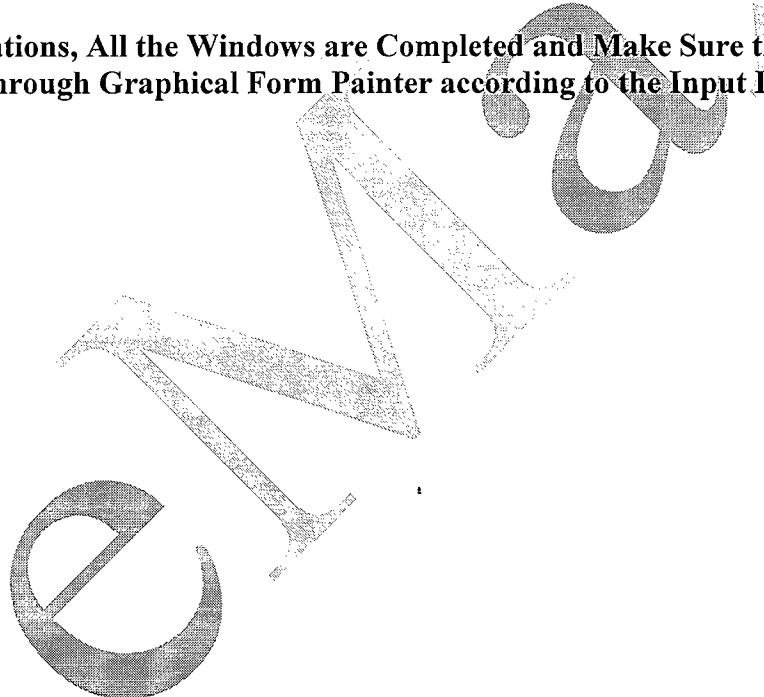
Right Click on TEXT16 -> Create GRAPHIC -> General Attributes

Enter the GRAPHIC Details

SAP Form Builder: Change Graphic %GRAPHIC2

Graphic	%GRAPHIC2
Description	New Graphic 2
General Attributes Output Options Conditions	
Name	SIGN_MGR
Object	GRAPHICS
ID	BMAP
<input type="radio"/> Black and White Bitmap Image (BMON) <input checked="" type="radio"/> Color Bitmap Image (BCOL) <input type="radio"/> Determine Dynamically (BMON, BCOL)	
Technical Attributes	

Congratulations, All the Windows are Completed and Make Sure that the Layout is Adjusted through Graphical Form Painter according to the Input Layout Provided.



Driver Program :

Note : Make Sure that the Driver Program can transfer all the Credit Card Expenses through the Internal Table and Customer Details through WA_KNA1.

REPORT ZGDEMO_CREDIT_CARD_PAYMENTS.

*BLOCK B1

SELECTION-SCREEN BEGIN OF BLOCK B1 WITH FRAME
TITLE TEXT-000.

PARAMETER P_KUNNR TYPE KUNNR.

SELECTION-SCREEN END OF BLOCK B1.

DATA : WA_KNA1 TYPE KNA1,
IT_DETAILS TYPE TABLE OF ZCREDIT_PAYMENTS,
v_fm type RS38L_FNAM.

* START-OF-SELECTION. *

START-OF-SELECTION.

SELECT SINGLE * INTO WA_KNA1 FROM KNA1
WHERE KUNNR = P_KUNNR.

SELECT * INTO TABLE IT_DETAILS FROM ZCREDIT_PAYMENTS
WHERE KUNNR = P_KUNNR.

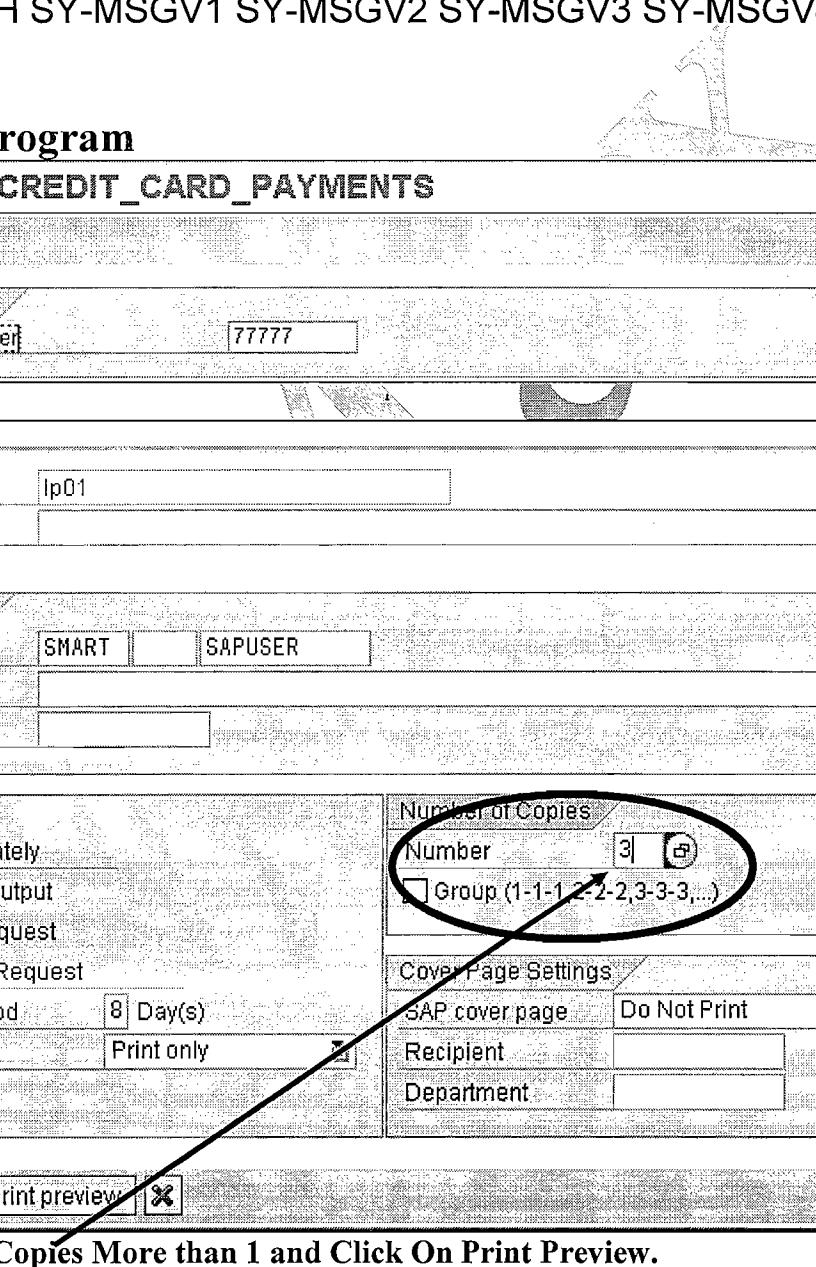
call function 'SSF_FUNCTION_MODULE_NAME'
exporting
formname
= 'ZGDEMO_CREDIT_CARD_PAYMENTS'
importing
fm_name
= v_fm

*CALL THE SMARTFORM FM DYNAMICALLY
 CALL FUNCTION V_FM
 EXPORTING
 wa_kna1 = WA_KNA1
 tables
 it_details = IT_DETAILS

IF sy-subrc <> 0.

* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
 * WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
 ENDIF.

Execute Program



ZGDEMO_CREDIT_CARD_PAYMENTS

Customer Details
Customer Number: 77777

Execute

Print:
OutputDevice: lp01
Page selection:

Spool Request
Name: SMART SAPUSER
Title:
Authorization:

Spool Control
 Print immediately
 Delete After Output
 New spool request
 Close Spool Request
Spool retention pd: 8 Day(s)
Storage Mode: Print only

Number of Copies
Number: 3
 Group (1-1-1, 2-2-2, 3-3-3,...)

Cover Page Settings
SAP cover page: Do Not Print
Recipient: Department:

Print | Print preview |

Enter No Of Copies More than 1 and Click On Print Preview.

Notice that Text ORIGINAL is Printed for COPY1.

Print Preview of LP01 Page 00001 of 00002 **Copy 001 of 003**

Archive Print and Archive | Back Forward | Print | Preview | Exit | Help | Options |

ORIGINAL

citibank	
-----------------	--

PAGE 1 Of 2

Card NO : 018301004245	
Customer No	77777
Customer Name	Adimulam
Street/Hno	Sanath Nagar
City	Sanath Nagar
ganapati.adimulam@yahoo.com	

Bank Key : 88991122	
Bank Name : City Bank INDIA Ltd	
City	Mumbai
Street	Andheri
Web	WWW.citibank.com

SNO	Trasaction Date	Trasact ion ID	Trasaction Amount	Trasaction Summary
1	07.01.2008	T1	1,000.00	SHOP @ RS BROS
2	07.01.2008	T2	3,000.00	SHOP @ CHERMAS
3	07.01.2008	T3	200.00	PETROL @ INDIAN OIL, ERRA GADDA
4	10.01.2008	T4	200.00	PETROL @ INDIAN OIL, ERRA GADDA
5	11.01.2008	T5	800.00	SWEETS @ ABHIRUCHI , AMEERPET
6	12.01.2008	T6	5,000.00	SAREES @ KALA NIKETAN , AMEERPET
Total Is : 10200.00				

Print Preview of LP01 Page 00001 of 00002 (Copy 001 of 003)

Archive Print and Archive | Back Forward | Print | Preview | Exit | Help | Options |

3	07.01.2008	T3	200.00	PETROL @ INDIAN OIL, ERRA GADDA
4	10.01.2008	T4	200.00	PETROL @ INDIAN OIL, ERRA GADDA
5	11.01.2008	T5	800.00	SWEETS @ ABHIRUCHI , AMEERPET
6	12.01.2008	T6	5,000.00	SAREES @ KALA NIKETAN , AMEERPET
Total Is : 10200.00				

Page Totals

Print Preview of LP01 Page 00002 of 00002 (Copy 001 of 003)

**ORIGINAL****PAGE 2 Of 2**

Card NO :	018301004245
Customer No	77777
Customer Name	Adimulam
Street/Hno	Sanath Nagar
City	Sanath Nagar
ganapati.adimulam@yahoo.com	

Bank Key : 88991122
 Bank Name : City Bank INDIA Ltd
 City : Mumbai
 Street : Andheri
 Web : www.citibank.com

7	13.01.2008	T7	1,000.00	BAGS @ PALEM LEATHERS , PANJA GUTTA
8	13.01.2008	T8	1,000.00	TICKETS @ KESINENI TRAVELS -

Print Preview of LP01 Page 00002 of 00002 (Copy 001 of 003)

Total Rs : 2 000.00

Grand Total : 12200.00**Signature Of Manager**

Notice that Signature Window is Printed Only in Last Page.

Click On Next , Displays next Copy as Copy1 is Completed.

COPY2 : Notice that the Copies Window , Prints COPY and in First Copy(Original) , Text ORIGINAL is Printed.

Print Preview of LP01 Page 00001 of 00002 **Copy 002 of 003**

Archive Print and Archive Back Forward

COPY

PAGE 1 OF 2

Card NO : 018301004245	
Customer No	77777
Customer Name	Adimulam
Street/Hno	Sanath Nagar
City	Sanath Nagar
ganapati adimulam@yahoo.com	

Bank Key : 88991122
Bank Name : City Bank INDIA Ltd
City : Mumbai
Street : Andheri
Web : www.citibank.com

SNO	Trasaction Date	Trasact ion ID	Trasaction Amount	Trasaction Summary
1	07.01.2008	T1	1,000.00	SHOP @ RS BROS
2	07.01.2008	T2	3,000.00	SHOP @ CHERMAS



