

SAP Smart Form Overview Brief

Additional Smart Form Information Sources	2
What Skills Do I Need to Work with SAP Smart Forms?	2
What are Smart Forms?	3
What About SAPscript?	3
SAP Smart Form Process Overview	3
Programming Flow.....	4
Smart Form Architecture Diagram	4
Smart Form Components	5
SAP Smart Form Builder	5
Smart Form Builder Components	6
Specifics about Smart Form Development	7
Important Objects for Form Development	7
Graphics.....	8
Templates and Tables	8
Procedures to Familiarize Yourself with Smart Forms	9
Getting Started: Creating a Form	9
Working with Nodes in a Form	11
Testing Your Form	12
Appendix.....	12

Purpose

The purpose of this document is to provide a high-level overview of the SAP Smart Form print solution, its components and architecture, and some short cuts. This document is **not** a substitute for the SAP Online Documentation available on CD or through SAP's help portal, <http://help.sap.com>.

Additional Smart Form Information Sources

- **Smart Form development team web site on the Service Marketplace:**

<http://service.sap.com/smartforms>

- **SAP Online Documentation**

- **CDs:** SAP Online Documentation CD's should come with your installation CDs. Additional copies may be ordered from the Service Marketplace.

- **SAP Help Portal:** The content is also accessible through SAP's Help Portal at <http://help.sap.com>. Instructions to reach the SAP Help Portal are:

In your web browser, navigate to <http://help.sap.com>.

Call the online documentation **for your release** using the left navigation frame and open the SAP Library. The SAP Smart Forms online documentation can be found by choosing:

FOR SAP R/3 4.6C:

Basis components → *Basis Services / Communication Interfaces (BC-SRV)* → *SAP Smart Forms (BC-SRV-SSF)*

FOR SAP Web Application Server 6.10:

mySAP Technology Components → *SAP Web Application Server* → *Basis Services / Communication Components (BC-SRV)* → *SAP Smart Forms (BC-SRV-SSF)*

SAP Training Course BC 470

For SAP Smart Forms training, SAP Educational Services offers BC470, a two-day course covering Smart Forms extensively.

To find out where the class is being offered closest to you, visit the online catalog in SAP Service Marketplace at <http://service.sap.com/education>.

What Skills Do I Need to Work with SAP Smart Forms?

SAP Smart Forms are significantly easier to use than SAPscript. Potentially, there may be no need for a user to have programming skills. It is very unlikely that the user will need to know SAPscript.

However, the skills required to work with Smart Forms vary depending on the role the user has, and the complexity of a customer's requirements. Required skills depend on whether the user's work comprises the following:



An explanation of the following terms will be explained in the Smart Forms architecture topic of this document

- Modifying an existing Smart Form print form template

In most cases where a user's work involves modifications to an existing Smart Form print form template, no programming skills are needed.

- Modifying an existing Smart Form print program

In cases where modifications to a print program are required, knowledge of the ABAP development workbench is recommended. However, we recommend that modifications that in the past were done in the Print Program be done in the Smart Form print form instead. In Smart Forms, you can modify the print program by using programming nodes that can be inserted into the form (this is explained in the SAP Online Documentation).

- Creating a new Smart Form print form where none exists

The creation of a new Smart Form print form where no template exists may not require ABAP programming skills if there is already a print program that retrieves all necessary fields from the application. However some ABAP programming may be necessary.

- Creating a new Smart Form print program where none exists

Creation of a new Smart Form print program where none exists will require ABAP skills and functional knowledge.

What are Smart Forms?

SAP's new print form solution, SAP Smart Forms, is the successor of SAPscript. All new development of forms at SAP will be done using the Smart Form print solution. However, SAPscript will still be supported for those forms already created in SAPscript, and for customers who have created their own forms in SAPscript.

SAP Smart Forms is a tool for the maintenance of forms on which application data from the SAP system can be output. Depending on the application, the number of forms to be printed can be very high, and they might have to be printed in a short period of time, in a **mass printing**. Examples of mass printing are monthly invoices sent by telecom companies or salary statements.

The output of application data is placed into a dynamically expandable table where the size and layout of the output table is determined by the number of records being retrieved.

What About SAPscript?

Customers can use either SAPscript or Smart Forms as their forms solution. SAP recommends you use Smart Forms for new developments (from 4.6C), because these forms are much easier to edit when using the Smart Forms graphical tools. Also, further development is planned for the Smart Form print solution.

For example, the customer can use a preconfigured Smart Form for the Purchase Order, and a SAPscript form for the Production Order (if no suitable Smart Form is available), or create the Smart Form.

While unnecessary, a user's prior knowledge of SAPscript forms can accelerate their Smart Forms learning because some tools are used both by Smart Forms and SAPscript (for example, font maintenance transaction SE73, SAPscript texts, and more).

SAP Smart Form Process Overview

The basic structure of SAP Smart Forms consists of the Smart Form Builder, the Smart Form print form template (which you create or is given to you as a preconfigured starting point), the Smart Form function module, and the Smart Form print program (also described as a driver program).

The Smart Form print programs are **not** the same as SAPScript programs, and you cannot use a SAPScript print program with a Smart Form print form.

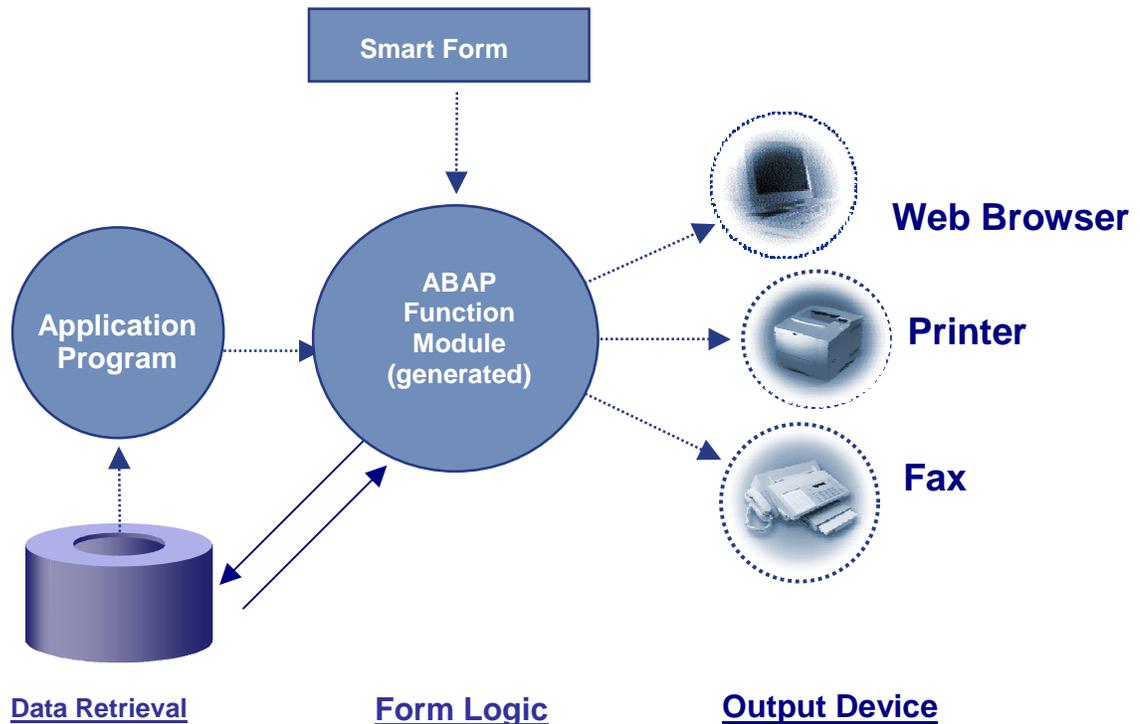
Programming Flow

When an SAP Smart Form template is created, a user creates the form layout, defines the required fields, conditions, and special programming instructions in the Smart Form template using the Smart Form Builder.

After the form design is complete, the form needs to be activated before it can be tested or accessible by print programs. Activating the form initiates the generation of a function module that handles **all** of the form's processing.

This function module interacts with the application program and print program to create the output in the user-defined output media for the specified device.

Smart Form Architecture Diagram

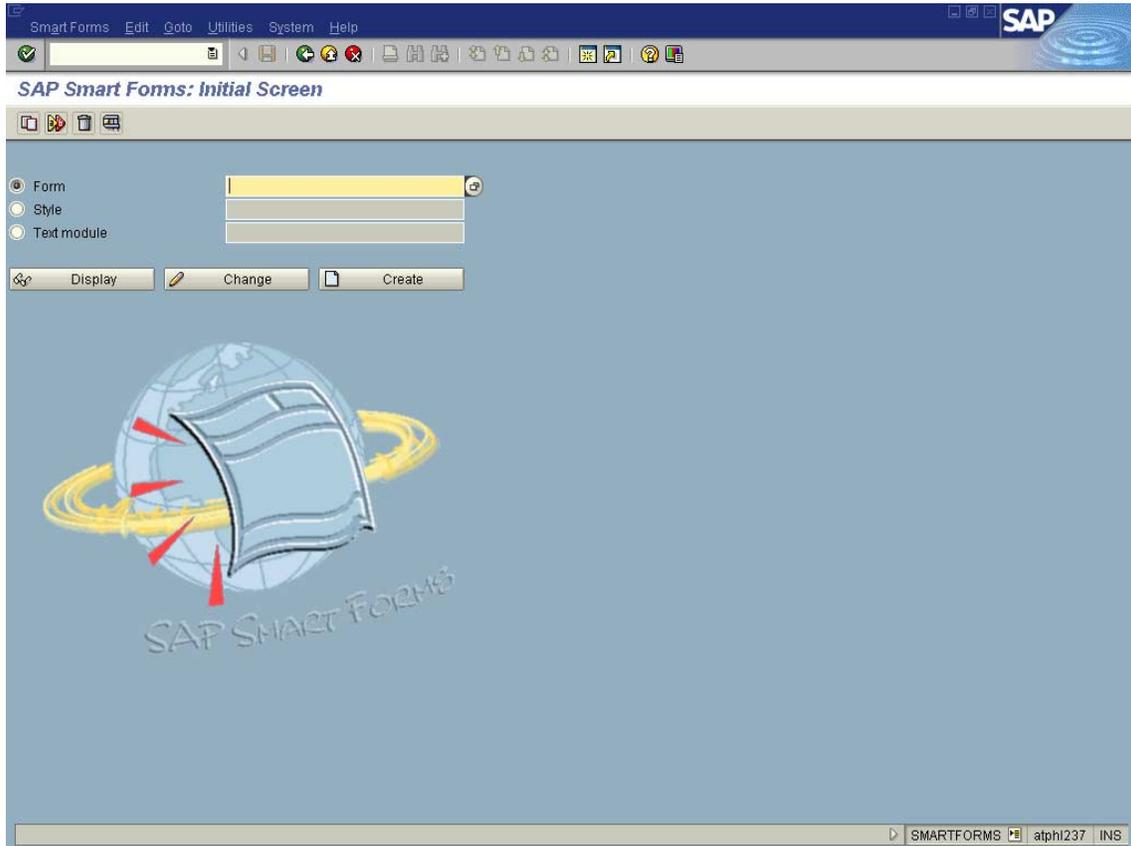


Smart Form Components

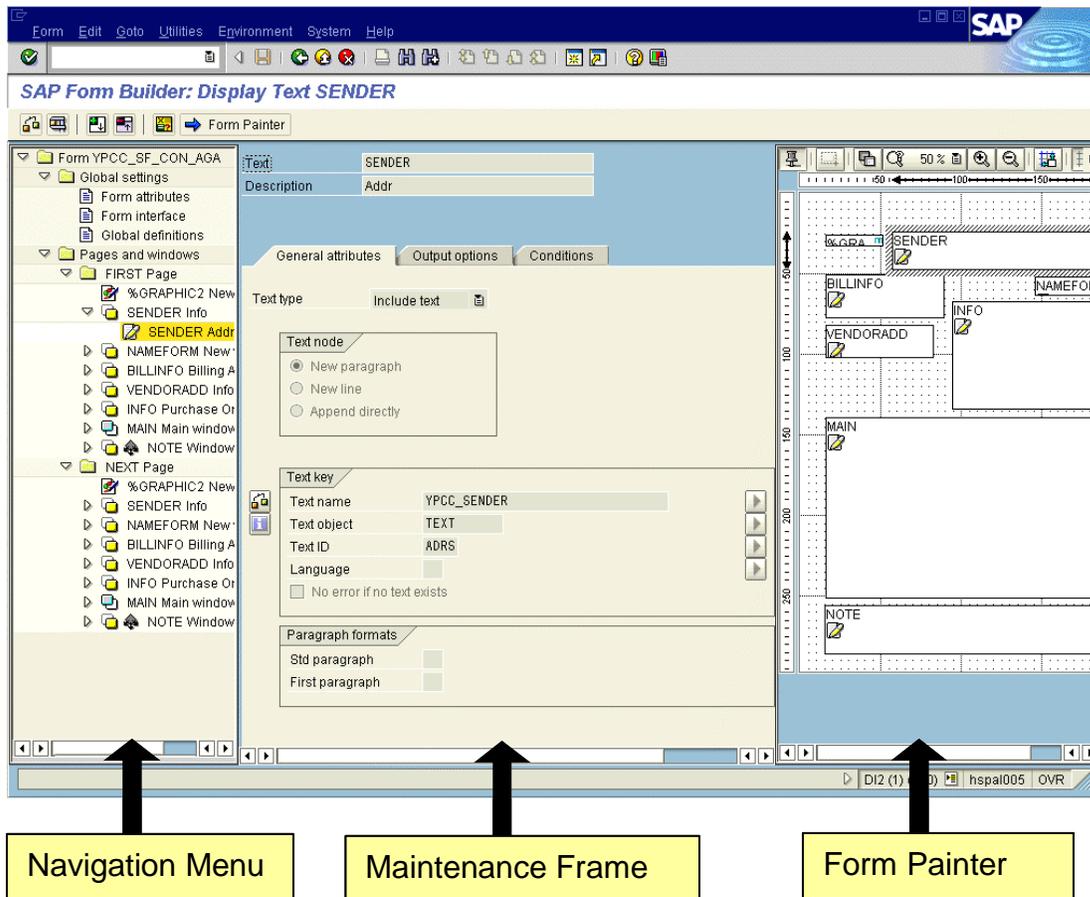
SAP Smart Form Builder

The main SAP Smart Form interface is the Smart Form Builder. Call this transaction by entering the transaction code **smartforms** in the *Command* field.

The initial screen allows you to enter the name of the form you want to modify or copy, or you can create a new form.



After entering the name of the form in the *Formular* (or form) field, select the desired button: either  *Display*,  *Change*, or  *Create*. The *SAP Form Builder* screen appears next.



Smart Form Builder Components

Smart Form Builder Screen

The three frames of Smart Forms Builder screen are the:

- **Navigation menu:** This frame appears on the left of the screen.
- **Maintenance frame:** This frame appears in the middle of the screen.
- **Form Painter frame:** This frame appears on the right of the screen. If your Form Painter is not visible on the screen, choose  *Form Painter*. You can control whether the Form Painter is visible or hidden using this button.

Each frame can be resized by clicking and dragging the borders.



By selecting a node in the navigation menu, the view in the other frames changes as follows:

- **Maintenance frame:** By double-clicking on the desired navigation menu node, the corresponding information (general attributes, options, and conditions) is selected.

- Form Painter: By single or double-clicking on the desired navigation menu node, the appearance of the Form Painter changes to highlight the corresponding **output area** (the box with the label based on the node label in the navigation menu) is highlighted.
- You can reposition the output area box by dragging and dropping it to a new position.

Smart Form Print Form Template

A Smart Form print form is designed using the Smart Form Builder. This form does not create the final output. The Smart Form function module, generated by the Smart Form print form, creates the output.

Smart Form Function Module

The Smart Form function module is the code generated automatically when you activate your Smart Form print form. Each time you make a change to your print form in the Smart Form Builder, the change needs to be activated in the Smart Form Builder before taking effect in your output.

Smart Form Print Program

The Smart Form print program is where the majority of data extraction occurs. Typically, one Smart Form print program should be associated with one type of form. For example, the Smart Form purchase order has a corresponding Smart Form purchase order print program that only handles data extraction for that form.

This one-to-one relationship improves the transparency for the user between what is being performed in the form versus the print program.

We recommend that customer-specific customization be performed in the Smart Form print form, and not in the Smart Form print program.



A major benefit of the Smart Form print solution over SAPscript is that if the layout or order of information changes in the Smart Form print form, you do not need to alter the print program. The layout of data on the form is controlled **only** in the Smart Form print form.

Specifics about Smart Form Development

Important Objects for Form Development

The following objects can be edited from the Smart Forms initial screen (transaction **SMARTFORMS**):

- **Smart Style:** Each form uses at least one Smart Style. A style is a collection of paragraph and character formats.
- **Text modules:** Forms can include text modules (encapsulated text), making it possible to reuse texts in several forms.
- **The form itself:** The form contains form processing logic, content, and layout, and refers to Smart Styles and text modules. The form is also called a Smart Form.

The objects are cross-client and have a translation connection (but until now no workbench connection). You can include old SAPscript texts in a Smart Form using include texts. However, all SAPscript commands will be ignored.

Graphics

Graphics to be used in forms must first be stored on the Business Document Server (BDS) of the SAP System (as a bitmap image or TIFF). The transaction code **SE78** is used for this purpose. Applications graphics include:

- Logos
- Preprinted forms that are scanned in as background pictures

Templates and Tables

SAP Smart Forms has a powerful new tool called the Table Painter. The Table Painter provides more powerful functionality over SAPscript. For example, table and template outputs that are rather difficult to create in SAPscript are much easier in Smart Forms. One exciting feature in Smart Forms is the ability to create dynamically expandable tables.

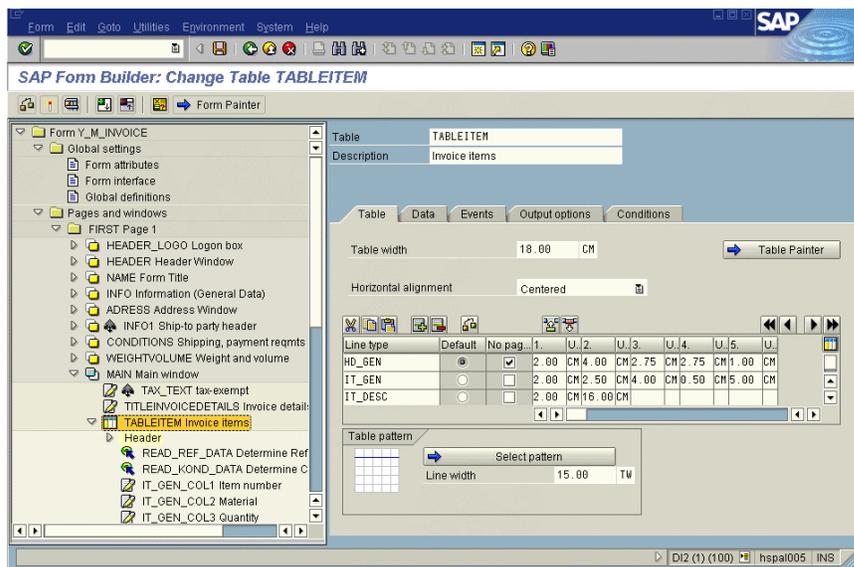
In Smart Forms, although templates still have a fixed number of rows, the number of rows in a Smart Form table can be dynamic. In both Smart Form templates and tables, you can determine how many cells a line of a table output should have using the Table Painter's row types.

Additionally, you can use a scanned-in image of a preprinted third-party form (such as a W2 form for Employee Earnings in the U.S.) as a background image in the creation of templates. A background image proves especially useful when laying out fields on a template. This background image can either be printed or not when the form is output.

Tables

When you work on a table node in your form, the node is marked with the table icon . In the maintenance frame, you get a new type of tab, the *Table* tab. See below.

On this tab, you can define the table characteristics, and turn on the Table Painter to help lay out the rows and columns. Turn on the Table Painter by choosing  *Table Painter*.



The display in the Table Painter does not exactly show what the actual output will be, because the depth of a table row output depends dynamically on the number of records received from the application. Instead, the rows displayed in the Table Painter all display the same table depth.

For tables, know that:

- The application data is selected in the application program and transferred to the form using the form interface.
- In contrast to the template node, the table node has another tab called the *Data* tab. On this tab, you enter the internal table whose data is to be output. The data is read to a work area in row format.
- The table layout is determined using row types (same for templates).
- A header area (for table headers), main area (for items of the table), and footer area (for whole sums, or similar) are table output areas. Table rows are created as subnodes of these areas.

Procedures to Familiarize Yourself with Smart Forms

Getting Started: Creating a Form

The steps listed in this section serve as an introduction to SAP Smart Forms. For this overview, we start with a Smart Form available in standard SAP R/3 4.6C called LB_BIL_INVOICE.



Before regularly using SAP Smart Forms, you should enter an output device in the user settings. Otherwise, you have to enter one in the spool dialog every time. You can enter the output device by choosing *System* → *User Profile* → *Own Data* and choosing the *Defaults* tab).

1. In the *Command* field, enter the transaction **SMARTFORMS** and choose
2. To create a new form by copying an existing form:
 - In *Form Name*, enter the form name (for example, **LB_BIL_INVOICE**).
 - Choose
3. In the *Copy form* dialog box:
 - In *Target form*, enter the name you would like for you new form. Your form name must begin with a Y or Z. Ensure that the name has not already been used in your system.
 - Choose to continue.

Source form	LB_BIL_INVOICE
Target form	Y_your name...

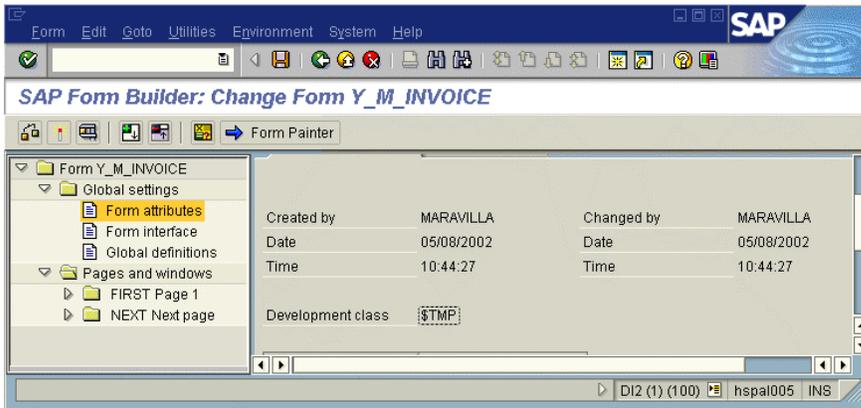
9. On the following screen, your form name now appears in the *Form name* box. Choose .



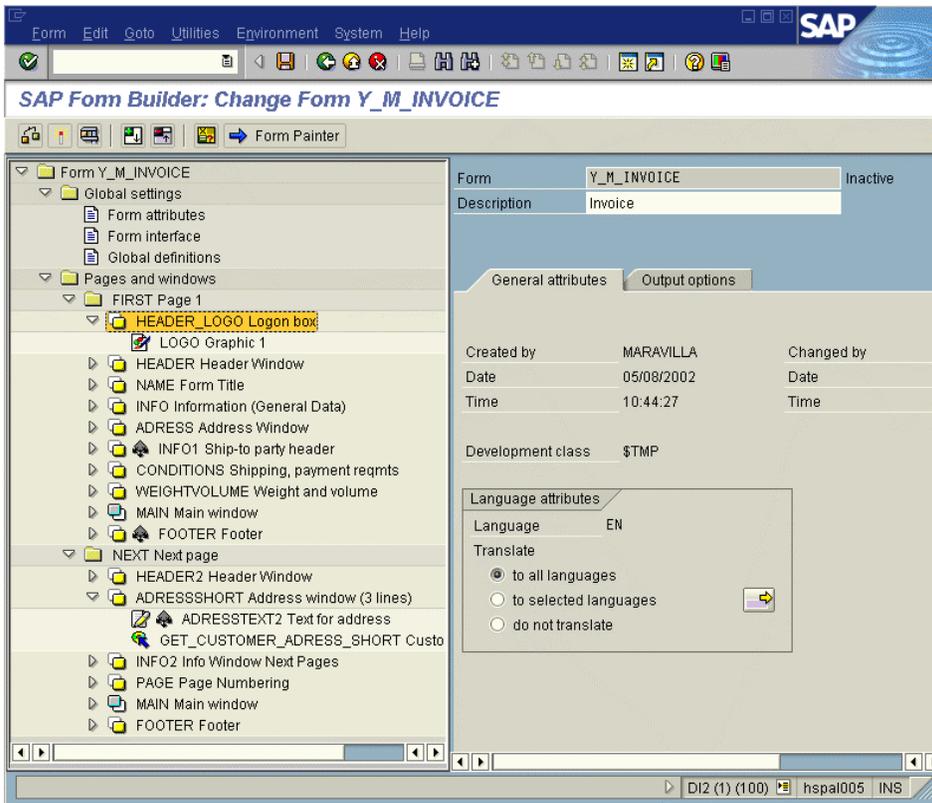
When the Form Builder starts, you can see a navigation menu in the left frame, in which the first draft page has been created with a main window.

10. At this point, you can see all the components of your new form, based on the standard SAP Smart Form for an invoice. When the form first opens, the menu items in the navigation menu will not be expanded (see below).

11. Choose a node to expand a menu.



With both *FIRST Page 1* and *NEXT Next Page* expanded, and one subnode below each further expanded, the navigation menu appears as follows:



Working with Nodes in a Form

All nodes can be created or moved using the navigation tree.

- **Node Order:**
 - The node order impacts the order in which form output is processed. Additionally, conditions that you can create on a specific node determined whether a node is processed.
 - You determine the order of output on the form based on the order that you arrange the nodes in the Navigation Tree.
- **Node Movement:**
 - To move a node, click and drag it to the new desired location in the navigation tree.
 - You can also move a node by double-clicking in the Form Painter, the right frame in the Form Builder screen. If your Form Painter is not visible on the screen, choose  *Form Painter*. You can control whether the Form Painter is visible or hidden using this button. You can navigate to the relevant nodes in the navigation menu by selecting the desired output area in the right frame.
- **Creation and Definition:**
 - For each node, you define the attributes for the node in the Form Builder's maintenance frame (for example, text, the technical name of the graphics to be included, or styles).
 - In the Form Builder's maintenance frame, the tabs for *Output options* and *Conditions* appear in all output node types.
 - A graphic is added as a separate node as well, and is positioned on the form using the Form Painter.



All texts of the form are output using text nodes. As an exception, addresses can also be output using address nodes (direct access to the Business Address Services, previously Central Address Management).

Node Creation Steps

1. Create a text node: Position your cursor where you want to add your text node and click.
2. Then right click and choose *Create* → *Text*
3. Enter the text `Hello world` in the text editing box in Form Builder's center frame, the maintenance frame.
4. Save the node.
5. To check the form, choose .
6. After the form has been checked, choose  to activate it.



When the form is activated, the Form Builder generates a function module. The name of the generated function module is unique within a system.

The module must be regenerated if it is transported to another system. You can view the Function Module name by pressing the F8 key, which brings you to the initial screen of the Function Builder.

Testing Your Form

For simple tests in which no application data has to be transferred, choose  or press F8.

- Press F8 once to call the Function Builder.
- Press F8 a second time for a single test of the Function Module.
- Press F8 a third time to execute the Function Module test screen.
- Press F8 a fourth time for the print preview.

Appendix

Transaction Codes:

Tool	Transaction	Use
SAP Smart Forms (initial screen)	SMARTFORMS	This central initial screen is the most important tool for form development. The relevant tool is started when creating, changing, or displaying an object (form, text module, or Smart Style): <ul style="list-style-type: none"> • Form Builder (edit form) • Style Builder (edit Smart Style) • Text module maintenance (edit text module)
Administration of form graphics	SE78	Import form graphics for use in SAPscript or SAP Smart Forms.

Example Forms Available in Standard SAP R/3:

Form Name/Report Name	Meaning
SF_EXAMPLE_01	Simple example; invoice with table output of flight booking for one customer
SF_EXAMPLE_02	Similar to SF_EXAMPLE_01 but with subtotals
SF_EXAMPLE_03	Similar to SF_EXAMPLE_02 , whereby several customers are selected in the application program; the form is called for each customer and all form outputs are included in an output request