

BC470

Form Printing with SAP Smart Forms

THE BEST-RUN BUSINESSES RUN SAP



© SAP AG 2007

- SAP NetWeaver 2004s
- 2006/Q2
- Material number: 50084556

Copyright 2007 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

© SAP AG 2006

- Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.
- Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.
- IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, System i, System i5, System p, System p5, System x, System z, System z9, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, POWER5+, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.
- Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.
- Oracle is a registered trademark of Oracle Corporation.
- UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.
- Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.
- HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.
- Java is a registered trademark of Sun Microsystems, Inc.
- JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.
- MaxDB is a trademark of MySQL AB, Sweden.
- SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

- The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG.
- This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice.
- SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.
- SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.
- The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.

- **Required:**
 - BC400 (ABAP Workbench: Basics)
 - ABAP programming knowledge
- **Recommended:**
 - Standard Level 2 application training courses are an advantage

© SAP AG 2002

- Note: This course teaches you the basic principles of developing forms with SAP Smart Forms. It does not teach any application-specific knowledge.

- **Participants:**
 - Project team members, developers and consultants responsible for form printing
- **Duration: 2 days**



© SAP AG 2002

Note:

- The training materials do not form an independent study program. They are intended for use in association with the course instructor's explanations.

Contents:

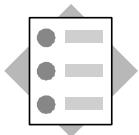
- **Course Goals**
- **Course Objectives**
- **Table of Contents**
- **Course Overview Diagram**
- **Main Business Scenario**

© SAP AG 2006



- **Become familiar with SAP Smart Forms and the interaction between the graphical tools**
- **Create and maintain forms**
- **Integration into application programs**
- **Use Smart Styles**

© SAP AG 2006



At the conclusion of this course, you will be able to:

- **Create and change SAP Smart Forms using the SAP Form Builder and its tools**
- **Create and change styles for forms using the Style Builder**
- **Explain how forms are implemented technically and how they are integrated into application programs**

© SAP AG 2006

Preface

Unit 1	Course Overview	Unit 6	Tables and Templates
Unit 2	SAP Smart Forms: Overview	Unit 7	Process Control
Unit 3	First Steps with the SAP Form Builder	Unit 8	Integration into Application Programs
Unit 4	Texts, Addresses, Graphics	Unit 9	Smart Styles
Unit 5	Data in Forms		

Appendix

© SAP AG 2006

Main Business Scenario

SAP

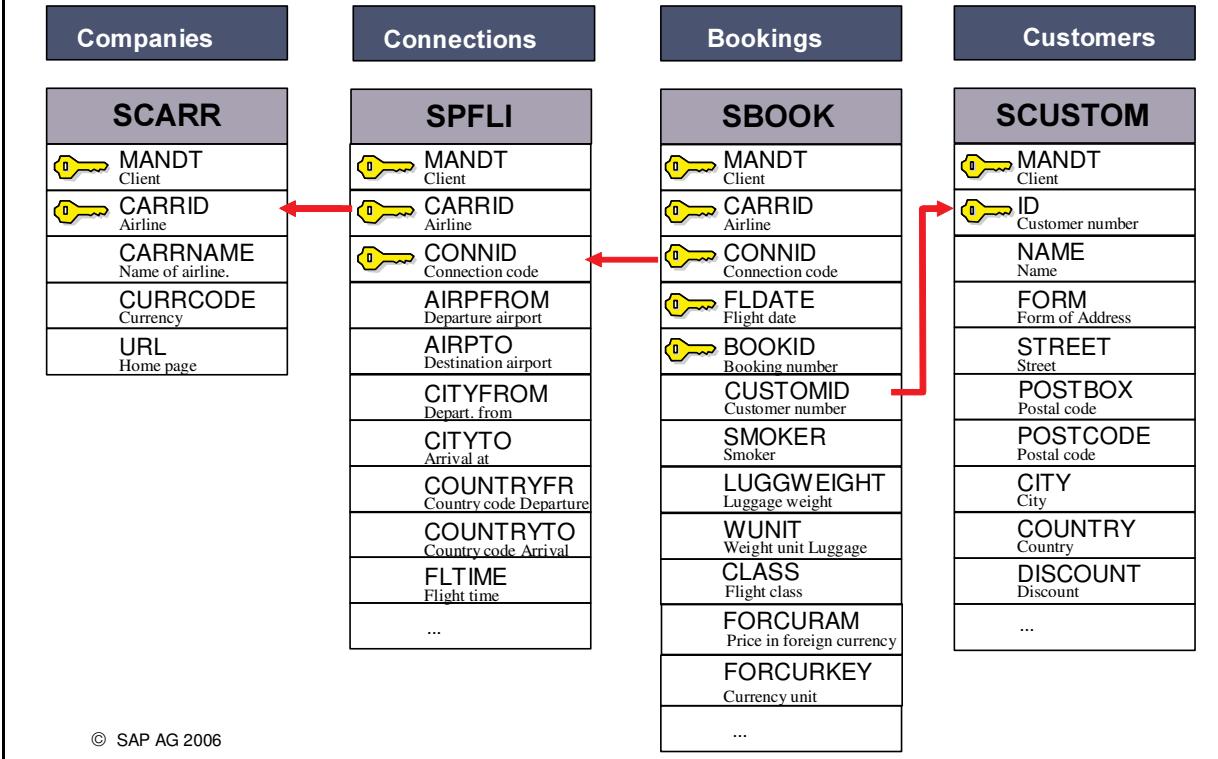


- You are an employee with the Fly & Smile travel agency. You are responsible for sending invoices for booked flights to customers.
- The invoice form is set up step by step.
- The business data is taken from the tables of the flight data model that is used in all BC4XX training courses.

© SAP AG 2006

Tables for the Flight Data Model

SAP



- The tables for the flight data model that are most relevant for this course are listed here, along with their key fields and their dependencies. For more information, see the ABAP Dictionary (transaction code SE11).

Package BC470 with the following naming convention:

- **Program**

- **Demos:** SAPBC470_#####D...
- **Templates:** SAPBC470_#####T...
- **Solutions:** SAPBC470_#####S...

- **Forms:**

- **Demos:** BC470_#####D...
- **Templates:** BC470_#####T...
- **Solutions** BC470_#####S...

where ##### = unit identifier

© SAP AG 2006

- The exercises are built on each other. A sample solution is available for each step completed, which you can use as a copy template for the next step.
- There may not be sufficient time to work through all the exercises during the course.
- All objects are in package BC470.
- Each unit has a four-character identifier:
 - 1. Overview
 - 2. SAP Smart Forms: OverviewSTEP 3. First Steps with the SAP Form Builder
TEXT 4. Texts, Addresses, Graphics
DATA 5. Data in Forms
TABL 6. Tables and Templates
FLOW 7. Process Control
PROG 8. Integration into Application Programs
STYL 9. Smart Styles

- The course material is based on SAP NetWeaver 7.0.
- Variations between this release and earlier releases are listed in the slides and in the notes.
- Greater variations between the releases are listed in the appendix.

© SAP AG 2006



BC470 Form Printing with SAP Smart Forms



1 Overview



2 SAP Smart Forms: Overview



3 First Steps with the SAP Form Builder



4 Texts, Addresses, Graphics



5 Data in Forms



6 Tables and Templates



7 Process Control



8 Integration into Application Programs



9 Smart Styles

...

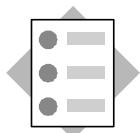
Appendix

© SAP AG 2006

Contents:

- Overview of how SAP Smart Forms can be used and how they function

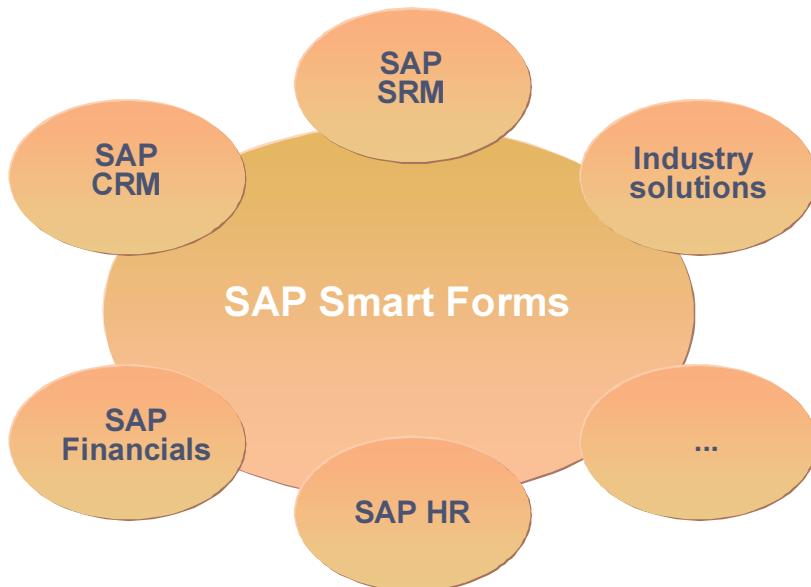
© SAP AG 2006



After completing this unit, you will be able to:

- **List the areas in which form printing with SAP Smart Forms can be used**
- **Explain the basic functions of SAP Smart Forms and how they are integrated into application programs.**

© SAP AG 2006



© SAP AG 2006

- Every company regularly needs to print large numbers of documents with a consistent design, such as invoices or delivery notes. To do this, they must use their business application software. Documents can be output to a printer, a fax device or as e-mails.
- As of SAP R/3 4.6C, SAP provides a tool for form processing **SAP Smart Forms**. This tool includes utilities for designing forms and defining the interface to the application programs that use forms for their data output.
- With the SAP Web Application Server (WAS), SAP Smart Forms can also be generated in HTML format and made available in the intranet or Internet for user entries.
- In addition to the tool itself, the SAP solutions come with a number of forms for central business processes. These include forms for industry solutions, SAP Customer Relationship Management (CRM), SAP Supplier Relationship Management (SRM), SAP Financials, and SAP HR.
- See Note 430621 for an overview of available forms and how they are integrated into applications.
- Note: This training course **does not teach you any application-specific knowledge**. Rather, you learn the basics of form development, which will enable you to make changes to all forms and/or application programs once the course is completed.



Minimum time required for form customizing



Consistent use of graphical tools



Separation of data retrieval and form logic

select



No special scripting language

HTML

Separation of text and layout/logic

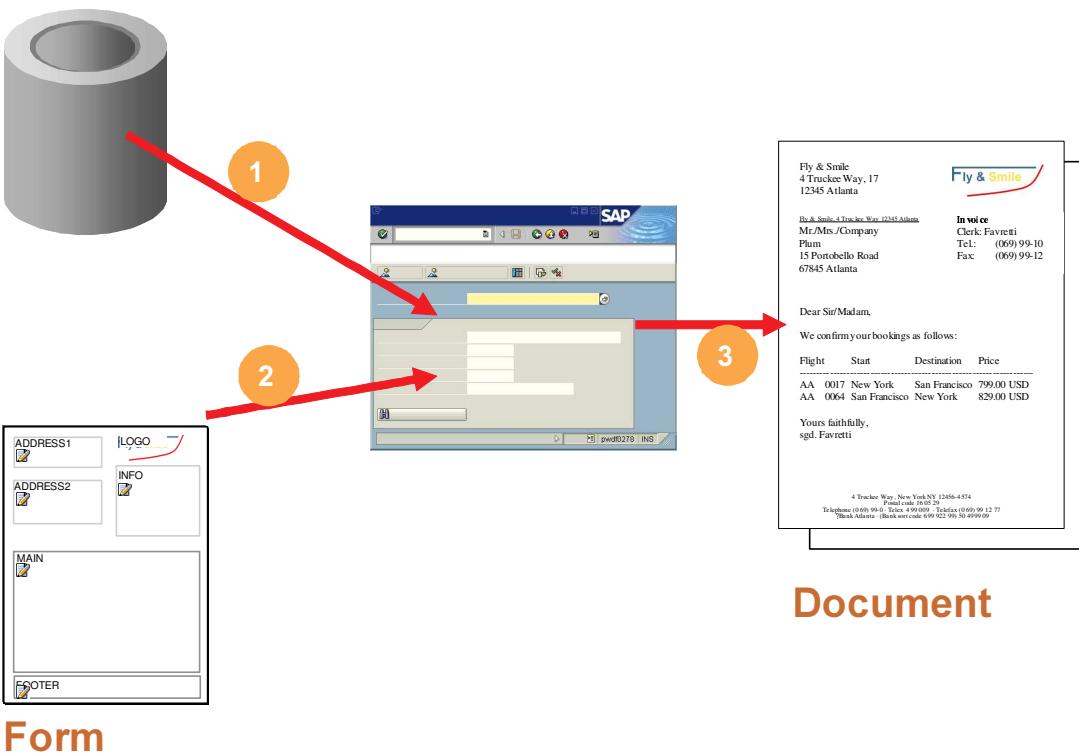
Integration into Internet applications

- **Migration of SAPscript forms and styles is supported.**
- **SAPscript texts can be inserted into SAP Smart Forms.**

© SAP AG 2006

- Compared with SAPscript, the classic form processing tool, SAP Smart Forms provide considerable advantages:
 - It is considerably easier to customize forms, on the one hand because the tools offer more functions (for example, you can very easily create tables with the Table Painter), on the other hand because the interface between the forms and the application programs has been redesigned.
 - Particular tasks, such as retrieving additional data within forms, do not require you to use special scripting language commands, as you had to with SAPscript. You can also insert normal ABAP code.
 - SAP Smart Forms objects are repository objects and are connected to the transport system to ensure that a transport to the production system is possible.
 - As of SAP Web Application Server 6.10, SAP Smart Forms can be output in HTML.
- The migration of SAPscript forms to SAP Smart Forms is supported. You can also migrate SAPscript styles to Smart Styles.
- SAPscript texts can be used directly in SAP Smart Forms.
- Depending on the application, SAPscript forms, SAP Smart Forms or (as of SAP NetWeaver) SAP Interactive Forms can be used. SAPscript texts can be used directly in SAP Smart Forms.

Creating Documents: Simplified View



- To allow an application program to output a document, two basic steps are required:
 1. Data retrieval. The data retrieval step is the central element of the program and can have any degree of complexity, with user interactions, and so on.
 2. Starting form processing. During this step, the data read is written into the form.

SAP Form Builder:



Object Navigator:

```
SELECT * FROM ...
```

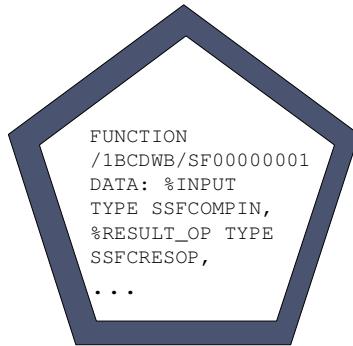
Customizing:

© SAP AG 2006

- Maintaining SAP Smart Forms comprises the following tasks:
 1. Customizing the form - your main task.
 2. Customizing the application program. This is only necessary in special cases, for example, if you want to modify spool settings or read additional data that is to be output in all documents of a printing operation.
 3. Saving the changes in Customizing.

Creation of SAP Smart Forms

SAP



SAP Form Builder:

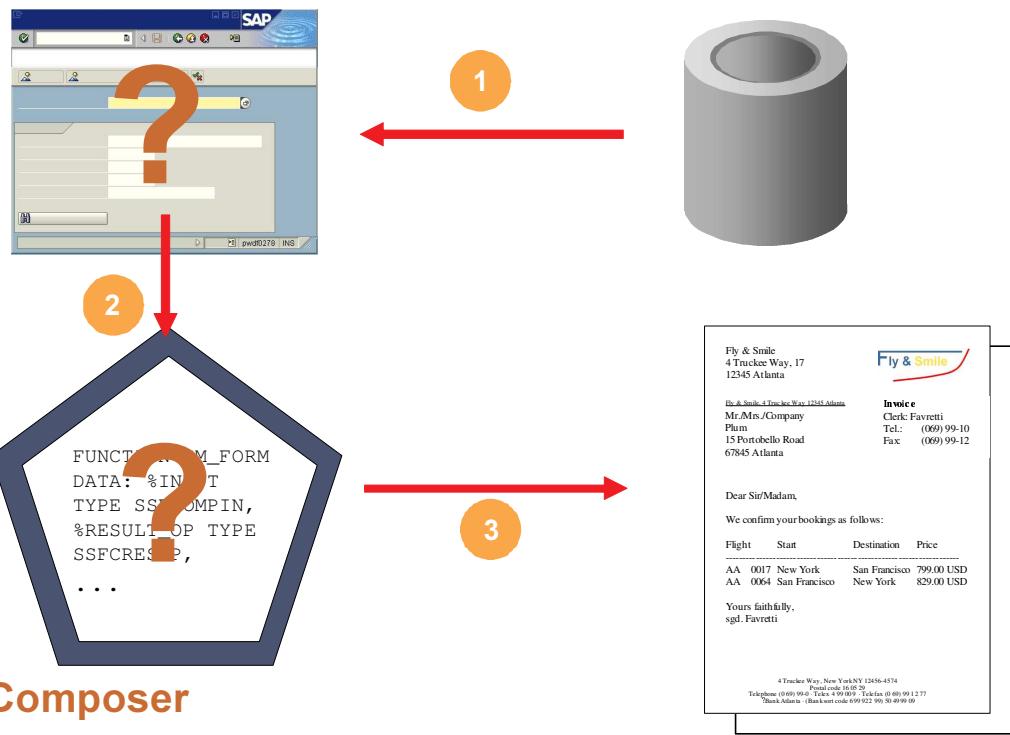
- What ?**
- Where?**
- What size?**
- How often?**
- Where from?**

Generated function module

© SAP AG 2006

- You use the SAP Form Builder to create and adjust forms (transaction: SMARTFORMS) and its tools. There you define the layout (such as the position and size of texts or graphics), the processing sequence of the form elements, and the interface, that is, the application data you want to output in the form.
- Once you have made any necessary adjustments, you must activate the form. During this process, the system first checks if the form contains errors and saves it automatically. The main step, however, is the generation of a function module. A function module is an encapsulated piece of ABAP code, that can be compared to a subroutine. The interface of the function module is the same as the one you defined for the form in the Form Builder. Since the function module is generated automatically, no ABAP knowledge is required.

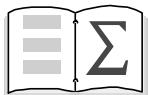
Creating Documents: Complex View



© SAP AG 2006

- The processes involved in document creation are now explained again in more detail:

 1. The transaction checks in Customizing which program to call. This program then reads the data.
 2. In Customizing, the transaction checks which SAP Smart Form to use for the scenario chosen, calls the appropriate function module generated, and thereby triggers the form processing process. The interface is filled with the data read.
 3. When the form processing process is started, the form processor (Composer) is automatically called in the background. The composer is responsible for formatting the texts according to the layout information stored in the form, filling fields with values at runtime, controlling the page breaks, and placing the completed document in the spool.



You are now able to:

- List the areas in which form printing with SAP Smart Forms can be used
- Explain the basic functions of SAP Smart Forms and how they are integrated into application programs.

© SAP AG 2006



BC470 Form Printing with SAP Smart Forms



1 Overview



2 SAP Smart Forms: Overview



3 First Steps with the SAP Form Builder



4 Texts, Addresses, Graphics



5 Data in Forms



6 Tables and Templates



7 Process Control



8 Integration into Application Programs



9 Smart Styles



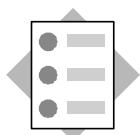
Appendix

© SAP AG 2006

Contents:

- **Creating and maintaining SAP Smart Forms using the SAP Form Builder**
- **Overview of the most important form elements**

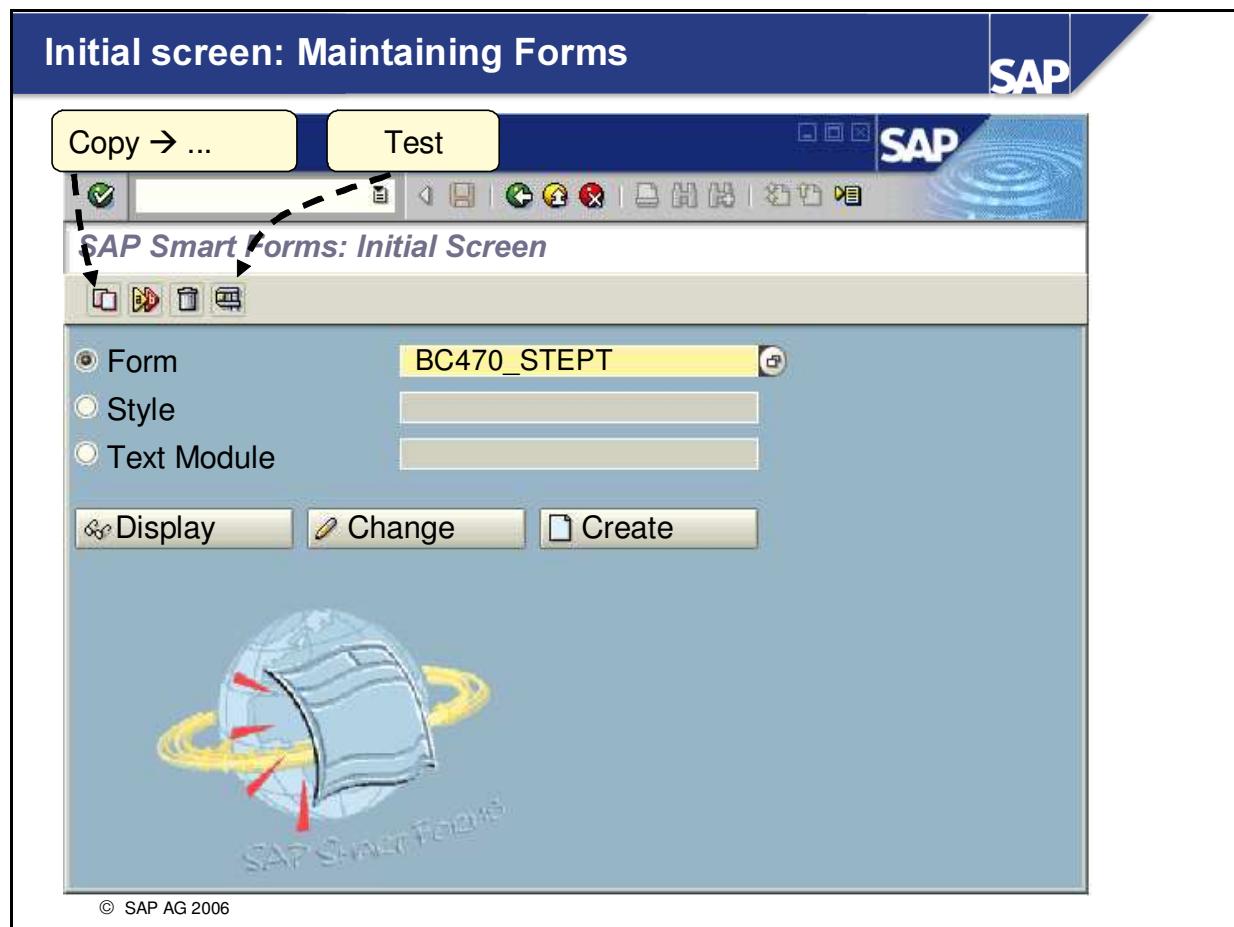
© SAP AG 2006



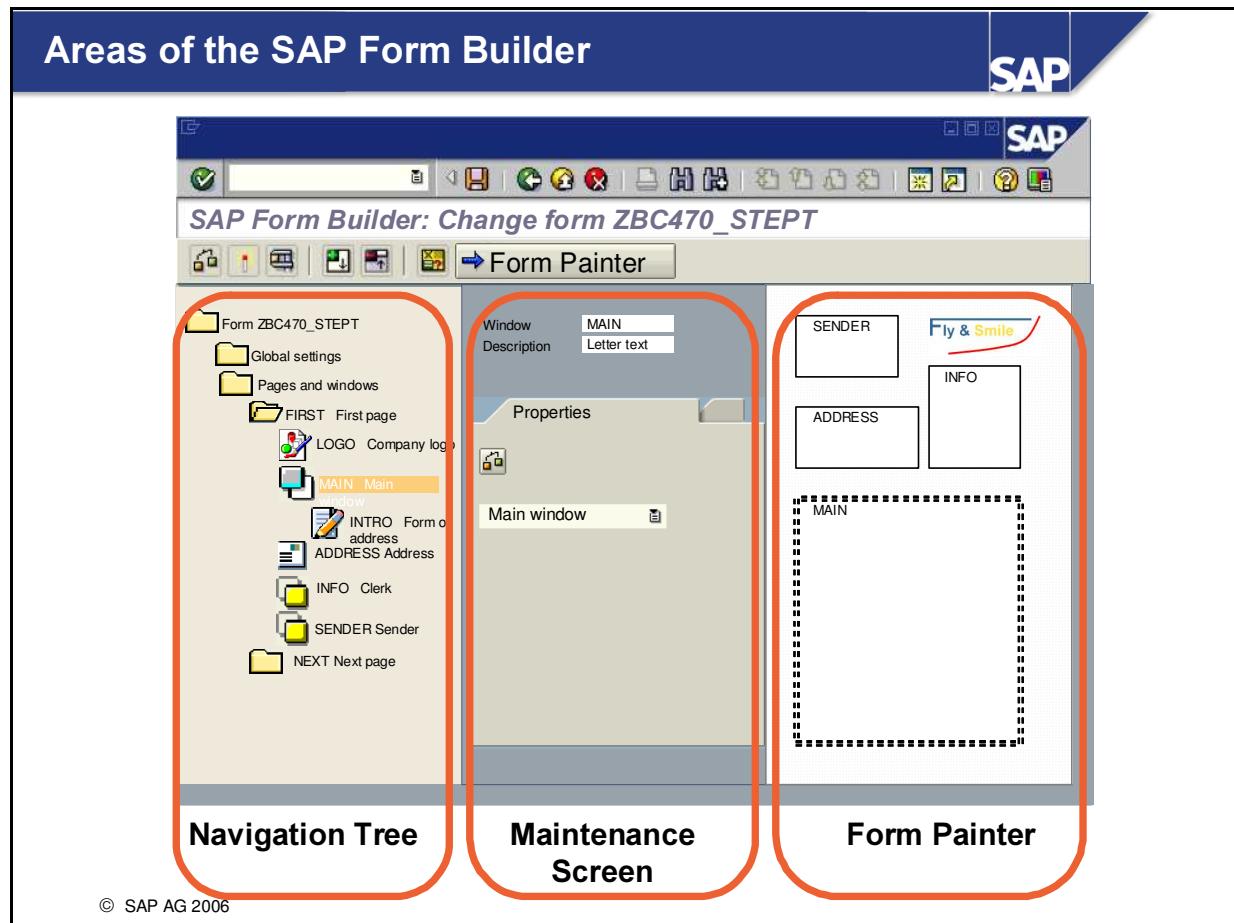
After completing this unit, you will be able to:

- **Work with the SAP Form Builder**
- **Create, copy and edit forms**
- **Create pages and windows**
- **Explain the different window types**
- **Use background pictures**
- **Set output options**
- **Test forms**

© SAP AG 2006

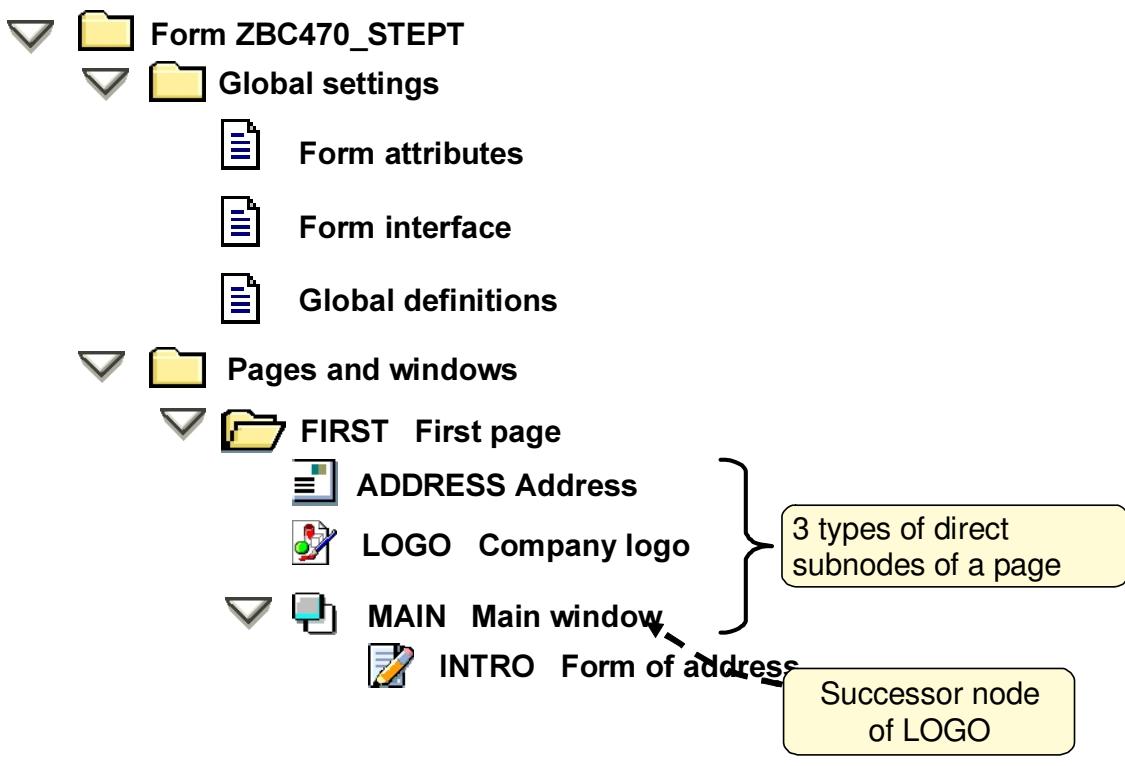


- To call the initial screen of the SAP Smart Forms maintenance transaction, either enter *SMARTFORMS* into the OK code field, or choose *Tools* ® *Form Printout* ® *Smart Forms* from the SAP Menu. Then choose which type of SAP Smart Forms objects you want to edit by selecting the relevant radio button.
 - Forms
 - Styles (see Unit 9: *Smart Styles*)
 - Text modules (see Unit 4: *Texts, Addresses, Graphics*)
- To work on a form, choose the *Form* radio button and enter the name of the form.
- You can create, display, and change forms. The system then takes you to the graphical editing tool, which is called the SAP Form Builder.
- Never change the original SAP forms as your modifications will be lost during the next upgrade. Instead, copy the original form into your customer namespace (starting with Y or Z) and then modify the copied form as required.
- You can also rename, delete, or test forms. To do this, choose the relevant pushbuttons or the options in the menu for *Smart Forms*. Testing a form from the initial screen requires that it has been activated in advance (in the SAP Form Builder).
- You can also make settings specific to the SAP Form Builder. You can create SAP Smart Forms based on existing SAPscript forms by choosing *Utilities* → *Migration* → *Import SAPscript form*.

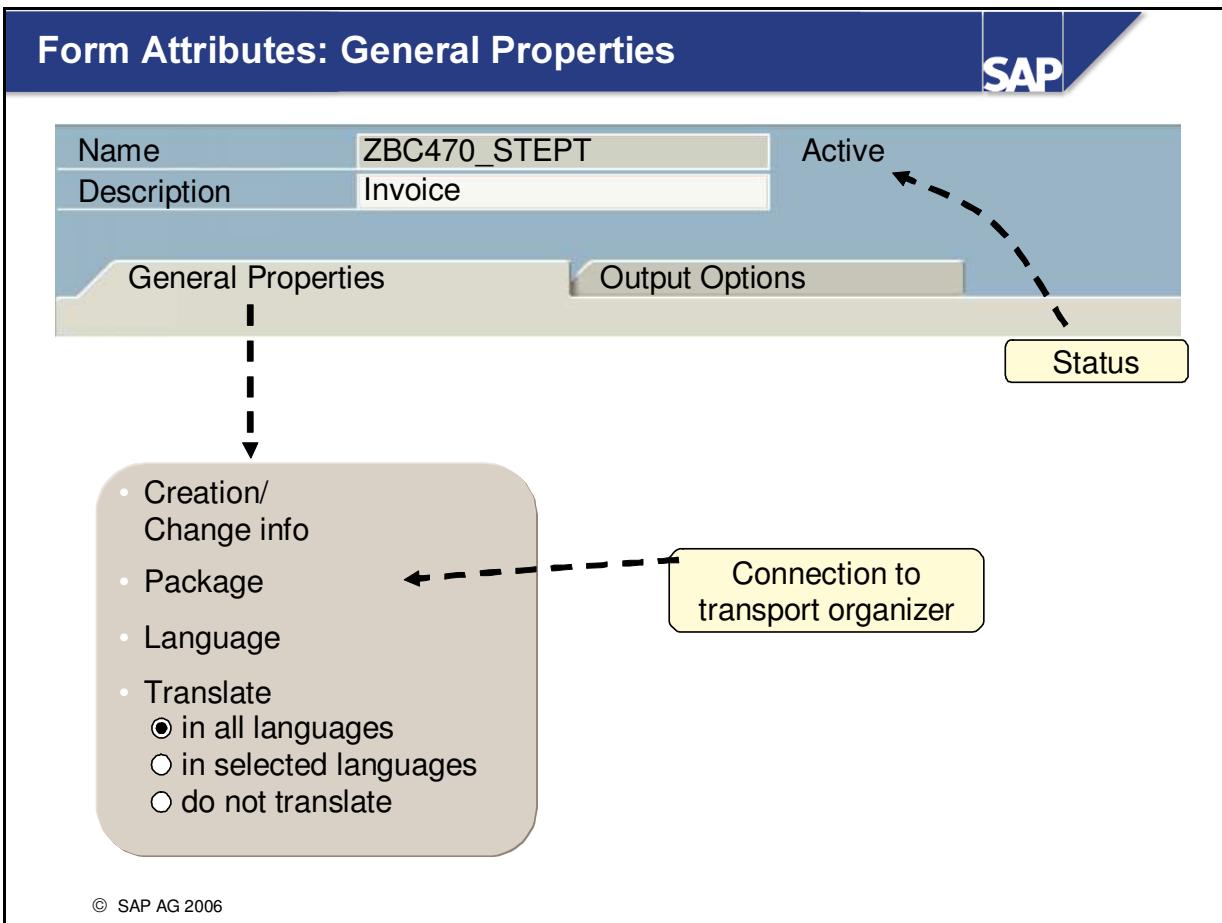


- You use the graphical SAP Form Builder to edit forms.
- The SAP Form Builder is divided into three areas:
 - left: The **Navigation Tree**. This tree graphically displays the hierarchy of the SAP Smart Form. The individual form elements (such as pages or graphics) are represented by nodes. You can also display the field list with variables below the navigation tree. For more information, see Unit 5: *Data in Forms*.
 - In the middle: The **Maintenance Screen**. This screen has several tab pages on which you edit the attributes of the node currently selected. You can also use the editor to enter text, or you can use the Table Painter to determine the layout of a table, for example. Any error messages that occur when the form is checked are displayed at the bottom of the maintenance screen.
 - Right: The **Form Painter**. The Form Painter is used to define the layout of a page, such as the position and size of text windows and graphics. The Form Painter can also be hidden by choosing: *Utilities → Form Painter on/off*.
- You can select nodes for editing by double-clicking them in the navigation tree or the Form Painter.

The Navigation Tree



- All elements of a form are represented by a specific node in the navigation tree.
- Subnodes** inherit properties of higher-level nodes, for example, the style. If a node is not processed, neither are its subnodes. However, a **successor node** of a node is independent. It is processed after the predecessor node.
- If a node has subnodes, you can expand its structure by clicking the triangle symbol beside the node icon. You can select a node for editing by double-clicking it. The system then displays the node in the maintenance screen and in the Form Painter (provided the Form Painter is switched on).
- Below the top node, you always find the following two nodes:
Global settings. These include:
 - Form attributes: These can be administrative information and basic formatting settings.
 - Form interface: Here you must define the fields to be filled by the application program or returned to the application program. (see Unit 5: *Data in Forms*.)
 - Global definitions: Here you can define additional fields to be used in the form. (see Unit 5: *Data in Forms*.)
 - Pages and windows (see later in this unit).



- The form attributes include not only the name and description of the form but also its current status: *active* or *inactive*. A form can exist in any of these two versions. Application programs always use the active version. This means that you can provisionally save your changes without directly affecting application processing. To activate a form, choose *Form → Activate*. Note that when you copy a form, the status of the copy is always set to *inactive*.
- Since SAP Smart Forms are connected to the transport system of the SAP System, they must be assigned to a package (Name under SAP R/3 4.6C: Development class). You do this when you first save your text module. You can subsequently change the package assigned from within the SAP Smart Forms initial screen by choosing *Goto → Object Directory Entry*. Be sure to use the customer namespace (begins with Y or Z).
- Each form has an original language. The *General Properties* tab page allows you to determine if you want to translate the form into other languages, and if so, which ones.
- As of SAP Web Application Server 6.20, you can indicate in the *Restricted Languages* field that the logon language is not to be used for missing texts.

Form Attributes: Output Options

SAP

Name	ZBC470_STEPT
Description	Invoice

General Properties Output Options

CH LN

- Page format
- Characters per inch
- Lines per Inch
- Style
- XML Output

© SAP AG 2006

- The page formats available include the page formats provided in spool administration. The orientation (portrait or landscape) is set individually for each page.
- You can determine which page format is proposed for a new form as follows: Go to the initial screen of transaction SMARTFORMS, choose *Utilities* → *Settings* → tab page *General*.
- You must assign a style to each form. A style is a collection of different character and paragraph formats, which are then used in the form. However, you can specify a separate style for most subnodes, which then overrides the default setting of the form. See Unit 9 - *Smart Styles*.
- *Characters per inch, CPI*. This field allows you to determine the CH unit of measure that you can use for horizontal length specifications (such as window widths) in the form. If you enter the default value 10, 1 CH is equivalent to one tenth of an inch, that is, approximately 2.5 mm.
- Similarly, the *Lines per inch* field allows you to determine the LN unit of measure that you can use for vertical length specifications (such as window lengths) in the form.
- The standard output format for printing is OTF (Output Text Format). A certified XML interface is provided, called the SAP Smart Forms XML Interface (XSF). The XSF data stream does not contain formatting specifications.
- As of SAP Web Application Server 6.10, HTML output is also possible, for example for Web forms. XDF format, a special XML format that also contains specifications for the dictionary types used for the interface parameters, can also be used.

Pages

SAP

Top page in tree
→ First page in document

Technical name:
A-Z, 0-9, _, %

Description

Create
Cut
Copy
Paste
Delete

Expand
Collapse

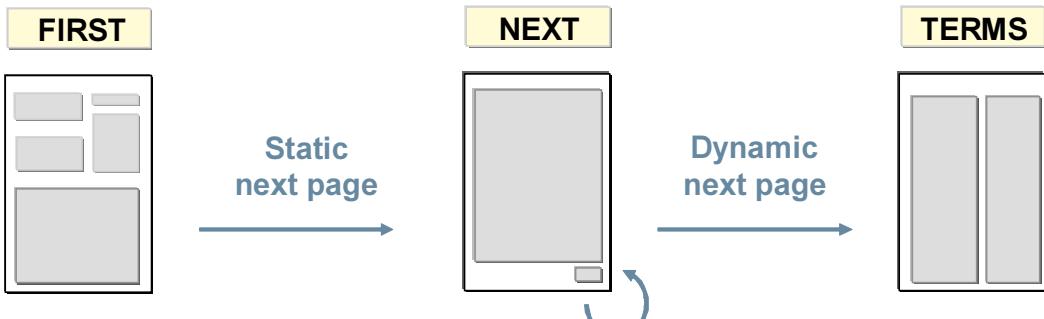
© SAP AG 2006

- Each form consists of at least one page.
- A page is represented by a node in the navigation tree. As with any other node types (such as texts or tables), right-clicking the mouse on an existing page opens a context menu with the available options:
 - Create or delete (in change mode only). When you create a new page, the system proposes a unique technical name, which you can change if required. Note that when you delete a node, all subnodes on the respective page are also deleted.
 - Copy to clipboard; cut and insert into clipboard; paste from clipboard. All subnodes are also affected.
 - Expand or collapse the page in the tree.
- As of SAP Web Application Server 6.10, the following options are also available for actions in the navigation tree: Delete, clipboard (Ctrl-C, Ctrl-V, Ctrl-X), expand (arrow right) and collapse (arrow left), first and last node (Pos1 and End), page up/down.
- The same functions are available in the menu *Edit → Node* or *Edit → Subtree*.
- Like all subnodes, each page has a technical name and a description. Only letters (without umlauts), numbers and underscores are permitted for the names. The first character must be a letter. As a special case, the percentage sign is allowed as the first character. The percentage sign is used by the SAP Form Builder to generate names automatically.

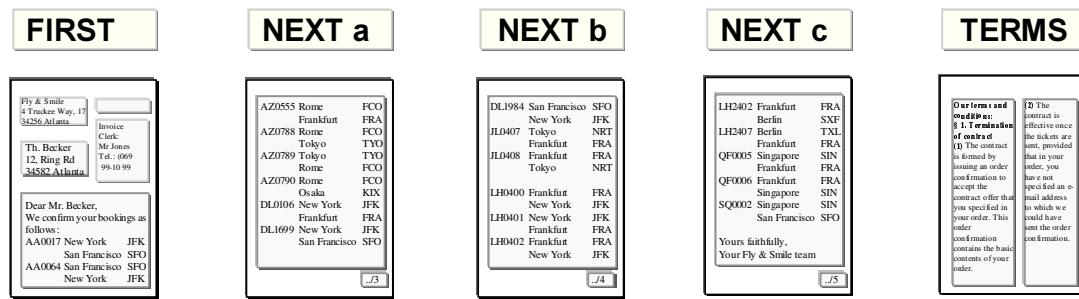
Pages in Forms and Documents

SAP

Form

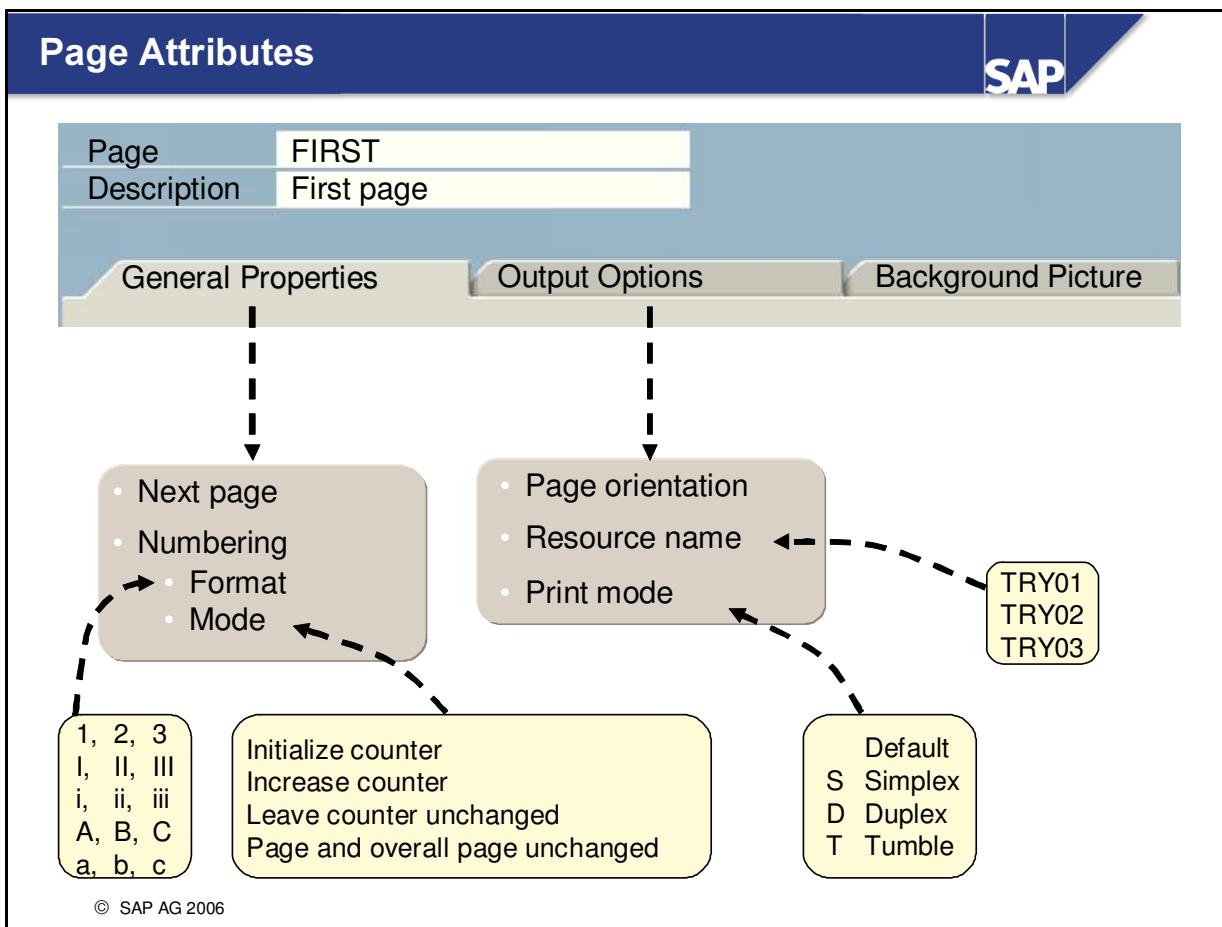


Document



© SAP AG 2006

- You can define one or more pages with different layouts for a form.
- The top page of the navigation tree is processed first. You then control the processing sequence by specifying the next page (on the *General Properties* tab page for the page), which is then processed automatically after the top page. (Alternatively, you can have the next page determined dynamically based on conditions. For example, you can process a page with line items as many times as needed to output all data records and then force the system to change to the page containing the general terms and conditions of business. See also Unit 7 - *Process Control*.)
- Depending on the amount of data to be processed, a form page can be used more than once in a document.



- You can make settings on the following tab pages:

General Properties

The next page. The default value is the page itself.

The type of automatic page numbering. You can choose between Roman and Arabic digits, and uppercase and lowercase format. You can also determine how the pages are numbered. Note that if you make page number settings, this does not mean that the pages are automatically numbered. To have your pages numbered, you must output the variable SFSY-PAGE in a text window. See the slide *System fields* in Unit 4 - *Texts, Addresses, Graphics*

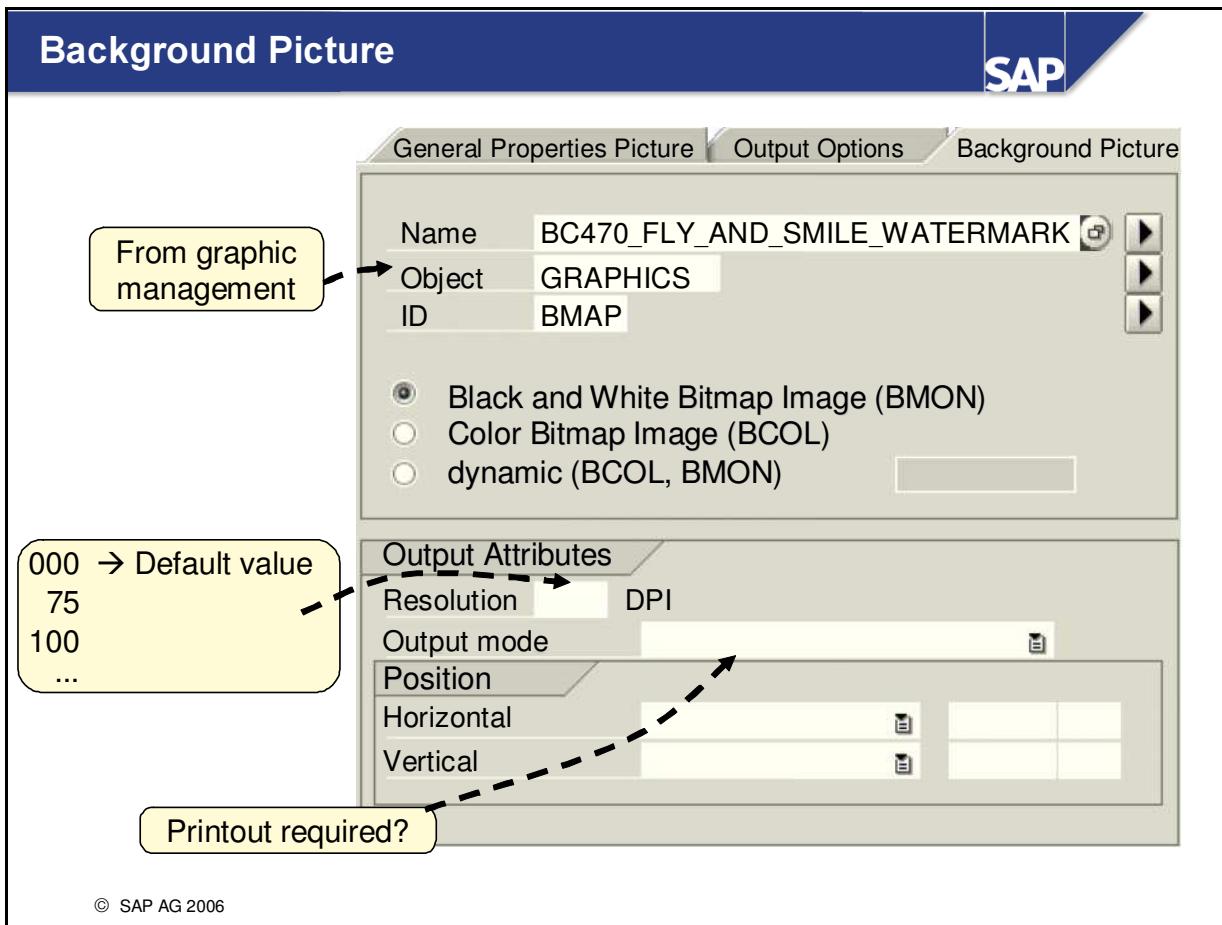
Output Options

While the page format you specify applies to the entire form, set the orientation (portrait or landscape) at page level only.

You can assign different paper trays to pages - provided your printer supports this feature. This makes sense, for example, if you want to use your company letter paper for the first page of a form and normal typing paper for all other pages.

You can set double-sided print mode if required. Your printer supports this feature.

Background Picture The required picture already exists in the system (after you use transaction SE78 for an import). Text is printed over the picture.



- Specify the name of the graphic and its description (object and ID). The default values are GRAPHICS and BMAP. The F4 Help lists the graphics that are available in the system.
- Determine whether it is a *black and white picture* or a *color picture*.
- You can also select the background picture dynamically by entering fields instead of static names, which must have a value assigned to them at runtime.
- Under *Output Attributes*, you can specify the *resolution* in dpi (dots per inch). The smaller the value, the larger the graphic is displayed in the form. If you leave the field blank or enter *000*, the default setting of the graphic is used.
- The *output mode* allows you to determine whether the background picture should only be displayed in the Form Builder (which makes sense if you redraw a scanned form that you do not want to print) or whether the picture should be included in the actual form processing. You can choose between *Print preview* or *Print preview and print*. You can, however, still decide from within the print preview (before the request is sent to the spool) whether you want to print the background picture or not.
- You must also determine the horizontal and the vertical position of the background picture with regard to the page border.
- Update the preview of the Form Painter by choosing *Enter* in the maintenance screen.

Navigation Tree

- ▽ Pages and windows
 - ▽ FIRST First page
 - ADDRESS1 Address
 - LOGO Company logo
 - ▷ MAIN Main window
 - ▷ INFO Clerk

Processing from top down

Document

Fly & Smile
4 Truckee Way, 17
12456 Atlanta
Invoice
Clerk: Favretti
Tel.: (069) 99-10
Fax: (069) 99-12

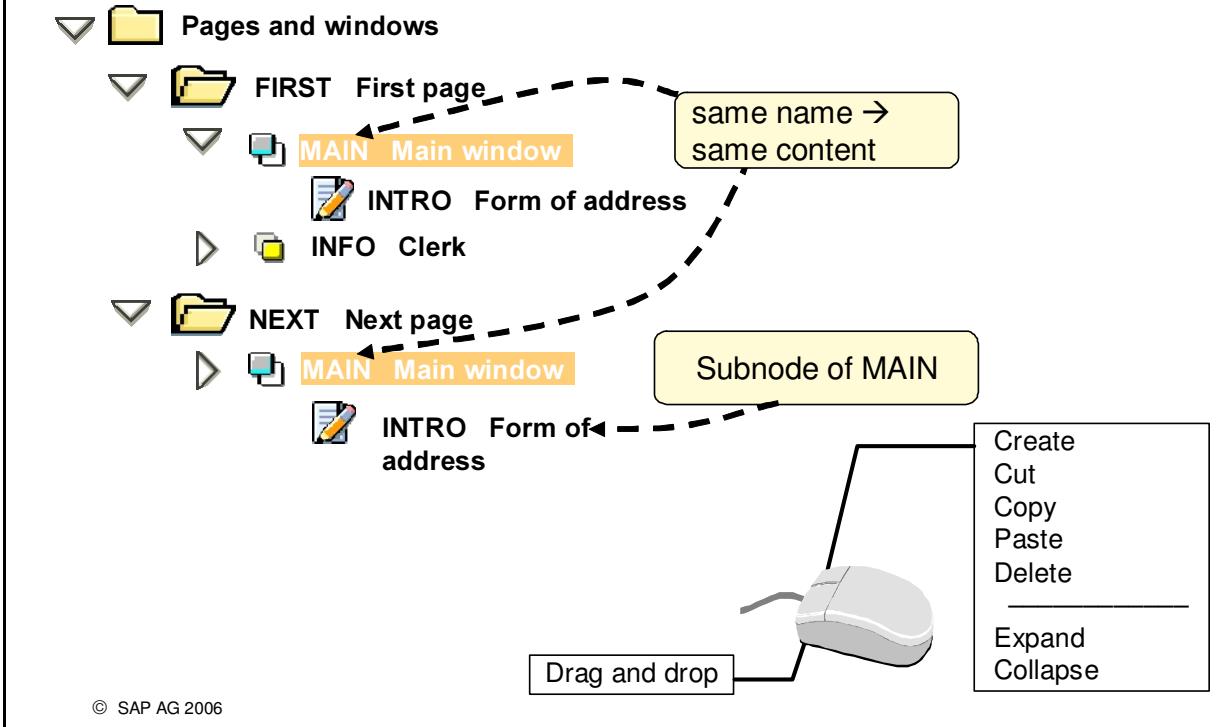
Dear Sir/Madam,
We confirm your bookings as follows:

Flight	Route	Price
AA 0017	NYC – SFO	799.00 USD
AA 0064	SFO - NYC	829.00 USD

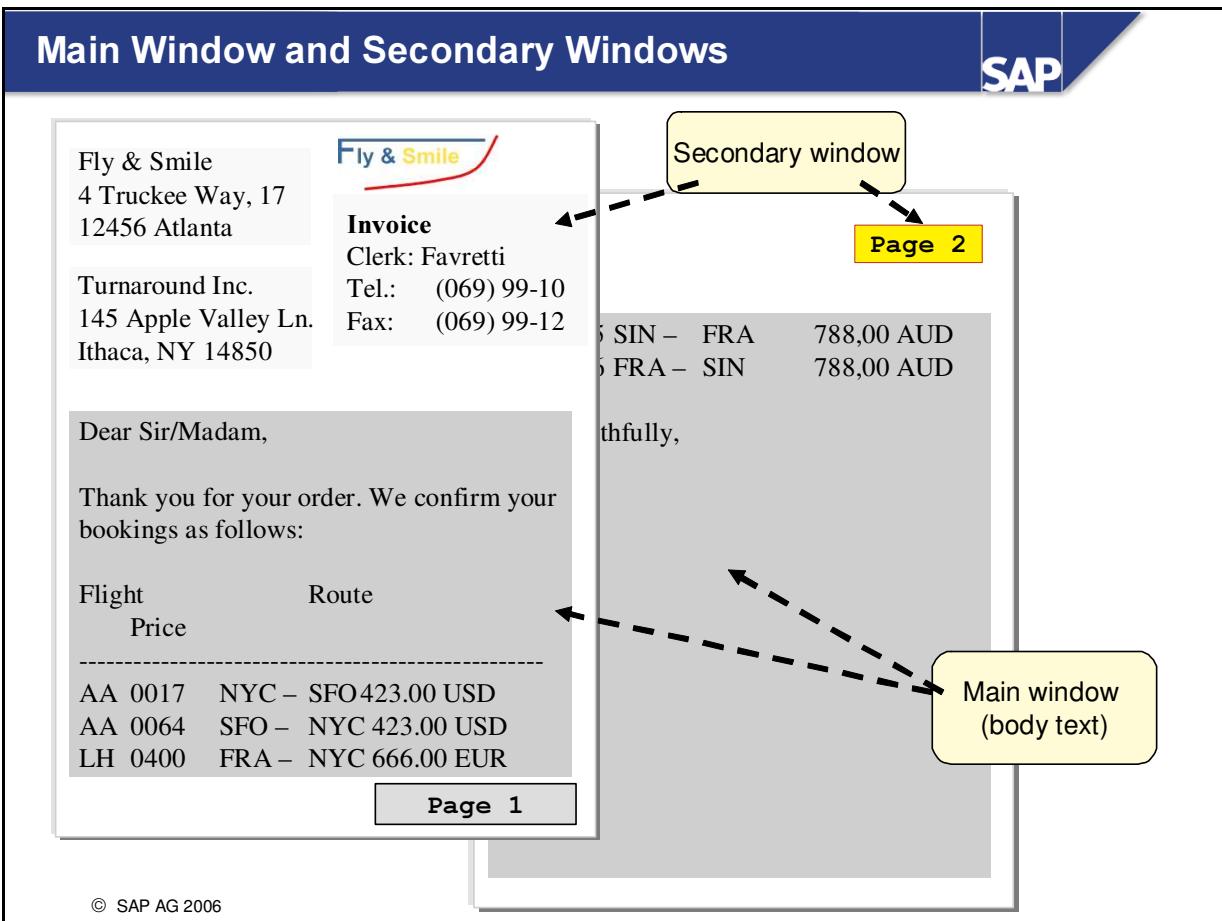
Yours faithfully,
sgd. Favretti

© SAP AG 2006

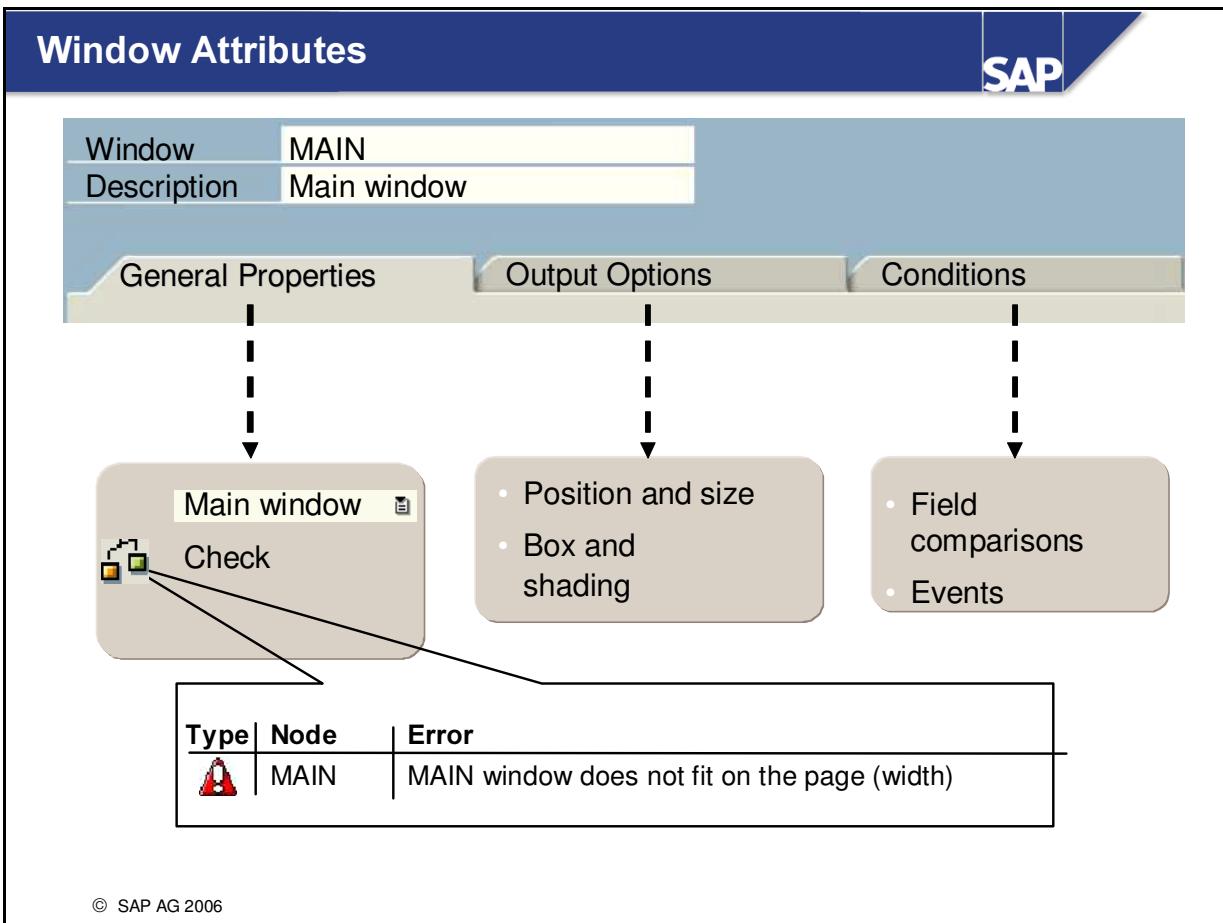
- To be able to output information in a form, you must create suitable output areas (windows) on the relevant page. The following output areas are available:
Subnodes of windows are used to output text and data.
Address window: If the application program uses Business Address Services (BAS), you can easily output formatted addresses in address nodes. (BAS was called *Central Address Management* → *(CAM)* in SAP R/3 4.6C).
Graphic window.
- The output areas of a form are represented as nodes in the navigation tree. The icon helps you to identify the three different node types (address, graphic, or window).
- The order of the page subnodes in the navigation tree does not affect their position in the form, but it does affect their processing: In each page, they are processed from the top down. It is useful to imagine that all nodes are expanded. If necessary, you must move subnodes using drag and drop (left mouse button). The processing sequence is particularly important if you use fields (variables) that are filled only at runtime.



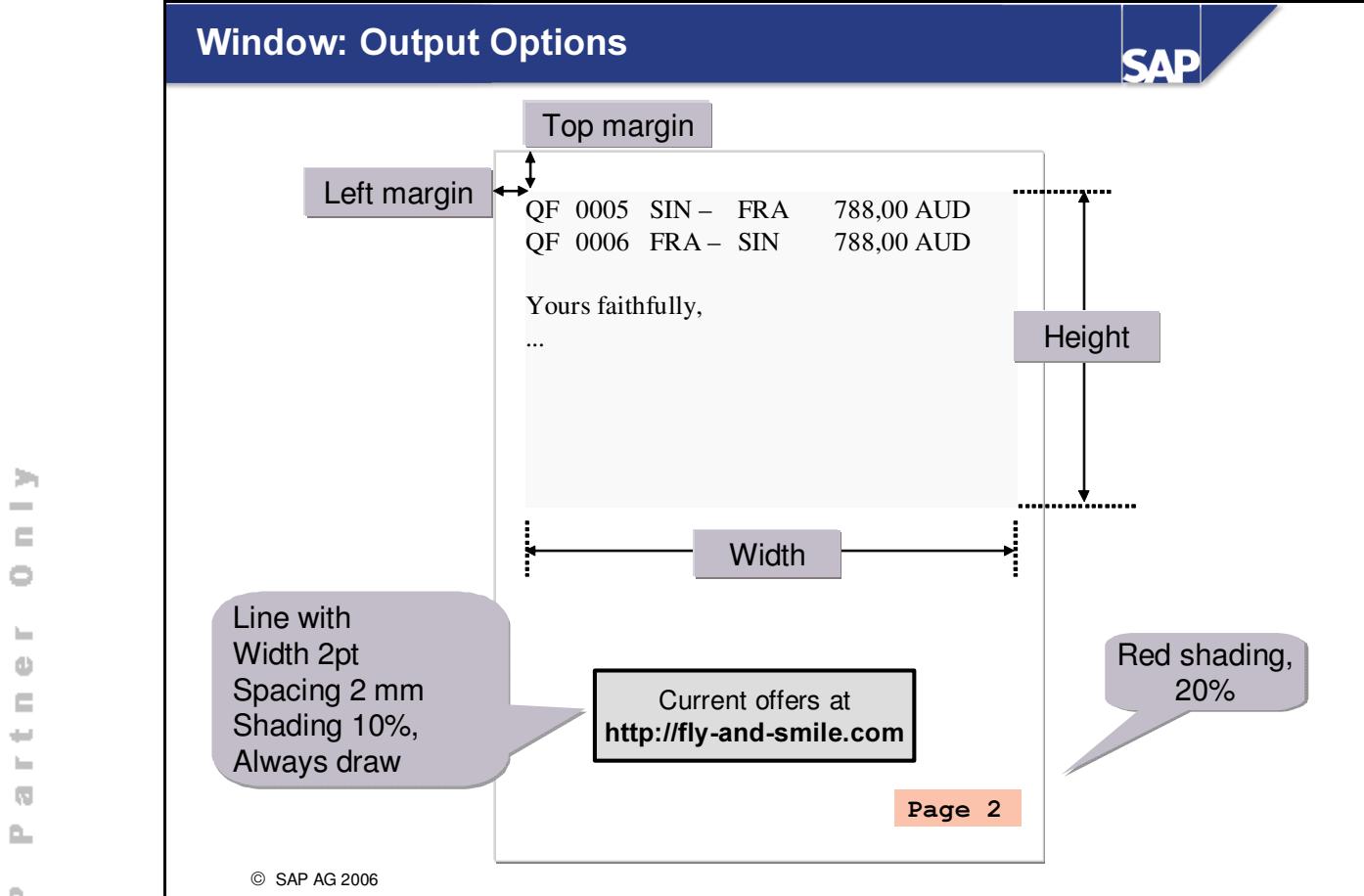
- You use the context menu (right-click the mouse on a page) to create output areas and other nodes. The system proposes a unique technical name which you can change if required.
- You can use drag and drop to move (left mouse button) or copy (Ctrl key and left mouse button) subtrees, including nodes with subnodes. Alternatively, you can use the clipboard (right-click the mouse: cut – copy – paste). For example, you can move or copy windows or text nodes from one page to another.
- As of SAP Web Application Server 6.10, you can also use the keyboard to copy (Ctrl-C), cut (Ctrl-X), and paste (Ctrl-V) nodes.
- If you drag a node A onto a node B, node A is inserted after node B. In some cases, you can also insert the node A to be moved as a subnode of node B. The system then displays a dialog box where you can choose to insert the node *below the node...* or *after the node...*. If you choose the second option, the node to be moved is inserted at the same level as node B, but after it.
- If you place output areas on **several pages** of a form, all changes made to the node contents (including the deletion of subnodes) affect all pages since the technical names of the nodes are identical. However, although the output areas have the same contents, their position may differ from page to page.
- If you place output areas several times **on the same page**, the system creates copies with the same contents, but with different technical names from the originals. Changes to the node contents therefore affect only the relevant area.



- There are basically two types of windows: Main windows and secondary windows.
 - In the subnodes of the **main window**, you enter content that may span several pages (called the body text), for example, a customer's bookings. As soon as the main window is filled with content, all secondary windows on the page that have not yet been processed are processed. There is an automatic page break on the next page. The nodes are processed again there in the sequence in which they are arranged in the navigation tree and the body text is continued in the main window.
- You can only define one window in the form as the main window.
- The main window must have the same width on each page. You can choose the height and position as required.
- A page without a main window may not refer to itself as the next page since this would cause an endless loop.
- In the subnodes of a **secondary window**, you output text and data in a predefined output area. This means that the text is not displayed as a body text with page breaks.
- Text that does not fit into the secondary window is truncated and not output.
- The height, width and position of a secondary window may be different for each page.
- Graphics are automatically set to the correct size.
 - As of SAP Web Application Server 6.10, there are two new window types: *copies windows* and *final windows*. They behave in the same way as secondary windows with regard to the body text, size, and position.



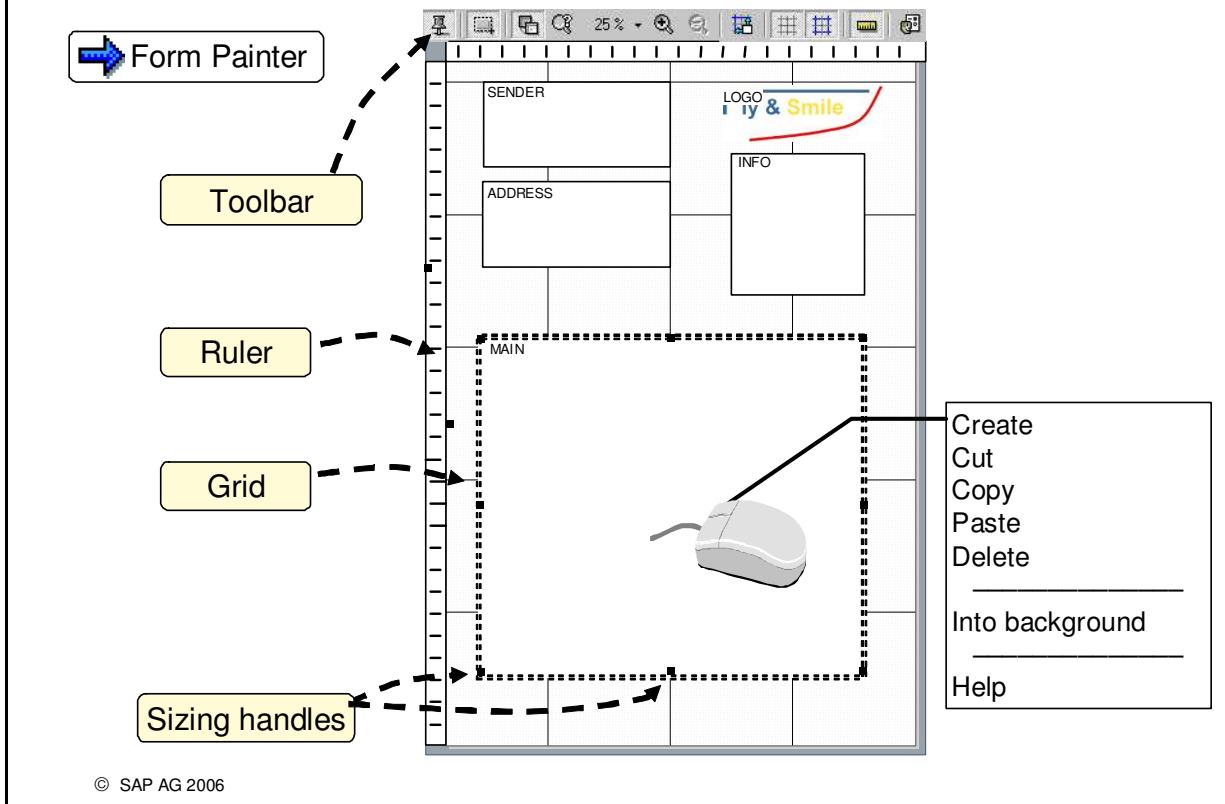
- You can execute checks at different levels, from the bottom node to the entire form. To ensure that all windows are free of errors, they must fit onto the respective page, and the main window must have the same width on all pages. All error messages are displayed in the bottom part of the maintenance screen. You can go directly to a node by clicking its name.
- You set the position and the appearance of a window using the output options.
- As with most node types, you can use conditions for windows to determine the time they are processed. You can choose from a number of processing events (such as *not on first page* or *only on page...*) and also control processing by means of specific values. For example, you may want to print text A for certain customers only, and text B for all other customers. For more information, see Unit 7 - *Process Control*. If the conditions set for a window are not fulfilled, neither the window nor its subnodes are processed. (The same applies for all other nodes and subnodes for which conditions have been specified.)
- If you use identical window nodes on different pages, each node has its own *Output options* and *Conditions* tab pages.



- You determine the position of a window by specifying the upper left margin, and its size by entering its height and width. If you draw a window in the Form Painter, the values you set are automatically copied to the maintenance screen, and vice versa.
- As with all other nodes that allow the output of text, you can define a box and a shading for windows.
- As of SAP Web Application Server 6.10, you can make detailed settings for the four window borders (top, bottom, left, right): You can determine lines in different widths and colors. The distance between the border and text can also be set for each window border individually. You can also choose the color for shading and its intensity for each window.
- With SAP R/3 4.6C, the chosen line width applies to each of the four borders of a window. Only the color black is available for shading, but in any intensity. You set the distance between the text and the window in the *Vertical spacing* or *Horizontal spacing* field.
- If you select the checkbox *Always draw box and shading*, the window is output in the format chosen even if it does not have any contents.
- You can use the following units of measure:
CM, MM, IN (inch = approximately 2,54 cm), PT (point = 1/72 inches), TW (twip = 1/20 points). You can also use LN for vertical length specifications, and CH for horizontal length specifications. You define these units in the form attributes.

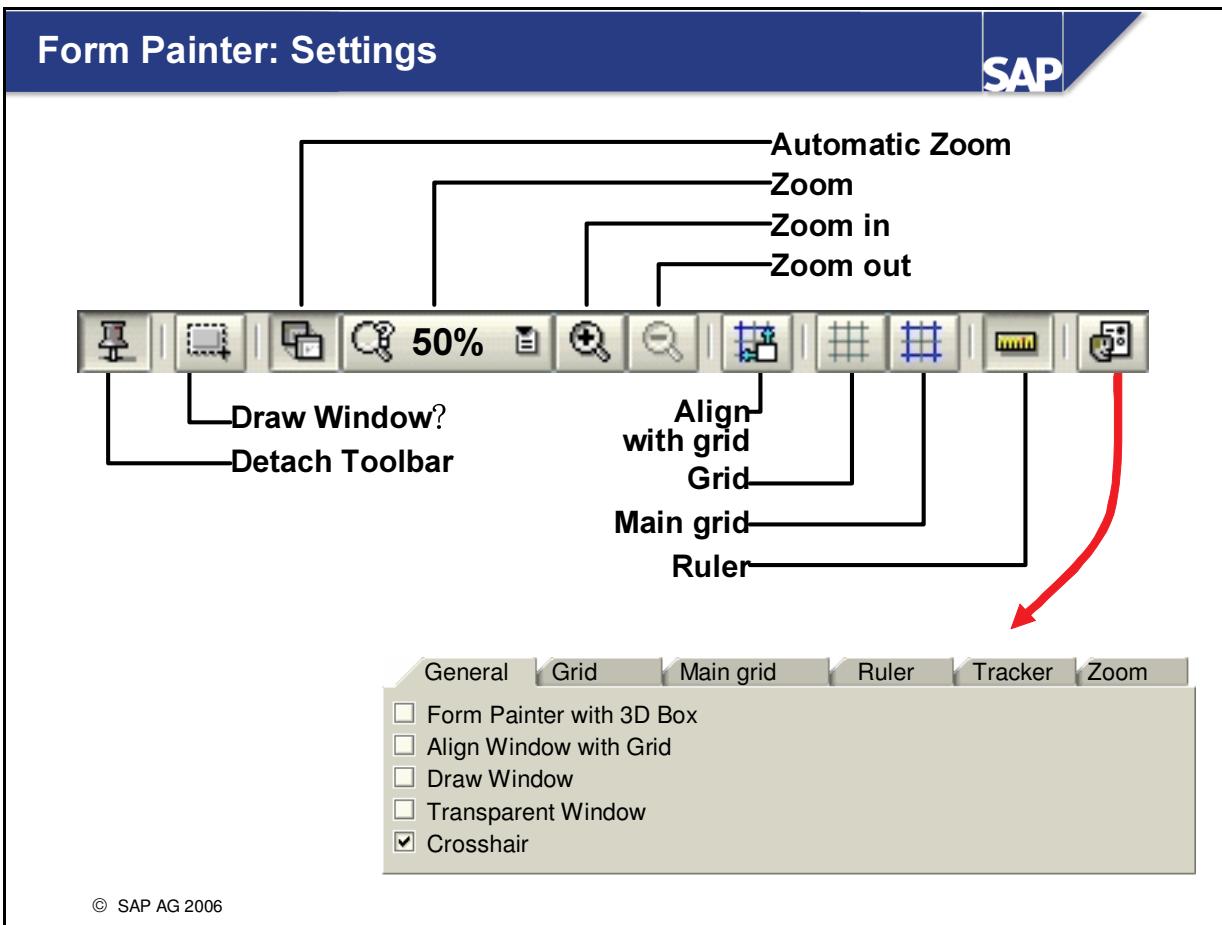
The Form Painter

SAP

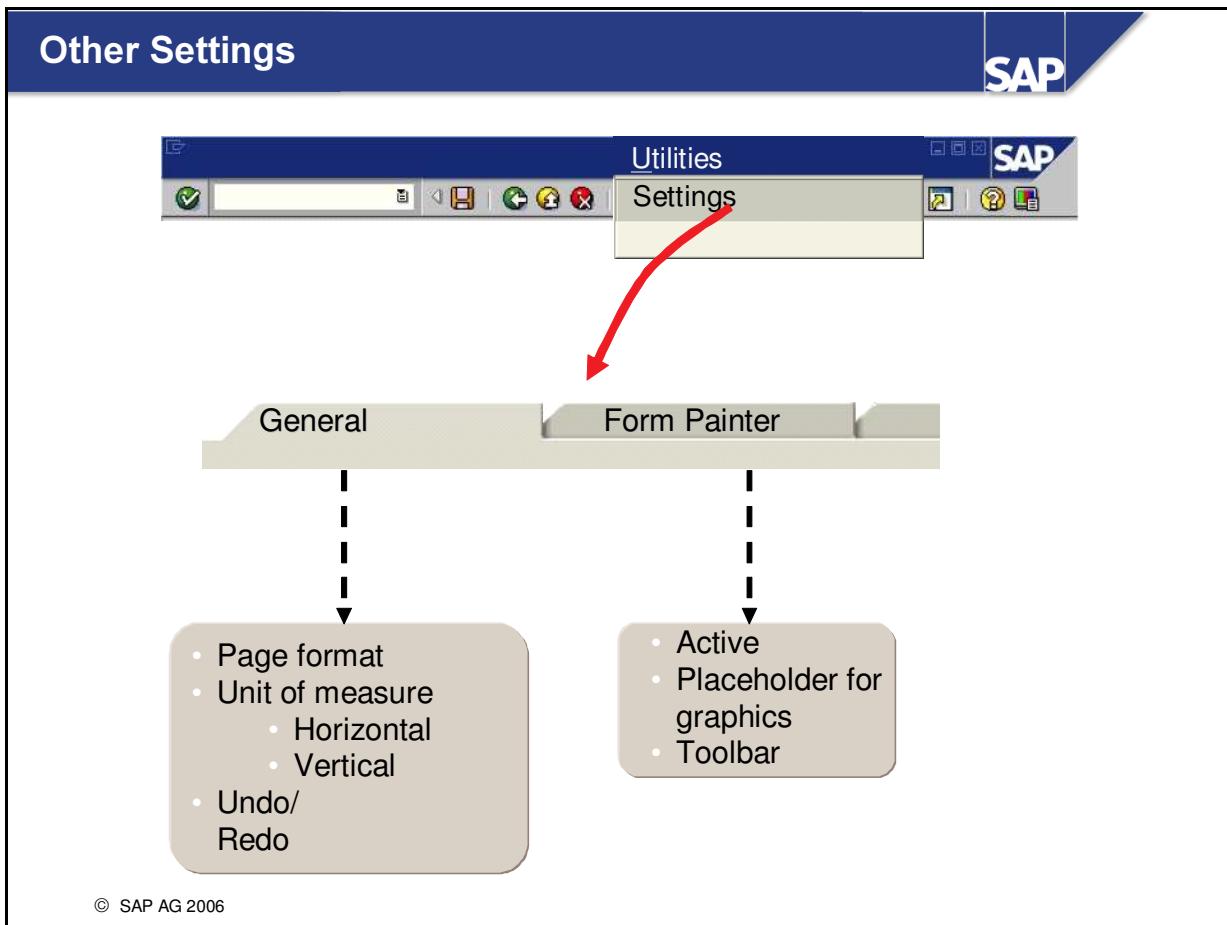


© SAP AG 2006

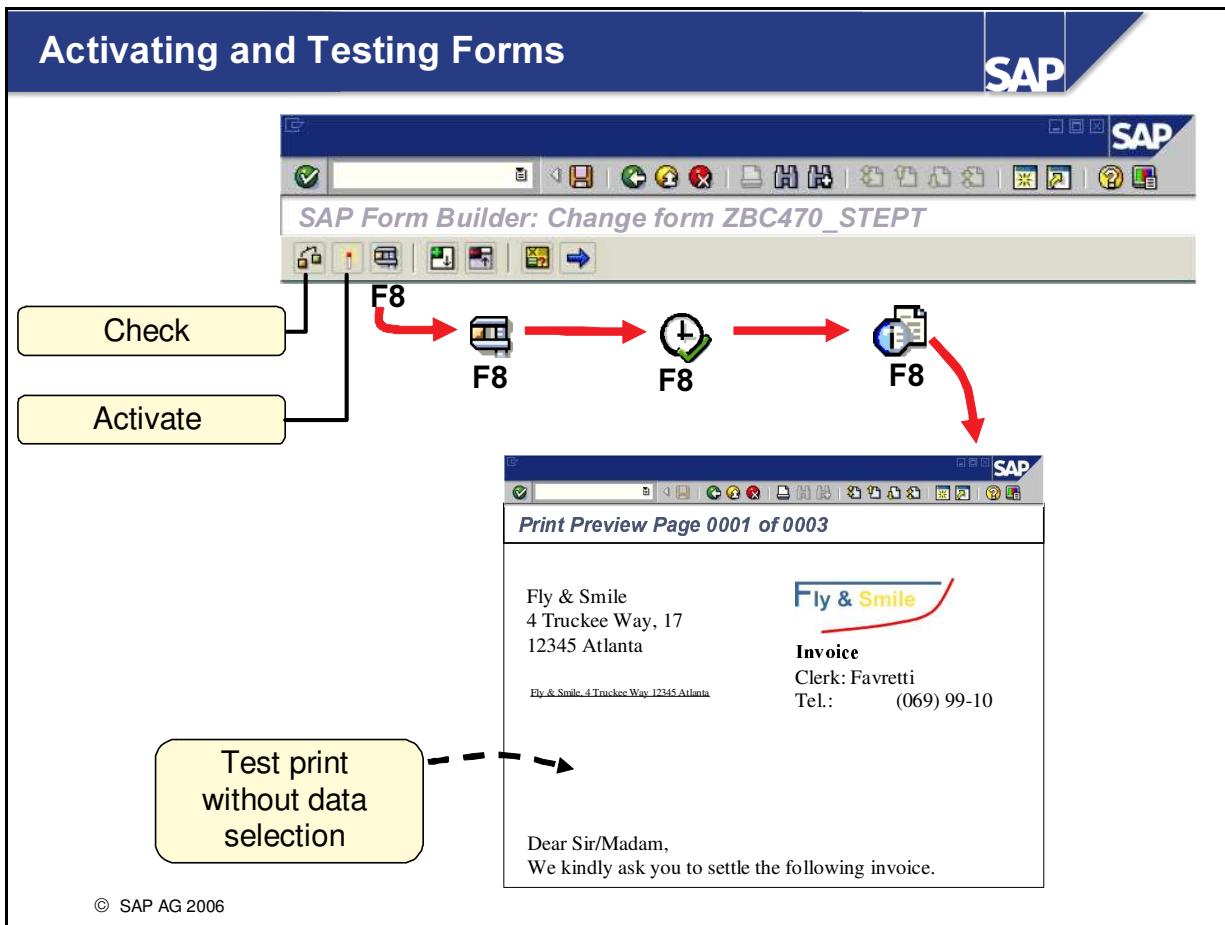
- You use the Form Painter to check/change the layout of a form. You can show or hide the Form Painter in the Form Builder by clicking the corresponding pushbutton or by choosing menu *Utilities→Form Painter on/off* from the menu. The Form Painter always displays the page selected in the navigation tree, including all output areas (windows, graphic windows, and address windows) and the background picture, provided there is one.
- To edit an output area, select it with a mouse click. The corresponding node is then also displayed on the maintenance screen. You can change the size of a window by clicking one of the sizing handles situated at the corners and the sides of the selection rectangle and dragging the handle to its new position while keeping the left mouse button pressed. If you want to reposition an output area, click the area and move it while keeping the left mouse button pressed (drag and drop). All size and position-related changes that you make are automatically copied to the maintenance screen.
- The context menu (right mouse button) is also available in the Form Painter. You can use this menu to create or delete output areas and perform normal clipboard functions (cut, copy, paste). Choose the *Into background* option if a small window is completely hidden by a larger one and you want to edit the small one. Using this option has no effect on the actual print output. (If windows, graphics, or texts overlap, they are printed one by one over another.)
- As of SAP Web AS 6.10, the keyboard can also be used for editing in the Form Painter, for example, Ctrl-C, Ctrl-X, Ctrl-V for temporary storage operations.



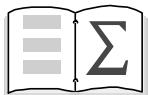
- The detachable toolbar of the Form Painter contains pushbuttons for the most important settings. For other option, choose *Utilities* → *Settings* or click the right pushbutton of the toolbar.
- If the *Draw Window* option is selected, you can draw a window directly by clicking a free area in the Form Painter and dragging the mouse with its button pressed until the window has reached the required size.
- Several zoom options are available to adjust the display. The most comfortable option is the *Automatic zoom*.
- To ensure that your output areas are correctly aligned, you can display a detail grid and/or the main grid. You can also make a setting in the Form Painter that ensures that output areas are automatically aligned with the grid when you move them with the mouse. You can set the step size of both grids. Using a crosshair cursor instead of the normal mouse pointer also helps you to align the nodes correctly. You activate the crosshair cursor on the *General* tab page of the Settings menu.
- The *Tracker* tab page of the Form Painter settings allows you to determine how the window that is currently selected is to be highlighted.



- If you choose *Utilities ® Settings* in the menu, you can make the following settings:
- Tab page *General*
 - Default value for the *page format* of new pages
 - Default values for horizontal and vertical *units of measure*
 - As of SAP Web Application Server 6.10, you can also set whether it is to be possible to undo or redo changes in the Form Builder.
- Tab page *Form Painter*
 - Activate/deactivate the Form Painter (checkbox *active*)
 - Placeholder for graphics* This setting is sometimes useful to increase performance, for example, if your workstation computer is too slow. If you select this checkbox, graphics are represented in the Form Painter simply as frames that are of the same size.
 - Activate/deactivate the *Toolbar* of the Form Painter.



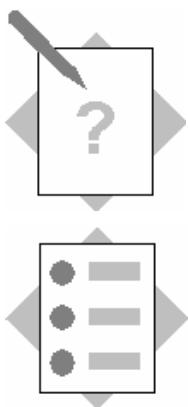
- You should test your form after making changes. You can make local checks at node level or check the entire form.
- Before you can use a form in programs, you have to activate it. Choose *Form → Activate* or click the second pushbutton in the toolbar of the Form Builder. Activating a form means that the entire form is checked and saved and the function module is generated.
- If you have changed and saved your form, but then want to revert to the active form version, choose *Utilities ® Return to active version*. Note that this deletes the inactive form version that you have edited.
- To test your form from the Form Builder (using Print Preview or Print), choose *Form ® Test* or click the third pushbutton of the SAP Form Builder. The system takes you to the Function Builder, which is the development environment for function modules. The name of the function module generated is already entered here. For testing purposes, you can also enter values in the interface of the function module. Then, choose *Test* (Menu: *Function module ® Test ® Single Test*), then *Execute*. In the *Output Device* field of the print attributes, enter your printer in the field and then choose *Print Preview* or *Print*. The quickest way to go to the print preview is to press the F8 function key four times in the SAP Form Builder.
- You can only test inactive forms as of SAP Web Application Server 6.10. For SAP R/3 4.6C, you must activate a form even just to test it.



You are now able to:

- Work with the SAP Form Builder
- Create, copy and edit forms
- Create pages and windows
- Explain the different window types
- Use background pictures
- Set output options
- Test forms

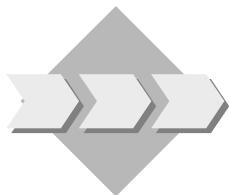
© SAP AG 2006



Unit: First Steps with the SAP Form Builder

After completing these exercises, you will be able to:

- Copy forms
- Navigate in the SAP Form Builder
- Create pages and windows
- Determine output options for forms, pages, and windows
- Check, activate and test forms



Your task:

Copy an existing form and make some changes to that form using the SAP Form Builder.

Copy template for the form:

BC470_STEPT

Package (for all exercises):

ZBC470_##

Name of the form to be created:

ZBC470##_STEPS

Model Solution:

BC470_STEPS

Application program for testing purposes:

SAPBC470_DEMO

1 Get an overview

The best thing to do at the beginning of the exercise is to run the program SAPBC470_DEMO to get an idea of the current layout of the template form BC470_STEPT.

2 Copy template

Copy the template form BC470_STEPT to ZBC470##_STEPS (## is your two-character group number).

3 Set general attributes

Specify that the form should be translated only into English, French and Spanish.

4 Choose page format

Change the page format to DINA4. This requires that you adjust the width of certain windows. Check the form and make any necessary corrections.

5 Set background picture

Use the black and white image BC470_FLY_AND_SMILE_WATERMARK, object GRAPHICS, ID BMAP as the background image for the page FIRST. Set a resolution of 300 dpi.

To ensure that the graphic is not hidden by a window in the Form Painter, you should set the *Transparent windows* option for the Form Painter.

The preview of the Form Painter is not precise. For a more precise display, you must generate the function module.

6 Create new window

On the page FIRST, create a new window with the name INFO and the description *Clerk*. The window should be at the same height as the existing window ADDRESS2. Define a box and shading for the window so that the window is always output even if it does not have any contents.

7 Create additional page

7-1 Create an additional page with the name NEXT.

7-2 NEXT should be the next page of FIRST and of NEXT itself.

7-3 On the NEXT page, create the main window at a height of 25 cm, and also the windows PAGE and FOOTER. Copy all three windows, including all subnodes, from page FIRST to page NEXT.

7-4 **Optional:** Page numbering should start on page NEXT, at II in Roman format.

8 Test form

8-1 Check and activate your form. Test the form by executing the function module that was generated automatically. What is the name of this function module?

8-2 Test your form again using the program SAPBC470_DEMO.



Unit: First Steps with the SAP Form Builder

1 Get an overview

Choose System → Services → Reporting. Enter the name SAPBC470_DEMO, and execute the program (function key F8). Enter BC470_STEPT as the form name, and execute the program (function key F8).

2 Copy template

Start transaction SMARTFORMS by entering the transaction code into the OK code field or by choosing Tools → Form printout → Smart Forms. Enter BC470_STEPT in the *Form* field, and choose the *Copy* icon in the toolbar. Enter ZBC470_##_STEPS as the new form name. The name of the form copied is then automatically transferred into the *Form* field. Choose the *Change* pushbutton. The system then takes you to the SAP Form Builder.

3 Set general attributes

You make these settings on the maintenance screen which forms the middle part of the SAP Form Builder. To set the language attributes, select the *Into selected languages* option in the *Language attributes* group box. Click the icon to the right and select the languages on the dialog box that appears next.

4 Choose page format

The page format is defined centrally for the entire form (with the exception of the orientation, which can be set at page level). From the navigation tree, choose *Global Settings* → *Form Attributes*. Go to the *Output Options* tab and choose the page format DINA4.

Click the scales icon on the left in the toolbar to execute the form check. The system checks the entire form. The error message is displayed in the bottom part of the maintenance screen. The message indicates that the width of the window FOOTER is too large for the new paper format chosen. You can directly go to this window by clicking the name of the node in the error display. Change the width of the window FOOTER in the *Position and Size* group box on the *Output Options* tab. Alternatively, you can also activate the Form Painter (by clicking the pushbutton furthest to the right in the toolbar) and adjust the width of the window using the mouse. To do this, click one of the sizing handles and drag the window border to the desired size while keeping the left mouse button pressed. Check your form again. There should not be any errors.

5 Set background picture

Double-click the page FIRST in the navigation tree. The attributes of that page are then displayed on the maintenance screen. Go to the *Background picture* tab and enter the values specified for the name, the object, and the ID. Select *Black and White Bitmap Image (BMON)* and determine the output attributes for the background picture: 300 dpi, output mode *Print preview and print*, and a vertical and horizontal position of your choice. Update the screen by choosing *Enter*.

6 Create new window

From the context menu (right mouse button) of the page FIRST, choose *Create → Window*. The system displays a window with a default size and a default name in the top left corner of the Form Painter. Move this window with the mouse to the desired position. To ensure that this window is at exactly the same height as the window INFO, you can either choose the *Align with grid* option in the toolbar of the Form Painter, or enter an identical value for both windows in the *Upper margin* field on the *Output options* tab. Change the name of the new window to INFO on the maintenance screen and enter a description. You set the box and shading on the Output Options tab page. Select the *Always draw box and shading* checkbox.

7 Create additional page

7-1 From the context menu of the page FIRST, choose *Create → Page*. Change the name of the new page to NEXT on the maintenance screen and enter a description ("Next page").

7-2 On the *General attributes* tab, enter NEXT as the next page. Repeat these steps for the page FIRST.

From the context menu of the MAIN window, choose *Copy*. This automatically copies the contents of the entire window, including all subnodes, to the clipboard. From the context menu of the page NEXT, choose *Paste*. This inserts the contents of the clipboard, that is the main window, into the page NEXT. Repeat these steps for the windows PAGE and FOOTER. Now change the height of the window MAIN on the page NEXT.

7-4 **Optional:** The page numbers are output in the window PAGE. The simplest way to suppress the output of page numbers on the page FIRST is to delete the entire window PAGE on the page FIRST. (Do not delete the contents of the window PAGE. Since the contents of this window are identical on all pages, the page numbers would then also be lost for the page NEXT.) Go to the *General attributes* tab of the page NEXT and set Roman digits for the page numbers and the mode *Increase counter*.

8. Test form

8-1 Check your form as before using the left-most pushbutton in the toolbar. If the check does not return any errors, you should be able to activate the form. You can find the Activate pushbutton directly to the right of the Check pushbutton. When you activate a form, the system prompts you for a package. Enter ZBC470_##. The name of the function module is unique only in your system. You can find out the name by choosing *Test* or pressing the function key F8. The quickest way to go to the print preview is to press the function F8 another three times. Make sure that the output device entered is the printer your instructor told you.

8-2 See task 1.

If there is too little text in the text node DUMMY_TABLE, the page NEXT is not processed. In this case, you must either enter more text, or reduce the height of the window MAIN on the page FIRST.



BC470 Form Printing with SAP Smart Forms



1 Overview



2 SAP Smart Forms: Overview



3 First Steps with the SAP Form Builder



4 Texts, Addresses, Graphics



5 Data in Forms



6 Tables and Templates



7 Process Control



8 Integration into Application Programs



9 Smart Styles

...

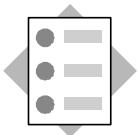
Appendix

© SAP AG 2006

Contents:

- **Texts (text elements, text modules, include texts)**
- **Addresses**
- **Graphics**

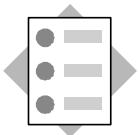
© SAP AG 2006



After completing this unit, you will be able to:

- **Create text nodes**
- **Create addresses**
- **Create graphics**

© SAP AG 2006



After completing this topic, you will be able to:

- **Create text nodes**
- **Decide which text node type (text element, text module, include text) you want to use**
- **Insert fields using the field list**
- **Format fields**

© SAP AG 2006

Text Nodes

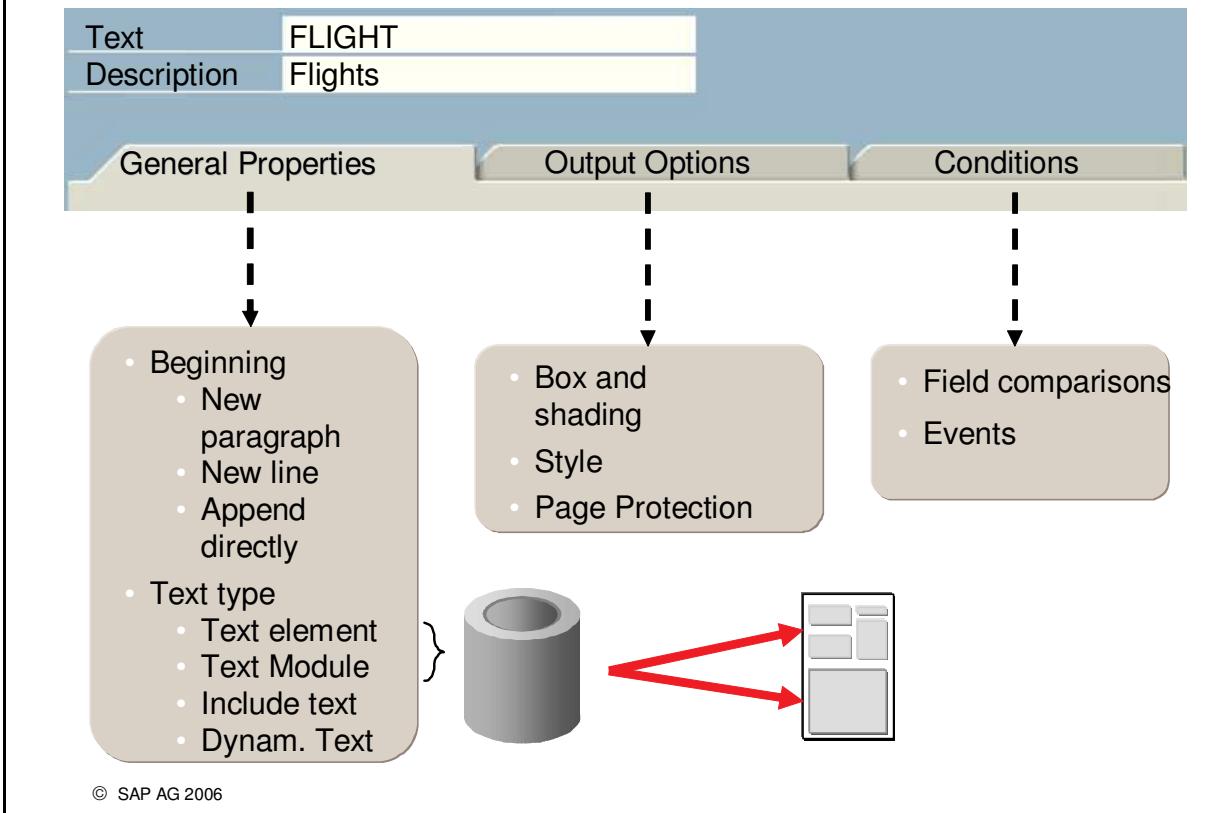
SAP

© SAP AG 2006

- You enter all texts in the form using text nodes (With the following exception: Address nodes.) Text nodes are subnodes of windows and their subnodes, for example folders or templates.
- For existing text nodes, you can use the context menu (right mouse button). You use this, for example, to create a further text node directly afterwards, on the same level. Alternatively, you can create text nodes using the context menu of the superordinate node. If you use this method, the new node is entered as the top subnode of this node.
- The same functions are available in the menu *Edit → Node*.
- Text nodes themselves cannot have subnodes.
- Note that text in secondary windows that no longer fits will be cut off. This is also the case for copy windows and final windows, as of SAP Web Application Server 6.10. On the next page, further processing only occurs in the main window.

Text nodes: Properties

SAP



- If you select a text node in the navigation tree or in the Form Painter, its properties are displayed on the maintenance screen.
- The three tab pages are similar to those for windows.

General Properties:

- The text type you choose allows you to determine whether the text should be saved and edited within the form (as a text element) or outside the form. In the second case, you can choose between a SAP Smart Form text module and an include text of SAPscript.
As of SAP Web Application Server 6.30, there is another text type as well as the dynamic text. This new text type interprets the contents of an internal table, which must have a special format with two columns, as a text.
- You also determine how two directly successive text elements should be combined.
The remaining fields you see on the tab page vary depending on the text type.

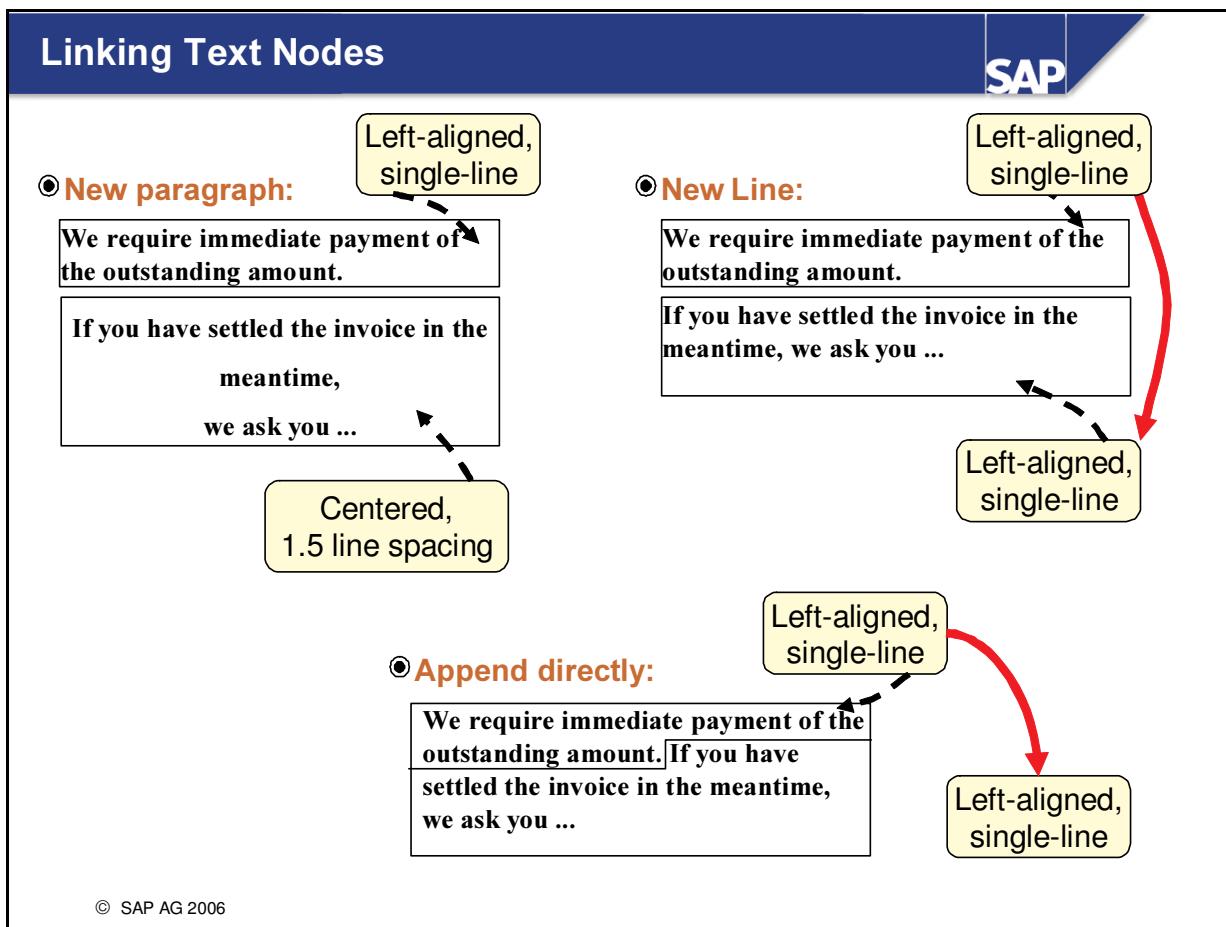
Output options:

The previous information on *boxes and shading* is also applicable to text nodes.

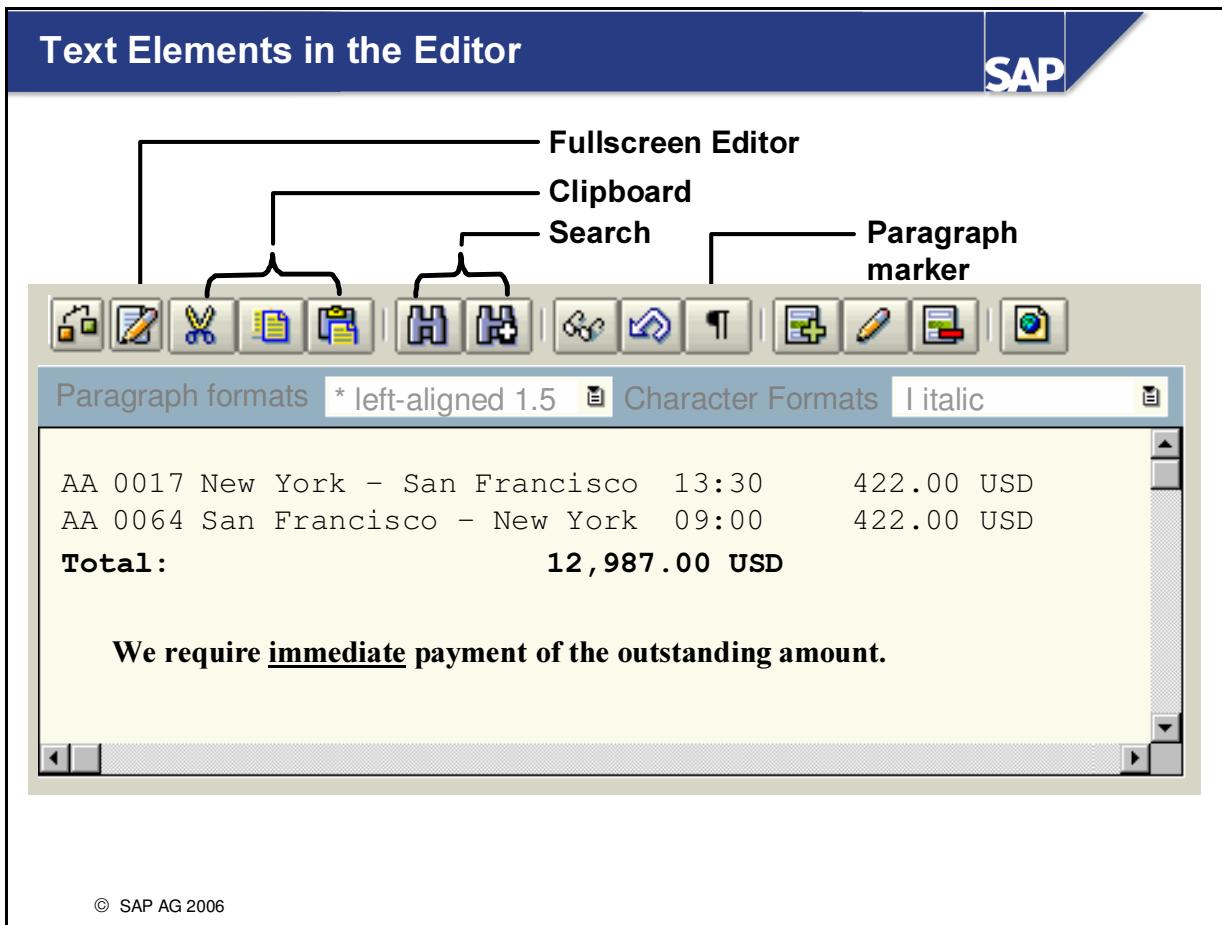
In addition to boxes and shading, you can also *assign a style* to the text node. A style is a collection of different character and paragraph formats. (See Unit 9 – *Smart Styles*.)

If the text node is in the main window, you can choose *Page protection*. This option prevents text from being separated by page breaks. If the protected text does not fit onto the current page, it is output on the next page.

Conditions: See conditions for windows and Unit 7 – *Process Control*.



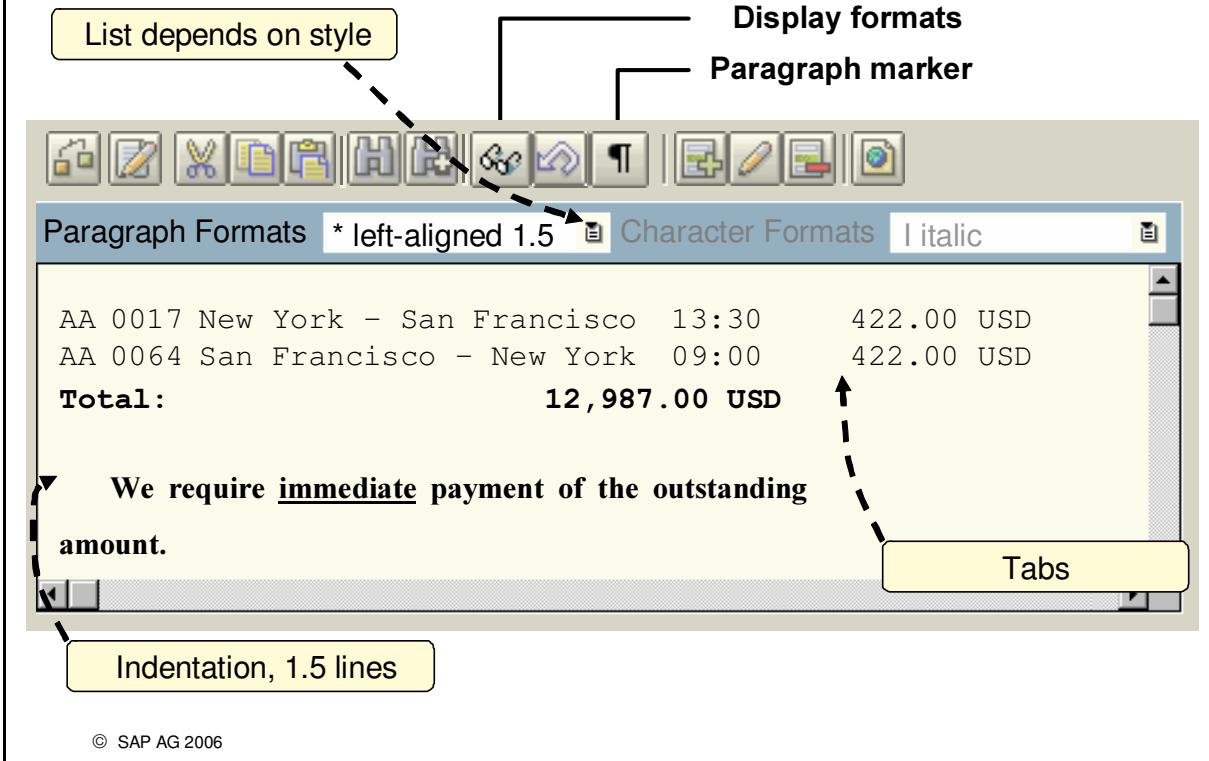
- On the *General Properties* tab page, you determine how two successive text nodes are combined in the same window:
 - If you choose *New paragraph*, the text of the second node begins in a new line based on the paragraph format that you specified for this paragraph. This means that the two text nodes are completely independent of each other (In **one** text node, choosing *Return* creates a new paragraph).
 - If you select *New line*, the text of the second node also begins in a new line. However, the format of the last paragraph of the first text node is used for the first paragraph of the second text. (In **one** text node, choosing *Return* creates a new line.)
 - You can also choose *Append directly*. In this case, two successive text nodes are combined without blanks or blank lines. The resulting paragraph is assigned the format of the first text element.
- If the first node has a box and/or shading, the second node is always appended using the *New paragraph* option, irrespective of which option you select.



- If you choose *Text element* as the text type, the inline editor is displayed on the *General Properties* tab page. You can enter text in the same way as you would in any common word-processing system. Alternatively, you can also switch to fullscreen mode by choosing *Text editor*. In Unicode systems, you can use Microsoft Word as the text editor in fullscreen mode. Read SAP Note 742662 for more details.
- You can use the clipboard by selecting text blocks with the mouse and then choosing *Cut*, *Copy*, or *Paste*. This way you can copy text sections between different windows or forms.
- Lines in text nodes are broken automatically depending on the window width. You can also use the *Enter* key in the editor to create a new paragraph that may then have a different format from the preceding one. *Shift-Enter* allows you to create a line break within a paragraph.
- The *Paragraph mark on/off* pushbutton allows you to determine whether you want to display nonprinting characters (blanks, tabulators, paragraph marks, line breaks)..

Formatting Texts: Paragraph Formats

SAP



- You can format selected text sections. They are then displayed as they appear in print (WYSIWYG = What You See Is What You Get).
- For each paragraph, you can choose a paragraph format from the selection list in the editor. A paragraph format is a collection of format settings, such as tabs, type of justification, and so on (see Unit 9 - Smart Styles). The paragraph formats available for selection in the list depend on the style you have chosen. If you have entered a style for more than one node (for example, for the form attributes and for a table), the following applies:
 - The style of a node overrides the style of the form attributes.
 - The style of a lower-level node overrides the style of a higher-level node. For text nodes, this means that the style of the text node is used first; if there is none, the style of the next higher node is used, and so on.
- If you do not choose a paragraph format, the standard paragraph format of the style is used.
- In the list of the paragraph format and of the character format, the format set in the current cursor position is displayed automatically. You therefore only need the *Display formats* function if you want to obtain detailed information on the format, or if text has been formatted using several character formats.

Formatting Texts: Character formats

SAP

Show paragraph markers

Reset Character formats

Display formats

List depends on style

The screenshot shows the SAP Fiori interface for 'Formatting Texts: Character formats'. At the top, there are three buttons: 'Show paragraph markers', 'Reset Character formats', and 'Display formats'. A callout bubble points to 'Display formats' with the text 'List depends on style'. Below these are several icons representing different document types and styles. A toolbar below the icons includes 'Paragraph formats' and 'Character Formats' tabs. The 'Character Formats' tab is selected, showing the 'Italic' option. The main area displays flight information and a total amount: AA 0017 New York - San Francisco 13:30 422.00 USD, AA 0064 San Francisco - New York 09:00 422.00 USD, Total: 12,987.00 USD. A red arrow points from the 'List depends on style' callout to the note 'We require immediate payment of the outstanding amount.' at the bottom.

Paragraph formats * left-aligned 1.5 Character Formats I italic

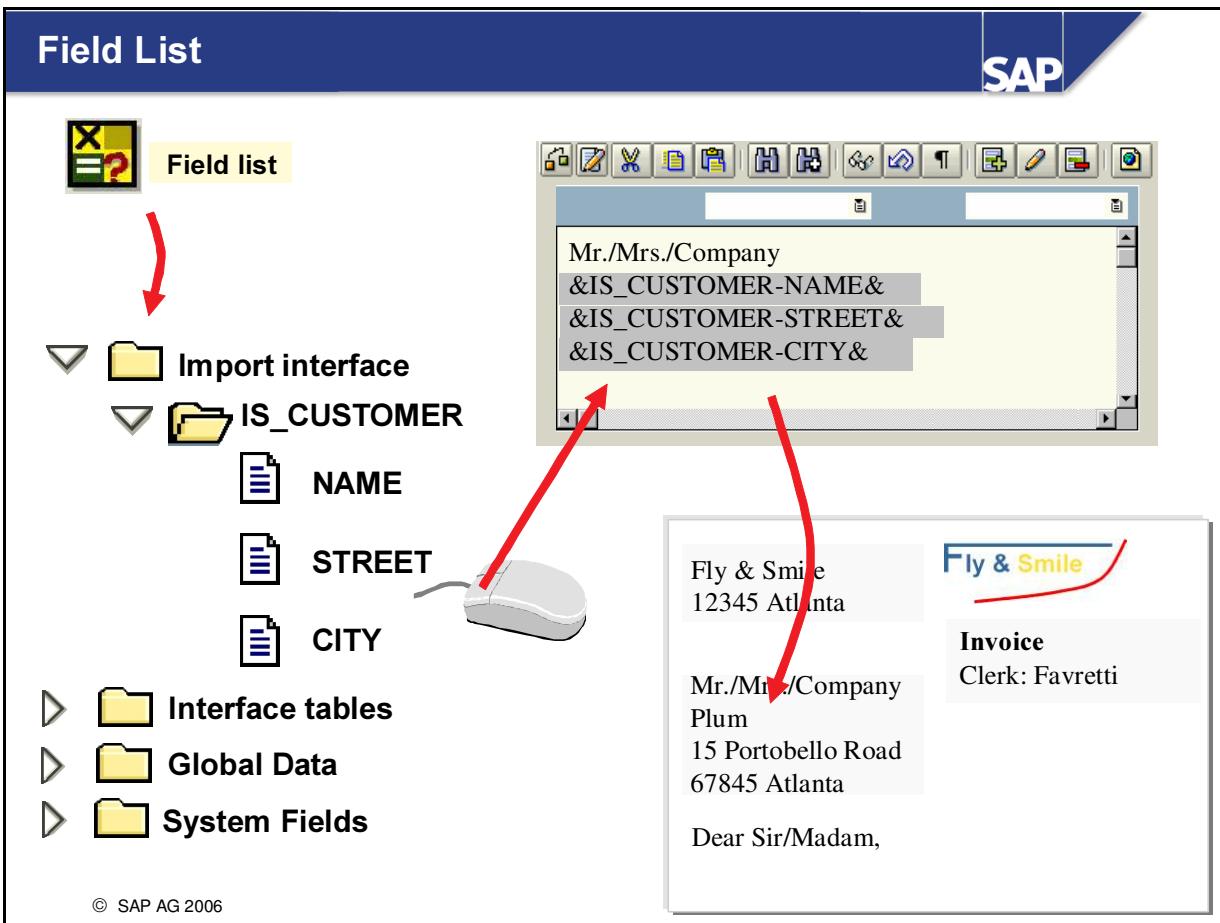
AA 0017 New York - San Francisco 13:30 422.00 USD
AA 0064 San Francisco - New York 09:00 422.00 USD

Total: 12,987.00 USD

We require immediate payment of the outstanding amount.

© SAP AG 2006

- You can also assign one or more character formats to selected text. A character format is a collection of format settings such as *font type* or *superscript*. The same is true for the character formats available as for the paragraph formats. The *Reset character formats* pushbutton resets all character formats for the selected text to the formats of the paragraph format.
- If you want to know which paragraph formats and character formats are valid at a specific cursor position, click the *Display formats* pushbutton. The system displays a dialog box containing the formats. For more information about a specific paragraph format or character format, double-click the respective format.



- Normally, your form contains not only static text but also variable data, referred to as fields, which are read from the database at application runtime or are entered by the user.
- The simplest way to insert fields is to use the field list. You can show or hide the field list by choosing the corresponding pushbutton, or in the menu by choosing *Utilities ® Field list on/off*.
- The following types of field are available for selection:
 - All fields that are recognized by the form as import, export or table parameters, through the form interface (which means they come from the application program)
 - All global data and field symbols you have created in the form in the global definitions
 - System fields that are filled automatically during program execution (for more information, see the next slide).
- Use the drag and drop function to integrate the fields: Drag a field name from the field list to the required position in the text element.
- If a field is structured, click the triangle icon to the left of the corresponding folder to access the individual subfields.

System Fields

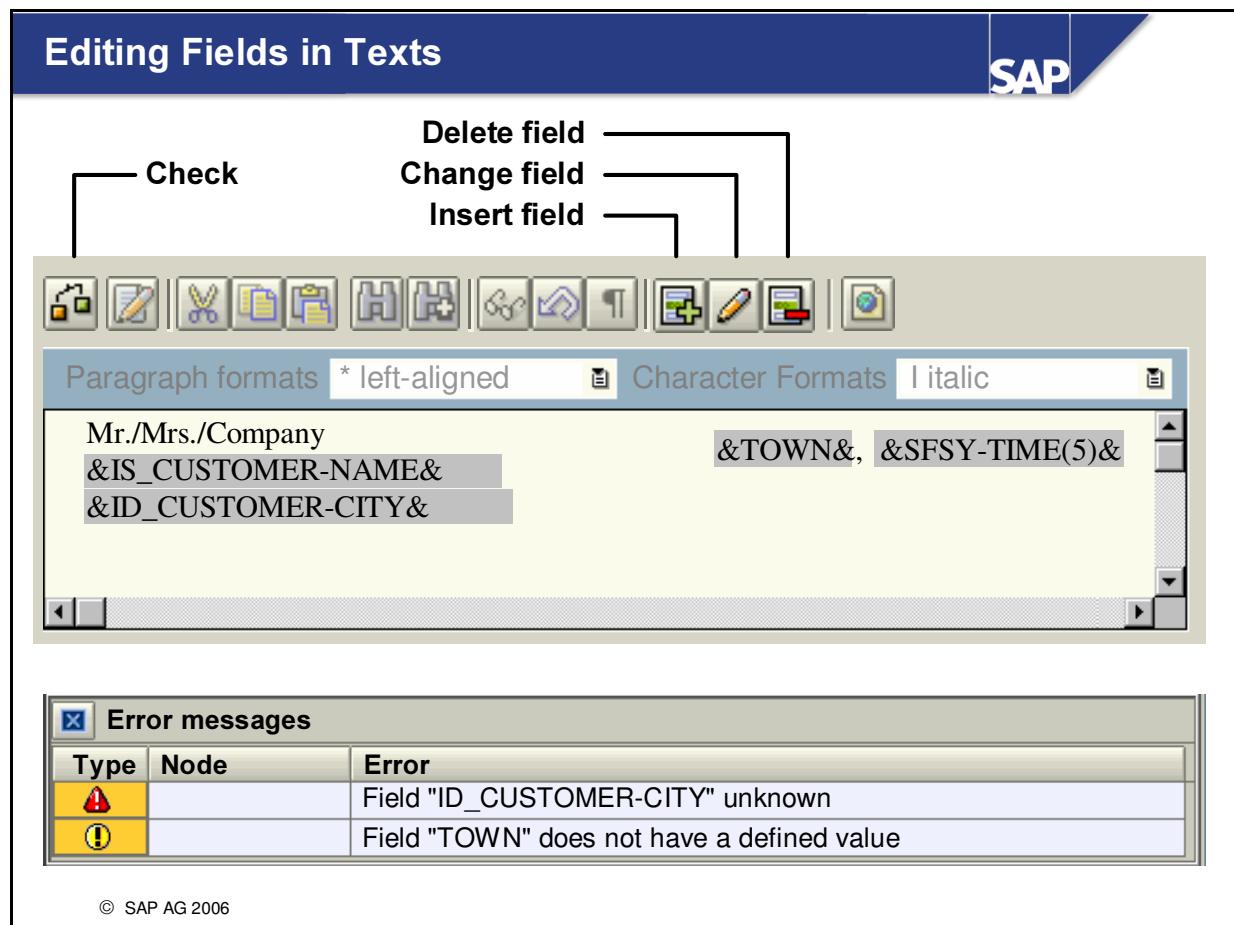
&SFSY-DATE&	Date
&SFSY-TIME&	Time
&SFSY-PAGE&	Number of current print page
&SFSY-FORMPAGES&	Total number of pages in document currently being processed
&SFSY-JOBPAGES&	Total number of pages of all documents in current print request
&SFSY-WINDOWNAME&	Name of current window
&SFSY-PAGENAME&	Name of current page

As of SAP Web Application Server 6.10:

&SFSY-XSF&	Indicator for XSF output
&SFSY-COPYCOUNT0&	Copy counter (0 = original, 1 = 1st copy)
&SFSY-COPYCOUNT&	Copy counter (1 = original, 2 = 1st copy)
&SFSY-USERNAME&	User name

© SAP AG 2006

- DATE: Date display. The display format is set in the user master record.
- TIME: The time is displayed in the form HH:MM:SS (HH: Hours, MM: Minutes, SS: Seconds).
- Page numbers:
PAGE: Number of current print page. You determine the format of the page number (for example, Arabic, or numeric) and the mode (increase, initialize, leave unchanged) on the *General Properties* tab page of the page node.
- FORMPAGES: Total page number for document currently being processed. You can print the page number in the form 'Page x of y', for example.
- JOBPAGES: Total number of pages of all documents in the current print request.
- WINDOWNAME: Name of current window
- PAGENAME: Name of current page
- As of SAP Web Application Server 6.10:
XSF: This indicator is set if the form is output in XSF format or HTML format.
COPYCOUNT and COPYCOUNT0: Query whether the original or the copy is output.
SUBRC: 0 if text module or include text exists, 4 if not.
USERNAME: Logon name of the user who is printing the form.



- You can also use the *Insert field* pushbutton to add fields to your text element. Enter the name of the field enclosed in ampersands (&). This also allows you to access the ABAP system fields of the structure SYST, such as &sy-uname& (user name). Field names are not case-sensitive.
- The fields are grayed out to distinguish them from normal text. They cannot be directly changed or deleted. To delete a field, select it and choose *Delete field*. To change a field, place your cursor on the field and choose *Change field*. You need this function, for example, to determine formatting options such as the output length for a field.
- To ensure that your field entries are correct, perform a check by clicking the corresponding pushbutton in the editor. The system notifies you of any errors that may exist (by means of an exclamation mark in a red triangle). If errors exist, you cannot activate the form.
- Only if you use the check function of the Form Builder (first pushbutton in the toolbar), does the system check if all fields used have been assigned a value when they are processed or if they are still initial. If the fields are initial, the system issues a warning (exclamation mark in a yellow circle) but you can still activate the form. Fields which are initial at application program runtime will be ignored. For more information, see the online documentation.

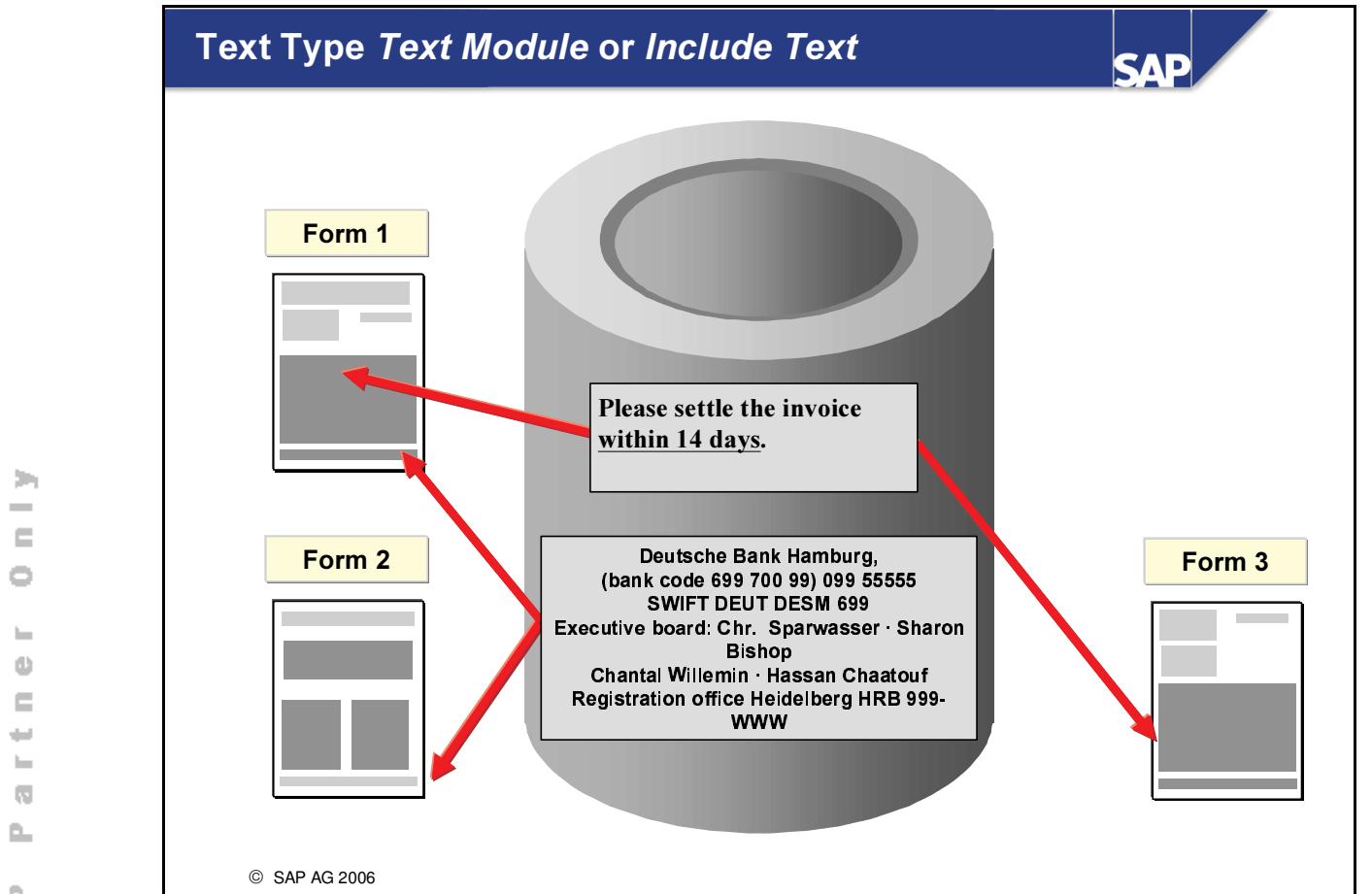
Formatting options

&WA_NAME&
SAP Smart Forms &WA_NAME+4 (5) &
Smart

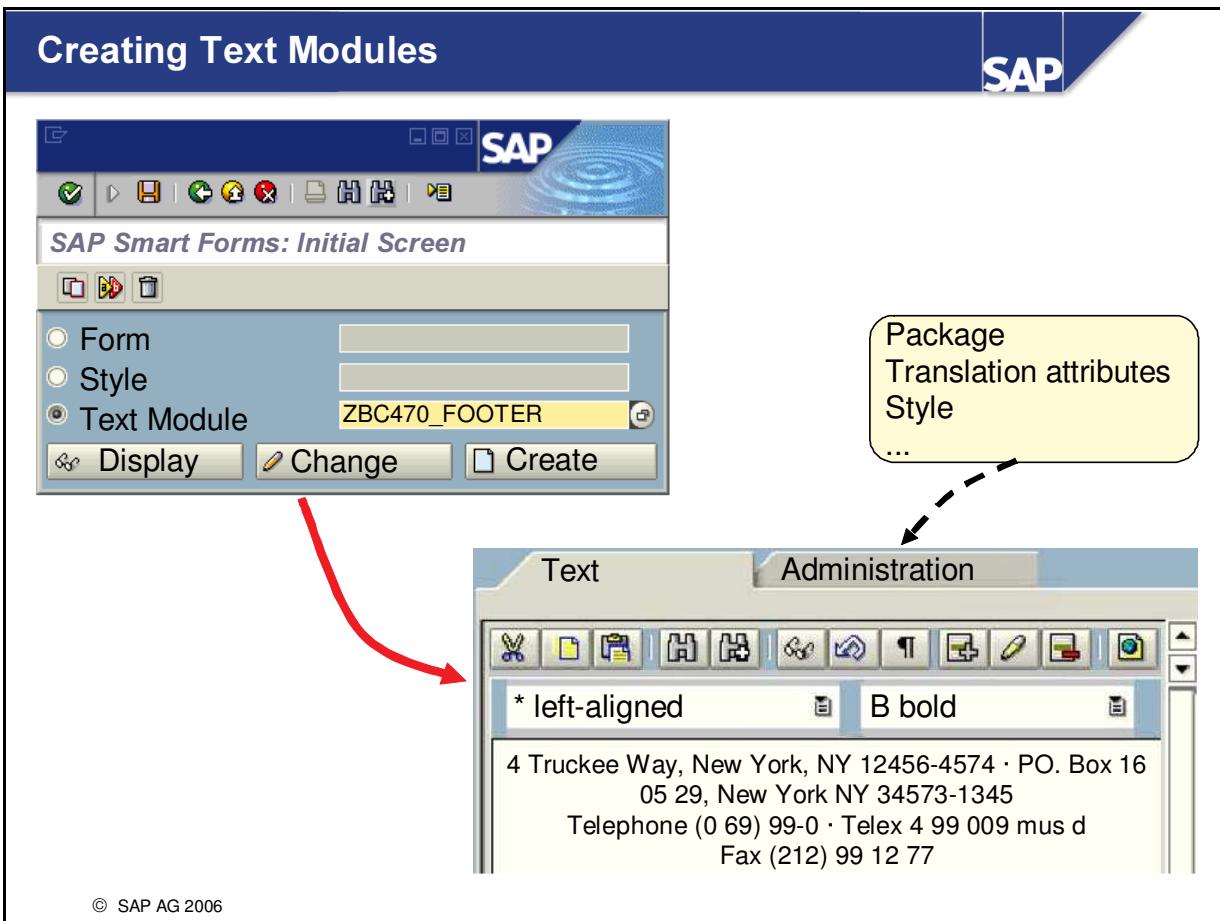
&field+6&	Offset (only for character fields; here: 6)
&field(9) &	Output length (here: 9)
&field(S) &	Suppress +/- signs
&field(<) &	Display +/- signs to the left of the number
&field(8.2) &	Decimal form (8 places, 2 after dec. point)
&field(T) &	Suppress thousands separator
&field(Z) &	Suppress leading zeros of numbers
&field(I) &	Suppress output of initial values
&field(R) &	Right-justified (only in combination with output length)
&field(F<Füllzeichen>) &	Replace left-justified blanks with fillers

© SAP AG 2006

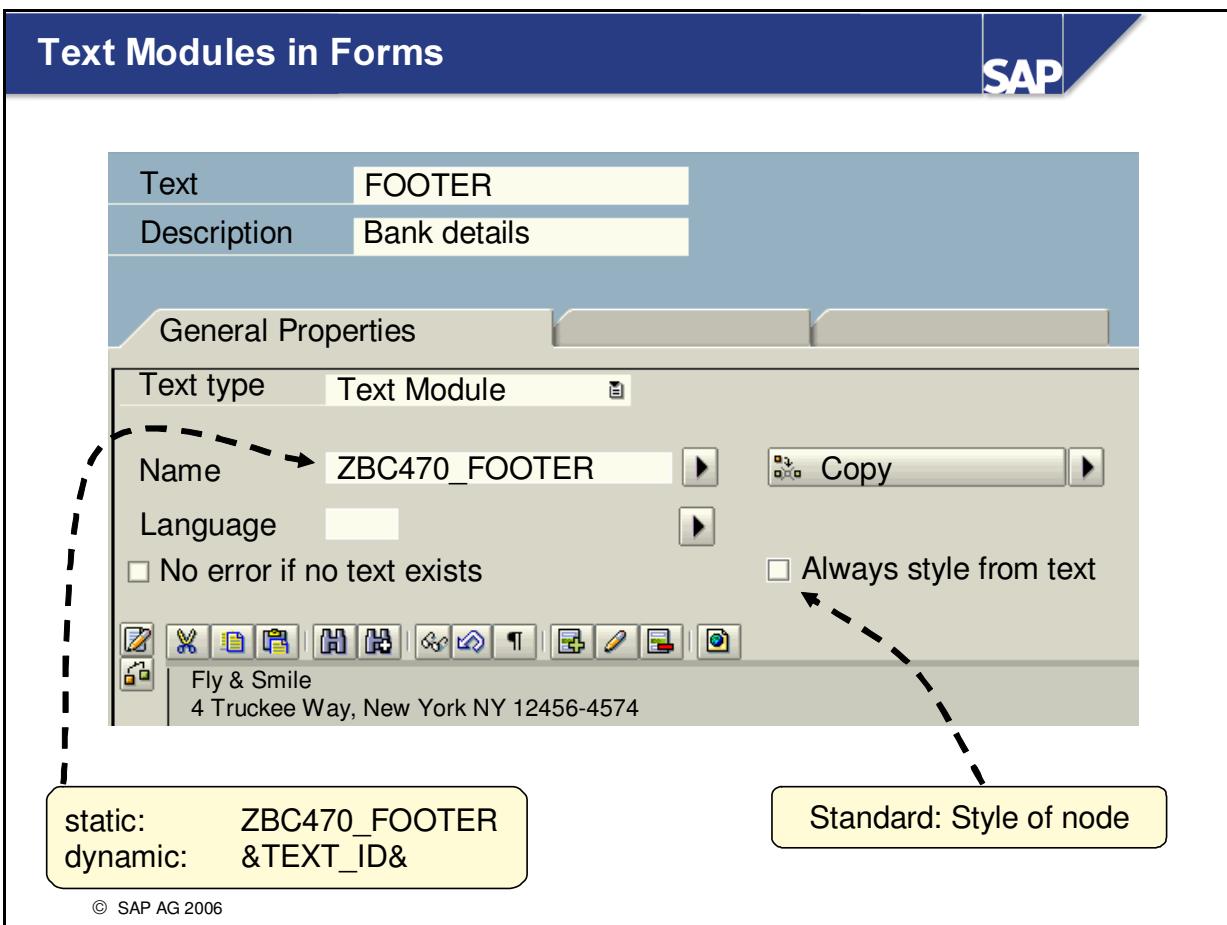
- Formatting options allow you to adjust the value of a field before it is output. You enter the corresponding shortcuts directly after the field name and **always in uppercase**. Some of the options can be combined, for example +6(9) or (8.2).
- The formatting options are not appropriate for all data types of a field (for example, you cannot display numbers with an offset). We distinguish between character and numeric fields:
- Character fields:
 - By default, the value of a field name is displayed in full. However, blank spaces at the end of the value are cut off.
 - Evaluation sequence: Suppress blanks (C), <offset> and (<length>), right-justified display (R), insert filler (F).
- Numeric fields:
 - The closing blank is interpreted as a plus sign. To suppress it, use the formatting option S.
 - Evaluation sequence: (<length>), +/- sign to the left (<), Japanese date (L), suppress blanks (C), right-justified display (R), insert filler (F).
- For information on further formatting options, see the online documentation.



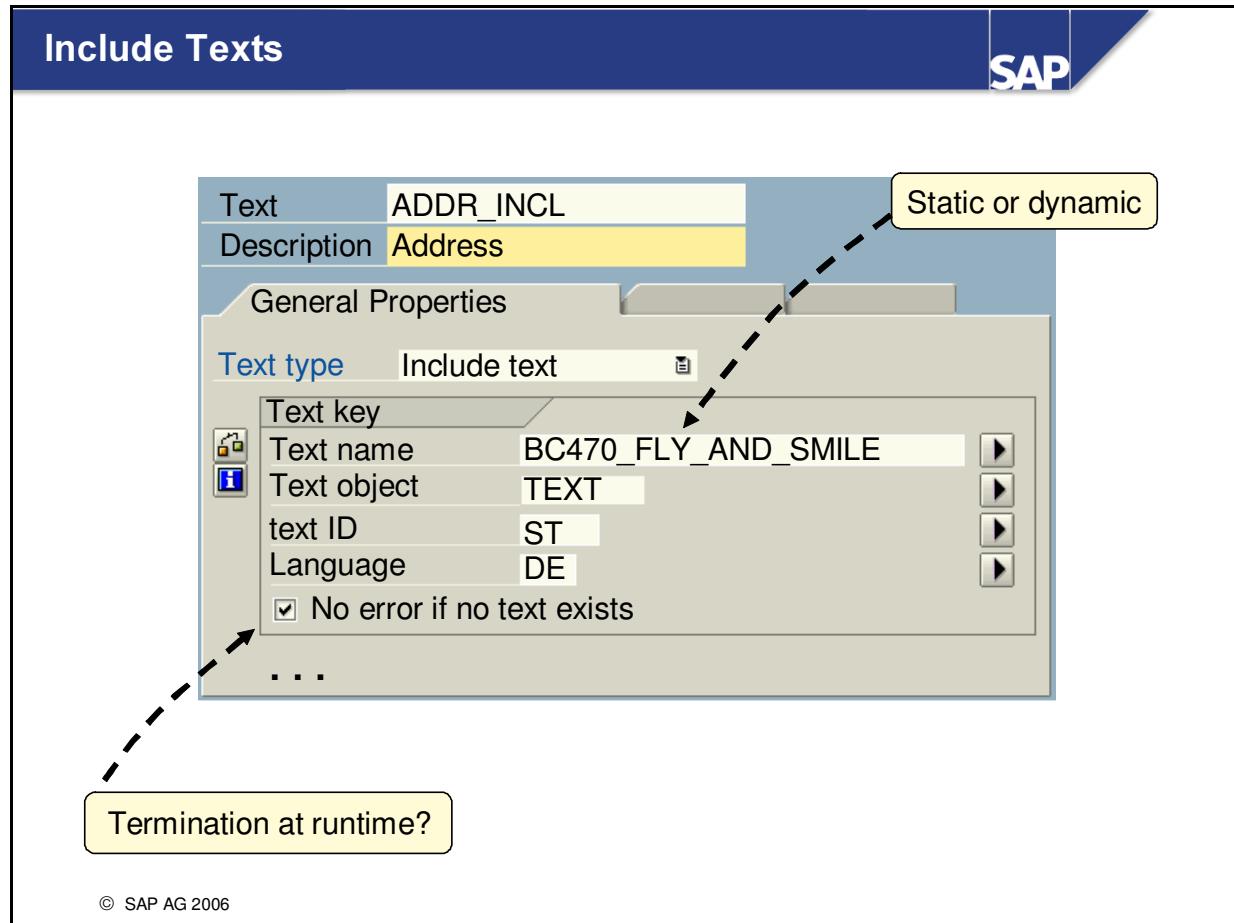
- In some cases, it makes sense to store texts not in the form itself, but centrally in the database. The form then contains only a reference indicating which text should be used at application program runtime. This method has the following advantages:
 - You need to create the texts only once and can then reuse them as required.
 - You make changes centrally only once without having to modify the actual forms. (The reference in the form remains unchanged.) Your bank details change.
- You can use text modules of SAP Smart Forms as well as include texts as you can use them in SAPscript. You set the text type on the *General Properties* tab of the text node. If you change the text type, the system displays a warning, since text already entered would be lost.
- The names of the paragraph and character formats used are saved in the text module or include text. In the form, however, they are interpreted as they are defined in the style of the text node.



- You go to the maintenance transaction for text modules by selecting the *Text module* radio button on the SAP Smart Forms initial screen and then choosing *Display*, *Change* or *Create* , depending on what you want to do. From this screen, you can also copy, rename or delete existing text modules. To do this, choose the appropriate pushbutton from the toolbar, or in the menu choose *Text module*.
- The name of a text module that you create must be in the customer namespace and therefore begin with Y or Z.
- Since text modules are integrated with the SAP transport system (like SAP Smart Forms), you must assign them to a package. You do this when you first save your text module. In SAP R/3 4.6C, subsequent changes are not possible.
- You work with the text module editor in the same way as you work with the inline editor of the form maintenance screen. However, neither the field list nor the check function is available since text modules can be inserted into various SAP Smart Forms with different interfaces. You can insert, delete and edit fields using the corresponding pushbuttons. However, note that the fields must be defined and filled in all forms that use the text module.
- You must assign a style to each text module on the *Administration* tab page. The default style is *System*

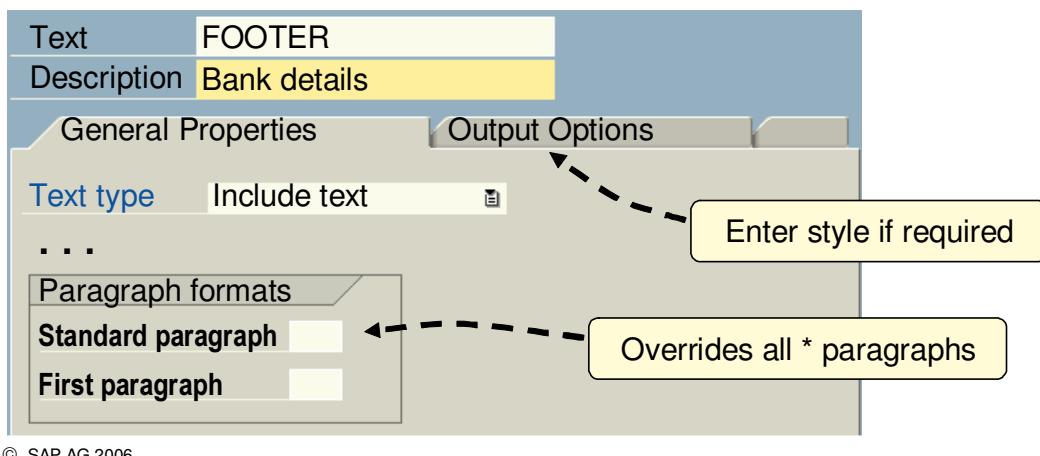


- To insert a text module into a form, create a text node, choose the *Text module* text type and enter a name.
- You can also determine the name of the text module dynamically at application program runtime. To do this, enter the name of a field recognized in the form, enclosed in ampersands (&). If the space is not sufficient to enter the full name, click *Return* or the triangle to the right of the *Copy* pushbutton.
- As of SAP Web AS 6.10, you can determine the language of the text module (statically or dynamically). In SAP R/3 4.6C, the text module is always entered in the same language as the form.
- The text of a text module is always grayed out in the inline editor of the Form Builder because you cannot change it directly from within form maintenance.
- If you determine the text module dynamically, its text cannot be displayed in the editor. Likewise, you cannot check from within form maintenance whether the field is filled with an appropriate value at application program runtime. In SAP R/3 4.6C, the current output request is terminated if the text module that you determined dynamically is not found when the program is executed. As of SAP Web Application Server 6.10, there is an extra field: *No error if no text exists*. You can check this field to prevent a termination occurring at runtime. SFSY-SUBRC contains the value 4 if a text module is not found. Otherwise it contains 0.
- If you choose the *Copy* pushbutton, the text of the text module is inserted into the form as a text element. Afterwards, there is no link to the text module.
- By default, the system uses the style of the text node or of a higher-level node. For a text module, however, you can also choose *Always copy style from text module*.



- You can also insert existing SAPscript texts. To do this, choose *Include text* as the text type of the text node and enter the necessary selection information in the *Text key* input area. You can also make these specifications dynamically (name of a field, enclosed in ampersands (&)).
- If you select *No error if no text exists*, the function module generated does not terminate if the text specified is not found at runtime. As of SAP Web Application Server 6.10, there is the system variable SFSY-SUBRC. You can query a value for this variable in a program line node. If the text is found, SFSY-SUBRC has the value 0. If an error occurs, it has the value 4.
- If you create new texts for use in SAP Smart Forms, you should always create SAP Smart Forms text modules rather than SAPscript texts. The reasons for this are:
 - You cannot maintain or preview SAPscript texts from SAP Smart Forms. As before, you can use transaction SO10 to maintain standard texts.
 - You cannot check whether fields of an inserted SAPscript text are actually defined in the form.
 - SAPscript texts are not automatically integrated with the transport system.
 - SAPscript texts are client-specific.
- Note that SAP Smart Forms ignore all SAPscript commands in SAPscript texts. If required, you must convert these commands into SAP Smart Forms logic.

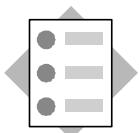
- Formats in include texts are interpreted in the same way as they are defined in the Smart Style of the node.
- Unknown formats in include texts are ignored.
- SAPscript styles in include texts are ignored.
- It is possible to override formats in SAPscript texts:



- All paragraph formats and character formats of an include text are interpreted as they are defined in the style (Smart Style) that applies for the text node. You can determine this style on the *Output Options* tab page of the text node. If you do not determine a specific style, the style of a superordinate node is used (for example, the style of a table) or the style that you determined in the form attributes.
Beispiel: Das Absatzformat B in einem SAPscript-Text bedeutet *Fett*, im Smart Style aber *Klein*. Each paragraph with format B in the text is therefore printed in small text.
- You can override the paragraph formatting of include texts:
In the *Standard Paragraph* field, you can select a paragraph format of the Smart Style that applies for the current node. This format is then used for all paragraphs of the SAPscript text that are formatted using the standard paragraph (*). If you are familiar with SAPscript commands: An entry in this field corresponds to the PARAGRAPH addition of the INCLUDE command.
You use the *First Paragraph* attribute to set a paragraph format for the first paragraph of the include text - regardless of how the paragraph is actually formatted in the include text (corresponds to the NEW PARAGRAPH addition of the SAPscript command INCLUDE). If the *Standard Paragraph* field remains empty, all standard paragraphs in the include text also adopt this paragraph format.

Smart Forms Text Module	SAPscript Text
Client <u>independent</u>	Client <u>dependent</u>
Package assignment → there is an automatic transport via the transport request	No package assignment → you must use report RSTXTRAN to manually assign the text to the transport request.
	Standard paragraph and first paragraph can be overridden.
	Old SAPscript commands may be misleading
Assigned styles may be used or overridden	Assigned styles are ignored

© SAP AG 2006



After completing this topic, you will be able to:

- **Create addresses from Business Address Services as separate windows or as subnodes of a window**

© SAP AG 2006

Addresses: Business Address Services

SAP

▼ FIRST First page

ADDRESS2 Customer address

Example 1:



Barbara McCloskey
74 Court Oak Road
Harborne
Birmingham B17 9TN
UNITED KINGDOM



- Business Address Services
- Formatted, country-specific, format



Peter Dennebaum
Alfred-Mumbächer-Str. 28a
55128 Mainz

Example 2:

Ms.
Karin Kottenhoff
Geschäftsleitung
Röderwiese 10
58093 Hagen

Karin Kottenhoff
Röderwiese 10
58093 Hagen

- Instead of using their own tables for address information, many applications now access the Business Address Services (BAS, which was known as CAM prior to SAP Web Application Server 6.10.) In the Business Address Services, addresses are identified by means of numbers.
- SAP Smart Forms allow you to use the BAS. You do not need to know the technical details of BAS, or worry about the correct formatting of the addresses. The addresses are formatted in accordance with country-specific conventions (based on ISO 11180 and the guidelines of the Universal Postal Union). If the space in the form is not sufficient, some fields may not be displayed. For detailed information, see the documentation on the function module ADDRESS_INTO_PRINTFORM.
- You normally create addresses as direct subnodes of a page (that is, as address windows). You can then position them as required in the Form Painter. Alternatively, you can also create an address node as the subnode of a (main or secondary) window, a loop, or a template.
- If you want to output formatted addresses without using the Business Address Services, you must call the ADDRESS_INTO_PRINTFORM function module in a program line node.

Addresses: Properties I

General Properties

- Company Address (1)
- Personal Address (2)
- Workplace Address (3)
- Determine dynamically

&ADRS_TYPE&

Address Number &ADRS_NO&
Person Number &PERS_NO&

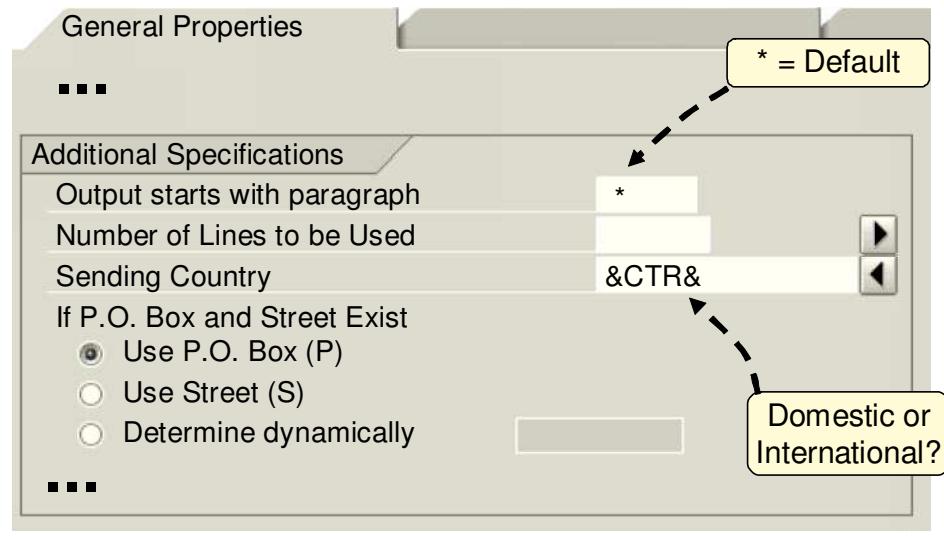
...

Only for types 2 and 3

© SAP AG 2006

- You make the basic settings for the address node on the *General Properties* tab page
- Address type:
 - Company addresses:* Typical examples are delivery addresses or company codes. These addresses are uniquely identified by their address number.
 - Personal addresses:* Addresses of this type are assigned to one natural person, along with other associated attributes, such as the form of address. Since a person can have more than one address, enter both the address number and the person number for identification.
 - Workplace addresses:* These are personal addresses in companies, which means they have additional attributes, such as the department or the room number. You identify such an address by means of the address number and the person number.
 - Dynamic:* If you want to determine the address type at runtime, enter the name of the field (enclosed in ampersands) that must be filled at runtime with 1, 2 or 3.
- You can also determine the address and the person number dynamically. If the length of the input field is not sufficient, click on the small triangle to the right of the field to make it larger. If the system does not find an address with the numbers specified in Business Address Services at runtime, the generated function module of the form terminates with an error message.

Addresses: Properties II



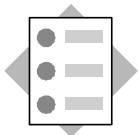
© SAP AG 2006

- In the *Output Starts with Paragraph* field, you set the paragraph format for the address node. This paragraph format must be defined in the style of the address node or, if you have not defined one in the output options, in the style of a higher-level node or the form. If you enter an asterisk, the system uses the default paragraph of the style.
- If, in the *Number of Lines to be Used* field, you set fewer lines than are required to output the full address, the Business Address Services suppresses address parts of reduced importance, such as the form of address or the function of the person in the company. The same applies if the window area you reserve for the address node is too small.
- To determine whether the address is a domestic address or a foreign address, you should always enter the ID of the sending country (if required, dynamically using a field name). In this case, the address country or its respective ID is printed in international addresses, but not in domestic ones.
- For addresses that have both a P.O. box number and a street address, use the radio buttons in the group box to determine which one to use.

The screenshot shows the SAP 'Addresses: Properties II' dialog box. The 'General Properties' tab is selected. In the 'Additional Specifications' section, there are two input fields: 'Fixed Language for Country Indicator' and 'Different Recipient Language', each with a right-pointing arrow button. Below these are two checkboxes: 'Country Indicator in Recipient Language' and 'Uppercase/Lowercase Spelling'. At the bottom is a 'Priority of Lines' field with a right-pointing arrow button.

© SAP AG 2006

- As of the SAP Web Application Server 6.10, the lower section of the *General Properties* tab page contains an additional range of parameters.
- In the *Priority of Lines* field, you can determine which part of the address is first suppressed if there is not enough space for the address node. For example, A stands for form of address, P for an obligatory empty line, and D for department. For more information, see the documentation for function module ADDRESS_INTO_PRINTFORM.
- Like other nodes, addresses have the *Output options* tab page where you can determine the style, the border and the shading. Also on this tab (as in the Form Painter), you can set the values for the position and size of the output area. Prerequisite for this is that you have created the address node as a separate address window, that is, as a direct subnode of a page.



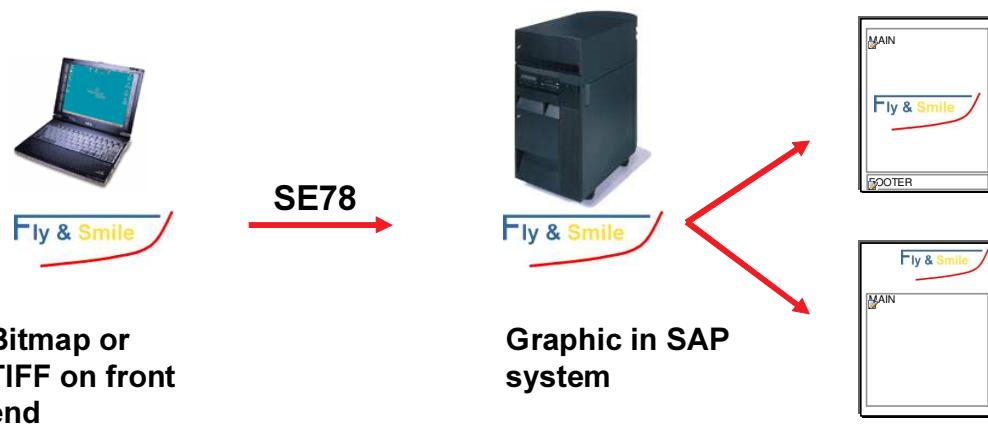
After completing this topic, you will be able to:

- Import graphics into the SAP system
- Create graphics as separate windows or as subnodes of a window

© SAP AG 2006

Addendum: Graphic Administration I

SAP

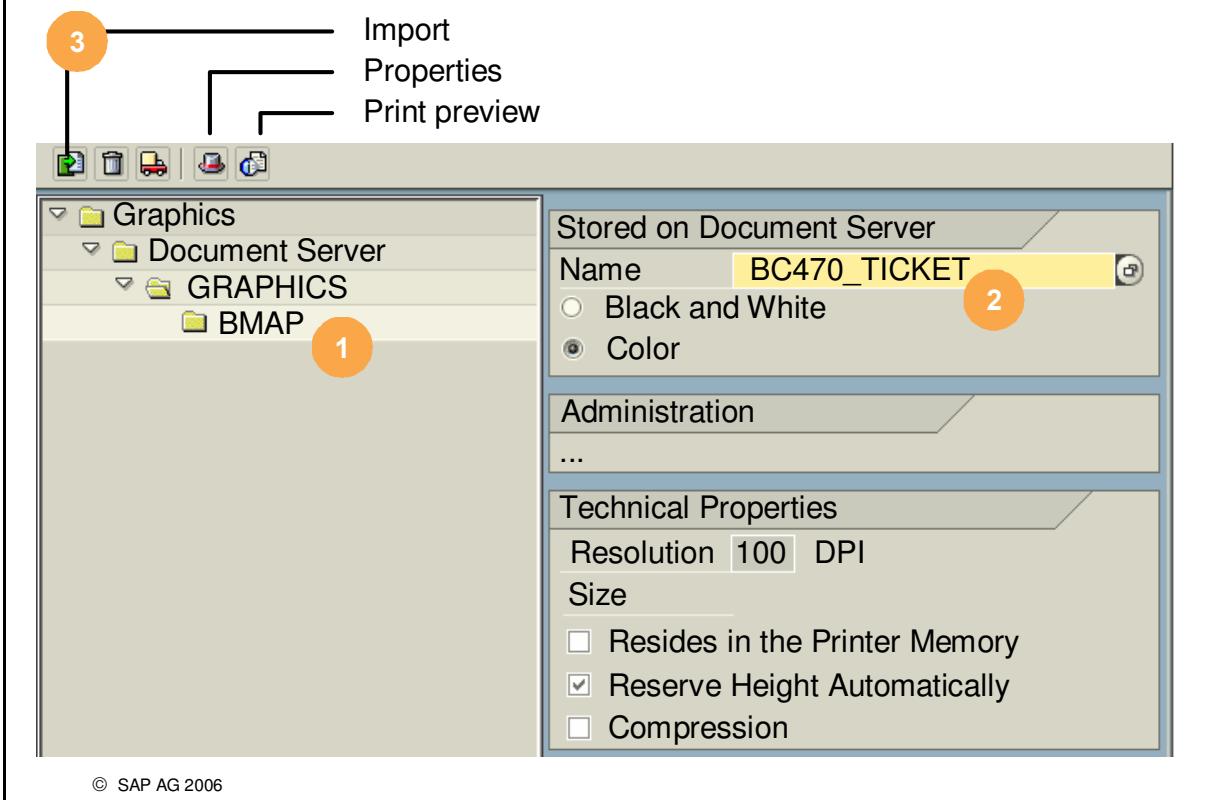


© SAP AG 2006

- Inserting a graphic, either as the background picture of a page or as graphic node, requires that the graphic has been imported from your front-end computer into the system using graphics administration. Graphics that you can use for SAP Smart Forms are stored in the Business Document Server.
- You can use transaction SE78 to display the graphics administration.
- You can import graphics in TIFF or bitmap format.
- A graphic that has been imported once can be integrated in any number of forms in the system.

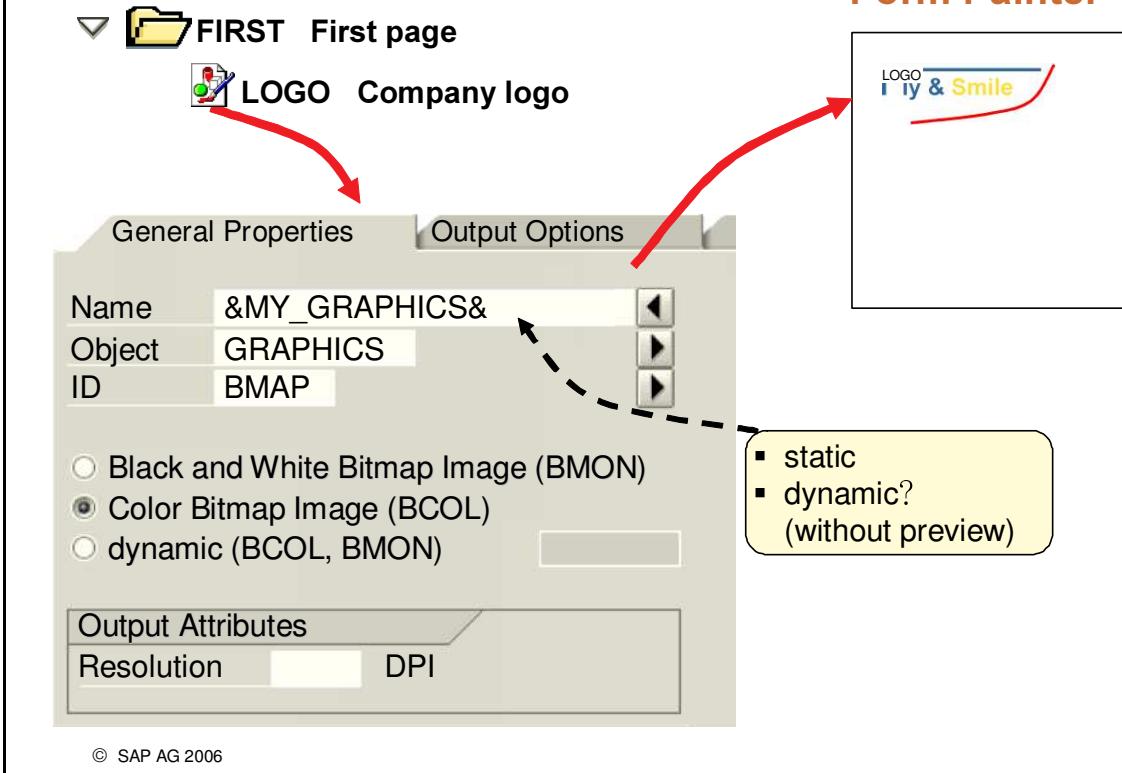
Addendum: Graphic Administration II

SAP



- The storage paths in the system are displayed on the left of the screen in transaction SE78. For SAP Smart Forms you can use only the *Stored on Document Server* option. You can determine which subfolders are displayed using transaction SE75.
- On the right of the screen, information is displayed for the picture or a preview is displayed (after you have selected a screen or after successful import).
- To transfer a graphic to the document server, follow the steps below:
 1. In the left screen area, select a subfolder.
 2. In the field on the right, enter a name for the graphic, and decide whether you want black and white, or color.
 3. Then click the left pushbutton in the toolbar. In the dialog box that appears, enter a description and determine whether the graphic should reside in the printer memory. If you select this checkbox, the graphic is stored in the printer memory during printout when it is used for the first time, so that it can be retrieved from there if it is needed again in the same print request. This may increase performance considerably. As of SAP Web AS 6.10, it is possible to compress graphics.
 4. Start the import by clicking the *Enter Key*.
- Once you have imported a graphic, you can delete it using the trash can pushbutton, or you can add it to a transport request using the truck pushbutton. You can also display the graphic properties or the graphic itself in the print preview on the right side of the screen.

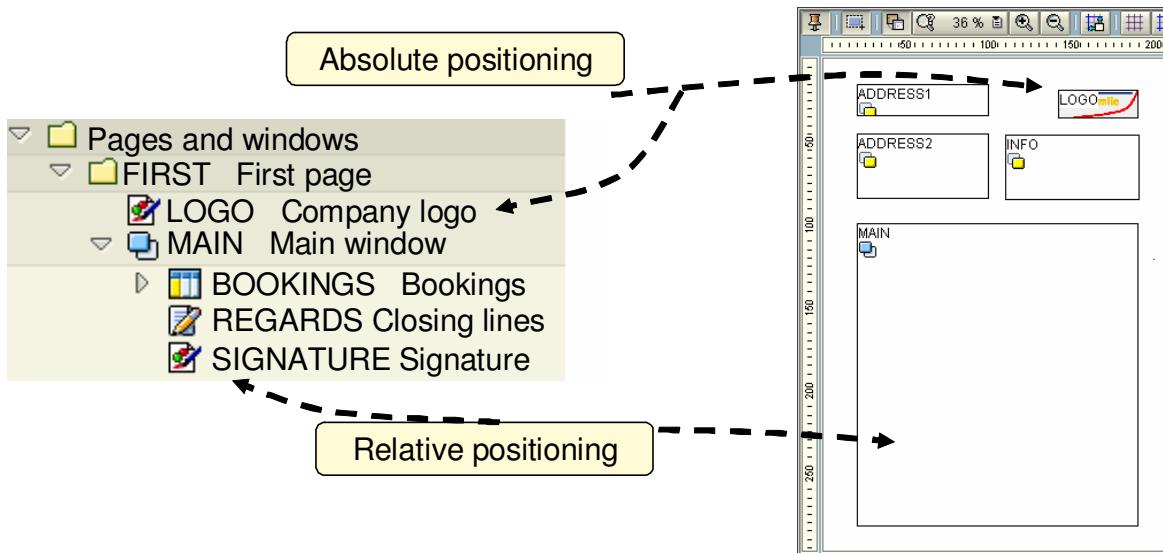
Form Painter



- You can embed graphics not only as background pictures, but also as separate graphic windows (direct subnodes of a page), or as subnodes of a window. A prerequisite for this is that the graphic already exists in the system.
- You can create a graphic node in the same way as other nodes: using the context menu (right-click the mouse to access the context menu) of the navigation tree or by choosing *Edit ® Node ® Create* from the menu.
- You make the settings for the name of the graphic, the object and ID, the type of the graphic (black and white or color) and the resolution in the same way as for a background picture. In particular, you can determine the name of the graphic dynamically at runtime by entering a field.
- If you do not create the graphic in a separate graphic window of the form but as a subnode, you must also determine its horizontal position on the *Output Options* tab page.

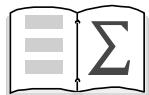
Position of Graphics

SAP



© SAP AG 2006

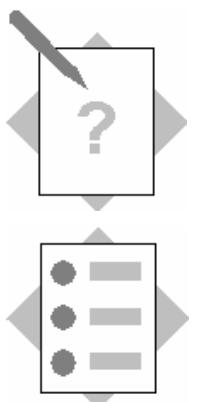
- If you create a separate graphic window for the graphic, the graphic is visible in the Form Painter (provided you have not selected the *Placeholder for graphics* checkbox in the Form Painter settings). You can then easily position the graphic on the page using drag and drop. The position depends on other elements of the page.
- If you create the graphic as the subnode of an existing node (for example, of a window or a template), it is not displayed in the Form Painter. You cannot position the graphic using drag and drop either. Its output depends on superordinate nodes.
An example is the scanned signature SIGNATURE shown above – as a result of the positioning, it is output after the closing lines depending on the contents of the table BOOKINGS, for example on the first or third document page.



You are now able to:

- **Create text nodes**
- **Create addresses**
- **Create graphics**

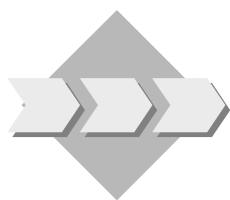
© SAP AG 2006



Unit: Texts, Addresses, Graphics

After completing these exercises, you will be able to:

- Create text nodes with different text types
- Use the field list
- Embed graphics



Your task: Enhance your previous invoice form by inserting text and your company logo.

Copy template for the form:

BC470_STEPS

Package (for all exercises):

ZBC470_##

Name of the form to be created:

ZBC470##_TEXTS

Model Solution:

BC470_TEXTS

Application program for testing purposes: SAPBC470_DEMO

Perform the optional tasks only if you have time left at the end of the exercise.

1. Copy template

Copy your form from the last exercise (ZBC470##_STEPS) to ZBC470##_TEXTS (where ## is your two-digit group number). Alternatively, copy the copy template (BC470_STEPS).

2. Enter information about the clerk

In the window INFO (on page FIRST), create a text node called INFO_TEXT (text type *Text element*) and output some information about the clerk: The clerk's name, telephone number and so on.

3. Write the start and the end of the letter

In the main window, create the following text nodes before or after DUMMY_TABLE (text type *Text element*):

- INTRODUCTION – Write a short introduction for your letter.
- REGARDS – The closing lines

4. Use field list for customer address

The field IS_CUSTOMER, which is filled by the application program, contains all the important information about the customer.

In the text node CUSTOMER_ADDRESS (which is in the window ADDRESS2), only the name of the customer is currently output. Use the field list to output the following fields of the import parameter IS_CUSTOMER: FORM (form of address), STREET, POSTCODE, REGION (if required), CITY. (Do not worry about formatting the address in accordance with ISO standards at this point).

5. Create a text module.

5-1 Create the text module ZBC470_##_FOOTER (preferably in a second session). Assign the style BC470. Enter the bank details of the travel agency in a small font size in the text module.

5-2 Save the text in your package ZBC470_##.

5-3 Embed this text module in the FOOTER window of the form.

6. Company logo

Insert the color graphic BC470_FLY_AND_SMILE, object GRAPHICS, ID BMAP in a separate graphic window in the top right corner of the page FIRST.

7. **Optional:** Include the include text for the address line of the envelope window

Insert the SAPscript text BC470_FLY_AND_SMILE, text object TEXT, text ID ADRS as a new text node in the window ADDRESS2. (This is a small text for the transparent window of the envelope.) Call this text node ADDR_INCL.

8. **Optional:** Determine page numbers

Output the current page and the total page number in the form "Page X of Y" in the window PAGE on the page NEXT.

9. Test the result.

Activate your form. Test your form using the program SAPBC470_DEMO. If required, modify your form so that the page NEXT is also processed.



Unit: Texts, Addresses, Graphics

1. Copy template
See exercise of the previous unit.
2. Enter information about the clerk

From the context menu of the window INFO, choose *Create → Text*. On the maintenance screen, change the name to INFO_TEXT and enter a description. In the editor (that is displayed in the lower part of the maintenance screen when you select the text node for processing), enter some text, such as:

Invoice

Clerk: Mrs. Favretti

and so on

To ensure that the text is not printed touching the frame of window INFO, you can set a margin of 0.1 cm, for example, on the 'Output Options' tab page of the window.

- 3 Write the start and the end of the letter

From the context menu of the main window, choose *Create → Text*. On the maintenance screen, change the name to INTRODUCTION and enter a description. In the editor, enter some text: "Dear ..."

To ensure that the other text node in the tree appears after the node DUMMY_TABLE, use the context menu of DUMMY_TABLE to create the node REGARDS. Alternatively, you can also move the nodes to the desired position using drag and drop (which means you keep the left mouse button pressed while moving the nodes).

Use the editor to enter text for REGARDS:"Yours faithfully,

Felix Müller"

- 4 Use field list for customer address

From the context menu of the window ADDRESS2, choose *Create → Text*. On the maintenance screen, change the name and enter a description.

Show the field list in the bottom left corner of the screen, for example by choosing *Utilities → Field list on/off*.

Expand the *Import interface* folder by clicking the small triangle to its left with the mouse. Use the same procedure to open the structure IS_CUSTOMER and move the required fields to the editor using drag and drop. Add some paragraphs.

5 Create a text module.

5-1 Open a new session and start the SAP Smart Forms maintenance transaction. Select the *Text module* radio button, enter ZBC470_##_FOOTER, and choose *Create*. The system displays the editor for the new text module. Enter the bank details. The default style SYSTEM, which is initially assigned to each new text module, does not use small font sizes. You must therefore assign another style, that is, BC470. You do this on the *Management* tab page. If you return to the editor, (tab page *Text*), you can select FT (footer) as the paragraph format.

5-2 Save your text module by clicking the disk icon. Enter your package ZBC470_##.

5-3 Now, return to the other session and choose *Create* → *Text* from the context menu of the window FOOTER.

On the maintenance screen, change the name to FOOTER_TEXT and enter a description. On the *General Properties* tab page of FOOTER_TEXT, change the text type to *Text module*. Confirm the change. The *General Properties* tab changes so that you can enter ZBC470_##_FOOTER in the field *Text name*.

Press *Return* - this should display the text module in the editor. You cannot directly modify the text module from within the SAP Form Builder.

6 Company logo

From the context menu of the page FIRST, choose *Create* → *Graphic*. Enter a name and a description. On the *General Properties* tab of the graphic window created, enter the name, the object, and the ID. Select *Color Bitmap Image (BCOL)*. Press *Return* to update the preview in the Form Painter. Then move the graphic with the mouse to the top right corner.

7 **Optional:** Include the include text for the address line of the envelope window

From the context menu of the window ADDRESS2, choose *Create* → *Text*. On the maintenance screen, change the name to read ADDR_INCL and enter a description. On the *General Properties* tab of ADDR_INCL, change the text type to *Include text*. Confirm the change. The *General Properties* tab changes so that you can enter the text name, text object and text ID in the *Text key* group box. Note that you cannot preview include texts from within the SAP Form Builder.

8 **Optional:** Determine page numbers

Go to the page NEXT. In the window PAGE, go to the text node PAGE. From the field list (see task 3) choose *System fields* → *SFSY*. Then drag the field PAGE into the editor of the text node. Change the text to "Page ... of".

9 Test the result.

Activate your form by choosing *Form* → *Activate* from the menu.

To test your form, choose *System* → *Services* → *Reporting*. Enter the name SAPBC470_DEMO, and execute the program (function key F8). On the selection screen, enter the name of your form and execute the program (function key F8).

If there is too little text in the text node DUMMY_TABLE, the page NEXT is not processed. In this case, you must either enter more text, or reduce the height of the window MAIN on the page FIRST.



BC470 Form Printing with SAP Smart Forms



1 Overview



2 SAP Smart Forms: Overview



3 First Steps with the SAP Form Builder



4 Texts, Addresses, Graphics



&WA&

5 Data in Forms



6 Tables and Templates



7 Process Control



8 Integration into Application Programs



9 Smart Styles

...

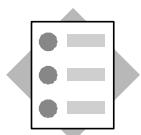
Appendix

© SAP AG 2006

Contents:

- **Integrating data into forms that is known only at runtime**

© SAP AG 2006

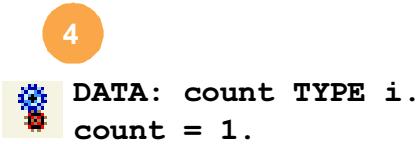
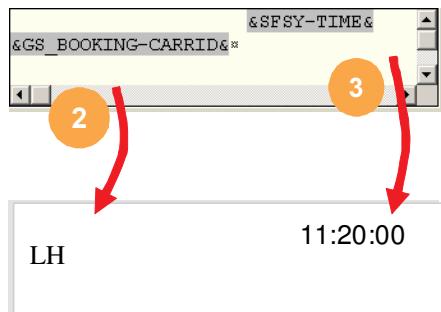
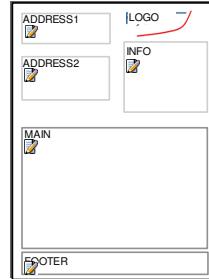


After completing this unit, you will be able to:

- Define the interface of a form
- Create global data and types

© SAP AG 2006

Data in Forms: Overview



© SAP AG 2006

- You have seen several times that you always need fields (variable data) in forms, for example, to output today's date, integrate values from a database table into the form, or define conditions for the processing of nodes.
- The following types of data exist in an SAP Smart Form:
 1. Interface data called parameters. They are passed to the form by the application program and vice versa.
 2. Global data that is known in all nodes of the form.
 3. System fields like the page number or the current time. (See slide *System Fields* in Unit 4 - *Texts, Addresses, Graphics*)
 4. Local data that you create in program line nodes using ABAP statements. (This type of data is not dealt with in this training course because it requires general ABAP knowledge not specific to forms.)

Interface Parameters I

SAP

The screenshot shows the SAP Smart Form interface with the following text and diagram:

CALL FUNCTION <SMART_FORM_FM>

EXPORTING ... →
 IMPORTING ... ←
 TABLES ... →
 EXCEPTIONS ... ←

Below the interface, there is a tree view under "Global settings" with "Form interface" expanded, and a red arrow points from the "Form interface" node to the "Import" tab in a navigation bar.

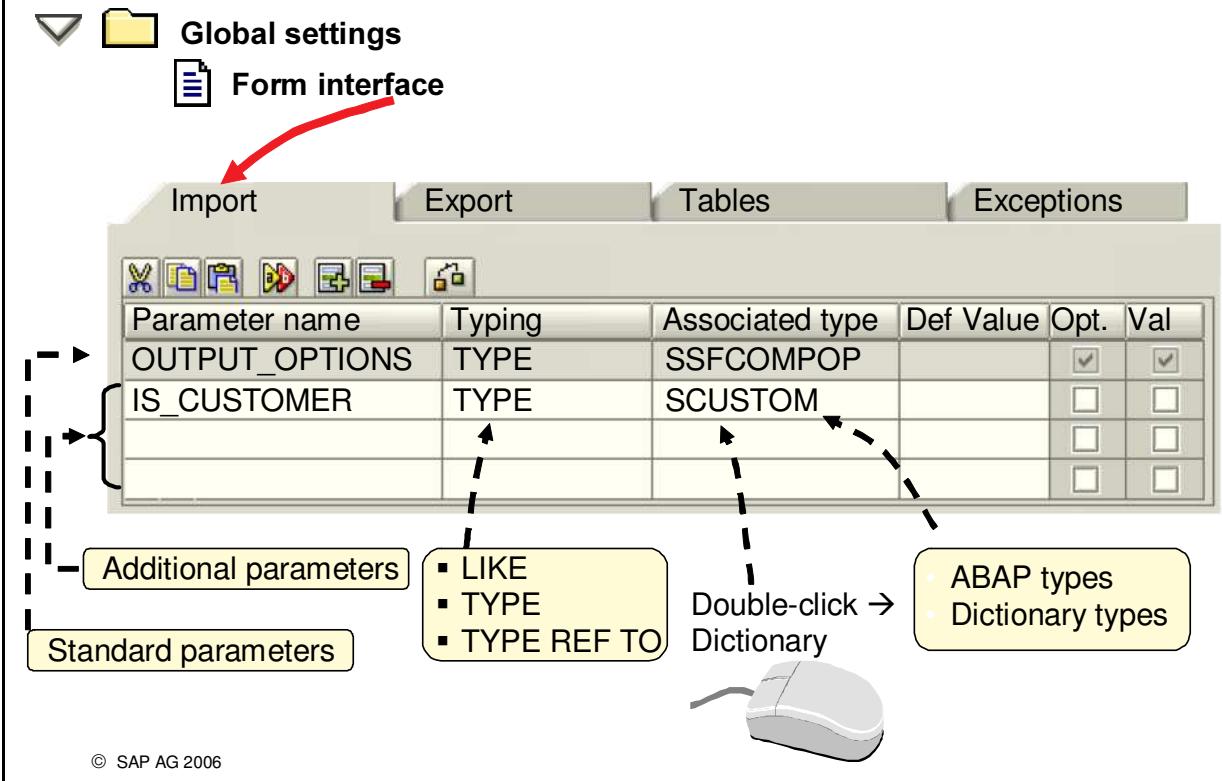
Import Export Tables Exceptions

© SAP AG 2006

- If an application program calls an SAP Smart Form (or more precisely: the generated function module of the form), there must be options to communicate with it. Data must be passed to the form and be returned by the form to the calling program. All data is exchanged through the interface of the form. You define the interface in the global settings of the Form Builder.
- All parameters of the interface are global, which means, they are known in all nodes of the form.
- The interface of the form/generated function module has the following parameters:

Import
 Export
 Tables
 Exceptions

Interface Parameters II



- *Import parameters* are read from the application program. There is a number of default parameters and an undefined number of others of any type. The default parameters - including the printout options - are covered in Unit 8 - *Integration into Application Programs*. Their fields are not ready for input in the interface. All other parameters differ from form to form and contain values provided by the application program, which means basically everything you want to output in the form. They are particularly important for the correct processing of the form.
- Assign names to all other parameters that comply with the ABAP naming conventions, that is, those that do not contain umlauts, blank spaces, or special characters.
- Import parameters must be typed. To do this, as with the DATA statement of an ABAP program, you can use **TYPE**, **LIKE** and (for ABAP objects) **TYPE REF TO**. You can choose one of the following as the *Associated type*:
 - ABAP types (C, N, D, T, X, I, P, F, STRING, XSTRING)
 - Dictionary types (data elements, structure types, internal table types)
 For more information, see the ABAP documentation on the keyword **TYPES**.
- You can set a default value for import parameters in the *Default value* field to ensure that they have a value even if they are not filled in the application program. This is particularly useful if you want to test a form without using an application program.
- If you check *Pass value*, the system passes a copy of the parameter to the form and not the parameter itself.
- As of SAP Web AS 6.10, you can set import parameters as optional.

Interface Parameters III

SAP

Parameter name	Typing	Associated type	Value
JOB_OUTPUT_INFO	TYPE	SSFCRESCL	<input checked="" type="checkbox"/>
SUM	TYPE	F	<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>

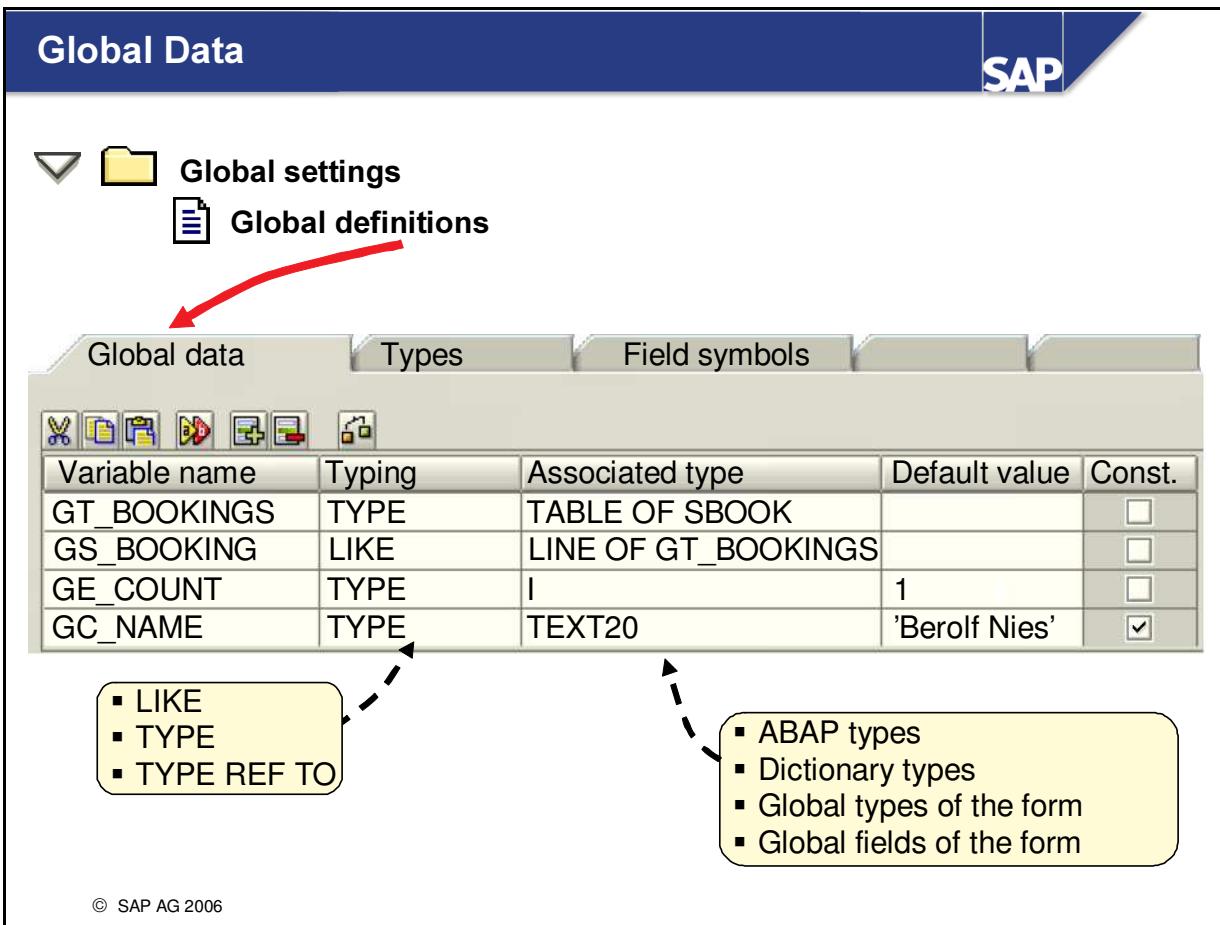
© SAP AG 2006

- *Export parameters* are returned to the application program. Again, there is a number of default parameters and an undefined number of others of any type. What we have said about the *Parameter names* and *Typing* of import parameters also applies to export parameters.

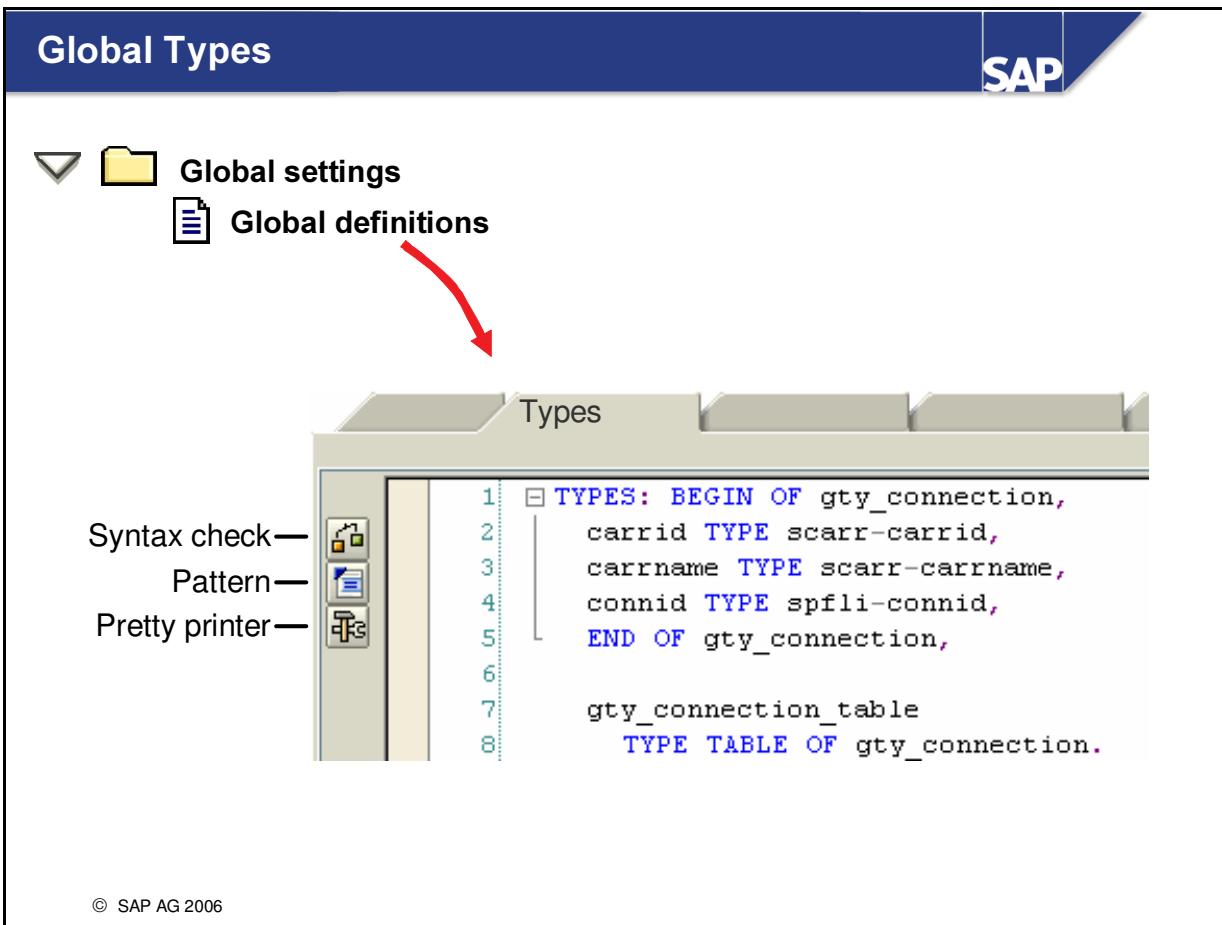
If you check *Pass by value*, the value is passed only at the end of the form function module to the parameter in the application program. If you do not check *Pass by value*, the values are passed by reference: If you make changes to export parameters in the form, the original value of the application program is changed **directly**. If form processing terminates for any reason, any value changes made so far to export parameters are also visible in the application program.

If values are passed by reference, an export parameter is turned into an import export parameter, which means that values can be passed in both directions: From the application program into the form and vice-versa.

- *Tables*: You can pass tables to the form. For typing, you can refer to a flat table type or (with LIKE) a flat structure type. (You can pass nested tables using appropriate export parameters.) The values are always passed by reference, which means that value changes in the form directly affect the values of the application program.
- *Exceptions*: Exception parameters are queried in the application program so that the program can respond to errors that occur during form processing.



- Variables that you enter on the *Global data* tab page of the global definitions are known in the entire form and can be used as work areas of tables or loops, for example. The best way to insert them into text nodes is to use the field list.
- The *Associated types* for the typing of the global data provide the same possibilities as the interface parameters and also:
 - Global types of the form (see next slide)
 - Global fields of the form (if referenced with `LIKE`)
- You can set an initial value for variables in the *Proposal* field.
- If you select the *Constant* checkbox, you generate a global constant instead of a variable, that is, you must then specify an initial value that can no longer be changed in the form.
- If you have entered an ABAP Dictionary associated type, you can double-click it to go to the ABAP Dictionary, where you can obtain information about the definition of the type.
- Check your entries using pushbutton *Check*.



- The *Types* tab page of the global definitions contains an ABAP editor which you can use to define types. You can then use these types in the entire form. In particular, you can reference them for the typing of the global data.
- Types are defined with the ABAP statement TYPES .
- For more information, see the ABAP keyword documentation for TYPES.

Field symbols

SAP

The screenshot shows the SAP Global settings interface under the 'Global definitions' section. A red arrow points from the text 'with field symbol' to the table row where the field symbol is defined.

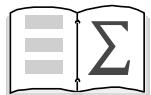
Field symbol name	Type Assignment	Associated type
<GS_BOOKING>	TYPE	SBOOK

Processing of an internal table:

with work area **with field symbol**

© SAP AG 2006

- In the global settings, you can create field symbols. The syntax corresponds to the ABAP command FIELD-SYMBOLS. This means you have to choose a name with an opening and a closing angle bracket, a typing method (TYPE, LIKE, TYPE REF TO) and an associated type
- Field symbols are placeholders for data objects, or more precisely: dereferenced pointers on data objects. Field symbols must be assigned to a data object at runtime and then contain the value of that object. This allows greater programming flexibility - however, at the expense of a restricted syntax and security check. This may result in runtime errors or wrong data assignments. You should therefore use field symbols only if other ABAP statements do not provide an adequate solution for implementing the operation you want.
- A good way to use field symbols is in loops over internal tables. The ABAP command LOOP AT <itab> INTO <workarea> copies each line of the internal table into the work area. If you use field symbols instead (LOOP AT <itab> ASSIGNING <<fieldsymbol>>), you do not have to copy the lines because the field symbol accesses the line contents of the internal table directly. Using field symbols for large tables in particular can therefore be better for runtime than work areas. SAP Smart Forms support field symbols for loops and tables.
- For more information, see the ABAP documentation for FIELD-SYMBOLS.

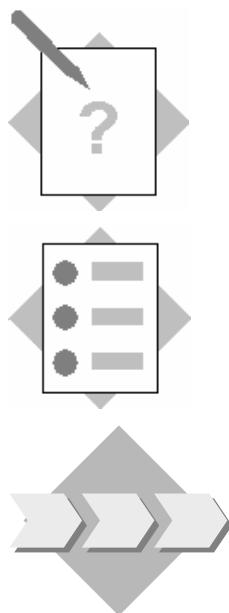


You are now able to:

- Define the interface of a form
- Create global data and types

© SAP AG 2006

Exercises



Unit: Data in Forms

After completing these exercises, you will be able to:

- Create interface parameters
- Create global variables and types (optional)

Your task: Extend your existing invoice form and make it more flexible by inserting fields. Perform the necessary preparatory steps to output the booking table.

Copy template for the form:

BC470_TEXTS

Package (for all exercises):

ZBC470_##

Name of the form to be created:

ZBC470_##_DATAS

Model Solution:

BC470_DATAS

Application program for testing purposes:

SAPBC470_DEMO

1 Copy template

Copy the form that you used in the last exercise (ZBC470_##_TEXTS) to ZBC470_##_DATAS. Alternatively, copy the copy template (BC470_TEXTS).

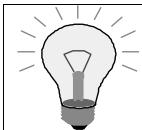
2 Enhance interface

2-1 On the selection screen of the print program, you can decide whether you want to insert the company logo in color (value = BCOL) or in black and white (value = BMON). The corresponding parameter is transferred when the generated form function module is called.

2-1-1 Add an import parameter called IE_COLOR of the Dictionary type TDBTYPE to the interface of your form.

2-1-2 Use this parameter for the company logo and make the necessary change on the *General Properties* tab page of the graphic.

- 2-2 Perform the necessary preparatory steps to output a real booking table for the next exercise:
Create a table called IT_BOOKINGS in the interface.
- 2-2-1 Refer to the Dictionary type TY_BOOKINGS.
 - 2-2-2 Perform a local check.
 - 2-2-3 TY_BOOKINGS is a table type. What is the name of the underlying structure?
Determine the name of the field in this structure which contains the price of the booking (in foreign currency).



Double-click TY_BOOKINGS to go to the ABAP Dictionary.

3 Create global variable

Also in preparation for the next exercise, create an appropriate work area for the table called GS_BOOKING. Type this work area with reference to the structure that you determined in 2.2.3.

4 Create and use global constant

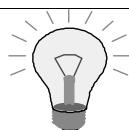
- 4-1 Create a global constant called GC_CLERK, and type this constant with reference to the Dictionary data element TEXT15. Assign a fine-sounding employee name to this constant.
- 4-2 Use this constant on page FIRST, window INFO, text node INFO_TEXT instead of the clerk entered there.
- 4-3 Add the name of the clerk to the closing lines (text node REGARDS of the main window) by inserting the constant.

5 **Optional:** Create global type

Create a global type called TEXT_FIFTEEN for a 15-digit character field. Perform a local check. Type the constant GC_CLERK with reference to that type instead of to TEXT15.

6 Test

Activate your form and test it using the program SAPBC470_DEMO. Verify, in particular, if the clerk is displayed correctly and if the color/black and white selection for the graphic on the selection screen functions properly.



If the program terminates, you may not have used the correct interface specifications.

Both the names and the types for interface parameters must be identical in the form and the application program. If required, read the text of the error message to track down the error in your program.



Unit: Data in Forms

1 Copy template

See the exercise in Unit 3.

2 Enhance interface

2-1-1 Choose *Global Settings* → *Form interface* and go to the *Import* tab. Add IE_COLOR TYPE TDBTYPE at the end of the list.

2-1-2 In the navigation tree of the page FIRST, select the graphic LOGO. On the maintenance screen, click the *General Properties* tab page and select the *Determine dynamically (BMON, BCOL)* radio button. Enter &IE_COLOR& there. At runtime of the generated function module, this variable is assigned the value of the selection screen. Depending on which option you choose, the company logo is printed in color or in black and white.

2-2-1 In the navigation tree, choose *Global settings* → *Form interface*. Click the *Tables* tab. Add IT_BOOKINGS TYPE TY_BOOKINGS at the end of the list.

2-2-2 Click the *Check* pushbutton of the maintenance screen.

2-2-3 The structure is called SBOOK. The name of the field which contains the price of the booking in foreign currency is FORCURAM.

3 Create global variable

In the navigation tree, choose *Global settings* → *Global definitions*. Enter GS_BOOKING TYPE SBOOK on the *Global Data* tab page.

4 Create and use global constant

4-1 In the navigation tree, choose *Global settings* → *Global definitions*. Enter GC_CLERK TYPE TEXT15 on the *Global Data* tab page. Enter the name in the *Default value* column. Select the *Constant* checkbox at the end of the line.

4-2 Choose the text node INFO_TEXT in the window INFO in the navigation tree. Delete the existing clerk name in the editor and use the field list to insert the global constant GC_CLERK. (The field is inserted as &GC_CLERK&.)

4-3 Repeat these steps for the text node REGARDS. (This node is in the MAIN window.)

5 **Optional:** Create global type

In the navigation tree, choose *Global settings* → *Global definitions*. On the Types tab page, enter the following: TYPES TEXT_FIFTEEN(15) TYPE c. Then, in the navigation tree, choose *Global settings* → *Global definitions*. On the *Global data* tab page, change the type GC_CLERK TYPE TEXT_FIFTEEN.

6 Test: See previous exercise.



BC470 Form Printing with SAP Smart Forms



1 Overview



2 SAP Smart Forms: Overview



3 First Steps with the SAP Form Builder



4 Texts, Addresses, Graphics

&WA&

5 Data in Forms



6 Tables and Templates



7 Process Control



8 Integration into Application Programs

ab_cd

9 Smart Styles

...

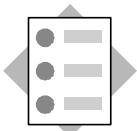
Appendix

© SAP AG 2006

Contents:

- **Tables**
- **Templates**
- **Table Painter**

© SAP AG 2006



After completing this unit, you will be able to:

- Explain the difference between tables and templates
- Use the Table Painter to create tables and templates
- Create control levels
- Create headers and footers

© SAP AG 2006

Tables and Templates: Differences

SAP



Table

Flight	Date	Price
AA017	12.16.2007	1,200.00 USD
AA017	12.31.2007	1,200.00 USD
Total for AA		2,400.00 USD
LH400	11.17.2007	581.00 EUR
LH402	11.17.2007	669.00 EUR
LH403	12.12.2007	610.00 EUR
Total for LH		1,860.00 EUR
Sum total		2,400.00 USD
		1,860.00 EUR



Template

Name of passenger (not transferrable)			Date of Issue		
YILMAZ/E MS			6NOV07		
To	Airline	Flight	Cl.	Date	Time
FRANKFURT	LH	2362	L	27NOV	1840
BERLIN TXL	LH	2351	L	28NOV	1910
Flight Price 350.00 EUR	Form and serial number 3344563125667				
Tax 102.59 EUR	Please do not write in or stamp this field				

- Layout
- Size

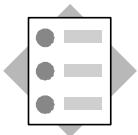


Only at
runtime

- Layout fixed
- Size fixed

© SAP AG 2006

- This unit will introduce you to two other types of nodes: tables and templates. Tables and templates have several things in common. For example, they are both designed with the Table Painter, and they use different line types.
- The most important difference between them is how their layout is determined:
The precise layout and the length of a table can only be determined at runtime, depending on the type and the number of records read by the application program from the database.
Templates, however, are defined completely in the Form Painter. This means that the type and the number of their cells cannot be modified at application program runtime. You therefore use templates primarily for external forms.



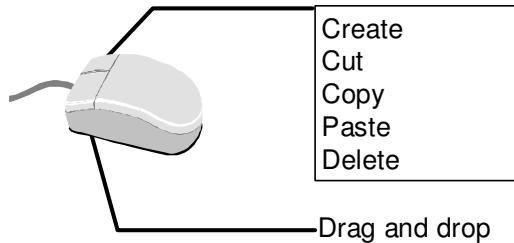
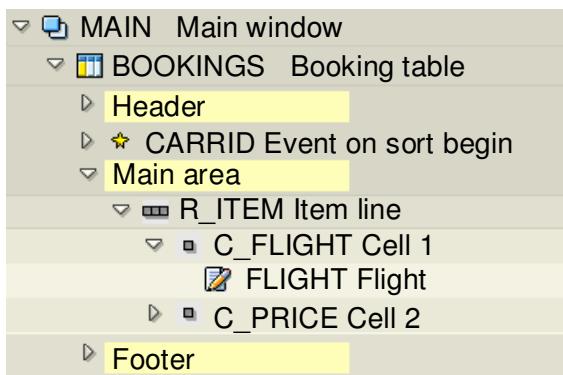
After completing this topic, you will be able to:

- **Create tables and their line types**
- **Explain how tables are processed**
- **Output data in tables**
- **Create control levels**
- **Perform calculations**
- **Create headers and footers**

© SAP AG 2006

- The table function has existed since SAP R/3 4.6C, but was changed for SAP Web Application Server 6.10. You can still use and edit forms with old table types, but you cannot create old table types. Tables of the old type have their own icon (see Appendix).
- **This unit only describes the procedure as of SAP Web Application Server 6.10.** The procedure with SAP R/3 4.6C is described in the Appendix.

Table: Overview



Flight **Price**

<i>Bookings for American Airlines</i>	
AA0016	1,200.00 USD
AA0017	1,200.00 USD
<i>Bookings for Lufthansa</i>	
LH0400	581.00 EUR
LH0402	669.00 EUR
Sum total	
	2,400.00 USD
	1,250.00 EUR

© SAP AG 2006

- Forms are frequently used to output data in tables. Tables in SAP Smart Forms are subnodes of windows and, like all other subnodes, they are created using the context menu (right mouse button) of the navigation tree.
- Since the length of tables is dynamic, you should use them only in main windows because they may be truncated in secondary windows.
- Tables provide functions to output headers and footers, sort the table, and calculate subtotals.

For each line type:

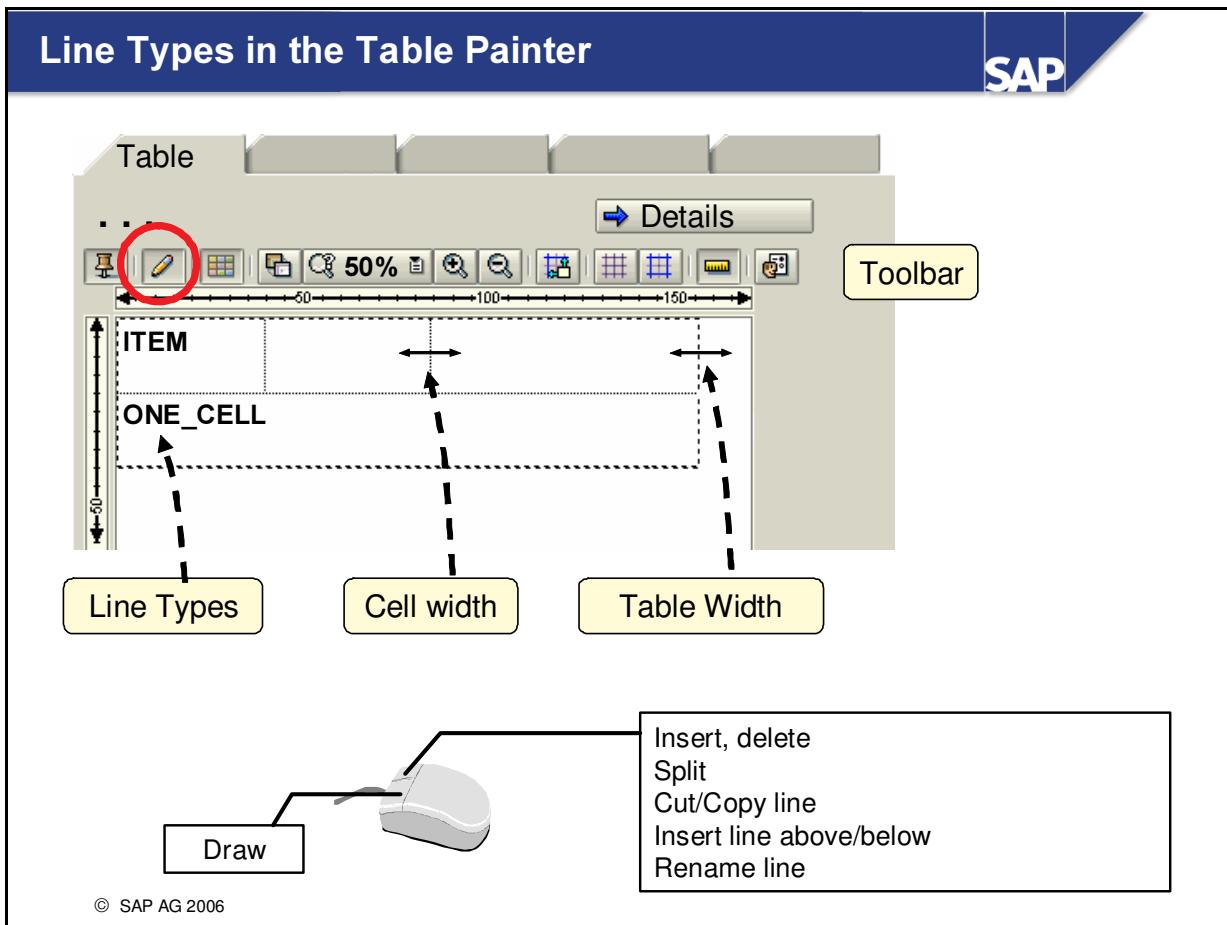
- Definition of number and width of cells
- Boxes/shading (also cell by cell)
- Height dynamic
- Contents dynamic

Bookings for Lufthansa			
LH0400	06/11/2007	581.00	EUR
LH0402	06/11/2007	669.00	EUR
Total for LH		1,250.00	EUR
Sum total			
			2,400.00 USD
			1,250.00 EUR

The diagram illustrates the application of different line types to specific rows in a booking table. The first two rows (LH0400 and LH0402) are grouped under 'Line type 1'. The third row ('Total for LH') is grouped under 'Line type 2'. The last three rows ('Sum total' and the currency conversion lines) are grouped under 'Line type 1'. The final row ('1,250.00 EUR') is grouped under 'Line type 3'.

© SAP AG 2006

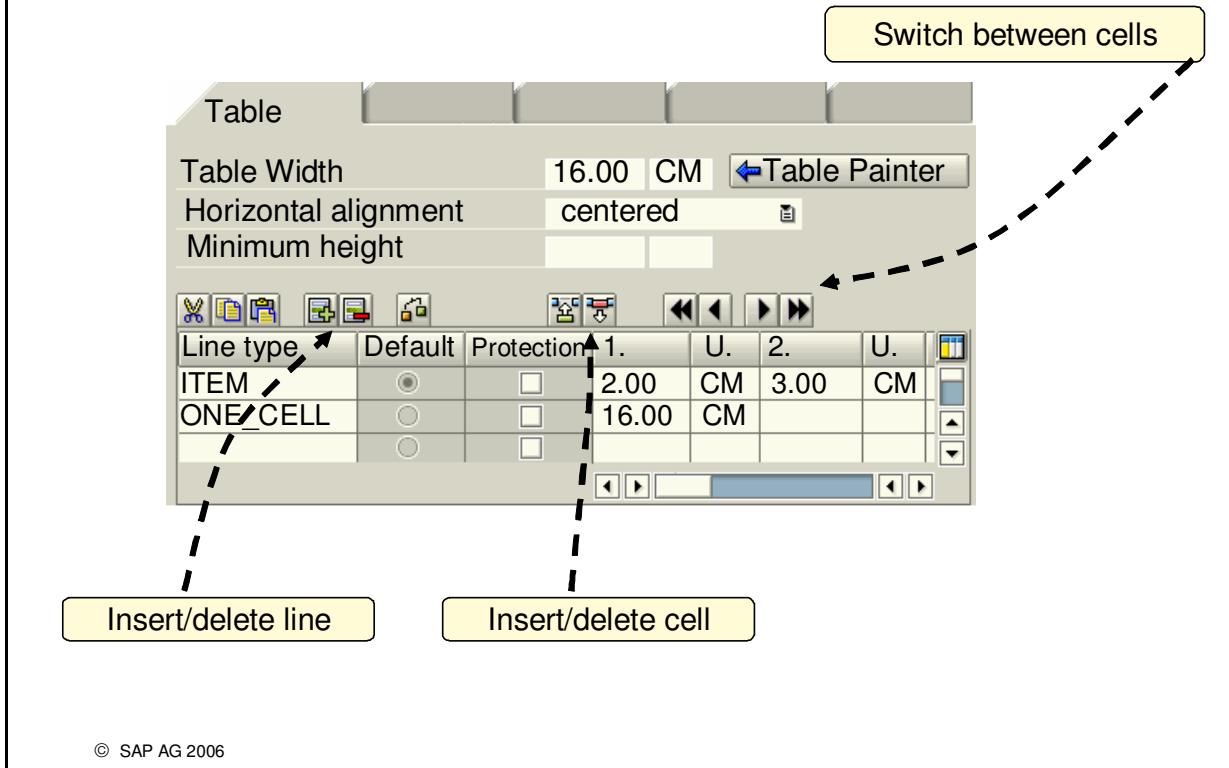
- Before you can fill tables with text, you must define the line types on the *Table* tab page of the maintenance screen. You also specify how many cells are to be in one table line and what width these cells should be. (The height is determined automatically at runtime.) For simple applications, a single line type is sufficient. However, you can also create different types for hierarchical (multi-level) tables, for example. You do this, for example, if you want to print the different bookings for a flight in the next lines or if you want to use subtotals.
- In the output options of the table lines, you specify when each line type is used.
- You can maintain line types graphically or alphanumerically. However, if maintained alphanumerically, boxes and/or shading cannot be defined.



- Line types are maintained on the *Table* tab page using the Table Painter as standard.
- To edit line types, ensure that **pushbutton *Draw Lines and Columns*** is activated.
- You insert a **new line** by horizontally dragging the mouse pointer, which is in the shape of a pencil, to the desired position. Alternatively, you can use the context menu (right mouse button ® *Insert* ® *Blank line above/below*). To assign a new name to a line, position the mouse on it and from the context menu, choose *Rename Line*.
- You insert a **new cell boundary** by vertically dragging the mouse pointer to the desired position while keeping the left mouse button pressed. Alternatively, you can split the cell in which the mouse pointer is positioned using the context menu (right mouse button ® *Split* ® *Cell*).
- You change the **width of a cell** by placing the mouse on a cell boundary and dragging it to the desired position while keeping the left mouse button pressed. (The mouse pointer has the shape of a double arrow.)
- You change the **table width** by placing the mouse on the right table border and dragging it to the desired position while keeping the left mouse button pressed.
- To **position the whole table** within the window, deactivate the *Draw Lines and Columns* pushbutton and drag the table to the required position with the left mouse button (the mouse pointer is in the shape of a double arrow).
- You cannot set the **height of individual line types** because the height is determined dynamically at application program runtime, depending on the data that is output.

Line Types in the Detail View

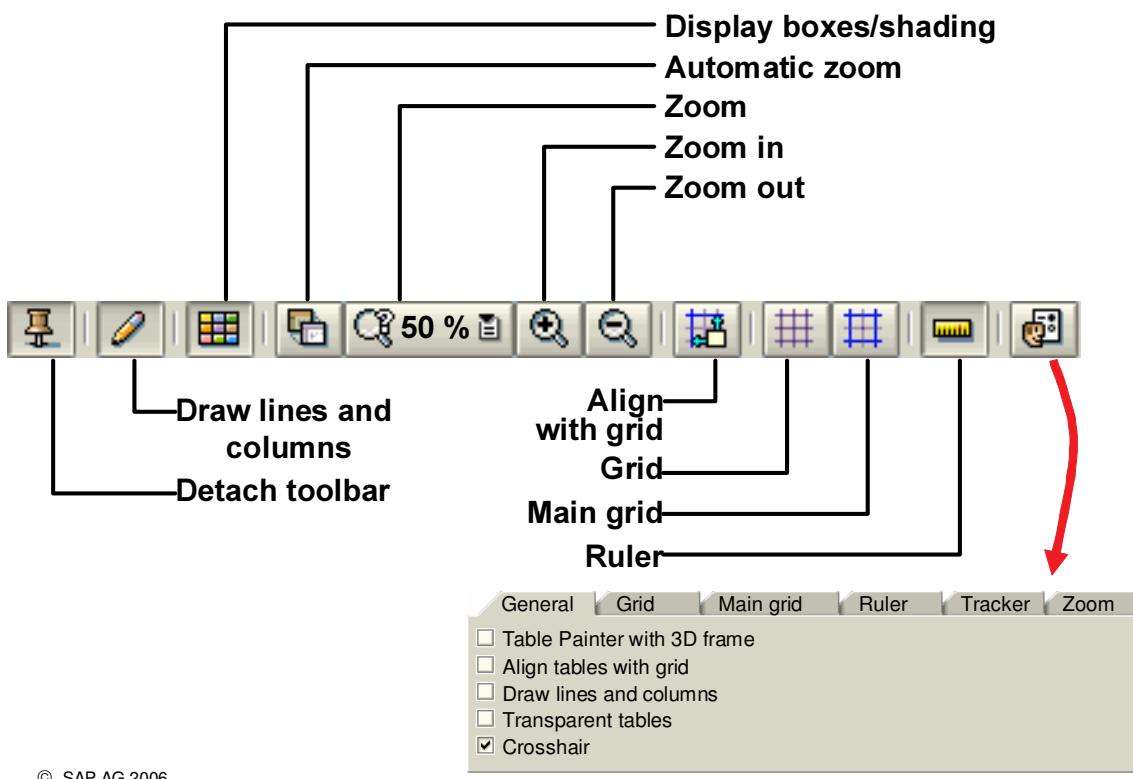
SAP



- Instead of entering the line types in the Table Painter, you can also define them numerically in the detail view. The entries you make in the Table Painter are automatically copied to the alphanumeric overview and vice versa.
- The following information is required for line types:
 - Name
 - Protection against page breaks. Since the height of a line type is dynamic, a page break may occur at runtime. If you select this option, all nodes of a table line of this line type are output together on one page.
 - Number and width of the cells
 - Default type: without relevance
 - The table width must be identical to the total width of all cells for each line type.
- You use the *horizontal alignment* to determine how the table is to be aligned with reference to the window margin. You can choose between *left*, *right* and *centered*. If you choose *left* or *right*, you have the option of specifying a distance from the respective window margin.
- As of SAP Web Application Server 6.20, there is another input field: *Minimum height*. If you enter a value in this field, the table is started on the next page if the minimum height is no longer available in the current window.

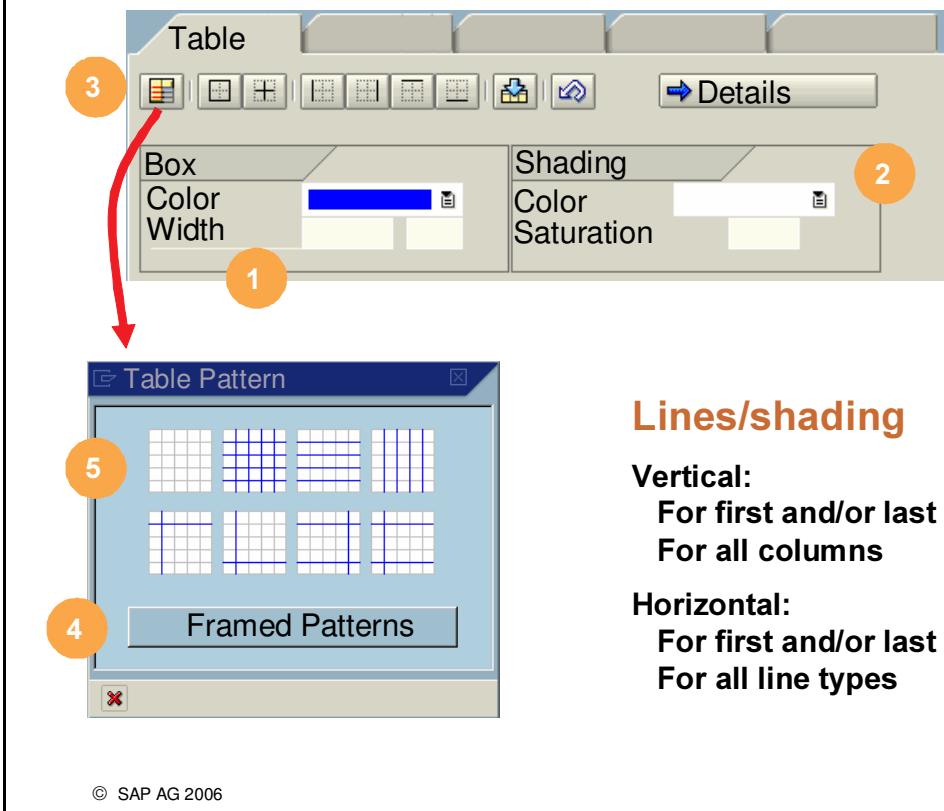
The Table Painter: Toolbar

SAP



© SAP AG 2006

- The most important settings (which are basically the same as those in the Form Painter) are displayed in the detachable toolbar of the Table Painter. For other options, choose *Utilities* → *Settings* → tab page *Table Painter* or click the right pushbutton of the toolbar.
- If the *Draw lines and columns* checkbox is selected, you can draw the cells of the line types directly with the mouse (as described on the previous slide). If you do not select this checkbox, you can move the table to the left or to the right within the window width - provided you have not chosen *centered* as the horizontal alignment of the table in the detail view.
- Several zoom options are available to adjust the display. The most comfortable option is the *Automatic zoom*.
- To make the drawing of cells easier, you can display a grid and/or the main grid. You can also make a setting in the Table Painter that ensures that vertical cell boundaries are automatically aligned with the grid when you move them with the mouse. You can set the step size of both grids. The crosshair cursor, which you can set instead of the normal mouse pointer on the *General* tab page of the Table Painter settings, also facilitates the drawing of cells.
- The *Tracker* tab page of the Table Painter settings allows you to determine how the table is to be highlighted against the background.
- To hide the toolbar, choose *Utilities* → *Settings* → tab page *Table Painter* and deactivate the *Toolbar* checkbox.



Lines/shading

Vertical:

- For first and/or last column
- For all columns

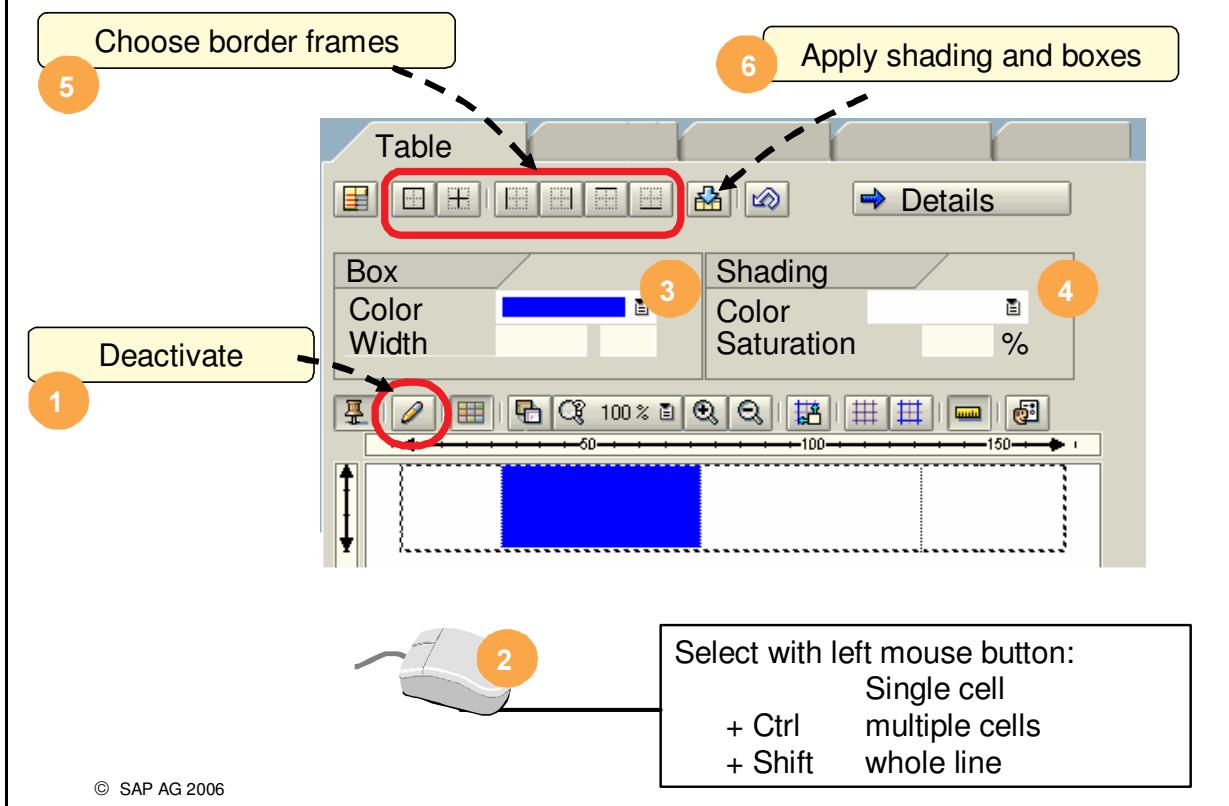
Horizontal:

- For first and/or last line type
- For all line types

- If you want to frame and/or shade your table, you can choose a pattern on the *Table* tab page of the Table Painter. Follow the instructions below:
 1. Choose the required **box color** from the list of available colors and set the **width**.
 2. Alternatively, or in addition, you can choose a **color** and a **degree of saturation** for the **shading**. This color is used for the whole table.
 3. Choose the **Select pattern** pushbutton.
 4. Decide whether you want the line types to be framed or not (pushbutton **Display Framed Pattern**.)
 5. Select the pattern you want to use by clicking it with the mouse. You can choose whether the first, the last, or all columns are to be separated by vertical gridlines and/or whether the first, the last, or all line types are to be separated by horizontal gridlines.

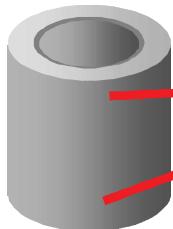
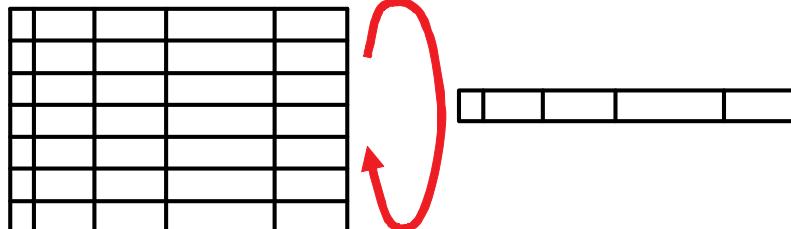
Boxes and Shading: Single Cells

SAP



- To add lines and/or shading to individual cells and lines of tables, proceed as follows:

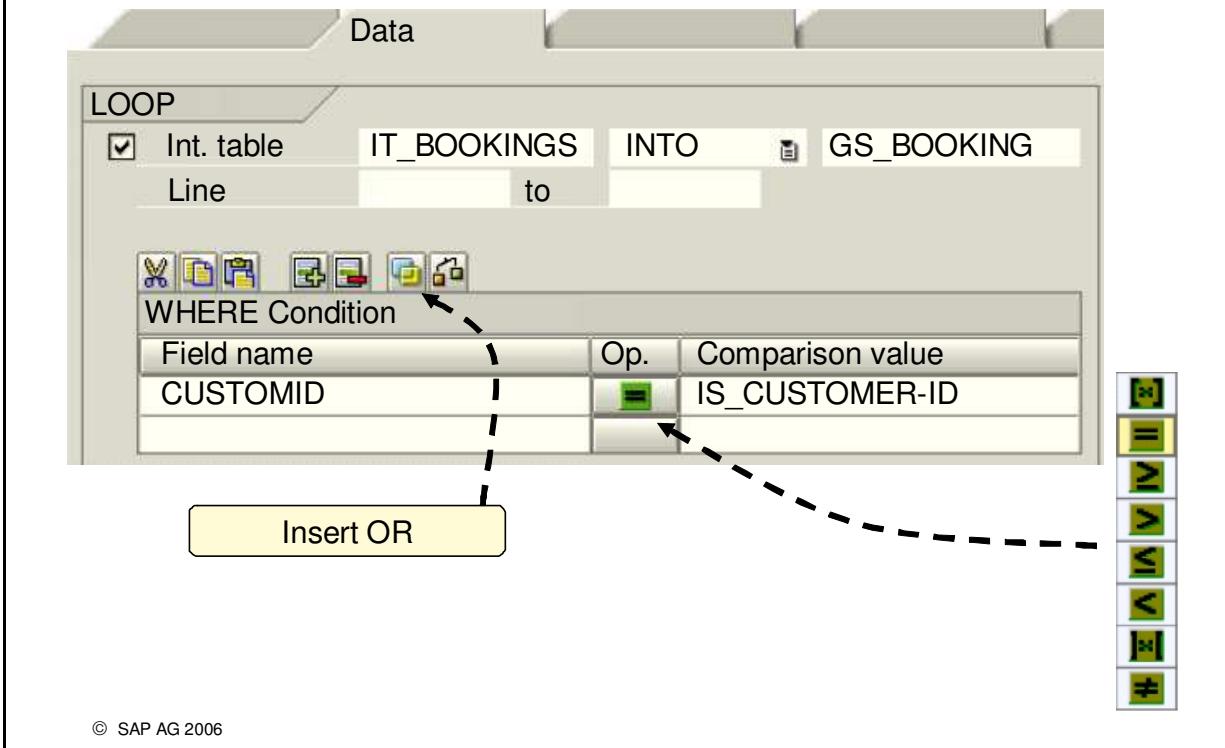
1. Deactivate **pushbutton Lines and Columns**.
2. In the Table Painter, use the left mouse button to **select** the cell(s) for which you want to set a frame and/or shading. To select multiple cells, click the left mouse button in all required cells whilst holding down the Control button (Ctrl) on your keyboard. To select a whole line, position the mouse pointer in the line and click with the left mouse button whilst holding down the Shift key on your keyboard. Press Ctrl-A to select the whole table.
3. Choose the required **box color** from the list of available colors and set the **width**.
4. Alternatively, or in addition, you can choose a **color** and a **degree of saturation** for the **shading**.
5. Choose the **page(s)** of the selected cells to which a line is to be added. Repeated clicking switches between the line being shown/hidden. Note that shading is not visible until you deselect.
6. If you then want to add the same color and frame to other cells, all you have to do is select the cell as described above and choose *Apply shading*

Data Retrieval**Loop in Form****Database Table(s)****Internal Table****Work Area**

© SAP AG 2006

- From a technical point of view, a table in an SAP Smart Form is filled by processing an internal table on a line-by-line basis. This is referred to as a *loop*. The respective lines are copied into a work area that has the same structure as the table. The internal table must be filled beforehand in the application program (which is the usual case) or in the form (See Unit 7 – *Flow Control*.) The data is normally taken from database tables. If the data is read in the application program, the internal table must be defined in the interface of the SAP Smart Form.
- If the internal table is very large, we recommend that you process the table using field symbols instead of a work area since field symbols can directly access the individual lines and therefore the lines do not need to be copied.

Filling Tables

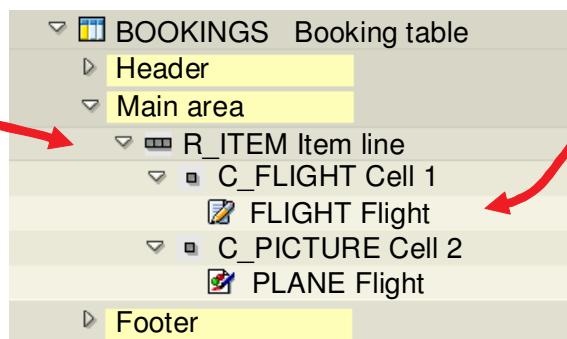


- After defining the table design on the *Table* tab page, you should determine on the *Data* tab page how the table is to be processed.
- Select the *Internal table* indicator and enter a name for the table and for the work area. Both the internal table and the work area must be known in the form. This means that they must have been defined from the interface or as a global field. If you do not select the *Internal table* indicator, no loop processing takes place. This may be useful, for example, if you want to output text in parallel columns.
- Possible assignment types are *into* and *assigning*. If you use *into*, the lines are copied from the table into the work area. If you use *assigning*, the lines are assigned to a field symbol. If you want to use tables with header lines, enter the name of the table as the work area.
- It is possible to use only a specific line range of the internal table. To do this, specify the rows in the *Line ... to...* fields.
- You can also use logical conditions to determine which lines of the internal table are to be processed. This corresponds to the *WHERE* clause of the ABAP command `LOOP AT <itab>`. Enter the name of a field of the work area, a relational operator, and the comparison value or comparison field. The relational operators that you know from normal selection screens are available: *With/without pattern*, *Equal to*, *Not equal to*, *Greater than or equal to*, *Greater than*, *Less than or equal to*, *Less than*. If you do not enter an operator, *Equal to* is automatically used. You link multiple conditions with *AND*, but you can also use the *OR* pushbutton.

Contents in Table Lines

SAP

- 1 Create table line (context menu)
- 2 Set line type (tab page *Output Options*)
- 3 Cells are created automatically
- 4 Create subnodes of cells



© SAP AG 2006

- To output contents in a table, proceed as follows:
 1. Right-click the relevant area (header, main area, or footer) and choose *Create table line* from the context menu.
 2. On the *Output Options* tab page for the inserted line, enter a title and a description. Choose an appropriate **line type** from the list.
 3. The number of cells that you defined for the selected line type are inserted automatically in the navigation tree.
 4. You now create the actual contents, for example text or graphics, as normal **subnodes of cells**. Since the height of cells is dynamic, you can create more than one subnode for a cell. Page breaks may then occur automatically.

The screenshot shows the SAP Fiori 'Headers and Footers' configuration interface. It displays two examples of table structures and their corresponding 'Output Options'.

Example 1:

- Table Structure:**
 - Header
 - R_HEADER
 - Main area
 - Footer
- Output Options:**
 - Style: [empty]
 - Page protection
 - Print time
 - At start of table
 - At page break

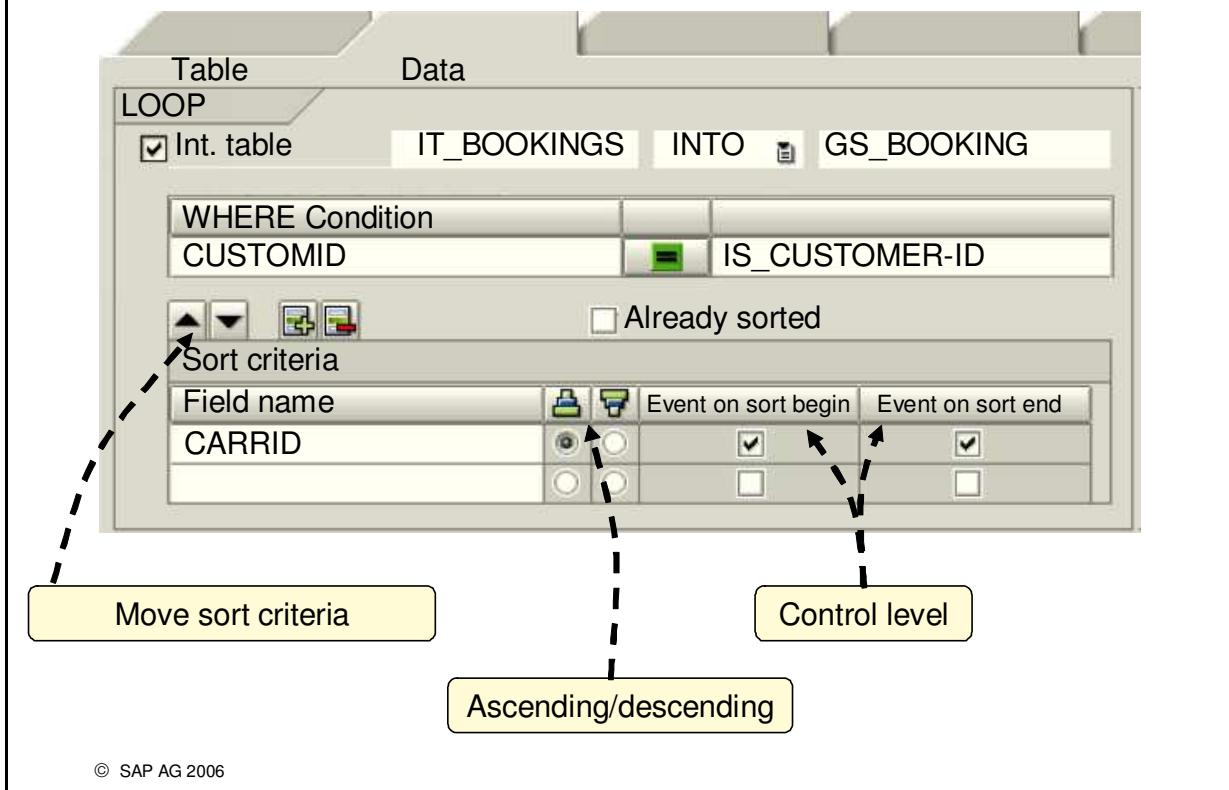
Example 2:

- Table Structure:**
 - Header
 - Main area
 - Footer
 - R_SUM
 - C_SUM
 - SUM
- Output Options:**
 - Style: [empty]
 - Page protection
 - Footer with height CM
 - Print time
 - At page break
 - At end of table

© SAP AG 2006

- You can output contents for headers and footers in the same way as for the main area of a table. Whereas the main area may be used for items, for example, the header may be used for column headers and the footer for page totals and sum totals. The subnodes for the three areas are created automatically as soon as you create the table.
- On the *Output Options* tab page of the header, you determine whether the header is to be output and if so, where it is to be output: at the start of the table and/or after a page break.
- On the *Output Options* tab page, you determine whether the footer is to be processed at the end of the table and/or before a page break. You must also specify a height for the footer to enable the form processor to reserve sufficient space.
- If all three areas appear on one page, they are processed from top to bottom, that is, first the header, then the main area, and finally the footer.
- Note that page protection for multiple table lines is possible only as of SAP Web AS 6.20. You have to create a folder, set page protection for this folder (on the *Output Options* tab page), and move the relevant table lines to it.

Sorting a Table



- You can also sort the internal table within the form. To do this, enter the names of the fields to be used as the *Sort criteria*. The order of the fields in this list determines the sort sequence. You can subsequently change the sort sequence by placing your cursor on a field and moving it up or down a line by clicking one of the two black triangles displayed above the sort criteria. To the right of the field, you can choose whether the table is to be sorted in ascending or descending order.
- For technical reasons, the system cannot recognize whether the internal table has already been sorted (for example, in the data retrieval program). If this is the case, you still have to enter the sort criteria but then select the *Already sorted* checkbox. (Otherwise, the table will be sorted again).
- Sorting is required for subtotals and subheadings.

Control Levels I

Data Records in Internal Table:

AA	0017	12/15/2007	1,200.00	USD
AA	0017	01/07/2008	1,200.00	USD
AA	0026	03/31/2007	1,400.00	USD
LH	0400	11/17/2007	581.00	EUR
LH	0400	12/19/2007	581.00	EUR

© SAP AG 2006

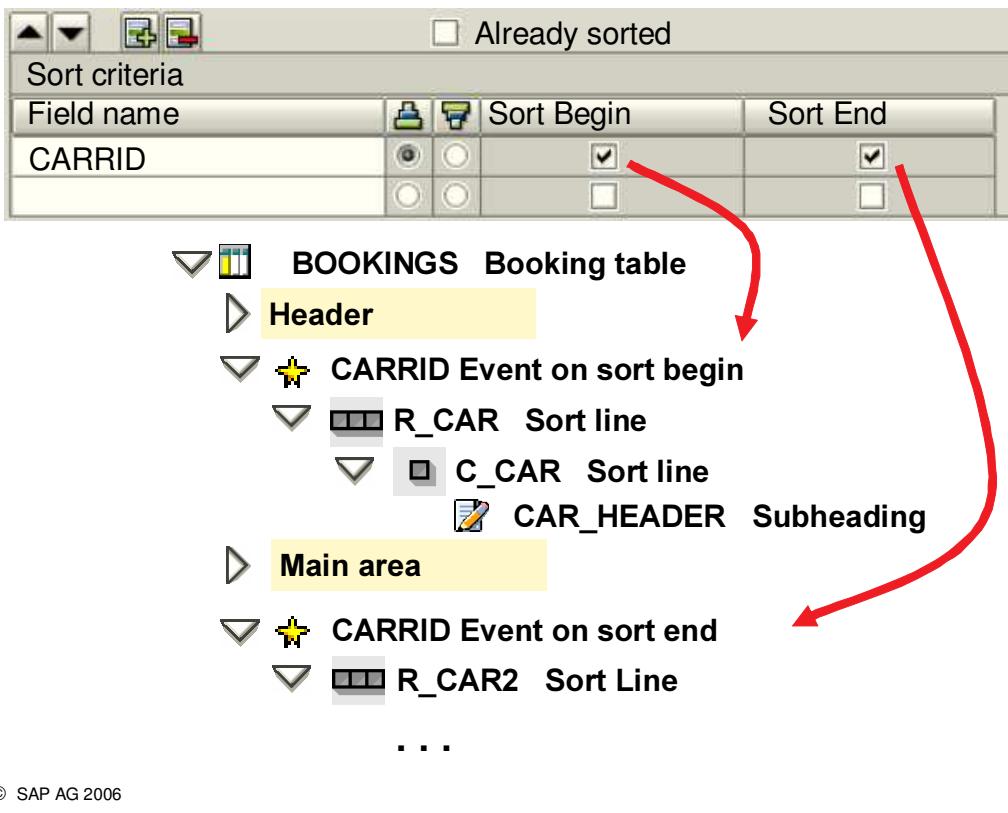
Table in Form:

Bookings for AA				
0017	12/15/2007	1,200.00	USD	
0017	07/01/2008	1,200.00	USD	
0026	03/31/2007	1,400.00	USD	
Total for AA:				3,800.00 USD
Bookings for LH				
0400	11/17/2007	581.00	EUR	
0400	12/19/2007	581.00	EUR	
Total for LH				1,162.00 EUR

- Tables are often not output in exactly the same structure in which they are filled. For example, it should be possible to group data records and to output subheaders or subtotals. Grouped data records that have certain identical values are called *control levels*. SAP Smart Forms enable you to create any number of control levels in a table. In the above example, there is a control level for airlines.

Control Levels II

SAP



- If you select *Event on sort begin* and/or *Event on sort end* for a sort criterion, the corresponding control levels are inserted into the navigation tree of the table. A control level contains all records of the internal table that have the same value in the sort field. In the example above, all records of an airline belong to one control level.
- You decide what is output in the control levels. You can insert table lines as subnodes of the sort levels and then fill the cells with contents for output, for example, program lines for complex subtotal calculations or text nodes.
- The node of a control level, called an event node, has only one tab page on the maintenance screen, the *Output Options* tab page. You can only set a style on this tab page.
- You can define control levels for all sort fields. This means you can set up a hierarchical table that contains, for example, one control level for airline and one for flight connections.
- You cannot create control levels directly as nodes in the navigation tree. You must always proceed as follows: Determine the sort criterion and then select *Event on sort begin* and/or *Event on sort end*.

Calculations in Tables I

SAP

Table

Calculations

Flight	Booking	Luggage
AA 0017	47001	13.2 KG
AA 0017	47002	25.0 KG
AA 0017	47003	17.2 KG
AA 0017	47004	11.7 KG
AA 0017	47005	19.5 KG
AA 0017	47006	22.9 KG
AA 0017	47007	10.0 KG
Total number:		7
Total luggage:		119.5 KG
Average luggage:		17.1 KG

© SAP AG 2006

- As of SAP Web Application Server 6.10, there is another tab page for tables: *Calculations*. On this tab page, you can specify that different calculations are to be performed when the table is processed, without having to enter code for them.
- You can configure the system so that all fields of a table are automatically counted and totals and average values are calculated from all numeric fields.
- Note that calculations do not take currencies or units of measure into account. If, for example, you have prices in different currencies for flight bookings (Euro and US Dollars, for example), you cannot use the *Calculations* tab page for the prices since there is no automatic conversion. Instead, you would have to perform the calculations manually using the program line nodes.

Operation	Field name	Target field name	Event	For field name
-----------	------------	-------------------	-------	----------------

Initialize	Reset	For field name
------------	-------	----------------

Operation:

- Mean value
- Number
- Total

Field name: From where is the total/average to be calculated?

Target field name: Contains variable for result

Event:

- Before loop (at start of main area)
- After loop (at end of main area)
- Before sorting
- After sorting

Initialize: Target field (once before table processing)

Reset: Target field (for a new level)

© SAP AG 2006

- There are three possible operations for calculations:

Number: The target field is increased by 1 for each table line.

Total: The value of the field entered in the *Field name* column is added to the value of the target field.

Mean value: The values of the field entered in the *Field name* column are used line-by-line to calculate the mean value, which is put in the target field.

- You choose the events *before loop* or *after loop* if a calculation is to be performed for every processed line of the table.
- You use the events *before sorting* and *after sorting* primarily to count sublevels. If you choose one of these events, you have to enter the sort criterion (the variable of the control level) in the *For field name* field without the name of the output area, for example, CARRID and not WA-CARRID.
- If you select the *Initialize* indicator for a calculation, the target field is reset at the start of the table.
- Normally, all values in a column are used for calculations. If, however, you choose the *Sort Criterion* entry for *Reset* and then enter a field in the *For Field* column, the corresponding counter variables are reset each time a change occurs as soon as the content of the sort criterion changes - therefore at every control level change. However, it is prerequisite that you have selected *Event on Sort Begin* on the *Data* tab page.

Calculations in Tables: Example I

Total Luggage Weight:

AA 0017	47001	13.2 KG
AA 0017	47002	25.0 KG
AA 0017	47003	17.2 KG
AA 0017	47004	11.7 KG
AA 0017	47005	19.5 KG
AA 0017	47006	22.9 KG
AA 0017	47007	10.0 KG
Total luggage:		119.5 KG

Entries on the *Calculations*:tab page:

Operation	Field name	Target field name	Event	Initial.
Total	WA-WEIGHT	GE_TOTAL	After loop	<input checked="" type="checkbox"/>

→ Output of GE_TOTAL, for example, after table

© SAP AG 2006

- Example: Calculation of total luggage weight:

Prerequisite: A loop in a work area (here: WA) that has a field for the total luggage weight (here: WEIGHT). You require a global help variable that is suitable for the luggage weight of the work area. Settings on the *Calculations* tab page, see above. If you select the *Initialize* indicator, the GE_TOTAL field is set to zero before processing.

Calculations in Tables: Example II

Number of bookings for each airline:

AA	0017	12.15.2007	13.2 KG
AA	0017	12.31.2007	25.0 KG
AA	0017	01.07.2008	17.2 KG
AA	0026	02.12.2007	11.7 KG
AA	0026	03.31.2007	19.5 KG
Bookings for AA: 5			
LH	0400	11.17.2007	22.9 KG
LH	0400	12.19.2007	10.0 KG
Bookings for LH: 2			

Entries on the *Calculations:tab* page:

Operation	Target field name	Event	Reset	For field name
Number	GE_CNT	Before loop	Sort Criterion	CARRID

→ Output of CNT in node CARRID Event on Sort End

© SAP AG 2006

■ Example: Number of bookings for each airline:

Prerequisite: You require a global counter variable (for example, type I). Here: GE_CNT.

Your table must be sorted by CARRID on the *Data* tab page

It does not matter whether you choose *before loop* or *after loop* for the event since you do not output the counter until the end of each control level and not for each line.

The counter variable has to be reset at each level change (every time there is a change in the CARRID field).



After completing this topic, you will be able to:

- **Create templates**
- **Output data in templates**
- **Create templates by redrawing**

© SAP AG 2006

Templates

▼ TICKET Flight ticket

- C_NAME Name
- C_DATE Date
- C_ARR1 1st Dest.
- ...

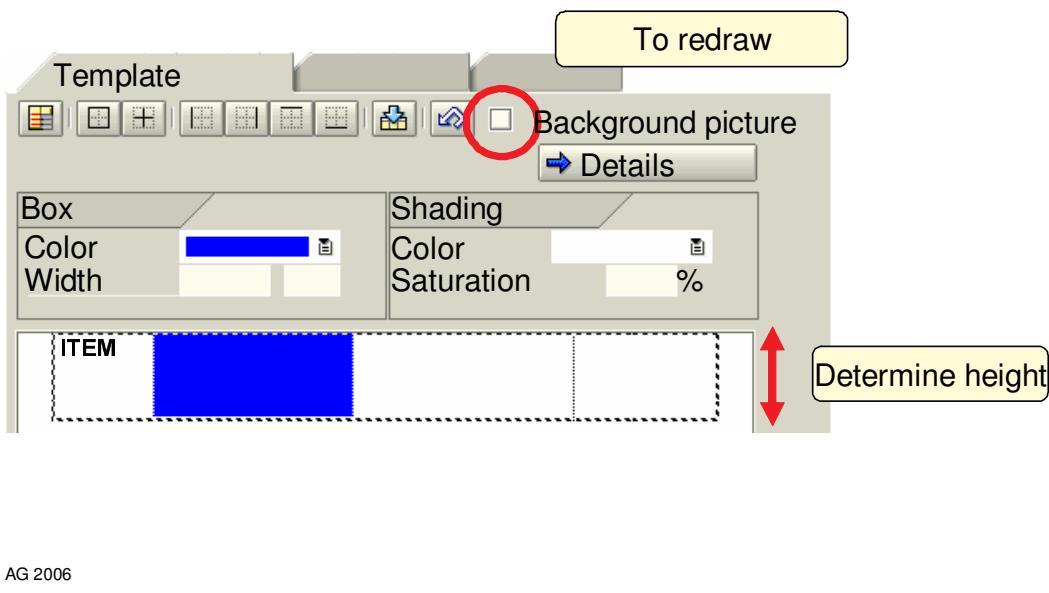
Name of passenger (not transferrable) RAHN/U MRS			Date of issue 06NOV07			
To	Airline	Flight	Class	Date	Time	Status
FRANKFURT	LH	2362	L	27NOV	1840	OK
BERLIN TXL	LH	2351	L	28NOV	1910	OK
Flight Price 350.00 EUR	Form and serial number 3344563125667					
Tax 52.59 EUR						
Total 402.59 EUR	Do not write on and stamp this field					

© SAP AG 2006

- Layout fixed
- Width and height fixed
- Different Line types

- You use the *Template* node type to output tables with a fixed layout and size. Templates are used, for example, for printing data on predefined forms, such as flight tickets (see above) or tax forms.
- Like all other nodes, templates are created as subnodes of windows, that is, using the context menu (right mouse button) in the navigation tree.
- Templates cannot be nested.
- You can create different node types as subnodes of templates. Text that does not fit into the cell selected is not output because the layout of the template is fixed.
- Graphics that you create as subnodes of templates are not visible in the Form Painter. You only see them in the print preview.

Usage: Same as with tables

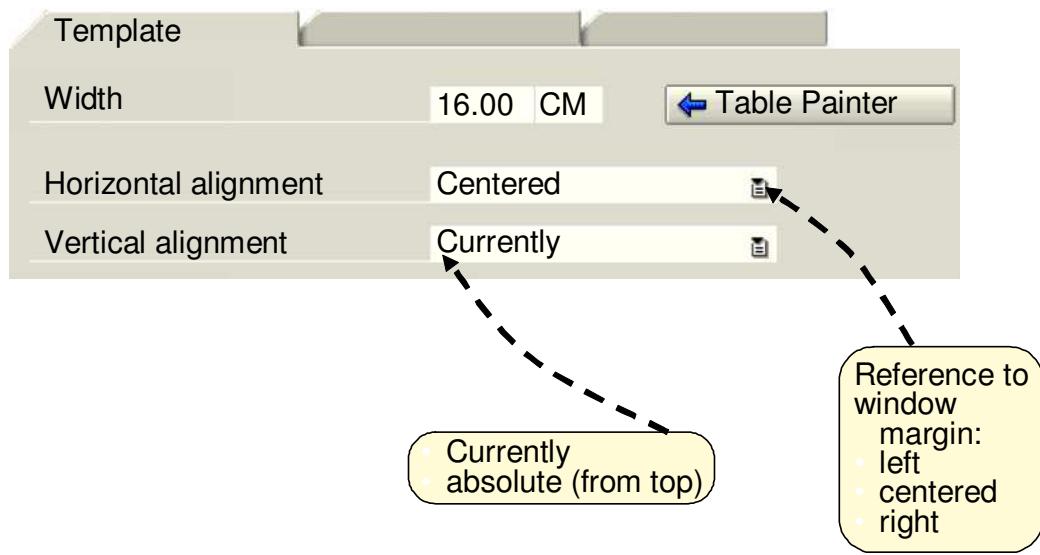


© SAP AG 2006

- The usage of the Table Painter is almost the same for templates and tables. You draw your lines and cells with the mouse and add frames and possibly also shading.
- However, one difference is that you can show or hide the background picture for the page (assuming there is one). This is useful if you have scanned in a template and you want to redraw it with the Table Painter. It is prerequisite that for the vertical alignment of the template in the detail view, you set *absolute (from top)*.
- You also have to specify the height of the individual lines because it is not dynamic, unlike with tables.

Drawing templates: Details I

SAP

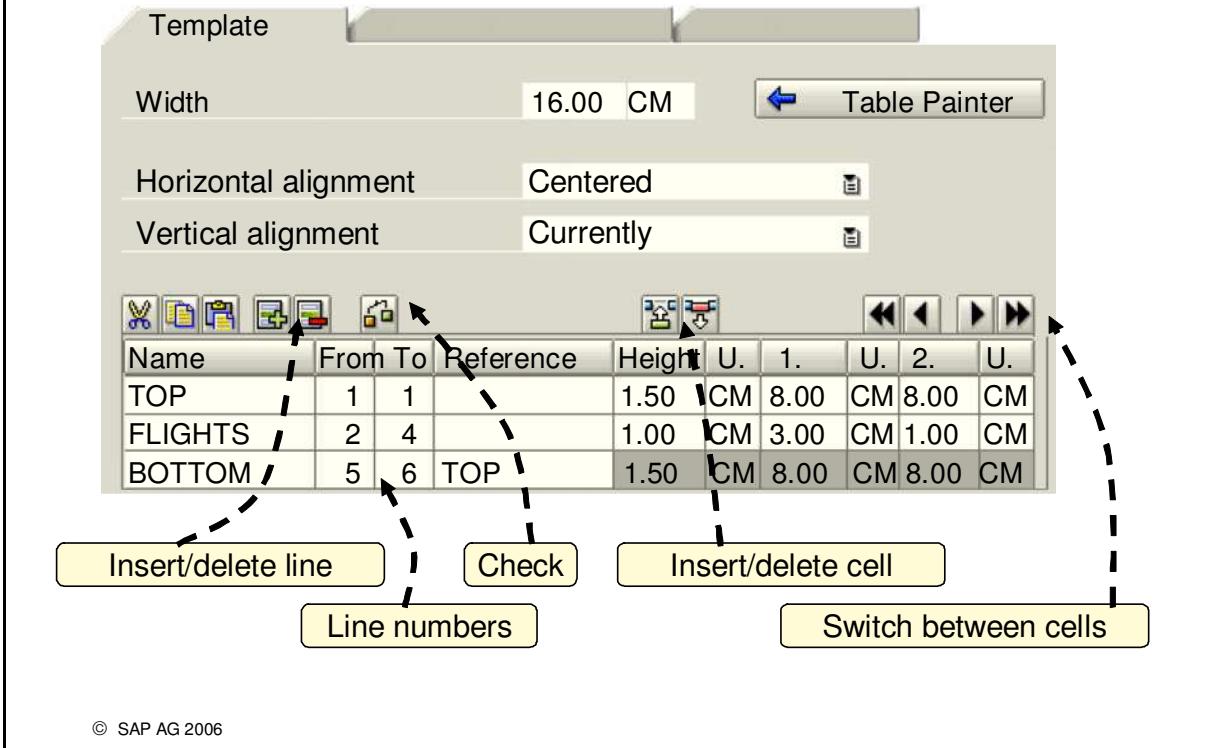


© SAP AG 2006

- If you choose the detail view on the *Template* tab page, you can make the following settings in the upper area:
- The **Width**. The width of the template must not exceed the width of the window into which the template is embedded.
- For **Horizontal alignment**, you can choose *left*, *centered* and *right*. These values refer to the window margin. If you choose *left* or *right*, the system displays an additional input field where you can enter the distance to the window margin. If you do not enter a value here, the template is placed directly on the margin.
- For **Vertical alignment**, you can determine the distance of the template from the top window margin. Choose *absolute (from top)* and enter the desired distance in the input field that appears on the right side. This enables you to place several templates side by side in the same window. (This is useful, for example, if you want to print labels or put several templates on top of one another to define complex line structures.) For vertical alignment, you can choose *current*. This places the template in the window directly underneath the node that precedes the template in the navigation tree. The vertical position of the template in the form is then determined by the number of nodes processed before the template at the time of output.

Drawing templates: Details II

SAP



- The lines and cells that you can draw with the mouse in the Table Painter can also be created and edited in the detail view. The procedure is similar to that for tables.
- First define a unique symbolic name, then the range of lines that use this line type. If several lines that are not successive use the same line type, you only need to define the line type once and can then specify it in the *Reference* field each time it is used. In the above example, lines 1, 5, and 6 have the same type. Since the line type BOTTOM refers to the type TOP, which has already been declared, the fields for the line height and the width of its cells are not ready for input.
- In the *Height* field, you set the height for the entire line.
- You can create any number of cells for each line. Enter the width of these cells. The sum of the values for the width of the cells must be the same as the width set for the template.

Template layout: Example

Name	From	To	Reference	Height	U.	1.	U.	2.	U.	3.	U.
TOP	1	1		1.50	CM	8.00	CM	8.00	CM		
FLIGHTS	2	4		1,00	CM	3,00	CM	1,00	CM	12,00	CM
BOTTOM	5	6	TOP	1.50	CM	8.00	CM	8.00	CM		



1. TOP	1.	2.
2. FLIGHTS	1.	2.
2. FLIGHTS	1.	2.
4. FLIGHTS	1.	2.
5. BOTTOM	1.	2.
6. BOTTOM	1.	2.

© SAP AG 2006

- The above slide shows a possible layout definition and the result of the print preview.
- You need the line numbers and cell numbers to output the contents in the cells.
- Tip: Since you cannot define cells with different heights in one line, you can put two or more tables on top of each other. To do this, enter absolute values for the vertical and the horizontal position.

Outputting Contents in Templates

TICKET Flight ticket

- C_NAME Name**
- C_DATE Date**
- C_ARR1 1st Dest.**

1.	1.	2.
2.	1.	2.
3.	1.	2.
4.	1.	2.

Template Output Options Conditions

Style

Output Structure

Line	2
Column	3

© SAP AG 2006

- After defining the layout of the template, you can use the context menu (right mouse button on the template) to create subnodes in which contents are output. Alternatively, you can create subnodes for the contents by choosing *Edit ® Node ® Create* and radio button *Sub*.
- On the *Output options* tab page of the new node, determine in the *Output structure* area in which line and cell of the template you want to append the node. If you do not make an entry here, the node is output in the current cell, that is, the cell in which data was last output. Note, however, that text that does not fit into a cell is not output.
- You can also assign multiple nodes to a cell. The output order in the cell is then determined by the order of the nodes in the navigation tree.
- Note that no settings are permitted for frames or shading on the *Output Options* tab page. You have to make these settings on the *Template* tab page.

Template

Name of passenger			Output date			
To	Airline	Flight	Cl.	Date	Time	Status
Flight Price	Form and serial number					
Tax						
Total						
Do not write on and stamp this field						



Background Picture for Page

Name of passenger			Output date			
To	Airline	Flight	Cl.	Date	Time	Status
Flight Price	Form and serial number					
Tax						
Total						
Do not write on and stamp this field						

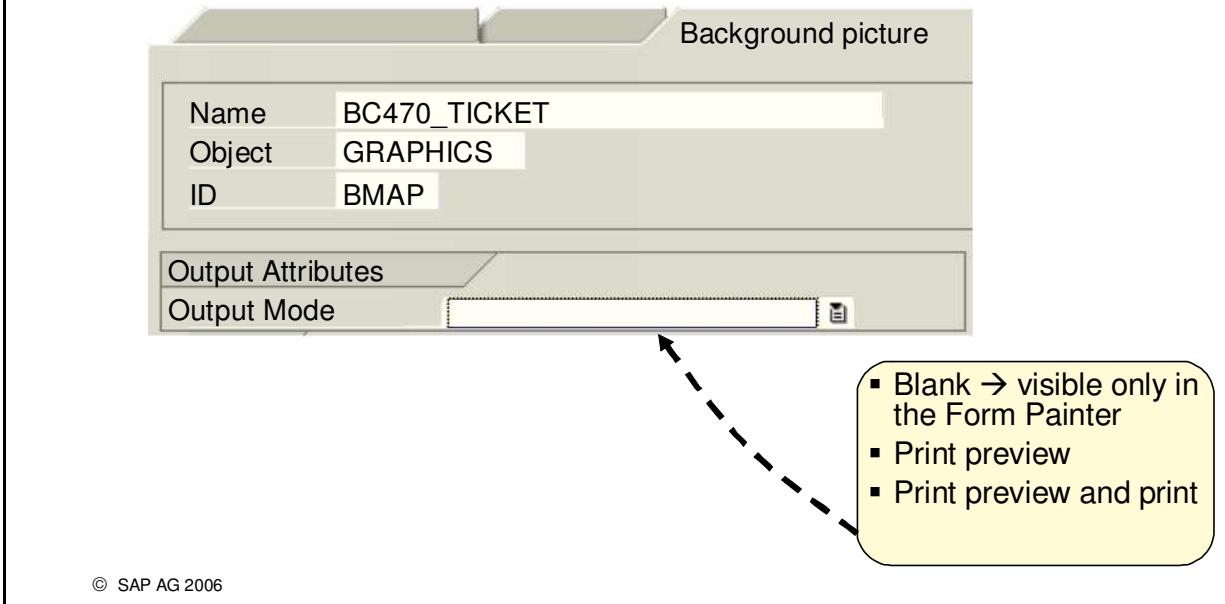


SE78

- You can create templates following the procedure we have just described. An alternative procedure, which is useful in the case of complex templates, is to scan an external form, import it using graphics administration (transaction SE78), set the graphic as the background picture for a page, and then redraw the form in the Table Painter.
- To do this, follow the steps described on the next slides. (It is assumed that the form has already been imported into the SAP System as a graphic.)

1

Graphic as background picture for page



1. On the *Background picture* tab page of the page in which you want to position the template, select the scanned form, that is, the graphic (see also Unit 3). Determine in the *Output mode* combo whether you want to print the scanned graphic or not. If you want to print your data in an existing form, leave the field initial. If you want to print the background picture on blank paper, choose *Print preview and print*.

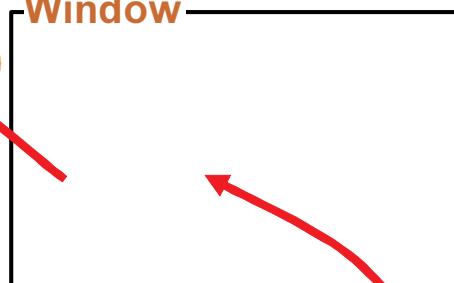
Update the preview of the Form Painter by choosing *Enter* in the maintenance screen.

Creating Templates by Redrawing III

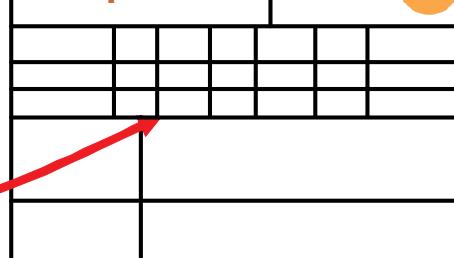
Background picture

Name of passenger		Out date					
To	Airline	Flight	Cl.	Date	Time	Status	
Flight Price	Form and serial number						
Tax							
Total	Do not write on and stamp this field						

Window



Template



→ Table Painter

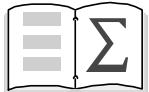
**Setting:
Transparent tables**

**Vertical alignment:
Absolute (from top)**

© SAP AG 2006

2. Use the Form Painter to place a window exactly over the graphic.
3. Create a template in this window. The template width should be the same as the window width. On the *Template* tab page, choose *absolute (from top)* as the *Vertical alignment*. Start the Table Painter from this tab page. Ensure that *Display background picture* is selected. In addition, the *Transparent tables* checkbox must be selected on the *General* tab page of the Table Painter. You can navigate to this tab from the Form Painter settings. Click the last icon in the toolbar of the Form Painter.
4. Now redraw the original form you scanned. To make redrawing easier, you can deselect the *Align tables with grid* checkbox in the properties for the Table Painter.

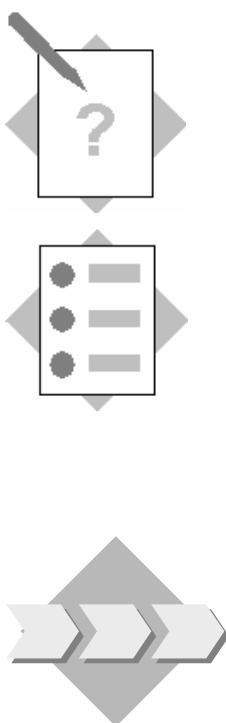
If the original form is very complex, you may need to create several templates above or next to each other.



You are now able to:

- Explain the difference between tables and templates
- Use the Table Painter to create tables and templates
- Create control levels
- Create headers and footers

© SAP AG 2006



Unit: Tables and Templates

Topic: Tables

After completing these exercises, you will be able to:

- Create a Table
- Create line types
- Output text in tables
- Define control levels for tables

Your task: Enhance your existing invoice form and create a "real" table to output actual customer bookings.

- Copy template for the form:** BC470_DATAS
Package (for all exercises): ZBC470_##
Name of the form to be created: ZBC470_##_TABLS
Model Solution: BC470_TABLS
Application program for testing purposes: SAPBC470_DEMO

1 Copy template

Copy the form that you used in the last exercise (ZBC470_##_DATAS) to ZBC470_##_TABLS. Alternatively, copy the copy template (BC470_DATAS).

2 Define table

Note that you do not need the Form Painter for tables. Therefore, you can hide it to allow more space for the maintenance screen.

Output the table of bookings for each customer in the main window instead of the existing text node DUMMY_TABLE. The application program passes the data as an internal table (IT_BOOKINGS) to the form.

2-1 Create table and define data for table

In the MAIN window, create a table node called BOOKINGS after the node INTRODUCTION. The system fills this form table by reading each line of the internal table IT_BOOKINGS into the work area GS_BOOKING. (You defined GS_BOOKING as a global variable in the previous exercise.)

Note that IT_BOOKINGS contains the data for **all** customers selected on the selection screen. In the WHERE condition, define that the field CUSTOMID of the internal table is identical to IS_CUSTOMER-ID. (Note that IS_CUSTOMER contains all important information about the customer for whom the invoice is to be created. It is an interface parameter which is filled in the application program.)

Perform a local check.

2-2 Determine table layout

The table should have the following structure:

<i>Flight</i>	<i>Flight date</i>	<i>Price</i>
<i>Bookings for AA</i>		
AA0017	10/16/2007	1,200.00 EUR
AA0017	10/17/2007	1,200.00 EUR
AA0026	11/18/2007	700.00 USD
<i>Number of bookings for AA: 3</i>		

2-2-1 Use the Table Painter to create the three line types required:

- Line type DETAILS with three cells. This is required for the column heading and the items. Use the table above to determine the cell widths. It is useful to display the ruler.
- Line type SUBHEADING with one shaded cell (for the subheadings).
- Line type SUBTOTAL with one cell (for the end of a control level).

2-2-2 View the result in the detail view. If you like “pretty” numbers, you can adjust the cell widths as follows, for example:

- Set the table width to 15 cm.
- The cell widths for the line type DETAILS should be 2.5 cm, 4.0 cm and 8.5 cm.

2-2-3 Prevent page breaks from occurring within each line type.

2-2-4 Perform a local check.

- 3 The following is to be output for each booking:
- Carrier ID and connection number
 - Flight date
 - Price and currency
- 3-1 Create a table line for the main area of the table and choose the type DETAILS. As a consequence, three cells should be automatically inserted in the navigation tree. Assign suitable names to the table line and cells. Create a text element FLIGHT in the first cell, a text element DATE in the second cell, and PRICE in the third cell.
- 3-2 Include the carrier ID and the connection number in FLIGHT. Use the field list and drag the fields CARRID and CONNID of the global field GS_BOOKING into the text node. Select the paragraph format TB ("Cell in table body").
- 3-3 Output the flight date (GS_BOOKING-FLDATE) in DATE. Choose the paragraph format TB.
- 3-4 Output a tabulator as well as the price and currency (GS_BOOKING-FORCURAM and GS_BOOKING-FORCURKEY) in PRICE. Choose the paragraph format TB. The decimal tabulator is already contained in paragraph format TB.
Define the following formatting options for GS_BOOKING-FORCURAM:
- Output length: 13
 - Decimal places: 2
- 4 Define column headings
- The table is to have the column headings "Flight", "Flight date" and "Price".
- 4-1 Create a table line for the table header and choose the type DETAILS. As a consequence, three cells should be automatically inserted in the navigation tree. Assign suitable names to the table line and cells. Create a text element H_FLIGHT in the first cell, a text element H_DATE in the second cell, and H_PRICE in the third cell.
- 4-2 Enter "Flight" in the text node H_FLIGHT, "Flight date" in the text node H_DATE, and "Price" in the text node H_PRICE.
Choose the paragraph format TH ("Cell in table header") for all three text nodes.
Collapse the column headings in the navigation tree again to obtain an overview.

5 Define control levels

The table is to have control levels for the airline carriers. Select the beginning and the end of the bookings for each carrier.

- 5-1 Make the necessary entries on the *Data* tab to ensure that the table is sorted by the carrier (CARRID) and event nodes are created for the beginning and the end of the control level in the navigation tree.

5-2 Insert subheading

- 5-2-1 To output a subheading for each airline carrier, create a table line for "CARRID Event on Sort Begin" and choose the type SUBHEADING. As a consequence, a cell should be automatically inserted in the navigation tree. Assign suitable names to the table line and cell.
- 5-2-2 Create a text node called H_CARRIER_HEADING in the new cell. Enter the text "Bookings for <carrier ID>". Use the field list to insert the appropriate ID (GS_BOOKING-CARRID). Choose the Italic character format and the TH paragraph format for the subheading.

5-3 **Optional:** Insert intermediate counter

The relevant number flight bookings is to be output after each airline carrier.

- 5-3-1 Create a global counter variable GE_CNT as an integer variable. On the *Calculations* tab page of the table, make appropriate entries so that GE_CNT increases by 1 with each booking .Make sure the counter is reset after each change of airline.
- 5-3-2 Create a table line for "CARRID Event on Sort End" and choose the type SUBTOTAL. As a consequence, a cell should be automatically inserted in the navigation tree. Assign suitable names to the table line and cell.
- 5-3-3 Insert a text node called CARRIER_SUBTOTAL as a subnode of the new cell and output (in paragraph format TO) the intermediate counter.

6. Delete the old text node DUMMY_TABLE since it is no longer required.
7. If you reduced the height of the main window on the page FIRST in one of the previous exercises and now use this form template, increase the height again.
8. Test your form using the program SAPBC470_DEMO.



Unit: Tables and Templates

Topic: Tables

1 Copy template

See the exercise in Unit 3.

2 Define table

2-1 Create table and define data for table

Choose *Create → Table* from the context menu of the text node INTRODUCTION. Change the name to BOOKINGS and enter a description.

On the *Data* tab page, select *Internal table* and enter: IT_BOOKINGS INTO GS_BOOKING. Enter the following for the WHERE condition: CUSTOMID = IS_CUSTOMER-ID.

Choose the 'Check' pushbutton on the maintenance screen.

2-2 Determine table layout

You define the layout on the *Table* tab.

2-2-1 Make sure that the *Draw Lines and Columns* option is activated (pencil pushbutton in the bottom toolbar of the Table Painter). A line type is automatically created. Create two other line types by holding down the left mouse button and drawing horizontal lines in the Table Painter. To create the required names DETAILS, SUBHEADING and SUBTOTAL, choose *Rename Line* from the context menu of a line type. Hold down the left mouse and draw two vertical lines for the line type DETAILS to create the required cells.

To add borders to the table, deactivate the *Draw Lines and Columns* option (pencil pushbutton in the bottom toolbar of the Table Painter). Choose an appropriate box color and line width, and then choose black with an intensity of 0% for the shading. Select all line types (with Ctrl-A) and choose the *Outer Frame* and *Inner Frame* buttons in the top toolbar of the Table Painter.

To shade the line type SUBHEADING, position the cursor on the line type SUBHEADING and choose a shading. Choose the *Use Shading* pushbutton (in the top toolbar of the Table Painter).

2-2-2 Choose *Details* (top right in the Table Painter) to go to the detail view and, if necessary, change the entries that have been copied by the Table Painter.

2-2-3 In the detail view, select the *No page break* checkbox for all line types.

2-2-4 Choose the 'Check' pushbutton on the maintenance screen.

3 Output contents in tables

- 3-1 Use the context menu of the main area of the table to create a table line, and call it R_ITEM. On the *Output Options* tab page, choose ITEM as the line type. Name the resulting cells C_FLIGHT, C_DATE, and C_PRICE, for example. Using the context menu, create a text node for each of these cells and call them FLIGHT, DATE, or PRICE.
- 3-2 Select the text node FLIGHT and go to the *General attributes* tab. Use the mouse to drag the fields GS_BOOKING-CARRID and GS_BOOKING-CONNID from the field list into the editor of the text node. Choose the paragraph format TB from the list. (To assign paragraph formats, you simply have to place the cursor in the paragraph that is to be formatted. You do not need to select anything in particular.)
- 3-3 Repeat these steps for the text node DATE and GS_BOOKING-FLDATE. Choose the paragraph format TB from the list.
- 3-4 Repeat these steps for the text node PRICE and GS_BOOKING-FORCURAM and GS_BOOKING-FORCURKEY. Choose the paragraph format TB from the list. Set a tab by pressing the tab button on your keyboard. Place your cursor on GS_BOOKING-FORCURAM. Choose the *Change Field* pushbutton (the pencil pushbutton in the Table Painter). On the subsequent dialog box, change &GS_BOOKING-FORCURAM& into &GS_BOOKING-FORCURAM(13.2)&.

4 Define column headings

- 4-1 Use the context menu of the header (right mouse click) to create a table line, and call it R_HEADING. On the *Output Options* tab page, choose DETAILS as the line type. Name the resulting cells CH_FLIGHT, CH_DATE, and CH_PRICE, for example. Using the context menu, create a text node for each of these cells and call them H_FLIGHT, H_DATE, or H_PRICE.
- 4-2 Enter the texts in the editor for the new text elements (*General attributes* tab page). Choose the paragraph format TH from the list.

5 Define control levels

- 5-1 On the *Data* tab page of the table BOOKINGS, enter CARRID as the field name for the sort criteria and select the *Event on Sort Begin* and *Event on Sort End* checkboxes. This automatically inserts the nodes for the control levels in the navigation tree.
- 5-2 Insert subheading
 - 5-2-1 Use the context menu (right mouse button) of the event node "CARRID Event on Sort Begin" to create a table line. Call it R_CARRIER_SUB_HEADING, for example, and choose SUBHEADING as the line type on the *Output Options* tab page.
 - 5-2-2 Create a text node called CARRIER_SUBHEADING in the cell that has now been automatically created. Enter your text in the editor of the text node CARRIER_HEADING (*General attributes* tab page).

5-3 Optional: Insert intermediate counter

- 5-3-1 In the global definitions, create the variable GE_CNT by entering: GE_CNT TYPE I. On the *Calculations* tab page of the table, select or enter the following:
- Operation: Number
 - Target field name: GE_CNT
 - Event: Before loop
 - Reset: Sort criteria
 - For field name: CARRID.
- 5-3-2 You have already created the event node "CARRID Event on Sort End" in exercise 5-1. Use the context menu (right mouse button) to create a table line, specify a name (for example, R_CARRIER_SUBTOTAL), and define SUBTOTAL as the line type on the tab page. This automatically inserts a new cell in the navigation tree. Name it C_CARRIER_SUBTOTAL.
- 5-3-3 Create a text node called CARRIER_SUBTOTAL as a subnode of the cell C_CARRIER_SUBTOTAL. Use the field list to output the intermediate counter GE_CNT.

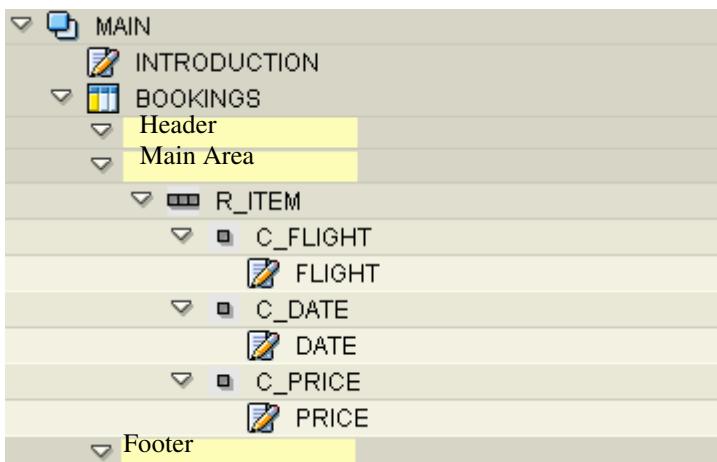
- 6 Delete the text node DUMMY_TABLE using its context menu (right mouse button).
- 7 If required, increase the size of the window MAIN on the page FIRST using the Form Painter.
- 8 Test form

Proceed as usual.

You may notice that the background picture is covered by the line shading. You can remove or position the background picture in such a way that it is not covered by the line shading.

The following three screen shots should help you:

When completely expanded, the table BOOKINGS should appear in the navigation tree as follows at the end of task 3:



When completely expanded, the table BOOKINGS should appear in the navigation tree as follows at the end of task 4:



When completely expanded, the table BOOKINGS should appear in the navigation tree as follows at the end of task 5:





BC470 Form Printing with SAP Smart Forms



1 Overview



2 SAP Smart Forms: Overview



3 First Steps with the SAP Form Builder



4 Texts, Addresses, Graphics

&WA&

5 Data in Forms



6 Tables and Templates



7 Flow Control



8 Integration into Application Programs



9 Smart Styles

...

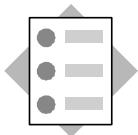
Appendix

© SAP AG 2006

Contents:

- **Node conditions**
- **Alternatives**
- **Program lines**
- **Initialization**
- **Global form routines**
- **Final window**
- **Command nodes**
- **Loops**
- **Folders**

© SAP AG 2006



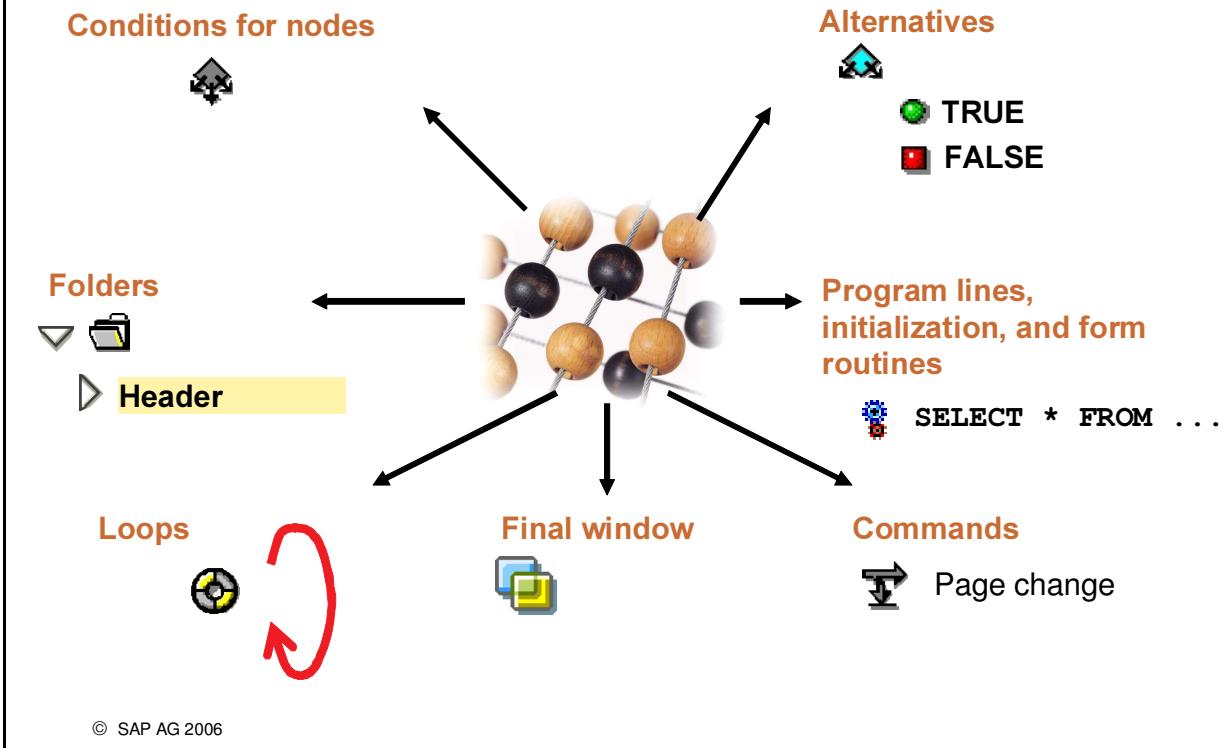
After completing this unit, you will be able to:

- **Explain how the various nodes can be used for flow control**
- **Create flow control nodes**

© SAP AG 2006

Flow Control: Overview

SAP



- The form elements presented to you up to this point are processed in a predefined order. Starting with the first page, the nodes of the tree structure are processed from top to bottom. It is helpful to imagine that all nodes are expanded.
- In some cases, however, the system can only determine at runtime which parts of a form should be processed. You have already seen an example with tables: Table lengths and sequences of line types depend on the data records that are read.
- This unit deals with other flow control options provided by SAP Smart Forms:
Output conditions for nodes.

Alternatives: An alternative is a condition that controls two nodes. One node is processed if the condition is fulfilled, the other is processed if the condition is not fulfilled.

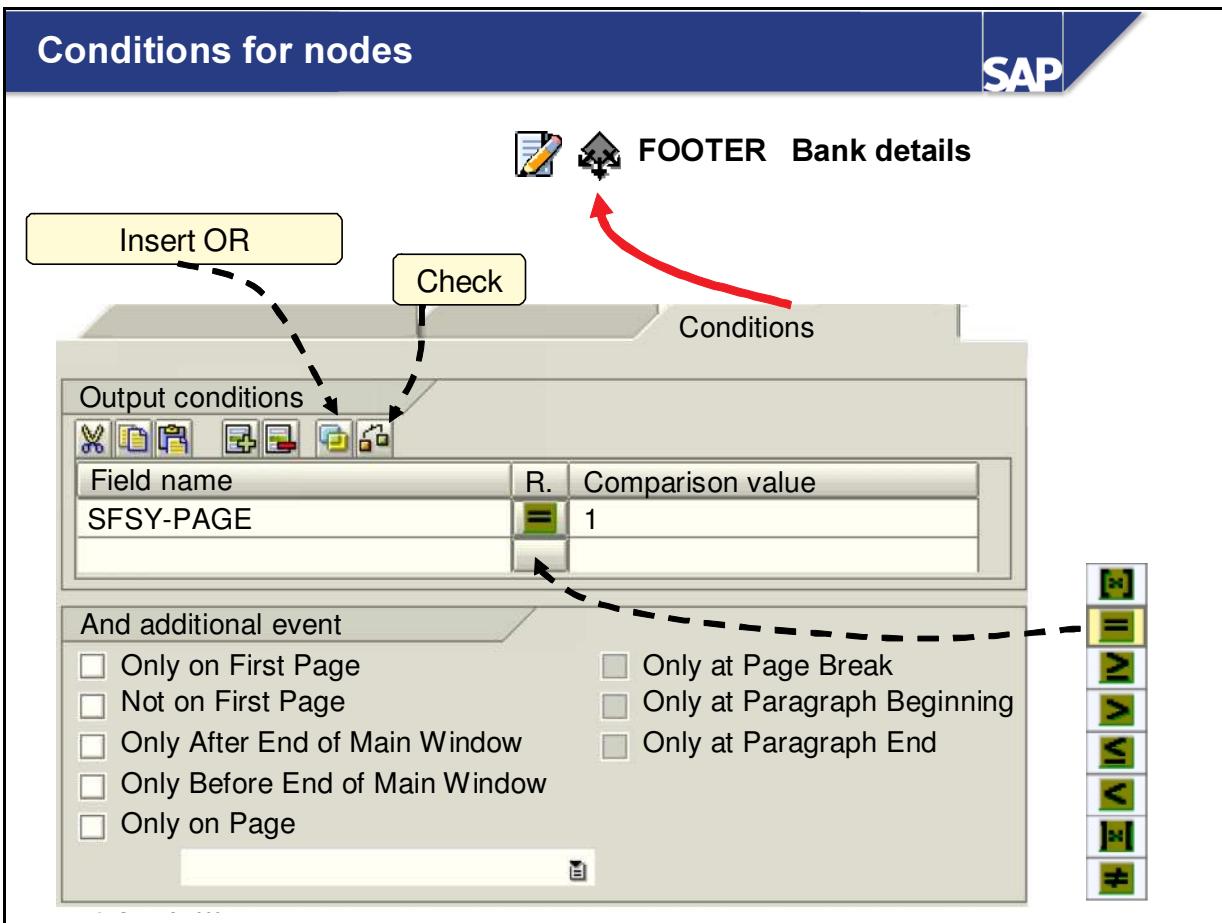
Program lines, initialization nodes and **global form routines** allow you to integrate ABAP statements into your form without having to adjust the application program.

Final windows are only processed in a second repetition.

Command nodes are used for dynamic page breaks, for example.

Subnodes of **loops** are processed several times..

Folders are used for grouping nodes



- Most nodes have the *Conditions* tab page. You can define two types of conditions for processing each node and all of its subnodes:

Field comparisons:

Enter a field name without ampersands for each line. Then select a relational operator (the default operator is *Equal to*), by clicking the pushbutton between the two columns, and enter a comparison value. This value can be a field or a fixed value.

The fields must be defined in the form interface or in the global definitions, or they must be system fields of SAP Smart Forms (SFSY-...). In final windows, you can only use system fields FORMPAGES and JOBPAGES in conditions, because otherwise their value is not set.

If you insert several conditions, these are linked by a logical AND. Using the pushbutton highlighted above, you can also define an OR relationship.

Specific events: The options available depend on which node is selected. For example, *Only at start of paragraph* is available only for headers or footers of tables, and for complex paragraphs.

Multiple events are always linked with OR.

- Check your entries using the 'Check' pushbutton on the tab page.
- In the navigation tree, the icon shown above is added to the respective node if you have defined a condition for it.
- If you use identical windows, graphic windows or address windows on different pages, then each has its own *Conditions* tab page.

Alternatives

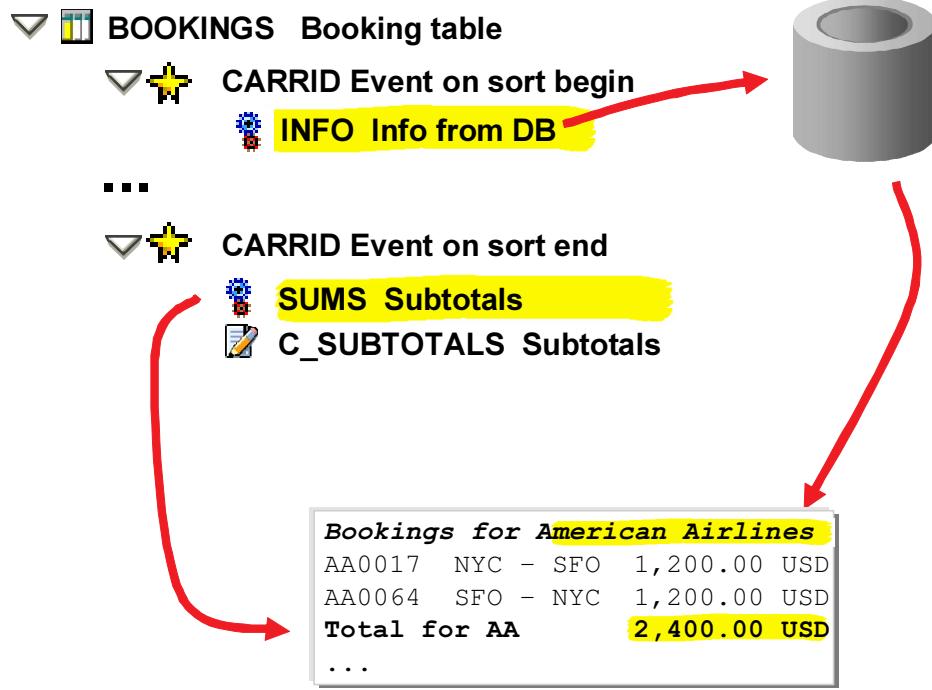
SAP

Field name	R.	Comparison value
IS_CUSTOMER-ID	<	50

© SAP AG 2006

- An alternative is a node with two subnodes (which each contain further subnodes). The condition(s) you enter on the *General Properties* tab page of the alternative determine which of the two subnodes is processed. If the condition is fulfilled, the TRUE node is processed including all of its subnodes. If the condition is not fulfilled, the FALSE node is processed together with its subnodes. This query is similar to that of the ABAP commands IF and ELSE.
- You can create an alternative in the same way as other nodes: using the context menu of the navigation tree or by choosing *Edit → Node → Create* from the menu.
- You can enter the same types of conditions as on the *Conditions* tab page of other nodes, such as field comparisons or specific events.
- Alternatives can be nested. This allows you to define complex queries.
- Please note the following important difference:
On the *General Properties* tab page, you set the conditions that determine whether the node TRUE or the node FALSE is processed.
However, on the *Conditions* tab page, you set the conditions that must be fulfilled for the alternative to be processed at all.

Program Lines I



© SAP AG 2006

- Nodes of type *Program lines* allow you to integrate ABAP code into your form. There are many situations where this is helpful, for example:
 - You need data that is not provided by the application program actually responsible for data retrieval.
 - You could modify the program, but this can be a very complex task and you would not be able to benefit from enhancements made to this program during an upgrade.
 - You want to reset counter variables.
 - You want to calculate complex subtotals and totals, for example, within a table.
- You create a program line node using the context menu or by choosing *Edit → Node → Create*. Note the processing sequence. Nodes are processed from top to bottom in the navigation tree. The results of the program lines are therefore only available for those nodes that are processed afterwards.
- Since program line nodes cannot generate output and cannot have subnodes, they do not have an *Output Options* tab page.

Program Lines II

SAP

F1 = ABAP Help!

Syntax check

Sample statement

Pretty printer

```

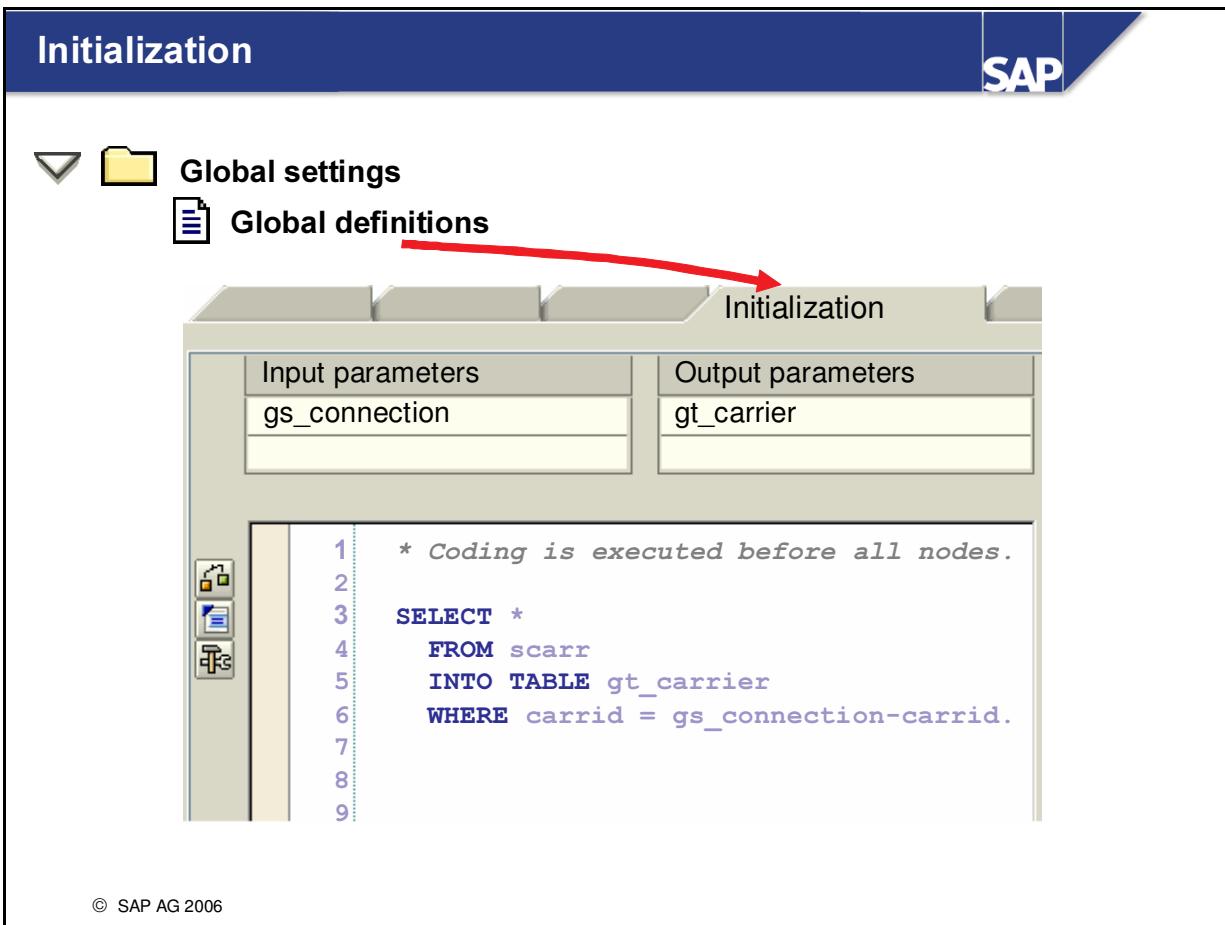
General Attributes
Input parameters: gs_booking-carrid
Output parameters: ge_name, ge_cnt

1 SELECT SINGLE carname
2   FROM scarr
3     INTO ge_name
4   WHERE carrid = gs_booking-carrid.
5
6   ge_cnt = ge_cnt + 1.
7
8
9

```

© SAP AG 2006

- Program lines in SAP Smart Forms are similar to subroutines in ABAP programs. This means that you must determine the interface, but you can also work with local variables that you create with the DATA statement. The input and output parameters must be globally recognized in the form, that is, they must be defined in the interface or in the global definitions.
- The distinction between input parameters and output parameters is only made for structuring purposes. It has no effect on the potential for modifying parameters, as both input and output parameters are transferred to the program line node for reference. As a consequence, changes to values in input parameters are permanent and do not only apply within the program lines.
- SAP Smart Forms system fields (sf...) or system fields of the ABAP system structure SYST (sy...) do not need to be declared, but can be used directly in the code. System fields should be read-only.
- If you have set the front-end editor as the ABAP editor for Smart Forms (in the following menu in the Form Builder: *Utilities* → *Settings* → tab page *Editor* → radio button *Text Edit Control*), in the general workbench settings, you can (as of SAP Web Application Server 6.20) specify which of the two front-end editors you want to use. If you would prefer to use the old, line-oriented ABAP Editor, you can set it from the Form Builder using the following menu: *Utilities* → *Settings* → tab page *Editor* → radio button *Text Table Control Editor*.



- On the *Initialization* tab page in global definitions, you can enter the ABAP code of your choice that you want to execute before the start page is processed. In particular, you can assign values to global data before the actual form formatting process starts.
- The handling is identical to that for normal program lines.

The screenshot shows the SAP Global Form Routines interface. At the top, there's a navigation bar with a downward arrow, a folder icon labeled "Global settings", and a document icon labeled "Global definitions". A red arrow points from the "Global definitions" link down to a code editor window. The window has a title bar "Form routines" and contains the following ABAP code:

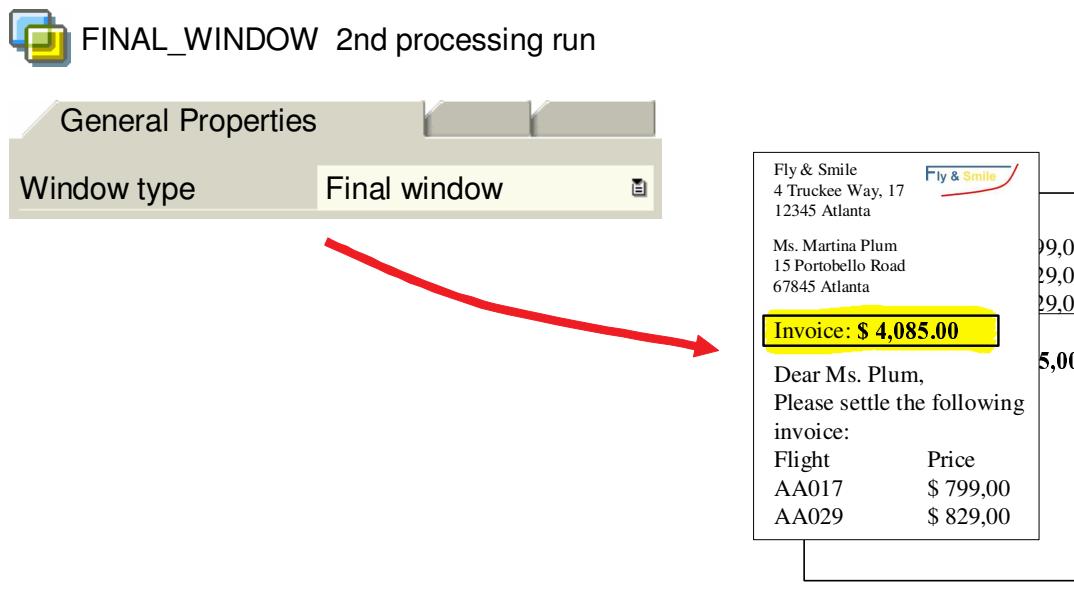
```
1 FORM frequently_used
2 USING      pv_one      TYPE i
3          pv_two      TYPE sy-datum
4 CHANGING  cv_three   TYPE p_debt.
5
6 * insert your coding here
7
8 ENDFORM.
```

Two red arrows point from the bottom of the code editor to two entries below it. Each entry consists of a small blue and red icon followed by text: "CODING1 Calculation of ..." and "CODING2 Calculation of ...".

At the bottom left of the interface, there's a copyright notice: "© SAP AG 2006".

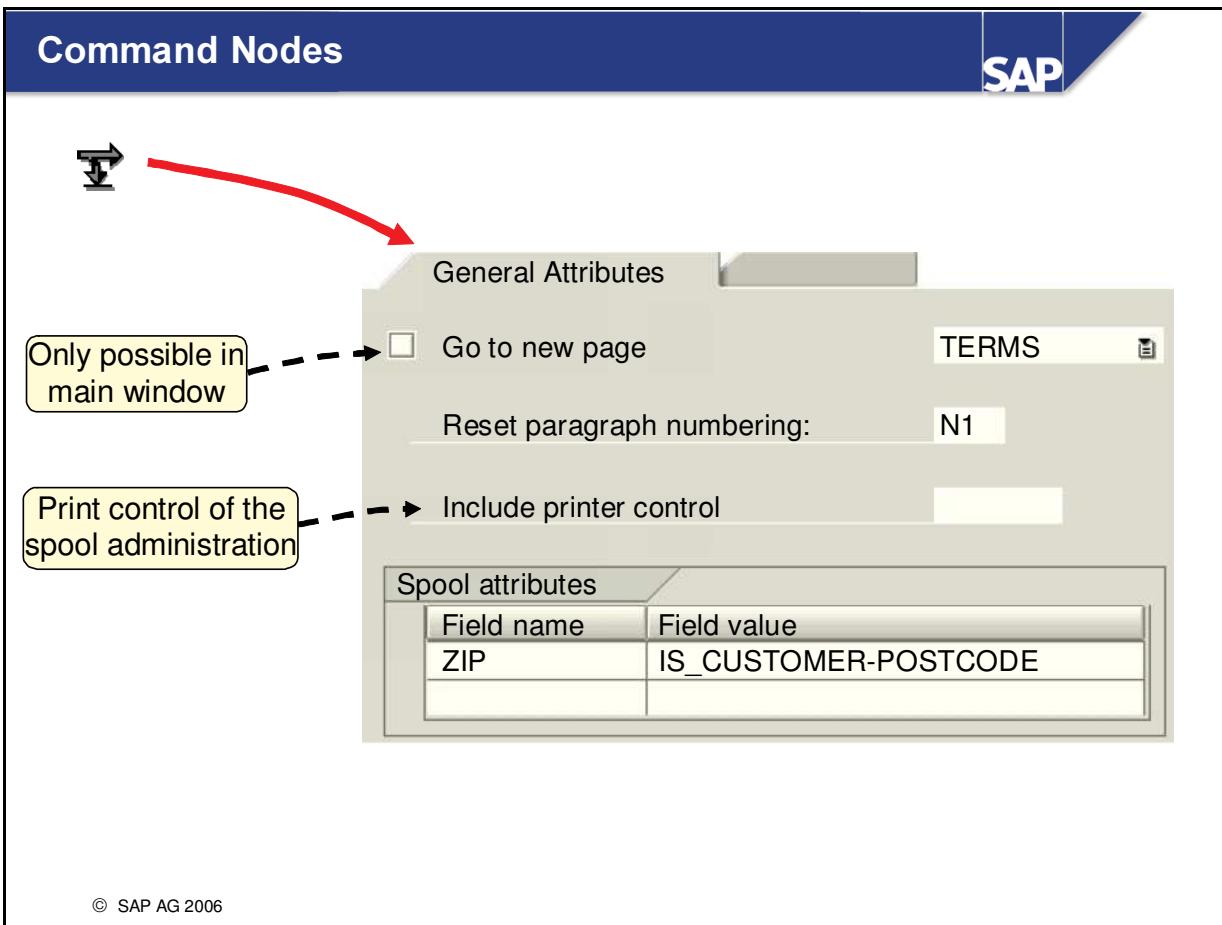
- If you have coding that is used in different program line nodes, it makes sense to move it into global subroutines (forms) and then call them as required. You create these subroutines in the editor of the *Form Routines* tab page of *Global definitions*.
- The syntax is normal ABAP syntax. You define the routines using `FORM <form> [TABLES ...] [USING ...] [CHANGING ...]`. You also use normal ABAP syntax for calling a defined form using `perform <form>`. For further information, see the ABAP documentation.
- Subroutines that you create in program lines are known only there and cannot be used in other program lines.

Final window



© SAP AG 2006

- You may sometimes need to use a condition on the first page to query the total number of pages of the form being processed (for example, for bar codes in machines for filling envelopes). The SFSY-FORMPAGES variable does already exist but its value is not determined until the end of form processing, and the form processor subsequently inserts it in a second processing run. Or you want to output a total on a page although it is only calculated during the form processing. The problem here is that fields for which a value has not yet been determined cannot be evaluated in a normal window.
- As of the SAP Web Application Server 6.10, there is a new window type: *Final window*. Windows of this type are only processed in a second processing run, by which time the values for all form variables are already known (including SFSY-FORMPAGES).
- Apart from the time of processing, final windows are identical to secondary windows. This means that text that does not fit into the window is truncated, and each page in a final window can have an individual height, width, and positioning.
- Note that using SFSY-FORMPAGES (and also SFSY-JOBPAGES) has a negative effect on performance, because all output pages are saved in the main memory until the end of the form, or until the end of the whole print output.



- A command node enables you to do the following:

Go to new page: A page break normally occurs if the main window of a page is full. The next page processed is the page that you entered on the *General Properties* tab page of the page. In some cases, however, you may want to process a different next page, possibly based on conditions. This is the case, for example, if a page is output several times (that is, is its own next page), but another page is to be processed afterwards. You then use this option and specify which page the system should process next. Note that this option is allowed only within main windows. Otherwise, the function module issues an error message. All nodes after a manual page break in the main window are not processed on the current page. All secondary windows of the current page are, however, still processed before the page break.

Paragraph numbering: If you enter an outline paragraph here (which must exist in the style used), the numbering of this paragraph and all associated paragraphs at lower-level outline depth is reset to initial. Paragraph formats without outline attributes are ignored. See Unit 9 - *Smart Styles*.

Include Printer Control: Here you can send a print control to the output device. This allows you to use special features of your printer. Print controls are managed in spool administration and are converted into printer-specific escape sequences during output.

You can also define free attributes for the spool request with a value of your choice. These are then evaluated using the table TSP02A or, for example, using the report RSOPRNT. For more information, see SAP Note 359379.

The screenshot shows a SAP Fiori application titled "Loops". At the top, there are two nodes: "LOOP_SUMS Sum Loop" (highlighted with a yellow bar) and "SHOW_SUMS Display Sums". Below them is a "Data" tab page. On the "Data" tab, there is a configuration for a "LOOP Loop". The configuration includes a checked checkbox for "Internal table", the table name "GT_TOTALS", and the assignment type "INTO" followed by a field symbol "GS_TOTAL". A red arrow points from the configuration area to a callout box containing the output: "Sum total:" followed by three currency amounts: "2,400.00 USD", "1,250.00 EUR", and "4,256.37 GBP".

- A loop is very similar to a table, because in loops, internal tables are also read line-by-line. However, you do not determine where the data is output, which means there are also no line types. Therefore, content nodes (for example, texts) are direct subnodes of a loop.
- Tab page *Data*: Enter the name of the internal table over which the loop is to be executed and the associated work area (assignment type *into*) or the field symbol (assignment type *assigning*). You can specify a line range, determine one or more WHERE conditions, and sort the internal table before it is processed. Sorting the internal table is a prerequisite for sort levels (control levels).
- Tab page *Events*: If you select *Header* and/or *Footer*, event nodes are automatically created in the navigation tree. You can, for example, enter text as a direct subnode of these event nodes. You can output headers at the beginning of the loop and/or after a page break. Similarly, footers can be output at the end of the loop and/or before a page break. You must specify a height for the footer so that the form processor reserves sufficient space.
- Loops can be nested. It is also possible to use tables or templates in loops.
- Examples for the use of loops:
 - Output of an internal table containing amounts sorted by currency
 - Output of all booked classes.

Common condition for multiple nodes:

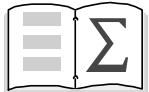
-    **ALL_OR_NOTHING** Flight, Date, Price
 -  **C_FLIGHT** Flight
 -  **C_DATE** Date
 -  **C_PRICE** Price

Clarity:

-   **DUNNING** Dunning texts
 -   **DUNN01** Dunning level 1
 -   **DUNN02** Dunning level 2
 -   **DUNN03** Dunning level 3
 -   **DUNN04** Dunning level 4

© SAP AG 2006

- The larger a form, the more complex its node hierarchy. For a clearer overview, you can create folders and group nodes in these folders.
- Examples of the usage of folders:
Nodes of a template that are assigned to a specific cell or line can be more easily identified (or moved) in the navigation tree if one folder exists for each cell or line.
A dunning form has different dunning texts of which only one is to be output depending on the reminder days exceeded. Create these texts (with conditions) and group them into folders.
Several nodes share the same condition. Instead of setting this condition individually for each node, you can create a folder and then assign the condition to this folder.
- Folders can also be used to output text with a footer and/or header. Like loops, folders have the *Events* tab page. If you select the *Header* and/or *Footer* checkbox on this tab page, one or two event nodes are displayed in the navigation tree. Create your text nodes as subnodes of these event nodes.

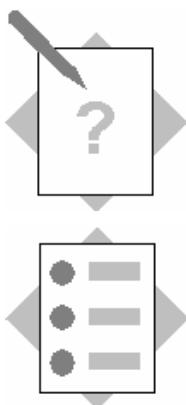


You are now able to:

- Explain how the various nodes can be used for process control
- Create process control nodes

© SAP AG 2006

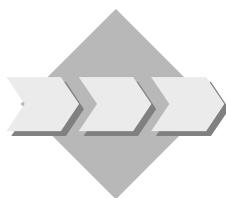
Exercises



Unit: Process Control

After completing these exercises, you will be able to:

- Create node conditions
- Use alternative nodes
- Create program nodes
- Create folders
- Create command nodes (optional)
- Create loops (optional)



Your task: Add more details to the invoice form.

Award a discount for some circumstances, transfer the English-speaking customers to a separate agent, and avoid inconvenient page breaks.

Optionally, you can also attach the terms and conditions of business and display the total sum of the invoice listed in order of currency.

Copy template for the form:

BC470_TABLS

Package (for all exercises):

ZBC470_##

Name of the form to be created:

ZBC470_##_FLOWS

Model Solution:

BC470_FLOWS

Application program for testing purposes:

SAPBC470_DEMO

1 Copy template

Copy the form that you used in the last exercise (ZBC470_##_TABLS) to ZBC470_##_FLOWS. Alternatively, copy the copy template (BC470_TABLS).

2 Award discount

Note that you do not need the Form Painter for this task. Therefore, you can hide it to allow more space for the maintenance screen.

In the customer table read by the application program, there is also an entry for discounts. If the customer has negotiated a discount, the following text is to be displayed at the end of the form: "P.S.: Please note: A discount of ... percent has been taken into account."

- 2-1 As a subsequent node to REGARDS, create a text node called DISCOUNT. Enter the reminder text detailed above.
- 2-2 Define the following condition for the text node: IS_CUSTOMER-DISCOUNT > 0.
- 2-3 Activate your form and test it using the program SAPBC470_DEMO.

3 Choose clerk

If, up to this point, only one clerk was responsible for creating the invoices, the travel agency Fly & Smile should now appoint two clerks: The first clerk would be responsible for the countries USA and Canada, and the second clerk would be responsible for all other countries.

- 3-1 Convert the global constant GC_CLERK into a variable and change the name to GE_CLERK.
Adjust the two positions where GC_CLERK was used up to now: In the text node INFO_TEXT of the window INFO and in the text node REGARDS of the main window.
- 3-2 Create an alternative node called WHICH_CLERK (at the correct position). If the customer lives in the USA or Canada (IS_CUSTOMERS-COUNTRY has the value 'US' or 'CA'), the system should process the node TRUE. In all other cases, the system should process the node FALSE.
- 3-3 Create a program node called CLERK_A in the node TRUE, and a program node called CLERK_B in the node FALSE. In this program node specify a suitable employee name each for the global field GE_CLERK.
To test whether your alternative node actually works, look in the Data Browser (transaction code SE16) for the table SCUSTOM to check which customers in the field COUNTRY have the value DE or US, for example.

4 Prevent page break

Ensure that the closing lines and the discount notification are on the same page. To do this, create a folder called PROTECT with the appropriate output option.

5 **Optional:** Create page for general terms and conditions of business

Create an additional page called TERMS for the general terms and conditions of business. Copy the main window onto this page. The page TERMS should be processed **after** (not: under) the folder PROTECT. Create a suitable command node called SHOW_TERMS. After the command node, create a text node called CONDITIONS, in which you can formulate business conditions as required.

6 Optional: Output totals by currency

For every booking line of the table BOOKINGS, collect the amounts in an internal table and output the content of this internal table before the greeting form in a loop.

- 6-1 Create a global variable GS_SUM of the type BC470_FORCUR, and an appropriate internal table GT_SUMS of the type IT_BC470_FORCUR. General information: The structure type BC470_FORCUR contains the two fields FORCURAM (price of the booking in foreign currency) and FORCURKEY (currency), in accordance with the fields of the same name in the table SBOOK.
- 6-2 In the main area of the form table (for example, after the text node in which you output the flight price), create a program line node with the name ADD_PRICES.

The currency and amount of each booking should be inserted into the internal table GT_SUM as appropriate.

Tip: This is best achieved using the syntax:

```
MOVE-CORRESPONDING gs_booking TO gs_sum.
```

```
COLLECT gs_sum INTO gt_sums.
```

- 6-3 Before the closing lines REGARDS, create a text node called SUMS with the text "Totals by currency".
- 6-4 After the text node SUMS, create a loop called SUM_LOOP for the totals by currency. Use the internal table GT_SUMS to loop into the work area GS_SUM. The output should be sorted by currency.
- 6-5 Create a text node called SHOW_SUM as a subnode of the loop, and output the total and the currency.

7 Activate your form and test it using the program SAPBC470_DEMO.



Unit: Process Control

1 Copy template

See the exercise in Unit 3.

2 Award discount

- 2-1 Use the context menu to create a text node called DISCOUNT as a subsequent node to REGARDS. Enter the text ("P.S.: Please note ...") in the editor (tab page *General Attributes* of the text node). To ensure the correct percentage is output, drag the import parameter IS_CUSTOMER-DISCOUNT to the appropriate point in the text node.

- 2-2 Enter the following on the *Conditions* tab page of the text node:

Field name: IS_CUSTOMER-DISCOUNT

Comparison operator: >

Comparison value: 0.

3 Choose clerk

- 3-1 On the *Global Data* tab page in the global definitions of the form, change the name of the field GC_CLERK to GE_CLERK and deselect the *Constant* checkbox.

- 3-2 Your task is to create an alternative node with program lines. The clerk should be determined within the program lines. It is important to place the program lines at the correct position in the navigation tree as it does not make sense to output the clerk earlier. You must ensure that the alternative node is processed before GE_CLERK is output for the first time, that is, before the text node INFO_TEXT (window INFO) and before the text node REGARDS (main window). To create the alternative node, use the context menu of the first of the two windows. Call it WHICH_CLERK.

On the General Attributes tab page of the alternative node, enter the two conditions: IS_CUSTOMER-COUNTRY = 'US' and IS_CUSTOMER-COUNTRY = 'CA'. (Note the uppercase and lowercase for the entry, and the single quotation marks.) Link these two conditions by an OR. To do this, place your cursor in the second condition line and then click the OR pushbutton (which is the second pushbutton on the maintenance screen).

- 3-3 Use the context menu of the TRUE node (which was generated automatically when you created the alternative node) to create a program lines node called CLERK_A. Define a value for the variable GE_CLERK (in the ABAP editor of the General Attributes tab page), for example: GE_CLERK = 'Mr. Miller'. Make the variable GE_CLERK known to the program lines node by adding GE_CLERK to the list of output parameters. Repeat these steps for the FALSE node and assign another name to this clerk.

Tip: Now, collapse the alternative node for a better overview of the navigation tree.

4 Prevent page break.

Use the context menu of the text node REGARDS to create a folder called PROTECT. Drag the text nodes REGARDS and DISCOUNT into the folder. (When you use drag and drop, the system asks whether to insert the nodes *under* or *after* PROTECT. Choose *under*). Select the *Page Protection* checkbox on the *Output Options* tab of the folder.

5 Optional: Create page for general terms and conditions of business

- Use the context menu of the page NEXT to create the page TERMS.
Copy the main window from the page NEXT to the page TERMS. You can do this, for example, using drag and drop and holding down the CTRL key. After (not under) the PROTECT folder, create a command node with the name SHOW_TERMS, activate the field *Go to New Page*, and enter TERMS as a new page.
 1. It does not matter whether you create the command node on the page FIRST, NEXT, or TERMS, because the content of the main window is the same on all pages.
 2. If you create the command node using the context menu of the folder PROTECT, it is created **under** the folder as standard – you therefore must move it to **after** the folder.
 Create a text node CONDITIONS after the command node, in which to enter your terms and conditions of business.
- Additional tip: If you have used another procedure and your TERMS page has no main window, you must ensure that no subsequent page is entered on the *General Attributes* tab page.

6. Optional: Output totals by currency

- 6-1 In the navigation tree, choose *Global settings* → *Global definitions*. On the *Global Data* tab page, enter GS_SUM TYPE BC470_FORCUR and GT_SUMS TYPE IT_BC470_FORCUR.
- 6-2 In the main area of the table, use the context menu of the text node with the flight price output to create a program line node with the name ADD_PRICES.
Enter the import parameter GS_BOOKING, and the output parameters GS_SUM and GT_SUM. Enter the following coding:


```
MOVE-CORRESPONDING gs_booking TO gs_sum.  
  
COLLECT gs_sum INTO gt_sums.
```
- 6-3 Use the context menu of the PROTECT folder to create a text node, and call it SUMS.
Enter the text.
- 6-4 Use the context menu of the text node SUMS to create a loop with the name SUM_LOOP.
Enter the following on the *Data* tab page: Internal table GT_SUMS INTO GS_SUM. In the lower section of the same *Data* tab page, enter FORCURKEY as a sort criterion.
- 6-5 Use the context menu of the loop to create a text node, and call it SHOW_SUM. Now, output the booking currency and the booking amount (GS_SUM-FORCURAM and GS_SUM-FORCURKEY).



BC470 Form Printing with SAP Smart Forms



1 Overview



2 SAP Smart Forms: Overview



3 First Steps with the SAP Form Builder



4 Texts, Addresses, and Graphics

&WA&

5 Data in Forms



6 Tables and Templates



7 Flow Control



8 Integration into Application Programs



9 Smart Styles

...

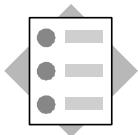
Appendix

© SAP AG 2002

Contents:

- **Generated function module**
- **Customizing the application program**

© SAP AG 2002

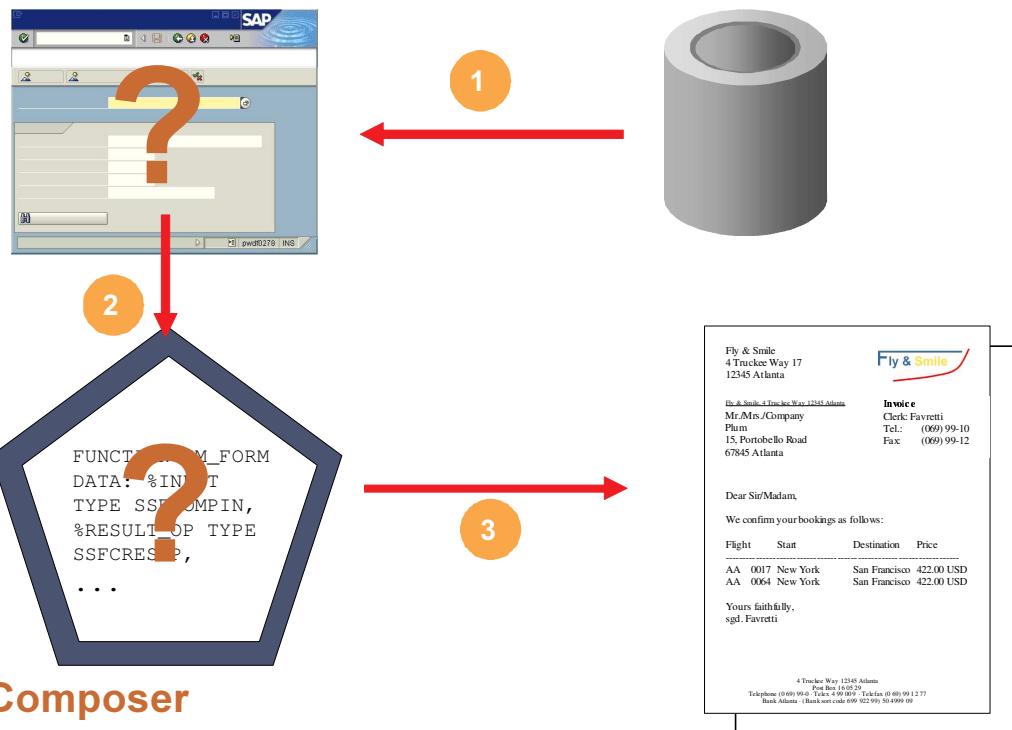


At the conclusion of this unit, you will be able to:

- Explain the interaction between application programs and SAP Smart Forms
- Fill the interface of the generated function module with parameters

© SAP AG 2002

Recap: Creating Documents



© SAP AG 2002

- Here is a recap of the document creation process:

- The transaction checks in Customizing which program to call. This program then reads the data.
- The transaction checks in Customizing which SAP Smart Form to use for the scenario chosen, calls the appropriate function module generated, and thereby triggers the form processing process. The interface is filled with the data read.

When the form processing process is started, the form processor (Composer) is automatically called in the background. The Composer is responsible for formatting the texts according to the layout information stored in the form, filling fields with values at runtime and controlling the page breaks.

a) Data retrieval

b) Name of the generated function module?

c) Calling the function module

```
PROGRAM ...  
DATA:  
    ssf_name      TYPE tdsfname,  
    func_mod_name TYPE rs38L_fnam.  
  
SELECT ... FROM ...  
...  
  
CALL FUNCTION 'SSF_FUNCTION_MODULE_NAME'  
    EXPORTING  
        formname = ssf_name  
    IMPORTING  
        fm_name   = func_mod_name.  
  
LOOP AT ...  
    CALL FUNCTION func_mod_name  
        EXPORTING ...  
        IMPORTING ...  
ENDLOOP.
```

© SAP AG 2002

- From the perspective of the SAP Smart Forms, the application program consists of three parts:
 - a) The data is selected from the database - which is by far the most comprehensive part.
 - b) The name of the function module generated for the form must be determined. Background: The name differs depending on the form and system.
Call the function module SSF_FUNCTION_MODULE_NAME and pass the form name to it. This name is typed as TYPE tdsfname. The return value (import parameter) you get is the name of the generated function module as TYPE rs38L_fnam.
 - c) Actual form processing starts. The generated function module is called once for each document to be created, that is, for example, once for each customer for which you want to create an invoice.

The Generated Function Module

SAP

Global Settings

Form Interface

Pattern!

Import Export Tables Exceptions

```

LOOP AT ...
  CALL FUNCTION func_mod_name
    EXPORTING ...
    IMPORTING ...
    TABLES ...
    EXCEPTIONS ...
ENDLOOP.
  
```

© SAP AG 2002

- Each generated function module has the interface which you defined on the four tabs of the global settings for the SAP Form Builder.
- To integrate function modules into programs, you can generally use the sample statement of the ABAP Editor. However, since the name of the function module differs depending on the form and system, you must use a workaround:
 - In the Form Builder choose *Environment → Function module name* and then use CTRL-Y and CTRL-C to copy the name to the clipboard.
 - Use CTRL-V to paste the name of the function module into the sample statement for CALL FUNCTION. The call of the function module with the correct interface is then inserted at the cursor position.
 - Replace the name after CALL FUNCTION with the variable which is filled by calling SSF_FUNCTION_MODULE_NAME and contains the current name of the generated function module at runtime.

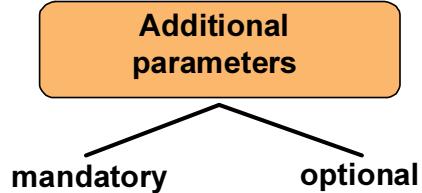
Interface Parameters

SAP

```
CALL FUNCTION func_mod_name
  EXPORTING
    * control_parameters      =
    * output_options          =
    * user_settings            = 'X'
    * is_customer               =
    * ie_color                  =
  IMPORTING
    * job_output_info           =
    * job_output_options        =
  TABLES
    it_bookings                 =
  EXCEPTIONS
    * formatting_error          = 1
    * others                     = 2.
  ...
  IF sy-subrc <> 0.  ... ENDIF.
```

© SAP AG 2006

Standard parameters:
optional



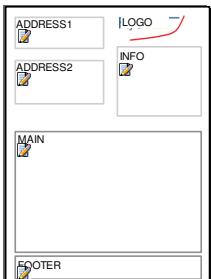
- The generated function module has both mandatory and optional parameters:
 - Optional** parameters. All standard parameters (available in every form) are optional. Their names and types cannot be changed. They include:
 - control_parameters : General control of the output
 - output_options : Output options (see structure ssfcompop in the Dictionary)
 - user_settings : If set to 'X', the user defaults for spool control are used. Otherwise, the output_options values for the printer, immediate output, and spool retention period are evaluated.
 - Parameters for archiving
 - Parameters of the Business Communication Interface for sending forms as e-mails.
 - document_output_info : Number of pages output (field tdfpages)
 - job_output_info, job_output_options : Structures with information about the output (for example, with XML output)
- As of SAP Web Application Server 6.10, you can also mark those separate importing parameters of the form without default values as **optional**.
- Exporting parameters of the form are always **optional**.
- Mandatory** parameters. All table parameters. For SAP R/3 4.6C, all importing parameters of the form are also mandatory, unless they have a default value.
 - For documentation on the standard parameters in SAP R/3 4.6C, see SAP Note 433988.
 - You handle errors as with other function modules by querying the return code (sy-subrc) directly after the call of the function module.

CONTROL_PARAMETERS**(Export parameters of the function module generated)**Type: **ssfctrlop**

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

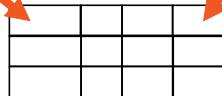
© SAP AG 1999

- One of the most important parameters of the generated function module is **control_parameters**. The following fields are available:
 - **no_open** and **no_close**: These parameters allow you to add several forms to a spool request. To do this, set the parameters as follows:
 - First call: **no_open** = space, **no_close** = 'X'.
 - All subsequent calls: **no_open** = 'X', **no_close** = 'X'.
 - Last call: **no_open** = 'X', **no_close** = space.
 - **device**: Output device ('PRINTER', 'TELEFAX', 'MAIL'). The default value is 'PRINTER'.
 - **no_dialog**: No dialog box for output.
 - **preview**: Print preview
 - **langu**: Language in which you want to print the form
 - **replangu1**, **replangu2**, **replangu3**: Alternative languages if the form does not exist in **langu**
 - **startpage**: Start page other than the top page in the navigation tree of the SAP Form Builder
 - **getotf**: No printout, display or faxing, but OTF (Output Text Format) output to the table **job_output_info-otfdata**.

SAP Form Builder OR Object Navigator?

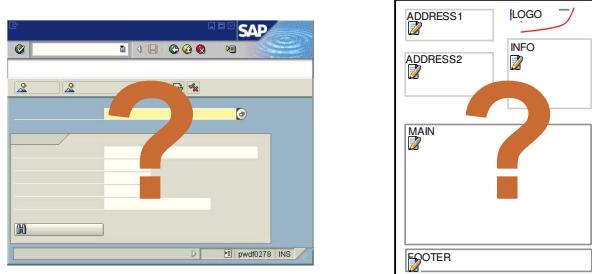
```
• SELECT general_data  
    FROM ...  
• output_options-tddest = ...
```

Customizing:



© SAP AG 2006

- When do you need to modify the application program?
You want to retrieve additional data to be used in all documents of the print run. For performance reasons, you should not make this data selection in the form because it would be executed numerous times.
You want to make output-related settings, for example, deactivate the print preview.
- The transaction determines which part of the application program you actually need to change (for example, a function module or a subroutine). In general: You should change original SAP programs in exceptional cases only.
- If you want to use your own modified copies instead of the SAP originals (programs, forms, or texts), you must make the appropriate entries and settings in Customizing. Note also: The application determines which settings you have to make in Customizing and in which part of Customizing you have to make them.

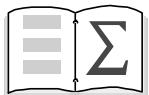


Customizing: When do you use which form?

Application	Scenario	Program	Form
XYZ	1	Z PROG1	Z SMARTIE 1
	2	Z PROG1	Z SMARTIE 2
	3	Z PROG1	Z SMARTIE 3
ABC	1	Z MY PROG2	Z MY SMARTIE 1
	2	Z MY PROG2	Z MY SMARTIE 2

© SAP AG 2002

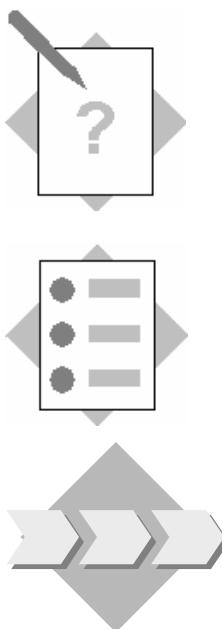
- The settings you have to make in Customizing depend on which application you use.
- Depending on the application, you have to decide between SAPscript or SAP Smart Forms.
- You also determine which form is to be used for the application.
The reason behind this is first of all that you should not directly modify the forms delivered by SAP but always copy them to your customer namespace and then change this copy. You must then inform the transaction about the name of your own form - you do this in Customizing.
Second, some applications allow you to choose between various forms for different scenarios. For example, you can determine in Customizing that for the dunning procedure an individual form should be used for each dunning level (amount and duration of late payment).
- Finally you have to add the print program to Customizing. For example, this can be a function module (in dunning) or a report with specification of the relevant subroutine (in the delivery note). For examples on Customizing, see the appendix. Also see the Customizing documentation on the relevant applications and SAP Notes, in particular, SAP Note 430621.



You are now able to:

- Explain the interaction between application programs and SAP Smart Forms
- Fill the interface of the generated function module with parameters

© SAP AG 2002



Unit: Integration into Application Programs Optional

After completing these exercises, you will be able to:

- Integrate the generated function module for the invoice form into a program
- Set output options for the function module (optional)

Your task: Add appropriate lines of code to the body of an application program so that the invoice form BC470_FLOWS is processed.

Copy template for the program:

SAPBC470_PROGT

Name of the program to be created:

ZBC470_##_PROGS

Package (for all exercises):

ZBC470##

Name of the form to be used:

BC470_##_FLOWS (You can of course also use your own form)

Model Solution:

SAPBC470_PROGS

We recommend that you work with two sessions in this exercise so that you can switch between the ABAP Workbench and the SAP Form Builder.

1 Copy template

In the ABAP Workbench, copy the template program SAPBC470_PROGT to ZBC470_##_PROGS. This program provides a selection screen and retrieves the data. You are responsible for the remaining parts.

2 Determine the function module name

2-1 The program must know the name of the generated function module for the form BC470_FLOWS. Call the function module SSF_FUNCTION_MODULE_NAME at the end of the program code. To do this, use the sample statement. This ensures that the interface is correct.

2-2 Create a variable called GE_FUNC_MOD_NAME of the type RS38L_FNAM. This variable should contain the name of the generated function module for the form after the function module is called.

- 3 Call the generated function module in a loop
 - 3-1 The customer data is stored in the internal table GT_CUSTOMERS. Create a loop in the program at GT_CUSTOMERS into the (existing) work area GS_CUSTOMER.
 - 3-2 Execute the generated function module in this loop. To do this, follow these steps:
 - 3-2-1 In the SAP Form Builder, determine the name of the generated function module for the form BC470_FLOWS (by choosing *Environment → Function module name* from the menu) and use CTRL-Y and CTRL-C to copy the name to the clipboard.
 - 3-2-2 Use CTRL-V to pass the name of the function module in the pattern for CALL FUNCTION. The call of the function module with the correct interface is then inserted at the cursor position.
 - 3-2-3 Add the relevant program parameters (GS_CUSTOMER, GE_COLOR und GT_BOOKINGS) to the mandatory interface parameters IS_CUSTOMER, IE_COLOR und IT_BOOKINGS.
 - 3-3 Test your program.
- 4 **Optional:** Set output options

You will have noticed that the system displays the printer settings dialog box for each customer. Avoid this by filling the parameters CONTROL_PARAMETERS and OUTPUT_OPTIONS with suitable values.

Create a variable called GS_CONTROL_PARAMS of the type SSFCTRLOP and a variable called OUTPUT_OPTIONS of the type SSFCOMPOP. Go to the Dictionary to get a description of the relevant fields or look in your course materials.

To ensure that your options are used you must set the parameter USER_SETTINGS to *space*.

- 5 **Optional:** Use only one spool request

Ensure that all invoices are output within a single spool request.

Assign appropriate values to the fields NO_OPEN and NO_CLOSE of the interface parameter CONTROL_PARAMETERS (see the corresponding slide in this unit).



Unit: Integration into Application Programs

The optional parts are printed in **bold**.

```
*****
* Solution for the exercise of unit 8 of BC470
*****  
  
REPORT    sapbc470_progs.  
DATA:  
  gt_bookings      TYPE ty_bookings,  
  gs_booking       TYPE sbook,  
  gt_customers     TYPE ty_customers,  
  gs_customer      TYPE scustom,  
  ge_color         TYPE tdbtype.  
  
DATA:  
  ge_func_mod_name TYPE rs381_fnam.  
  
DATA:  
  gs_output_options TYPE ssfcompop,           " optional part of exercise  
  gs_control_params TYPE ssfctrlop.          " optional part of exercise  
  
* selection-screen  
SELECT-OPTIONS:  
  so_cust FOR gs_customer-id      DEFAULT 1      TO 3,  
  so_carr FOR gs_booking-carrid  DEFAULT 'AA' TO 'LH'.  
  
* printing options  
SELECTION-SCREEN SKIP 1.  
PARAMETERS:  
  pa_prnt  TYPE tsp01-rqdest VALUE CHECK DEFAULT 'P280'  
            OBLIGATORY VISIBLE LENGTH 4.  
  
* graphics  
SELECTION-SCREEN SKIP 2.  
SELECTION-SCREEN COMMENT 1(30) text-se2.  
PARAMETERS:  
  pa_col    RADIOBUTTON GROUP col,  
  pa_mon    RADIOBUTTON GROUP col DEFAULT 'X'.  
  
*****  
START-OF-SELECTION.  
  
* set ge_color for company logo  
  IF pa_col = 'X'.  
    ge_color = 'BCOL'.  
  ELSE.  
    ge_color = 'BMON'.  
  ENDIF.
```

```

SELECT * FROM scustom
  INTO TABLE gt_customers
 WHERE id IN so_cust
 ORDER BY PRIMARY KEY.

SELECT * FROM sbook
  INTO TABLE gt_bookings
 WHERE customid IN so_cust AND
       carrid   IN so_carr
 ORDER BY PRIMARY KEY.

*****
* Your Coding here:

* find out the name of the generated function module
CALL FUNCTION 'SSF_FUNCTION_MODULE_NAME'
  EXPORTING
    formname      = 'BC470_FLOWS'
  IMPORTING
    fm_name       = ge_func_mod_name
  EXCEPTIONS
    no_form       = 1
    no_function_module = 2
    OTHERS        = 3.
IF sy-subrc <> 0.
  MESSAGE ID sy-msgid TYPE sy-msgtxt NUMBER sy-msgno
    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

* set output options (optional)
gs_output_options-tddest = pa_prnt.

gs_control_params-no_dialog = 'X'.
gs_control_params-preview = 'X'.

* process the form for every customer in it_customers
LOOP AT gt_customers
  INTO gs_customer.

* make sure only one spool request is used
  AT FIRST.
    gs_control_params-no_close = 'X'.
  ENDAT.
  AT LAST.
    gs_control_params-no_close = space.
  ENDAT.

```

```
* call the generated function module
    CALL FUNCTION ge_func_mod_name
        EXPORTING
* The following three parameters belong to the optional
* part of the exercise.
    control_parameters = gs_control_params
    output_options      = gs_output_options
    user_settings       = space

    is_customer          = gs_customer
    ie_color              = ge_color
TABLES
    it_bookings           = gt_bookings
EXCEPTIONS
    formatting_error      = 1
    internal_error         = 2
    send_error             = 3
    user_canceled          = 4
    OTHERS                 = 6.

IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgtxt NUMBER sy-msgno
        WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.
gs_control_params-no_open = 'X'.
ENDLOOP.
```




BC470 Form Printing with SAP Smart Forms



1 Overview



2 SAP Smart Forms: Overview



3 First Steps with the SAP Form Builder



4 Texts, Addresses, and Graphics

&WA&

5 Data in Forms



6 Tables and Templates



7 Flow Control



8 Integration into Application Programs



9 Smart Styles

...

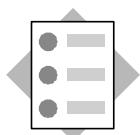
Appendix

© SAP AG 2002

Contents:

- **Smart Styles**
- **Style Builder**

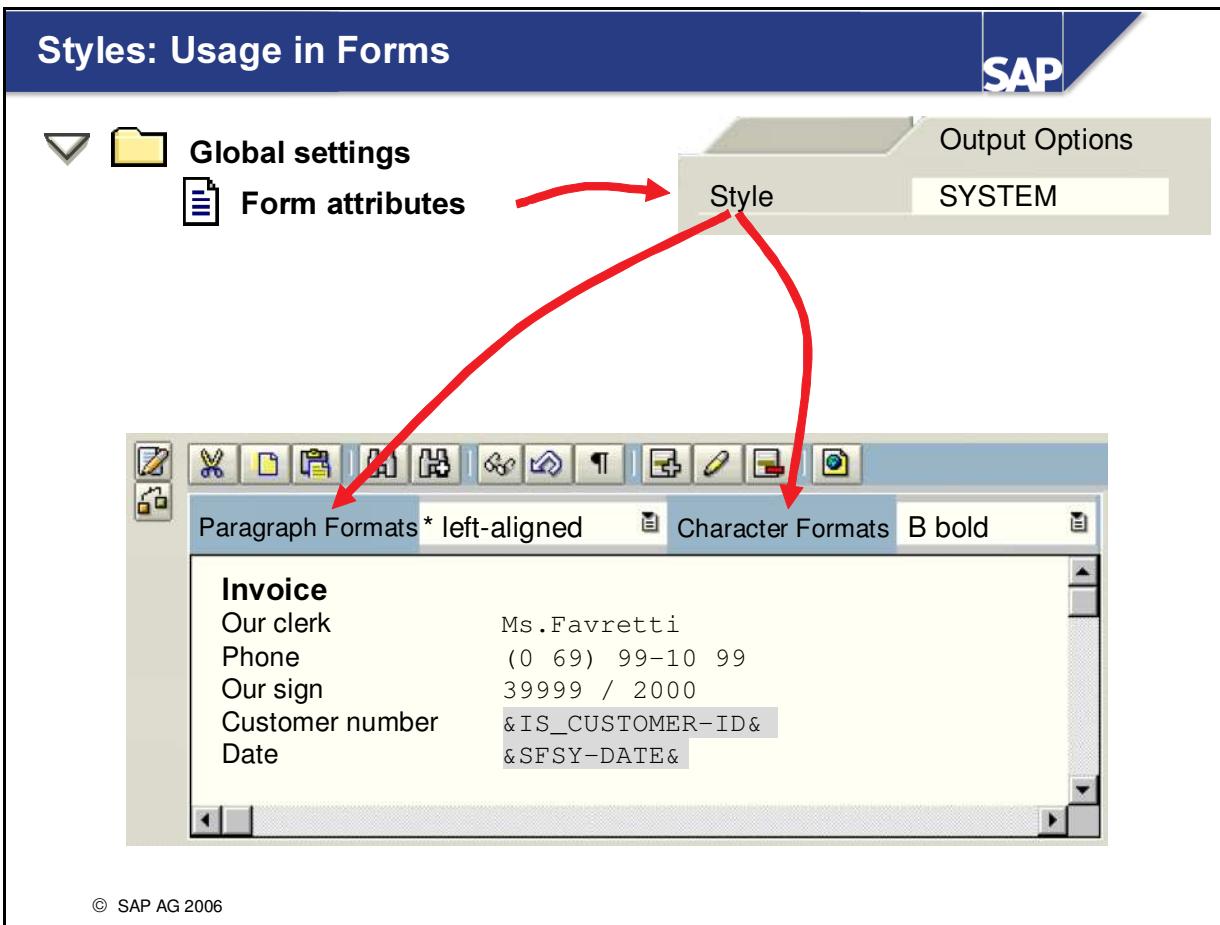
© SAP AG 2002



At the conclusion of this unit, you will be able to:

- Use the Style Builder to create/change Smart Styles
- Create paragraph and character formats
- Use formats in texts

© SAP AG 2002



- If you want to format texts in a form, select a character and/or paragraph format from the format lists in the editor. The formats offered in this list depend on the style (Smart Style) chosen. The following rules apply:
 - Each form must be assigned a style (on the *Output options* tab page of the form attributes). The default for new forms is the style *System*.
 - You can assign a different style to most nodes (on the *Output Options*) tab page). This style then applies to all lower-level nodes.
- Styles are collections of character and paragraph formats that are similar to format templates in common word processing programs. You cannot choose a format that has not been added to a style. This ensures that all forms using the same style have a consistent text design.
- Please note that SAPscript styles of include texts are not considered irrespective of whether they are set as dynamic or static. If formats are used in include texts that are not defined in the Smart Style which applies to the include text, then these formats are ignored.

Smart Styles: Initial Screen

SAP

The screenshot shows the SAP Smart Forms: Initial Screen. At the top, there is a toolbar with various icons. Below the toolbar, the title bar displays "SAP Smart Forms: Initial Screen". The main area contains three radio button options: "Form", "Style", and "Text Module". The "Style" option is selected, and the value "ZBC470" is displayed in a text field next to it. Below these controls are three buttons: "Display", "Change", and "Create". A red arrow points from the text "Style Builder" to the "Change" button. At the bottom left of the screen, there is a copyright notice: "© SAP AG 2006".

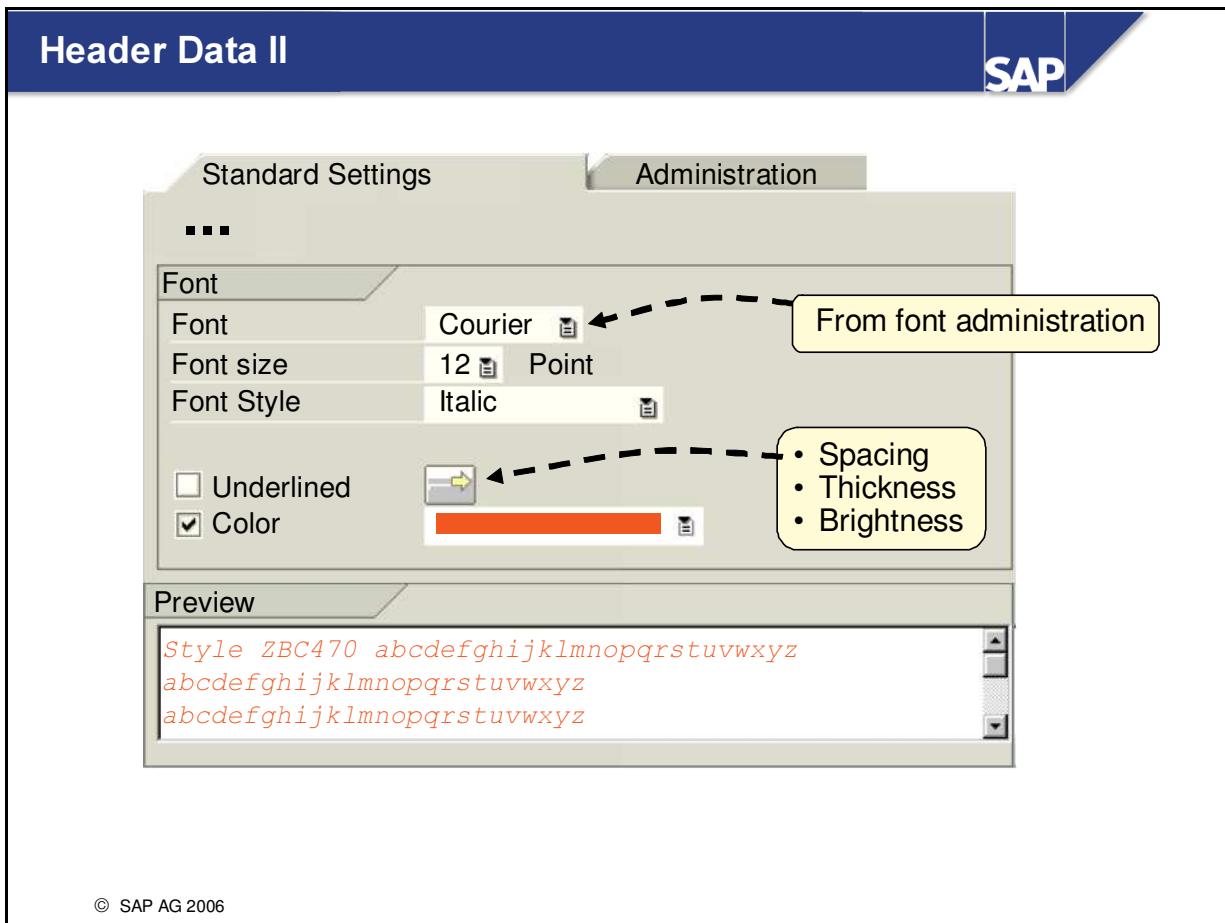
- To create a Smart Style, you can either select the *Style* radio button on the initial screen of the SAP Smart Forms transaction, or start transaction SMARTSTYLES directly. Enter the name of a style. We recommend that you only change styles in your customer namespace, that is, styles beginning with Y or Z. If required, copy the SAP styles to your customer namespace. To do this, click the corresponding pushbutton or choose *Smart Styles → Copy from the menu*.
- Like forms, styles are integrated with the SAP transport system. This is why the system prompts you for a package when you first save a style.
- You can see the package assigned - and other style-related information such as who created or changed the style - on the *Administration* tab of the style header data. It makes no sense to enter a style variant for SAP R/3 4.6C here.

The Style Builder/Header Data I

The screenshot shows the SAP Style Builder interface. On the left, a navigation tree displays a folder named 'Style ZBC470' which contains 'Header data', 'Paragraph formats', and 'Character Formats'. A red arrow points from the 'Header data' item to a preview window on the right. The preview window shows a style named 'ZBC470' with the status 'Active'. It has two tabs: 'Standard Settings' and 'Administration'. Under 'Standard Settings', there are fields for 'Standard paragraph ?' and 'Tab stop ?'. Under 'Administration', there are fields for 'Characters per inch 10,00' and 'Lines per Inch 6,00'. Red arrows highlight the values '10,00' and '6,00', which are enclosed in yellow boxes labeled 'CH' and 'LN' respectively. Below the preview window, the text '© SAP AG 2006' is visible.

- The maintenance tool for Smart Styles, called the Style Builder, consists of two areas. The navigation tree is on the left-hand side. Detailed information on the element you select in the tree is displayed on the right-hand side. You can also see a preview of the element there.
 - Like forms, styles can exist in an active and an inactive version. You activate a style by clicking the corresponding pushbutton or by choosing *Style ↔ Activate*.
 - The header data has two tab pages: *Standard settings* and *Administration*. Part of the *Standard Settings* can be seen on the above slide:
 - Standard paragraph*: Here you specify which paragraph format is to be used to format the text if the text has no explicit formatting. In the format list of the editor, this standard paragraph is marked with an asterisk (*). You must specify the standard paragraph (and the description of the style) if you want to activate the style. Before you can make an entry here, you must have defined at least one paragraph format.
 - Tab stop*: Here you set the distance between the standard tabs. (They are always left-aligned tab stops.)
- In the *Characters per inch* field, you enter the size of the CH unit of measure (for horizontal size specifications) in the style. Similarly, you determine the LN unit of measure (for vertical size specifications) in the *Lines per inch* field.

Header Data II



© SAP AG 2006

- The lower part of the *Standard Settings* tab page contains the remaining options. The selections you make in the *Font* group box apply to all paragraph and character formats of the style - unless you explicitly specify individual font attributes for these formats.
- The *Fon*ts available depend on the font administration settings.
- The *Font size* determines the height of the font in points (PT). The default is 12.
- For *Font style*, you can choose from: *bold*, *italic*, *bold and italic* or none.
- If you select the *Underlined* checkbox, you can make further settings by clicking the pushbutton on the right. (However, note that not all printers support this option.)
For *Spacing*, specify the distance between the underline and the base line. The default is 0, which means that the underline is on the base line. If you enter a negative value, you do not underline but strike through the text.
Thickness: The default is 1 point.
Brightness: Enter a percentage value. A brightness of 0% means the full shade.
For technical reasons, the various underlining types cannot be differentiated from each other in the preview of the Style Builder or the editor. This is only possible in the print preview of the application program or in the printout itself.
- The *Standard formats allowed* selection field for SAP R/3 4.6C does not have a function.

Editing Character and Paragraph Formats

SAP

The screenshot shows the SAP user interface for editing character and paragraph formats. At the top is a blue header bar with the title 'Editing Character and Paragraph Formats' and the SAP logo. Below the header is a toolbar with several icons. To the left of the toolbar is a navigation tree showing a folder structure under 'Style ZBC470'. A mouse cursor is shown pointing at the 'Character Formats' node in the tree. A context menu is open at the cursor position, listing actions: 'Expand', 'Collapse', 'Create', 'Delete', 'Copy', and 'Rename'. The bottom left corner of the screen displays the copyright notice '© SAP AG 2006'.

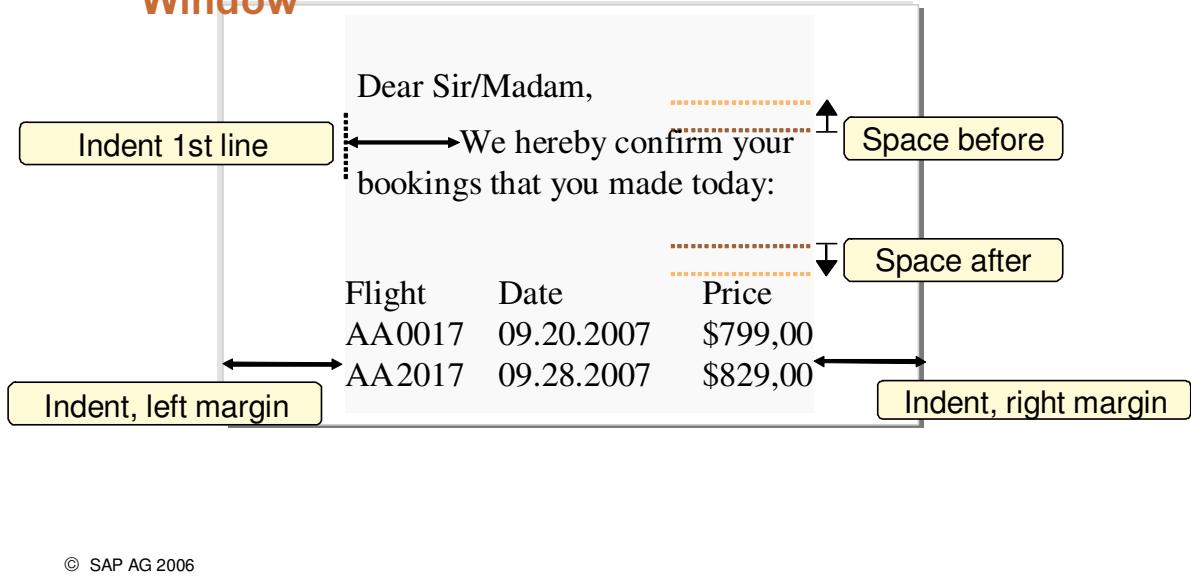
- Character and paragraph formats are displayed in the navigation tree. They always have a two-character technical name and a description.
- To create, copy, or rename a character or paragraph format, use the *Edit* menu, the pushbuttons, or the context menu (right mouse button). The action you choose refers to the node currently selected in the tree. You can select nodes by double-clicking them.

Paragraph Formats: Indents and Spacing

SAP

Indents and Spacing

Window



- On the *Indents and spacing* tab page of a paragraph format, you can make the following entries:

Indent:

Left/right margin of indent: Spacing between window and text margin.

For the left margin, you can add another indent to the first line of a paragraph.

Spacing:

Proposal: This indent is inserted in front of the first line of a paragraph. Similarly, for a *Space after*, an extra space is left after the last line of a paragraph.

Line spacing: Note that the line spacing is not adjusted automatically if you use a larger font size.

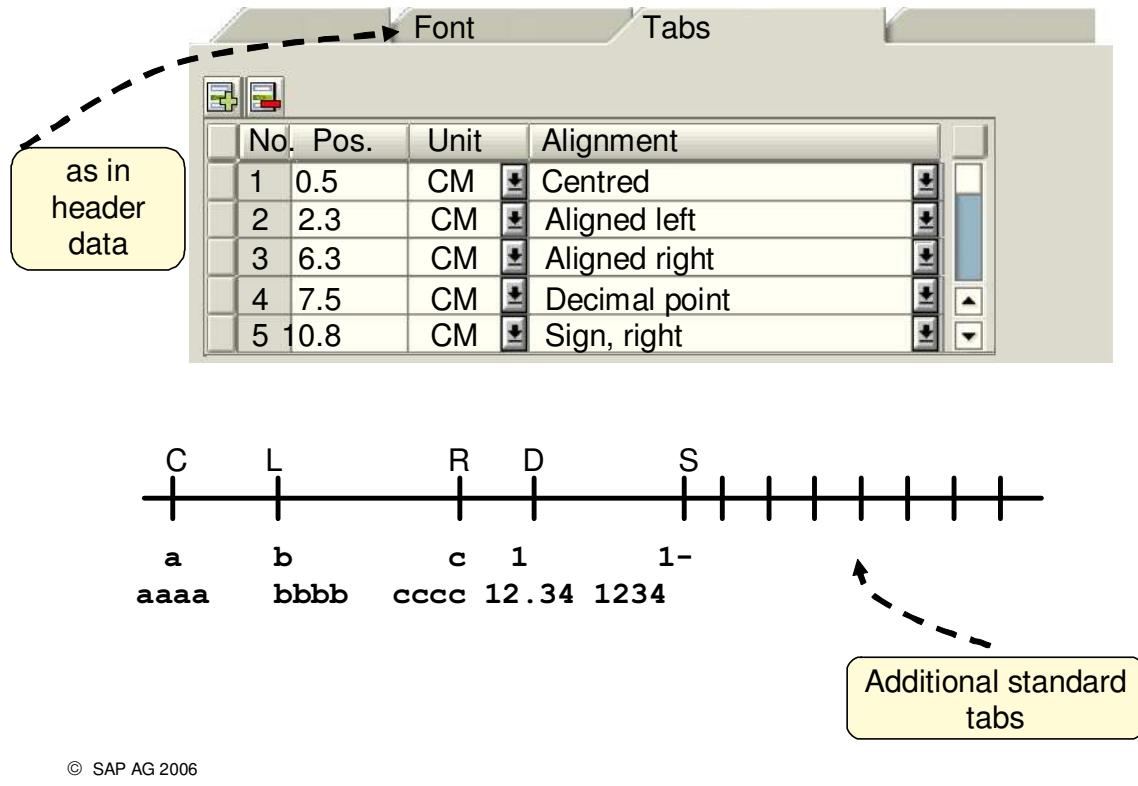
To avoid overlappings, you must therefore change the line spacing if required.

Text flow:

You can also specify that text within a paragraph should always be printed on one page (*Page protection* checkbox). If a paragraph does not fit onto a page, the system automatically inserts a page break before this paragraph.

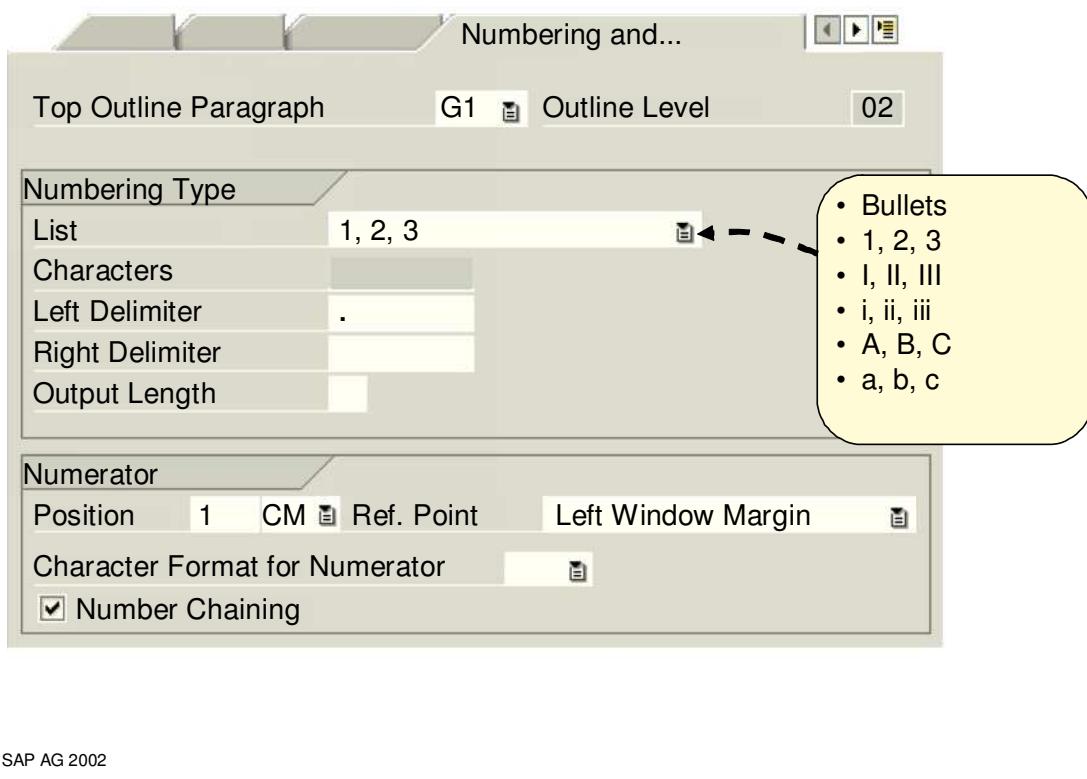
If you select the *Next paragraph same page* checkbox, the next paragraph is printed on the same page as the current paragraph.

Paragraph Formats: Tabs



© SAP AG 2006

- The *Font* tab has the same fields as the *Standard settings* tab of the style. If you do not make any settings, the system uses the settings of the header data: *Font family*, *Font size*, *Font style*, *Underlined*, *Color*. You can override the header data entries by specifying a different font. Note that standard settings such as the color or the font family are not displayed in the preview of a paragraph format.
- On the *Tabs* tab page, you define individual tab stops for the paragraph format.
 - The *Sign* alignment type lets you define numbers right aligned at the tab stop position, considering the minus sign or implied blank space at the end of the number.
 - The *Alignment with decimal point* alignment type lets you align numbers with decimal points printed at the tab stop position.
 - After the last tab stop, the standard tab stops of the standard settings (header data) are used.



- You can define paragraphs that have outline characters (for example °) or are numbered automatically. To do this, go to the *Numbering and outline* tab.
- If you want to use a multi-level numbering or outline (such as 1, 1.1, 1.2, 1.2.1 and so on), you must create a separate paragraph for each level (we recommend that you use the *Copy* function to do this). As the *Top outline paragraph*, you enter the top paragraph in the hierarchy which controls all other outline or numbering levels. The individual levels are inserted as subnodes of the top outline paragraph into the navigation tree, and the outline level is automatically entered in the corresponding field on the tab. An example is shown on the next slide.
- Choose a numbering type:
 - *List character*: The list character printed at the beginning of the paragraph is the one that you enter in the *Character* field. You can use eight digits at the most.
 - Arabic numbers, Roman numerals (lowercase or uppercase) or letters (lowercase or uppercase). You can also specify a *left* and a *right delimiter* such as an angle bracket to define numberings of the form "a), b), c)" and so on.
- In the *Position* field, you enter the space between the numbering/outline character and the left window margin. Make sure that the paragraph margin is not affected by the numerator margin. To avoid overlappings, define a paragraph margin which is large enough.
- If you select *Number chaining*, the system uses the previous numerator for multi-level numberings. Example: If level G1 = 1, then level G2 (with number chaining) = 1.2.

Example:

Paragraph Definition

G1

G2

G3

List	Left Delimiter	Right Delimiter	Number Chaining	Position
1, 2, 3				
1, 2, 3	.		<input checked="" type="checkbox"/>	0.5 cm
a, b, c	.)	<input checked="" type="checkbox"/>	1 cm

Text

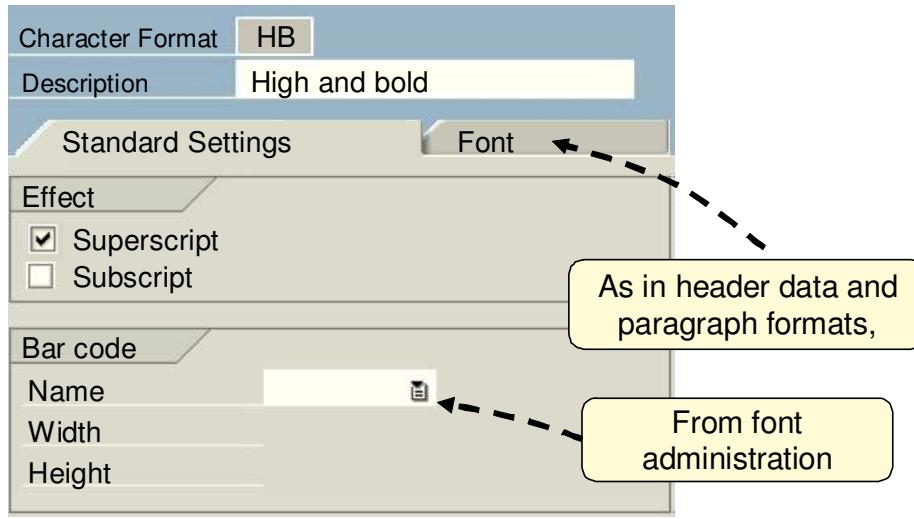
	Outline Level	Paragraph	Top Paragraph
1	1	G1	G1
1.1	2	G2	G1
1.1.a)	3	G3	G1
1.1.b)	3	G3	G1
1.2	2	G2	G1
2	1	G1	G1

© SAP AG 2002

- This is an example of a three-level outline. G1 is the top level.
- Outlines and numberings are not shown in exact WYSIWYG mode in the preview of the editor. For a real WYSIWYG display, you need to test the function module of the form.
- If you want to reset the numbering of a paragraph to its initial value, create a command node in your form and use the command *Reset paragraph numbering*. See Unit 7 - *Flow Control*.

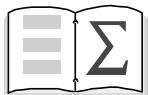
Character Formats

SAP



© SAP AG 2006

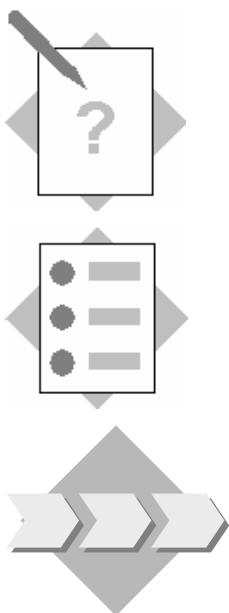
- A character format has two tab pages: *Standard Settings* and *Font*. The *Font* tab page has the same fields as the standard settings tab page for the style or as a paragraph format. The entries you make here override the settings of the header data or the paragraph format used in the text: *Font*, *font size*, *font style*, *underscores*, *color*. Note that font settings made in the standard settings or a paragraph format are not displayed in the preview of a character format.
- On the *Standard settings* tab page, you can set the attributes *Superscript* and *Subscript*.
- You can also choose a *Bar code*.
The height and the width of a bar code are automatically copied from font administration.
You cannot combine the *Superscript* or *Subscript* attribute with a bar code.
Bar codes are not displayed in the preview of the Style Builder. The print preview of the application program roughly displays them in their correct size as a pattern of lines.



You are now able to:

- Use the Style Builder to create/change Smart Styles
- Create paragraph and character formats
- Use formats in texts

© SAP AG 2002



Unit: Smart Styles Optional

After completing these exercises, you will be able to:

- Maintain Smart Styles

Your task: Copy and enhance your existing style and use it in the invoice form.

Copy template for the style:

BC470

Name of the style to be created:

ZBC470_##_STYLS

Package (for all exercises):

ZBC470_##

Name of the form to be used:

ZBC470_##_FLOWS

Only use your own form ZBC470_##_FLOWS if you have worked through the optional task in unit 7 where you had to create the page TERMS. Otherwise, copy the form BC470_FLOWS to ZBC470_##_STYLS.

Model solution for the style:

BC470_STYLS

Application program for testing purposes:

SAPBC470_DEMO

1 Copy template

Copy the style BC470 to ZBC470_##_STYLS.

2 Change standard settings

Set Times, 12 pt as the standard font type.

3 Change and create paragraph formats

3-1 Specify for the paragraph format L that the lines of a paragraph should not be separated by a page break.

3-2 Use the standard font for the paragraph format L.

3-3 For the paragraph format TO, change the positions for the two tab stops to 13 and 14 cm.

- 3-4 **Optional:** Create the paragraph formats G1 and G2. These formats should outline paragraphs as follows:

- I This paragraph has format G1
- II This paragraph has format G1
 - II a) This paragraph has format G2
 - II b) This paragraph has format G2
- III This paragraph has format G1

Note that the preview does not show the exact result.

- 4 Create at least one character format of your choice.
5. Activate your style.
- 6 Enter your style as the style for the text CONDITIONS on the page TERMS of your form.

Test your formats.

Optional: Format several paragraphs using the outline paragraphs G1 and G2 that you created. (If necessary, add a few lines of text.)

- 7 Test your form using the program SAPBC470_DEMO.

Smart Styles - Solutions



Unit: Smart Styles

1 Copy template

On the initial screen of transaction SMARTFORMS, select the *Style* radio button and enter BC470 as the name. Alternatively, you can start the separate maintenance transaction SMARTSTYLES and enter the name of the style there. It does not matter which point of entry you choose since the subsequent steps are all identical.

Copy the style by choosing the *Copy* pushbutton (first pushbutton in the application toolbar). Enter the new name (ZBC470_##_STYLS) and enter your package on the dialog box that is displayed next.

2 Change standard settings

Select the *Header data* node in the navigation tree. On the *Standard settings* tab of the maintenance screen, select *Times* as the font family and *12* as the font size.

3 Change and create paragraph formats

3-1 In the navigation tree, select the node *Paragraph formats* and then L. On the *Indents and spacing* tab of the maintenance screen, select the *Page protection* checkbox.

3-2 You must not enter a font on the *Font* tab page of the paragraph format L to ensure that the standard font of the header data is used.

3-3 In the navigation tree, select the node *Paragraph formats* and then the node TO. On the maintenance screen, change the tab stop positions on the *Tabs* tab.

3-4 **Optional:** From the context menu of the node *Paragraph formats*, choose *Create node*. The system displays a dialog box. Enter G1 and choose *Enter*. Enter a description on the maintenance screen (for example, "Outline, Roman numerals").

On the *Indents and Spacing* tab page, set a left margin of 1.2 cm, for example.

On the *Numbering and outline* tab, enter G1 as the top outline paragraph.

Choose Roman letters in uppercase as the numbering type. Leave the numerator position empty (0 cm).

Use the context menu of the paragraph format G1 to create the paragraph format G2. Enter a description.

On the *Indents and Spacing* tab page, set a left margin of 2.4cm, for example.

On the *Numbering and outline* tab, enter G1 as the top outline paragraph. This inserts G2 as a subnode of G1 in the navigation tree. Choose lowercase letters as the numbering type and a closing bracket as the right delimiter. Select the *Number chaining* checkbox to display the Roman numbers of the higher numbering level also on the level G2. Enter 1.2 cm as the position for the numerator (reference point: left window margin).

- 4 Create at least one character format of your choice.
You can do this, for example, using the context menu of an existing character format node. Assign a two-character ID and a description, and make some settings of your choice on the *Standard settings* and the *Font* tabs.
- 5 Activate your style by clicking the *Activate* pushbutton (which is the third pushbutton from the left).
- 6 Go to the SAP Form Builder for your form (in a second session). On the *Output Options* tab page of the node CONDITIONS (in the main window), enter your style ZBC470_##_STYLS. In the editor of the text node, you should now be able to select the new format for your terms and conditions of business in the selection list.
- 7 Test your form using the program SAPBC470_DEMO.



BC470 Form Printing with SAP Smart Forms



1 Overview



2 SAP Smart Forms: Overview



3 First Steps with the SAP Form Builder



4 Texts, Addresses, and Graphics

&WA&

5 Data in Forms



6 Tables and Templates



7 Flow Control



8 Integration into Application Programs

ab_cd

9 Smart Styles



...

Appendix

© SAP AG 2002

Contents:

- A A Orientation in the SAP Menu and in the Navigation Tree**
- B Changes: Customizing and Transport**
- C Fonts and Bar Codes**
- D Forms in Multiple Languages**
- E Other Release-dependent Enhancements**
- F Special Features for SAP R/3 4.6C**
- G Using SAPscript Objects**
- H Preview: SAP Interactive Forms by Adobe**

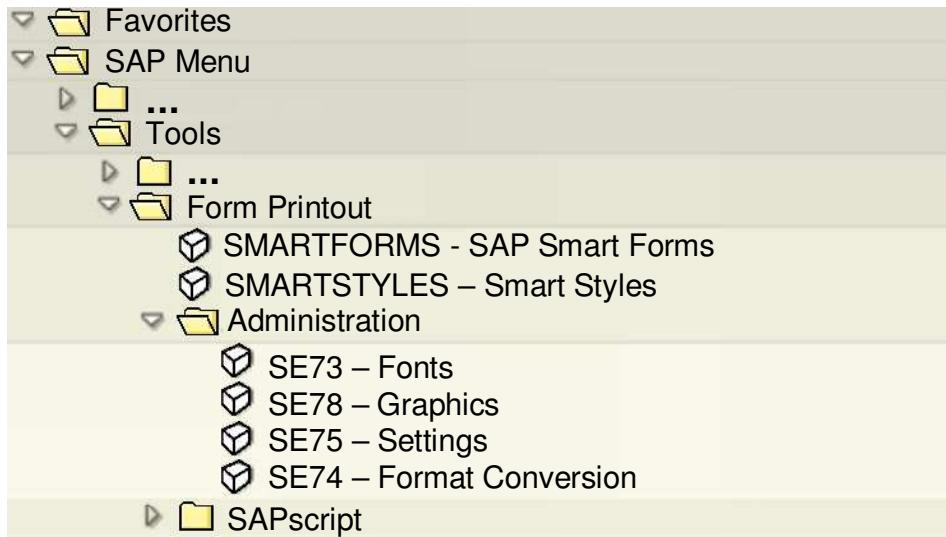
© SAP AG 2006

- **Menu paths and transaction codes**
- **Legend of icons in the navigation tree**

© SAP AG 2002

Menu Paths and Transaction Codes

SAP



© SAP AG 2002

Legend for Icons in the Navigation Tree

SAP

	Main window		Loop
	Secondary window		Folders
	Copies window/ Final window		Alternative
	Text		TRUE
	Address		FALSE
	Graphic		Command
	Table (as of SAP WAS 6.10)		Program lines
	Table (SAP R/3 4.6C)		Condition
	Control level		
	Template		Smart Style element
			Outline paragraph

© SAP AG 2006

- Note that the appearance of the icons depends on your GUI.

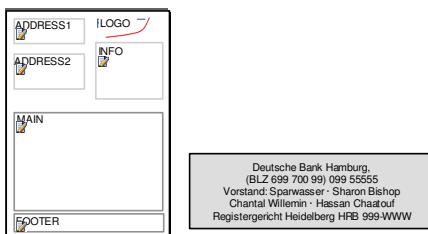
- **Customizing:**
 - Dunning
 - Delivery note
 - Invoice
- **Transport of SAP Smart Forms Objects**

© SAP AG 2002

1

SAPscript or SAP Smart Forms?

2

Form F150_DUNN_SF**Text modules****Graphic****Copy and adjust**

© SAP AG 2002

- We will use the dunning procedure as an example to illustrate all steps that are required to ensure that a transaction executes your coding and uses your SAP Smart Forms.

1. Ensure that SAP Smart Forms are used instead of SAPscript:

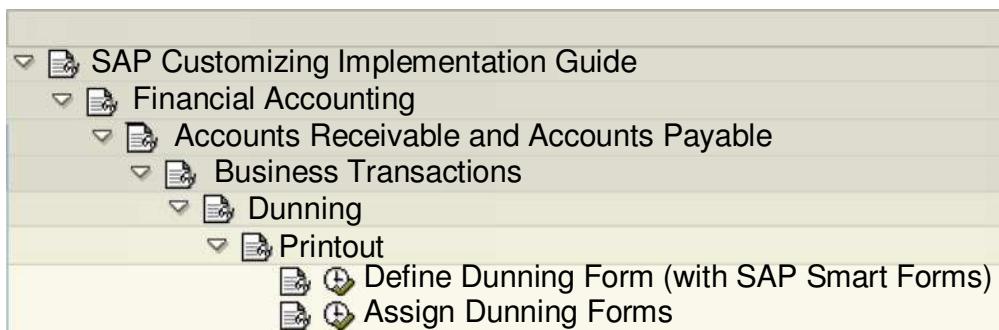
- Implementation Guide: *Financial Accounting -> Financial Accounting Global Settings -> Business Transaction Events*
- Menu *Settings -> P/S function modules -> of an SAP application*
- Change the function module to **FI_PRINT_DUNNING_NOTICE_SMARTF** for the Business Transaction Event 1720 with the application indicator FI-FI.
- Save

2. Adjust form and texts:

- Copy the form F150_DUNN_SF in the SAP Form Builder to your customer namespace and then adjust and activate it.
- Adjust the inserted texts (such as the address) and the company logo.

3

Which Form?



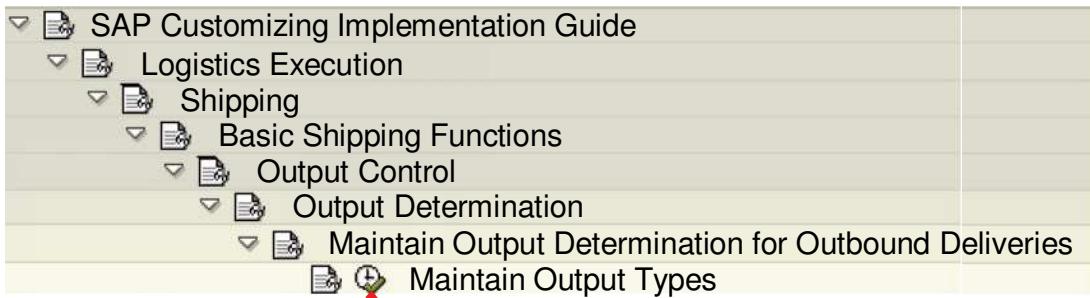
© SAP AG 2002

- 3. Enter the form in Customizing:
 - Implementation Guide: *Financial Accounting -> Accounts Receivable and Accounts Payable -> Business Transactions -> Dunning -> Printout -> Assign Dunning Forms*
 - Select the desired procedure (for example, *Four-level dunning, every two weeks*), then choose *Forms for normal or legal dunning procedure* in the tree and enter the company code.
 - Enter your copy of the dunning form for the desired dunning level. Save your data and ignore the warning saying that the form does not exist or is inactive (this warning is displayed because the system only checks for SAPscript forms).

- 1 Form LE_SHP_DELNOTE
Text modules
Graphic
Program RLE_DELNOTE

} Copy and adjust

- 2 Customizing entries

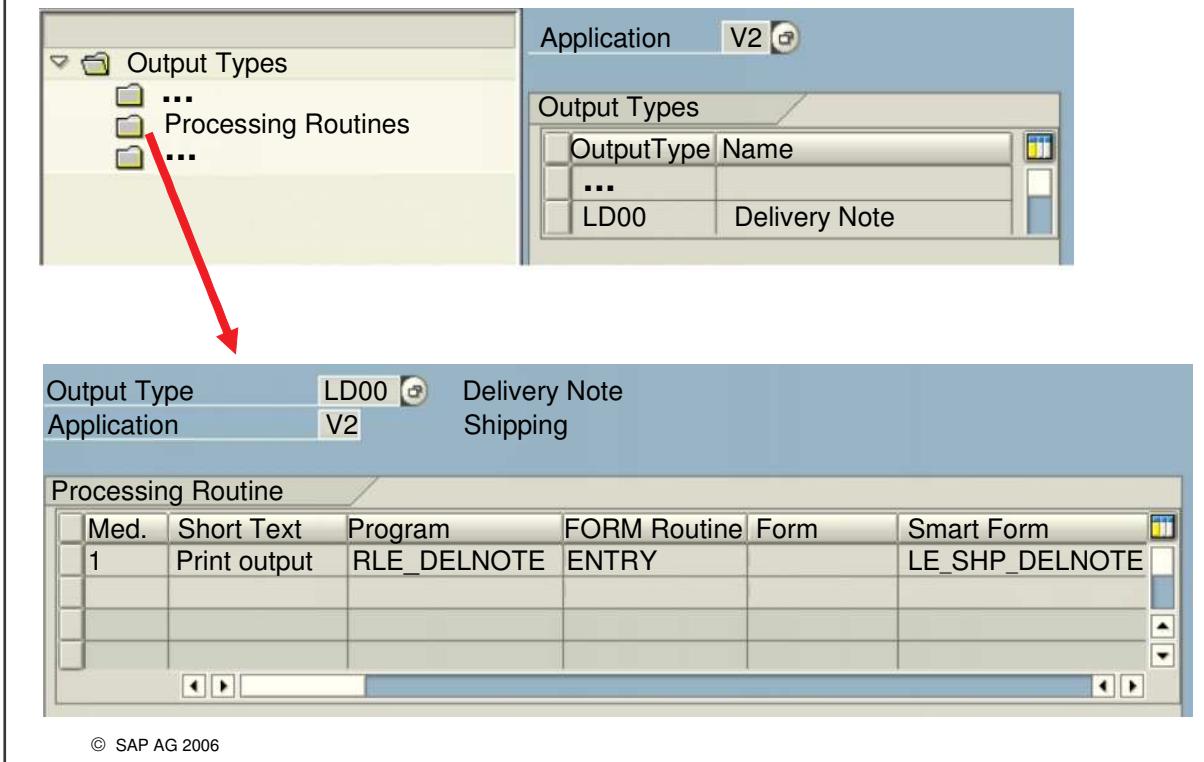


© SAP AG 2006

- The delivery note form provided by SAP is called LE_SHP_DELNOTE, the program name is RLE_DELNOTE. For company-specific adjustments, copy the form or the program to your customer namespace and then modify the copy as required.
- For the delivery note, you make a setting in Customizing that specifies whether SAP Smart Forms should be used, and which ones should be used.
- Implementation Guide: *Logistics Execution ® Shipping® Basic Shipping Functions ® Output Control ® Output Determination®* Maintain Output Determination for Outbound Deliveries(Maintain Output Types.

Change Procedure: Delivery Note Example II

SAP



- Select the output type LD00 (delivery note) and double-click Processing Routines. On the next screen, you have to make the following entries:
 - Medium (fax or printer)
 - Program: RLE_DELNOTE or the copy you made.
 - FORM routine of the program that contains the call of the SAP Smart Form.
 - Form: Do not enter anything here. (This field is for SAPscript forms.)
 - Smart Form: LE_SHP_DELNOTE or the copy you made.

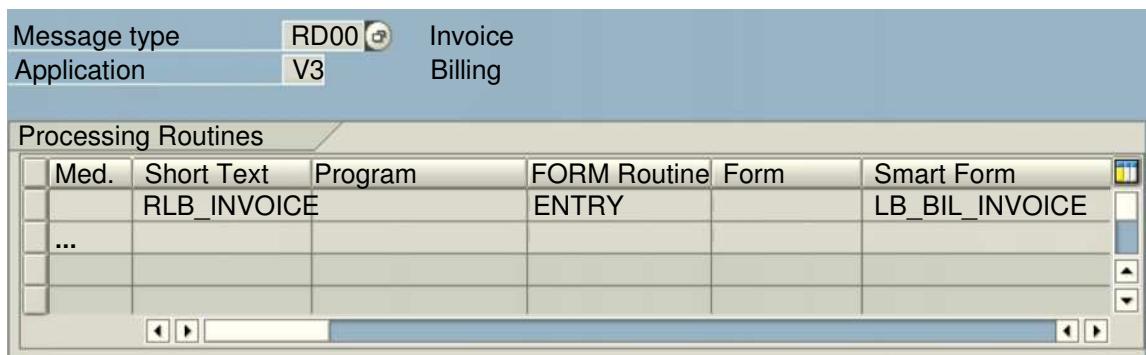
Change Procedure: Invoice Example

SAP

- 1 Form LB_BIL_INVOICE
Text modules
Graphic
Program RLB_INVOICE

} Copy and adjust

- 2 Customizing entries:
VOK2 → Output → Processing Programs → Billing Document
Output Type: RD00; Application: V3



- The name of the invoice form delivered by SAP is LB_BIL_INVOICE, the program is called RLB_INVOICE. For company-specific adjustments, copy the form or the program to your customer namespace and then modify the copy as required.
- You make a setting in Customizing for the invoice form which specifies that SAP Smart Forms should be used and which ones should be used.
- To go to Customizing, start the transaction VOK2 and choose Output → Processing Programs → Billing document. On the dialog box that appears next, enter RD00 as the output type and V3 as the application.
- On the next screen, you have to make the following entries:
 - Medium
 - Program: RLB_INVOICE or the copy you made.
 - FORM routine of the program that contains the call of the SAP Smart Form.
 - Form: Do not enter anything here. (This field is for SAPscript forms.)
 - Smart Form: LB_BIL_INVOICE or the copy you made.

Transport

SAP



- **Forms**
- **Text modules**
- **Smart Styles**
- **Programs**
- **Transport Organizer**

- **Include Texts**
- **Report RSTXTRAN**

- **Graphics**
- **SE78**

© SAP AG 2006

■ The following rules apply to the transport of SAP Smart Forms objects:

- Forms, text modules, Smart Styles and programs must always be assigned to a development class. This ensures that they are automatically integrated with the Transport Organizer and are transported like all other Workbench objects.
 - When forms are transported, the function modules generated are not included in the transport. The function modules are automatically generated in the system when the form is called for the first time.
 - Language versions are automatically included in transports.
- Include texts (SAPscript texts) are not automatically linked to the Workbench Organizer. You must add them to a development request either manually (with transaction SE09) or automatically using report RSTXTRAN. For more information, see the SAPscript documentation.
- Graphics must also be added to a development request. To do this, use the graphics administration transaction (SE78) and click the truck icon or choose Graphic → Import.

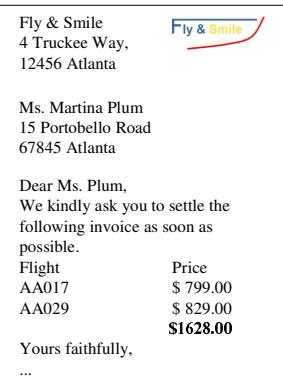
- **Adjusting Device Types**
- **Maintaining Font Families, System Fonts, and Printer Fonts**
- **Maintaining Bar Codes**

© SAP AG 2006

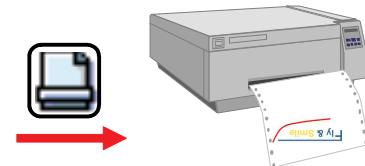
From Form to Device-Specific Output

SAP

System fonts System bar codes



Printer fonts Printer bar codes



Administration: SPAD, SE73

© SAP AG 2006

- Note: The following applies to both SAPscript and SAP Smart Forms.
- In a form or a style, the fonts are selected independently of printers.
- The user does not have to select a printer, and therefore a specific device type, until they print the form. A device type specifies which printer driver to use for formatting the output, and which printer character sets are required. The temporary format for spool administration (OTF = Output Text Format) already takes the printer into account (for page breaks, for example). OTF however, still has the symbolic name (print control) for the control characters of the printer.
- The print controls are converted to printer-specific commands by an output request.
- Because a device type specifies attributes that apply for all devices of a particular model, the type can be used by several device definitions. For example, all devices with printers compatible to the type Hewlett Packard LaserJet 4, use the device type HPLJ4 in the spool system.
- You can change any part of a device type according to requirements, for example, to install a printer font that is not a standard SAP font.
- Do not directly modify a device type delivered by SAP, make the changes in a copy instead - otherwise, your changes may be overwritten when an upgrade is carried out.

Example: Non-Printing Fonts

& Smile
4 Fly Truckee Way
NY, NY 12456-4574

Fly & Smile

Invoice

Clerk: Faetti
Tel.: (069) 99-10
Fax: (069) 99-12

Mr./Mrs./Company
Plum
15 Portobello Road
67845 Atlanta

Dear Sir/Madam,
Please settle the following invoice:

Flight	Date	Departure	Price
AA 0017	13:15	799.00 USD	
AA 2017	21:55	829.00 USD	
LH 0400	9:07	398.80 EUR	

Yours faithfully,
...

Required Steps:

- 1 Create new device type
- 2 Create new print controls for device type
- 3 Create font family
- 4 Define system fonts
- 5 Define printer fonts
- 6 Assign device type to a printer

© SAP AG 2006

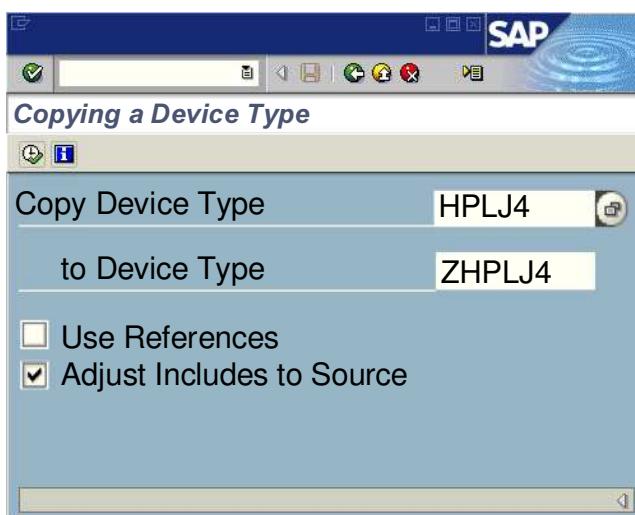
- Problem 1: Errors have occurred in printing an invoice. Some fonts have not been correctly printed.
- Cause 1: You are using a printer that is not yet supported by SAP.
- Problem 2: You cannot use all the fonts that the printer can offer.
- Cause 2: You have installed a new font in your printer for which the SAP standard device type (for example, HPLJ4) has not been set up, or you are using a printer that is not yet supported by SAP.
- Solution: Generate a device type with the additional fonts/font sizes.
- To set up additional fonts/font sizes, perform the following steps:
 - 1) In spool management (transaction SPAD), create a new device type in your namespace by copying an existing device type.
 - 2) Create new print controls for the new device type (transaction SPAD or SE73).
 - 3) If necessary, create a new font family.
 - 4) Define system fonts for the font family in the required sizes and bold/italic variants.
 - 5) For the device type, define printer fonts to correspond to the new system fonts, and assign the corresponding print controls to them.
 - 6) Assign the new device type to your printer (the output device). For example, for the printer P280, replace the original device type HPLJ4 with the new type, ZHPLJ4.

Copying a Device Type

SAP

1

Transaction SPAD:



© SAP AG 2002

- To create a new device type, copy an existing one into your customer namespace. For example, copy HPLJ4 to ZHPLJ4.
- In transaction SPAD (SAP Easy Access Menu: *Tools -> CCMS -> Spool -> Spool Administration*), choose *Utilities -> For Device Types -> Copy Device Type*.
- *Use References*: If you select this option, the formatting steps and font metrics contained in the source device type are not copied, rather, the target device type contains references to the source device type. The advantage of this is that the new device type always has the most current status when the SAP device type is updated. You should only activate this parameter if you are sure that the source device type exists in every system where the target device type is used. If your source device type is already a customer copy, (YXXX or ZXXX), do not select this option.
- When you copy a device type, the following are copied:
 - The device type definition
 - The print controls (you can add to the print controls at a later time)
 - Format types
 - Font metrics
 - SAP Smart Forms and SAPscript printer fonts and printer bar codes.

Adding Print Controls to a Device Type

SAP

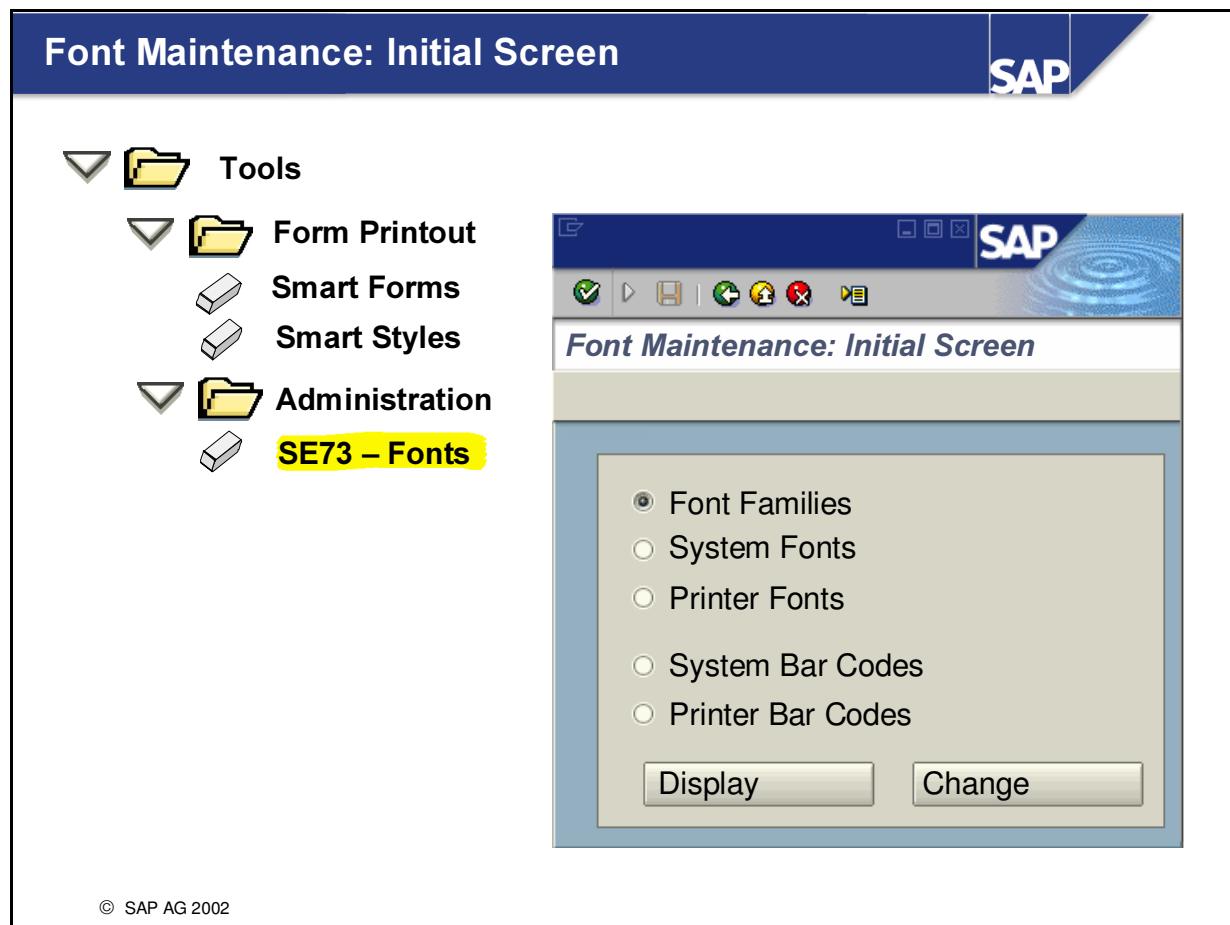
2

- **Process print controls for new device type**
Add new entries for required font sizes

PrintCtrl (Example)	Font size ($1/_{10}$ Point)	Control character sequence
SF910	180	1B28304E1B28733170313876307333623431303154
SF915	240	1B28304E1B28733170323476307333623431303154
SF920	360	1B28304E1B28733170333676307333623431303154

© SAP AG 2002

- You need to set up a new print control SFXXX for the device type ZHLPJ4. The device type contains the printer control characters to implement the required font according to the SAP Smart Forms font. See the printer handbook to find out the control characters. Also check which print controls are already defined for the device type. The XXX number of the SFXXX print controls is arbitrary.
- To add new print controls to your copy of a device type, start transaction SPAD, and click on *Full Administration*. Choose the *Device Types* tab page. Enter the name of the device type, and confirm your entry. Choose *Print Control*, and switch to change mode. Choose *Edit -> Insert Line*, and enter a new SFXXX print control.
- Enter the attributes and the control character sequence (that you found out in the printer handbook) for the print control. You can enter the command in hexadecimal format, or as text.



© SAP AG 2002

- All fonts and bar codes for SAP Smart Forms are administrated using Font Maintenance (transaction SE73). In the SAP Menu, choose *Tools -> Form Printout -> Administration -> Font*.
- In **font families**, the usable font names are entered, along with the attribute for proportional or not proportional.
- A **system font** (or SAP font) is the combination of font family, font size, and bold/italic attributes. The font selections used when entering text with SAP Smart Forms always refer to system fonts.
- A **printer font** is the combination of printer type, font family, font size, and bold/italic attributes. Printer fonts are fonts of the output device that are available for use with SAP Smart Forms. For printer fonts, certain metric data (information on the width of the letters) must be maintained, along with control data for setting the fonts on the output devices.
- **System bar codes**, like system fonts, are independent of the device type. To use these in SAP Smart Forms, choose a character format that has the bar code attribute activated.
- **Printer bar codes** contain the device-type-specific control character sequences that activate bar codes for suitable printers.

3 Font family:

Family	Description	P	Rpl.1	Rpl.2	Rpl.3
COURIER	Courier	✓	LETGOTH		0000

4 System font:

Family	Font Size	Bold	Italic
COURIER	060		
COURIER	060	✓	
COURIER	060		✓
COURIER	060	✓	✓

5 Printer font:

Device type	Family	Font Size	Bold	Italic	CPI
ZHPLJ4	PrtCtl.1	...			
ZHPLJ4	COURIER	060			17,00 SF025
ZHPLJ4	COURIER	060	✓		17,00 SF026
ZHPLJ4	COURIER	060		✓	17,00 SF027
ZHPLJ4	COURIER	060	✓	✓	17,00 SF028

© SAP AG 2002

- To create a new font family, choose *Font Family* then *Change* on the initial screen of transaction SE73. Then choose *Font Family -> Create*.
- Enter a *Replacement Family* if one exists in the system. The replacement family is used in an output request if the font family is not installed on the device type you are using.
- If required, enter the name of the SAP character set (code page).
- A new printer font requires the following information:
 - Device type
 - Font family (font name in the SAP system, for example Courier or Times)
 - Size (in $1/10$ point, for example 240. For printer drivers that support scalable fonts, 000 is entered here).
 - The font attributes *bold* and/or *italic*.
 - Characters per inch
 - Print control to use.
- If you are using a proportional font (such as Times), you have to maintain an AFM file, which contains width information for the individual characters. To maintain an AFM file, place the cursor on a printer font and choose *Edit Metric*. You can also copy the font metrics of an existing font: Choose *Edit -> Copy Metric*.

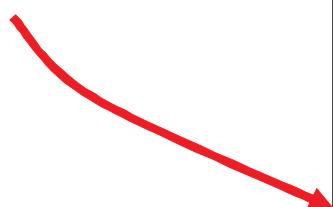
Result: Correct Printout of the Invoice

SAP

6

Last Step:

Assign new device type
to a printer



<i>Fly & Smile</i> 4 Truckee Way, 12456 Atlanta	Fly & Smile												
Invoice													
Mr./Mrs./Company Plum 15 Portobello Road 67845 Atlanta	Clerk: Favretti Tel.: (069) 99-10 Fax: (069) 99-12												
<p>Dear Sir/Madam,</p> <p>Please settle the following invoice:</p> <table border="0"><thead><tr><th>Flight</th><th>Departure</th><th>Price</th></tr></thead><tbody><tr><td>AA 0017</td><td>13:15</td><td>799.00 USD</td></tr><tr><td>AA 2017</td><td>21:55</td><td>829.00 USD</td></tr><tr><td>LH 0400</td><td>9:07</td><td>398.80 EUR</td></tr></tbody></table> <p>Yours faithfully, ...</p>		Flight	Departure	Price	AA 0017	13:15	799.00 USD	AA 2017	21:55	829.00 USD	LH 0400	9:07	398.80 EUR
Flight	Departure	Price											
AA 0017	13:15	799.00 USD											
AA 2017	21:55	829.00 USD											
LH 0400	9:07	398.80 EUR											

© SAP AG 2002

- Finally, use spool administration (transaction SPAD) to convert the device type of the output device (the printer) from the original SAP setting to your new type. For example, change the device type of the output device P280 from HPLJ4 to ZHPLJ4.

How a Bar Code Works

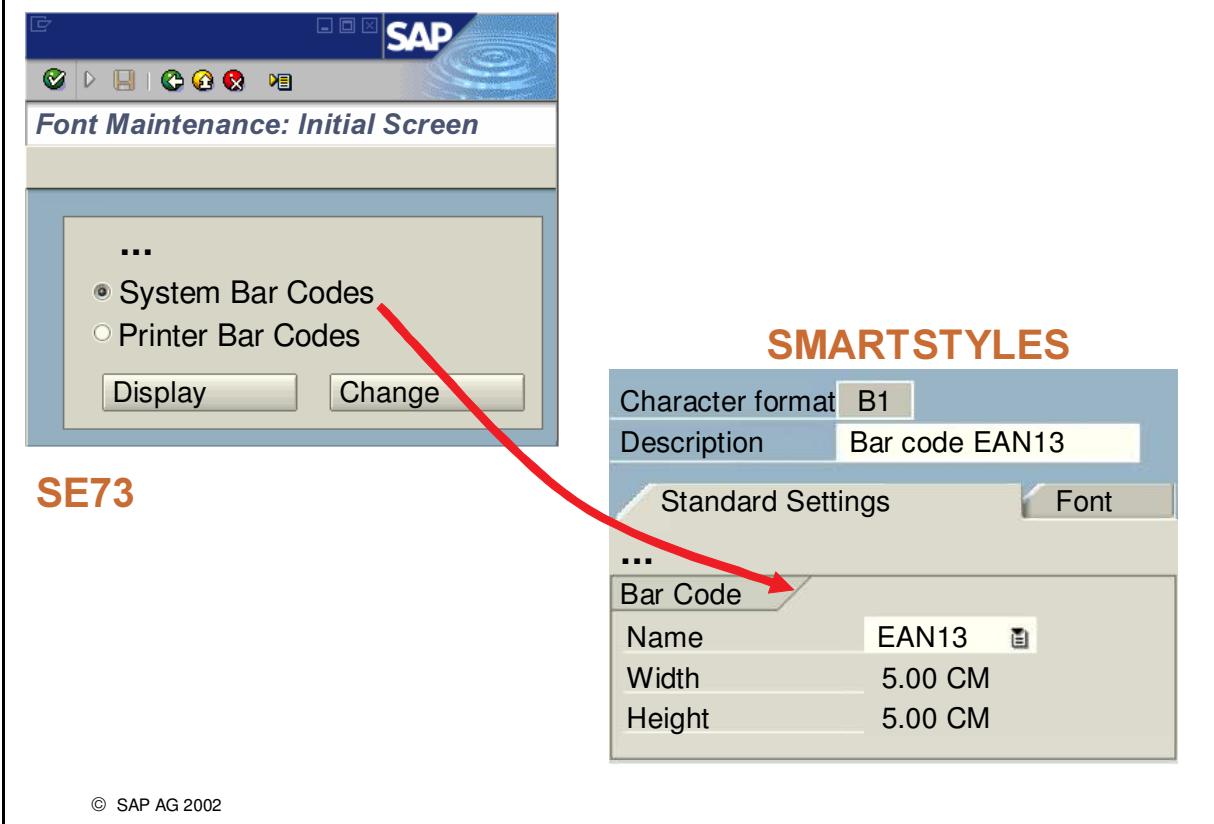
SAP

The diagram illustrates the conversion of a barcode string into a physical barcode and its internal structure. At the top, a screenshot of SAP Smart Forms software shows a text area containing the character string **4<AMCDPQ=ghijab>**. This string is highlighted with a red arrow pointing down to a sample barcode. Below the barcode, the string is broken down into its components: **Country identifier** (4), **Manufacturer number** (012345), **Article number** (678901), and **Check digit** (1). Dashed arrows point from each label to its corresponding part in the barcode.

- Bar codes are used for automatic identification. Bar codes enable information to be read quickly and precisely.
- A bar code consists of a series of parallel stripes and empty spaces. Predefined width patterns are used to encode data as a bar code. Scanners are used to read bar codes, and a decoder interprets the width of the stripes and empty spaces.
- In SAP Smart Forms, bar code data is handled in the same way as text data (for example, the character string **4<AMCDPQ=ghijab>**). This text data must be formatted using a bar code character format, which is a character format used by a system bar code. In this way, for example, the character format **B1** could be defined as a bar code format used by the bar code **EAN13**.

Maintaining and Using Bar Codes

SAP



- When you print a bar code, a print control called the bar code prefix is first sent to the printer, followed by the actual bar code data (for example, an eight-figure number). The printer then processes a print control called a bar code suffix. The naming convention is SBPXX for prefixes, and SBSXX for suffixes.
- Device-type-specific control character sequences (print controls) are maintained with the printer bar codes.
- For maintaining bar code print controls, system bar codes, and printer bar codes, the respective notes for maintaining system fonts and printer fonts apply.

- **0008928** List of supported printers/device types
- **0005196** Printing bar codes with SAPscript
- **0645158** New bar code technology for Smart Forms
- **0017054** How to copy or change a device type
- **0012462** How can I define a new printer font?
- **0317851** Printing PDF files in 4.6C/4.6B/4.5B/4.0B
- **0201307** TrueType fonts for Smart Forms/SAPscript

© SAP AG 2006

- **Forms in multiple languages**

© SAP AG 2002

Forms in Multiple Languages

Text elements:

In the application program, you set the language in which you want to print the form. To do this, use the field `langu` in the control structure `control_parameters`. This is a parameter of the generated form function module. You can specify up to three alternative languages in the fields `replangu1`, `replangu2` and `replangu3` of the control structure.

If the form or individual text elements do not exist in any of the specified languages, or if you have not set the language, text elements are printed in the logon language. If the form does not exist in this language either, the original language is used.

As of SAP Web Application Server 6.20, you can select *Restricted Languages* in the *General Attributes* of a form. This stops texts in the logon language and texts in the original language being used if these texts do not exist in the language in which the form is processed.

Text Modules:

Text modules are output in the language in which the form is output. As of SAP Web Application Server 6.10, you can set an alternative language for each text module.

Include texts:

With include texts, you can set a language explicitly in the corresponding text node of the form. If you do not specify a language, the language in which the form is output is used. If the include text does not exist in this language, the system searches for this include text in the logon language and then in the original language of the form.

Addresses:

To ensure addresses are formatted correctly for specific countries, use address nodes (with Business Address Services) or call the function module `ADDRESS_INTO_PRINTFORM` in program line nodes.

Format for decimal numbers and dates:

You set the format for decimal numbers and dates using the ABAP command `SET COUNTRY <Language>` in the application program. If you do not make an entry, the formats that were entered in the user master during system logon are used.

Form maintenance:

The form can only be maintained in the original language. You either log on to the system in this language or you are asked if you want to set the original language to a different language before you change the form.

Create translation:

You translate forms in transaction SE63. You can translate a form only in the languages that you have set in the form attributes.

The path in SAP NetWeaver 7.0 is:

- Short texts of forms and text modules: Menu Translation → *ABAP Objects* → *Transport Object*
Entry R3TR/SSFO/<Form>
- Long texts of forms and text modules: Menu Translation → *ABAP Objects* → *Other Long Texts* → *Forms and Styles* → *SAP Smart Forms*
- Styles: Menu Translation → *ABAP Objects* → *Transport Object*
Entry R3TR/SSST/<Style>

Copying/transporting forms:

All languages are always copied/transported.

- **Undo and Redo**
- **Copies Window**
- **Downloading Forms/Parts of Forms**
- **HTML Output and Web Forms**
- **Dynamic texts**

© SAP AG 2006

Undo and Redo

The screenshot shows the SAP Form Builder interface with the title bar "SAP Form Builder Form ZBC470_STEPT". A magnifying glass icon is positioned over the toolbar, specifically over the Undo and Redo icons. Dashed arrows point from the magnifying glass to two small icons below it: a blue arrow pointing left labeled "Undo" and a blue arrow pointing right labeled "Redo".

Since form was last saved

© SAP AG 2002

- As of the SAP Web Application Server 6.10, the Form Builder keeps a record of the changes you have made since the form was last saved. This includes changes to the navigation tree as well as the Table Painter, Form Painter, PC editor, and the input fields of the maintenance screen. You can undo these changes in steps. It is also possible to subsequently redo the changes you have undone.
- For this function, the Form Builder saves the intermediate statuses of the form. The status is only saved when you press RETURN or call an application function.
- After you have called a function, Smart Forms collapses all the nodes in the navigation tree of the Form Builder. Otherwise, the Form Builder would have to remember the exact status of the navigation tree after every activity. This would lead to performance problems when working with the Form Builder.
- For very large form descriptions, saving the intermediate statuses in the Form Builder runtime can have a noticeably negative effect on performance. To deactivate the function, choose: Menu *Utilities* -> *Settings*, tab page *General*, radio button *Undo/Redo Form Changes*.

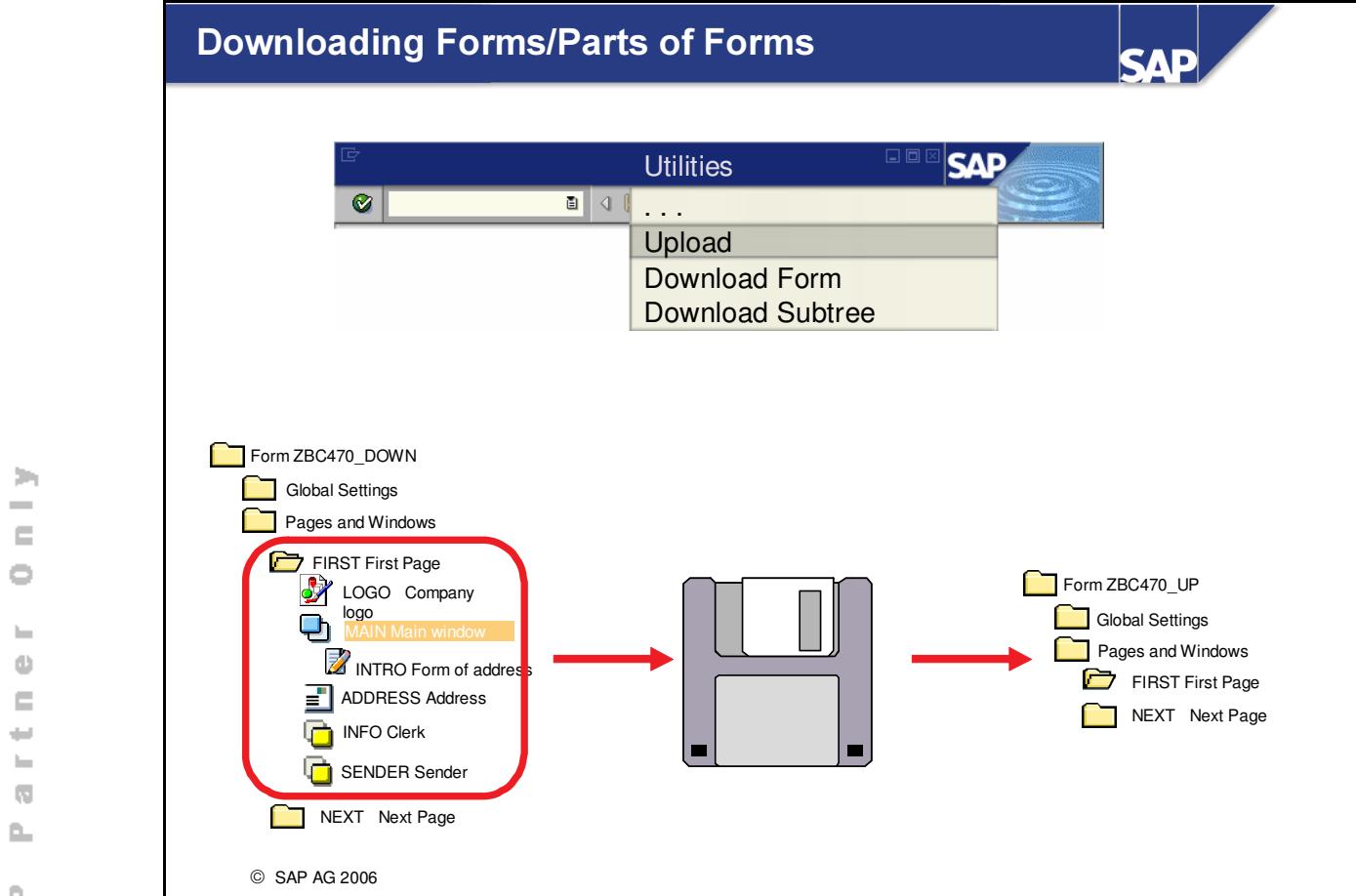
Copies Window

The screenshot shows the SAP Copies Window interface. On the left, there are two examples of forms. The top form is a booking confirmation for 'Fly & Smile' with details like '4 Truckee Way 17, 12345 Atlanta' and 'Mr/Mrs. /Company Plum'. The bottom form is similar. To the right of the forms is a 'General Attributes' panel. Under 'Window Type', it says 'Copies Window'. In the 'Output to' section, the radio button 'Original and Copies - Copies Differ*' is selected. Below this, there are five other options: 'Original and Copies - Copies Identical', 'Only Original', 'Only Copies - Copies Differ*', and 'Only Copies - Copies Identical'. A note states '* Query possible:' followed by a table:

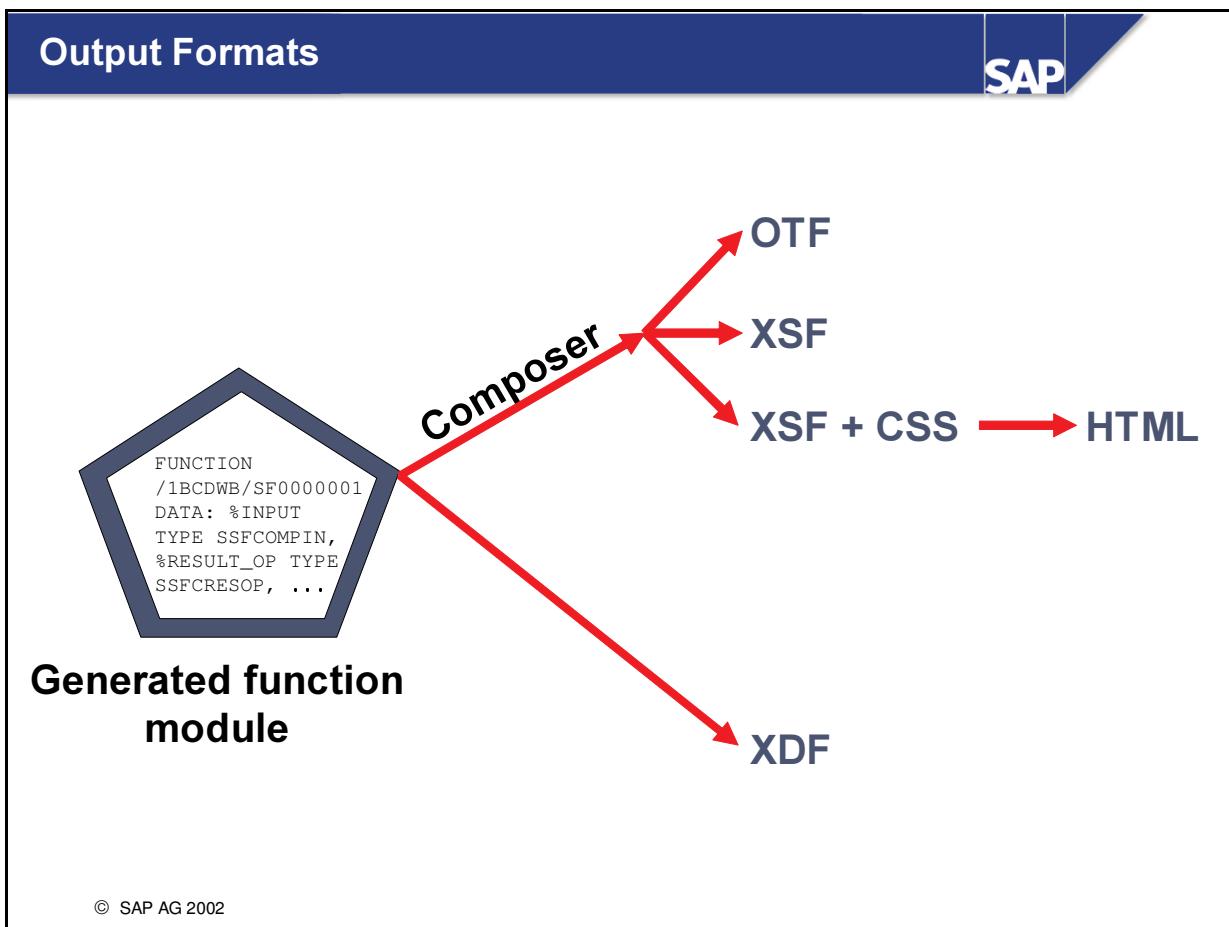
	Original	1st Copy	2nd Copy
SFSY-COPYCOUNT0	0	1	2
SFSY-COPYCOUNT	1	2	3

© SAP AG 2006

- As of the SAP Web Application Server 6.10, there is a new window type: The copies window. A copies window is similar to a secondary window, in that you can define multiple copies windows for one form with any size and positioning on the page, and a copies window cannot contain continuous text. A special feature of the copies window is the additional feature that you can decide to make processing depend on whether the form is printed as an original or as a copy (if you are printing more than 1 copy). It is therefore possible to identify copies individually.
- You have three main options for determining when the window is processed:
 - On the original and the copies.
 - Only on copies
 - Only on the original
- For the first two options, you can then further specify between *Copies Differ* and *Copies Identical*.
 - For identical copies, the window content is only entered in the spool once (with references to the pages on which the window is to be printed). This can notably improve performance, particularly for graphics.
 - For different copies, you can query the system variables SFSY-COPYCOUNT0 and SFSY-COPYCOUNT, for example, in conditions. SFSY-COPYCOUNT0 returns the value 0 for the original, followed by 1 for the first copy, 2 for the second copy, and so on. SFSY-COPYCOUNT is greater by 1, which means the original has the value 1, the first copy 2, the second copy 3, and so on.



- As of the SAP Web Application Server 6.10, it is possible to download a form or parts of a form in XML format to a PC, and subsequently upload it into the same form, or into a different form.
- To download, choose *Utilities* → *Download Form* or *Download Subtree*.
- To upload, choose *Utilities* → *Upload*.
- If you upload a form from your PC into an existing form, the existing form is overwritten (a warning is displayed first).
- If you upload the subtree of a form from your PC into a existing form, the subtree is temporarily stored on a clipboard. You can then select the appropriate position in the navigation tree, and choose *Paste* from the context menu (right mouse button).
- Note: When uploading subtrees of a form, you must ensure that the styles and fields in the downloaded form also exist in the target form.

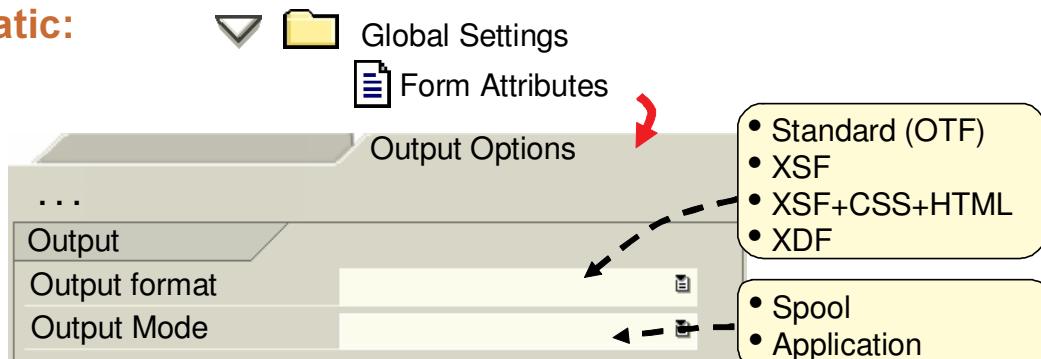


- Normally, a form and its flow logic are processed by the composer. The following output formats are possible:
 - OTF (Output Text Format). OTF is the standard format used in the spool for normal output. It already contains the symbolic names (print controls) for the control characters of the installed printer, and line and page breaks. OTF can be converted into PDF (Portable Document Format). See SAP Note 317851.
 - XSF (XML for SAP Smart Forms) is an XML format that only contains information on the content of the processed form, but not the layout. XSF is used so that external tools can access and further process the data (using the certified XSF interface).
 - As of the SAP Web AS 6.10, a CSS style sheet (Cascading Style Sheet) can be generated in addition to the XSF output. XSF and CSS are then automatically converted on the server side to HTML format, which can be displayed in every Web browser.
- As of SAP Web AS 6.10, you can also call a generated function module so that the form is not processed, but the content of the interface is read. This results in the XDF format (eXtensible Data Format). This format contains the field name of every field, along with the name, type, and time stamp of the dictionary type.

Determining Output Formats

SAP

Static:



Dynamic:

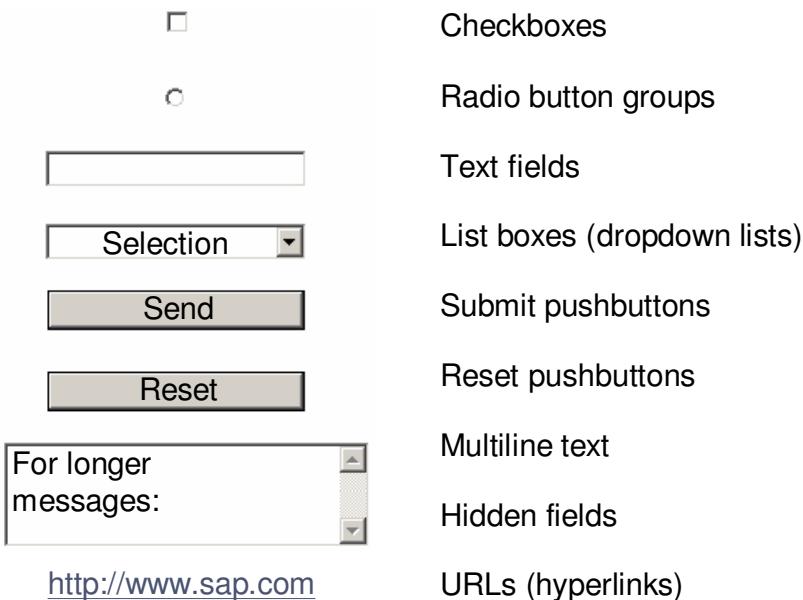
Fields of the import parameter OUTPUT_OPTIONS

	OTF	XSF	HTML	XDF
XSFCMODE	X	X	X	X
XSF	space	X	X	space
XSFOUTMODE	space	A/S	A	space
XSFFORMAT	space	space	X	space
XDFCMODE	X	X	space	X
XDF	space	space	space	X
XDFOUTMODE	space	space	space	S

© SAP AG 2002

- In the form, you can specify which output format to use. You specify this using the form attributes on the *Output Options* tab page. In the *Output Mode* field, determine whether the output can be forwarded to the spool, or to an internal table.
- You can also set the output format dynamically. To do this, make the settings in the fields of the import parameter OUTPUT_OPTIONS as specified above.
- After form processing, you can query the OTF output in the internal table OTFDATA of the export parameter JOB_OUTPUT_INFO. A prerequisite for this is that the GETOTF field of the import parameter CONTROL_PARAMETERS is selected.
- The XSF, CSS, and HTML data is contained in the fields of the internal table XMLOUTPUT, which is a component of the export parameter JOB_OUTPUT_INFO.

Form Elements for HTML Output



© SAP AG 2002

- You can also have fields ready for input in text elements or text modules of your form, which are then filled by the user when displaying the HTML output in a Web browser and then evaluated by a program.
- The following fields are available:
 - Checkboxes
 - Radio button groups: Only one radio button can be selected for each group of radio buttons.
 - Text fields (maximum length 255 characters)
 - Submit pushbuttons. If the user presses a key, the page is called that was either determined statically in the output options of the form or dynamically using the field XSFACTION of import parameter OUTPUT_OPTIONS. The value of the field used gives the labelling of the pushbutton.
 - Reset pushbuttons. When press you this pushbutton, all the entries in the form are reset.
 - Hidden fields: Invisible fields that contain data that is necessary for further processing of the form.
 - List boxes (dropdown lists)
 - Multiline text (text areas): For the input of longer and also multiline text (from SAP Web AS 6.10, Support Package 5).
- You can also mark text as a URL by clicking on the pushbutton on the far right in the Editor.

- 1 Define global fields
- 2 Output fields in text nodes
- 3 Set Web attributes for text nodes:

The screenshot shows the SAP interface for defining simple input fields. A table lists the following fields:

Field Name	Input Type	Field L.	Max.	Dsp Only	Group Name
CHECK1	Checkbox			<input type="checkbox"/>	
TEXT	Text	20	30	<input type="checkbox"/>	
NEXT	Submit			<input type="checkbox"/>	
CLEAR	Reset			<input type="checkbox"/>	

© SAP AG 2002

1. If you want to output input fields or hidden fields in a text node, you must first create these as global fields. (That is, you use fields from the interface.)
2. You then integrate the fields in the text node as normal, for example, using the field list.
3. Finally you have to determine how the fields are to be output on the *Web Attributes* tab page of the text node. These settings only apply to the HTML output. To suppress your output for other formats, you can query field SFSY-XSF (value X for XSF- and HTML output).
 - The parameters *Field length* and *Maximum length* are only of significance for the input type *Text*.
 - To generate multiline text (text areas), you have to create a separate text node and choose the entry *Text area* in the field *Text area*. You can then determine the width and height of the input area.
 - See the online documentation for information on which name/value pairs of the HTML form are transferred when the submit pushbutton is used.

Defining Grouping Input Fields for HTML

SAP

- Invoice
- Order confirmation

Your choice ▾

1 Define global fields

2 Output fields in text nodes

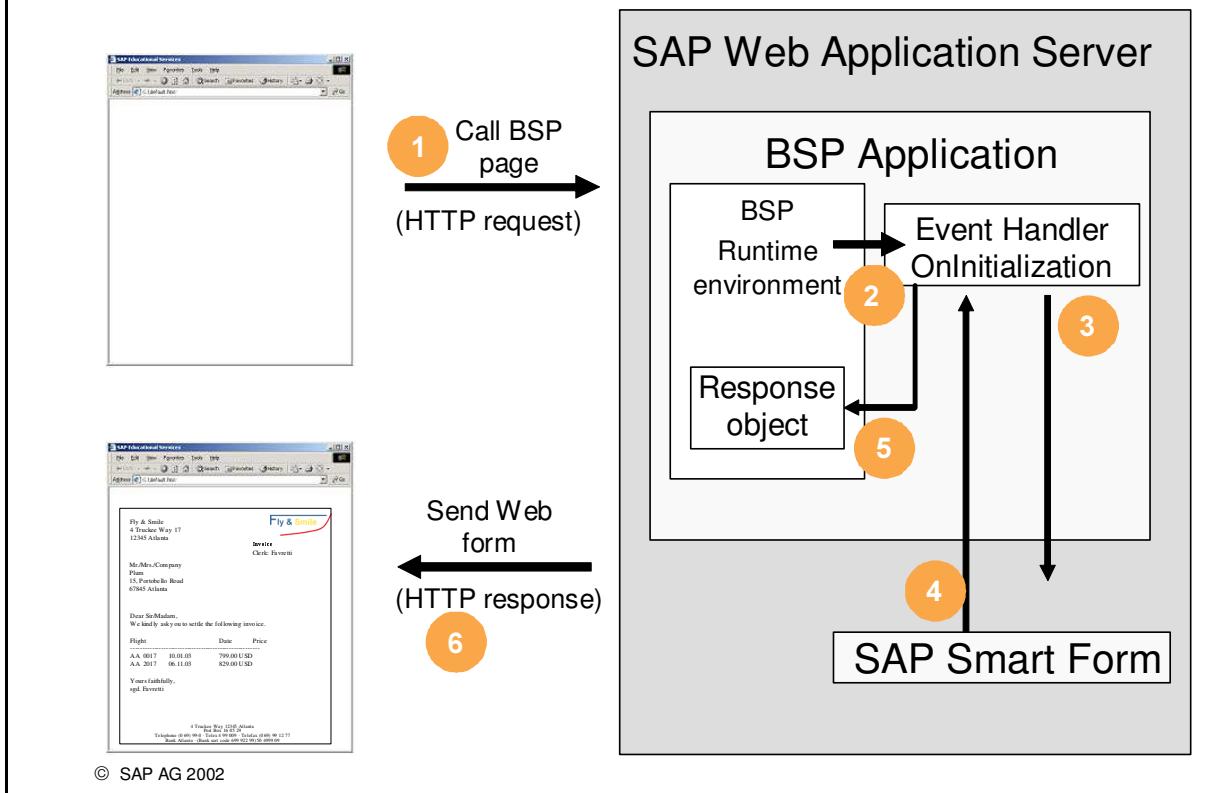
3 Set Web attributes for text nodes and group fields:

Text Entry						
Web Attributes						
Text Area	Text Fields					
Field Name	Input Type	F.	M.	Dsp Only	Group Name	Std Val.
RADIO1	Radiobutton	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'MY_RADIO'	<input type="checkbox"/>
RADIO2	Radiobutton	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'MY_RADIO'	<input checked="" type="checkbox"/>
LIST1	Listbox	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'MY_LIST'	<input type="checkbox"/>
LIST2	Listbox	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'MY_LIST'	<input type="checkbox"/>

© SAP AG 2002

- The procedure when grouping the input types (*Listbox* and *Radiobutton*) is similar - here you must, if necessary, also first define the fields and then integrate them in a text node. However, a difference is that two or more fields within a group are merged in both of the grouping input types. The user can then select exactly one entry for each group in the Web form. You enter the name (any) of a group in the column *Group name*.
- All fields that belong to a group must have the same input type (either listbox or radio button).
- You have the option of marking a value as the standard selection for each group by entering a tick in the column *Std Val.*.
- For radio buttons, the value of the field used as a text is displayed next to the selection button. However, you can also output normal text before it or after it.
- The possible values of a listbox correspond to the values of the fields used.

Web Forms in Business Server Pages



- The SAP Web Application Server is a further development of the traditional client server architecture. By using it, you can directly process requests from the Internet/intranet that were created using a browser with a HTTP log, for example. It is also possible to send a response to the browser.
- Web applications that use this technology are called Business Server Pages (BSP). A BSP application is completely integrated in the SAP Web AS, that is, data can be read from the database or SAP Smart Forms can be called.
- Here is an overview of the steps for displaying a Web form with the help of a BSP:
 1. The user calls the BSP application using a URL in the browser.
 2. The BSP calls the event `OnInitialization`. It dynamically defines the layout of the HTML page.
 3. The generated function module of the SAP Smart Form is called, whereby the HTML output must be activated (statically or dynamically).
 4. The function module returns the HTML data in the parameter `JOB_OUTPUT_INFO`.
 5. This must then (after a conversion) be transferred to the response object with the method `SET_DATA`, so that a HTTP response is generated.
 6. The Web AS sends this HTTP response to the browser so that the Web form can be displayed there.

```

DATA:
  fm_name      TYPE rs381_fnam,          "name of generated function module
  output_data   TYPE ssfcrescl,
  output_options TYPE ssfcompop,
  xmloffput    TYPE ssfxmlout,
  it_html_raw   TYPE tsfixml,
  xstring       TYPE xstring,
  xlenth        TYPE i,
  html_xstring  TYPE xstring.

* set output options
output_options-xsfcmode = 'X'.           "ignore static XSF settings
output_options-xsf      = 'X'.           "XSF: yes
output_options-xsfoutmode = 'A'.          "output: application
output_options-xssfformat = 'X'.          "with CSS
output_options-xsfaction = 'RESULT.HTM'. "URL for submit buttons

* find out name of generated function module (as always)
CALL FUNCTION 'SSF_FUNCTION_MODULE_NAME'
  EXPORTING formname = 'BSP_SMART_FORM'
  IMPORTING fm_name = fm_name.

```

© SAP AG 2002

- You find the most important coding for the BSP event OnInitialization here. Here, for the sake of simplicity, any business data was left out.
- So that input fields can be evaluated, your SAP Smart Form requires a submit pushbutton. By mouse-clicking on it, the page that you added to parameter OUTPUT_OPTIONS-XSFACTION is called. This can be a further BSP or for example another HTML page. Note: For each SAP Smart Form there is only one HTML form, that is, you can only specify one subsequent page.
- If you want to query the user entries on a subsequent BSP, call the method GET_FORM_FIELDS of the request object in the event OnInitialization there.
- For details see the online documentation for the Business Server Pages or the BSPs SF_WEBFORM_01, SF_WEBFORM_02, and SF_WEBFORM_03. For general questions on Web forms, see SAP Note 445605.

```
* Call the generated function module of the SAP Smart Form
CALL FUNCTION fm_name
  EXPORTING
    output_options      = output_options
    user_settings       = space
  IMPORTING
    job_output_info    = output_data.

  xmloffput           = output_data-xmloffput.
  it_html_raw          = xmloffput-trfresult-content[].

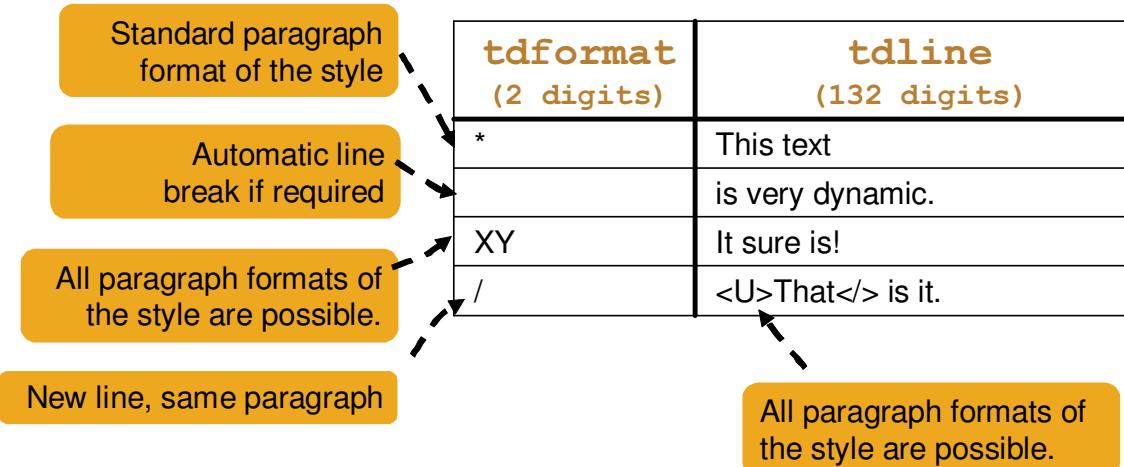
* SAP Smart Forms returns XML data island in raw data format.
* Method 'set_data' of the response object needs the output
* in XSTRING format. The next loop converts the raw table into XSTRING.
LOOP AT it_html_raw INTO xstring.
  CONCATENATE html_xstring xstring INTO html_xstring IN BYTE MODE.
ENDLOOP.

xlength = xstrlen( html_xstring ).

* Fill HTTP request
response->set_header_field( name  = 'content-type'
                             value = xmloffput-trfresult-type ).
response->set_data( data   = html_xstring
                     length = xlength ).
```

© SAP AG 2002

Internal table with line type TLINE



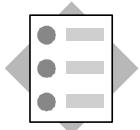
© SAP AG 2006

- In addition to text modules and SAPscript texts, as of SAP Web Application Server 6.30, you can integrate a third variant of long texts into a form: dynamic texts. In this case, you must specify an internal table with line type 'tline', which contains the text in ITF format at runtime. tline comprises two columns: the 'tdformat' column, which contains the name of the paragraph format, and the 'tdline' column, which contains the contents of the line. (In other words, a dynamic text looks exactly like a text module or a SAPscript text in the line editor.)
- If you enter an asterisk (*) as the paragraph, the default paragraph will be taken from the style (Smart Style) that you specify in the context.
- If you want to have continuous text, make sure to enter a paragraph format only for the first line. You can then add as many lines to the internal table as you like, using the tdline column. tdline can contain up to 132 characters. Regardless of how many characters you enter, the resulting text in the form will have one paragraph.
- If you want to apply character formats to some parts of the text, look up the name of a suitable format in the style to be used (it will always have a one- or two-digit name). Put this name in angle brackets, then write the text that needs to be formatted, and close the formatting with </>. For example, if a style contains the character format AB, you might format parts of a text like this: irrelevant <AB>formatted</> irrelevant.
- You include a dynamic text into a form as follows: Create a text node, select text type „Dynamic text“ and enter the name of the internal table in *Field name*.

Contents:

- **Tables**
- **Postprocessing**

© SAP AG 2003



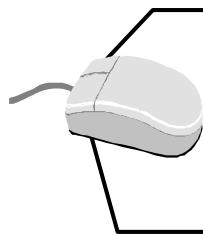
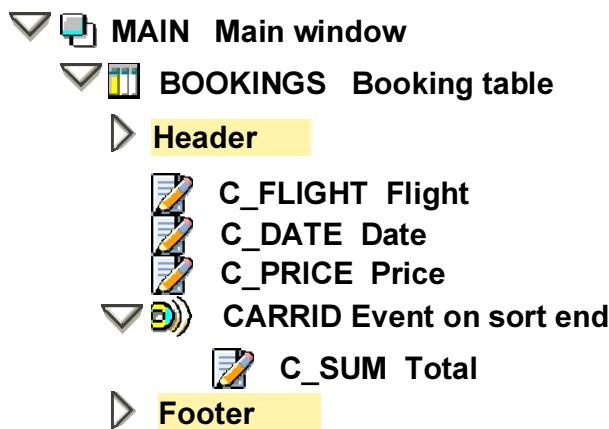
After completing this topic, you will be able to:

- **Create tables and their line types**
- **Output data in tables**
- **Create headers and footers**

© SAP AG 2006

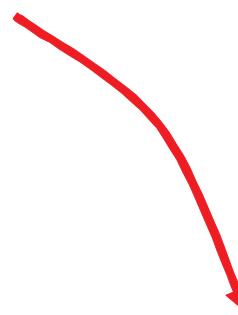
- The table function has existed since SAP R/3 4.6C, but was changed for SAP Web Application Server 6.10. You can still use and edit forms with old table types, but you cannot create old table types. Tables of the old type have their own icon (see Appendix).
- **This unit describes only the procedure for SAP R/3 4.6C.**

Tables for SAP R/3 4.6C: Overview



Create
Cut
Copy
Paste
Delete

Drag and drop



Flight	Date	Price
AA017	12.16.2007	1,200.00 USD
AA017	12.31.2007	1,200.00 USD
Total for AA		2,400.00 USD
LH0400	11.17.2007	581.00 EUR
LH0402	11.17.2007	669.00 EUR
Total for LH		1,250.00 EUR
Sum total		
		2,400.00 USD
		1,250.00 EUR

© SAP AG 2006

- Forms are frequently used to output data in tables. Tables in SAP Smart Forms are subnodes of windows and, like all other subnodes, they are created using the context menu (right mouse button) of the navigation tree.
- Since the length of tables is dynamic, you should use them only in main windows because they may be truncated in secondary windows.
- You can format the individual line types in the graphical Table Painter.
- Tables provide functions to output headers and footers, control levels, and subtotals.

For each line type:

- Definition of number and width of columns
- Height dynamic
- Contents dynamic

Bookings for Lufthansa

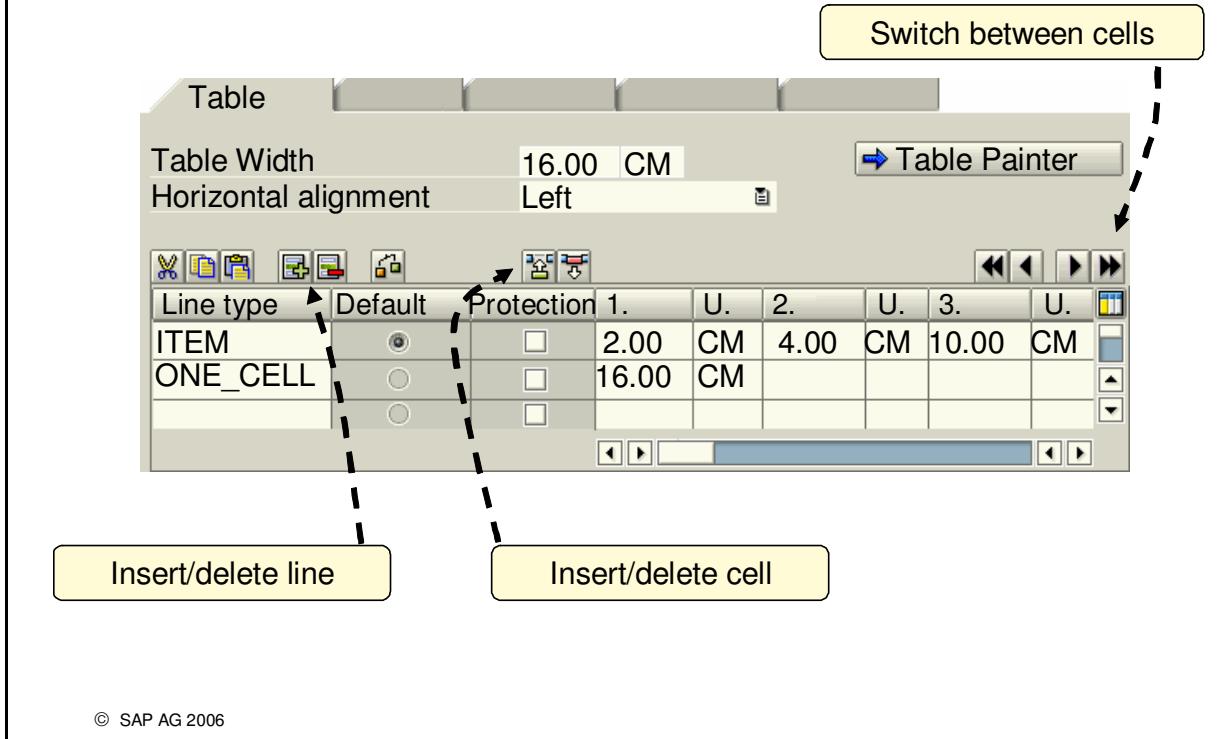
LH0400	11/06/2007	581.00 EUR
LH0402	11/06/2007	669.00 EUR
Total for LH		1,250.00 EUR
Sum total		2,400.00 USD
		1,250.00 EUR

© SAP AG 2006

- Before you can fill tables with text, you must define the line types on the *Table* tab page of the maintenance screen. You also specify how many cells are to be in one table line and what width these cells should be. (The height is determined automatically at runtime.) For simple applications, a single line type is sufficient. However, you can also create different types for hierarchical (multi-level) tables, for example. You do this, for example, if you want to print the different bookings for a flight in the next lines or if you want to use subtotals.
- In the output options of the table contents, you specify which line types are to be used when.
- You can maintain line types numerically or graphically.

Defining Line Types Numerically

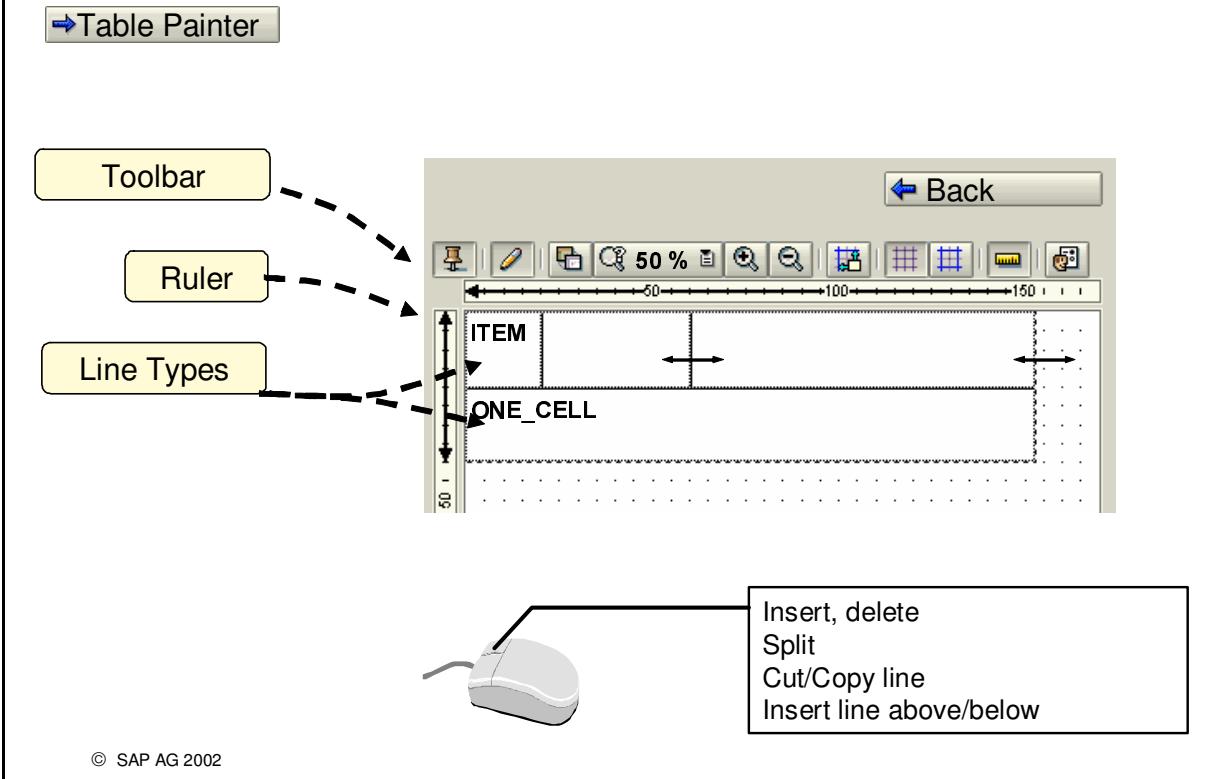
SAP



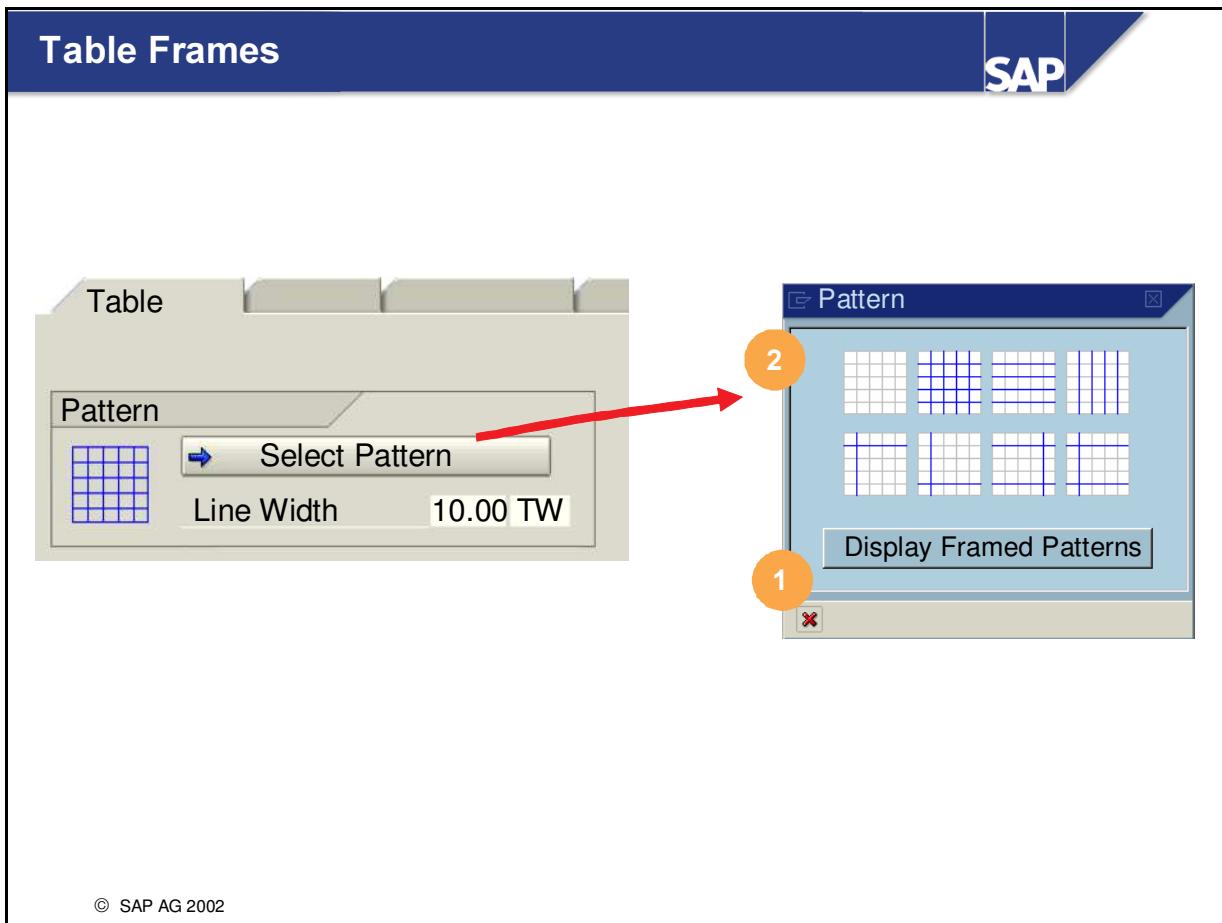
- The following information is required for line types:
 - Name
 - Default type: You can select only one type as the standard type. If no line type is assigned to a subnode of the table, the system uses the default type.
 - Protection against page breaks
 - Number and width of the cells
- The total width of the table must be identical to the total width of all cells for each line type.
- You use the *horizontal alignment* to determine how the table is to be aligned with reference to the window margin. You can choose between *left*, *right* and *centered*. If you choose *left* or *right*, you have the option of specifying a distance from the respective window margin.

Line Types in the Table Painter

SAP



- Instead of entering the line types in the table control, you can also define them in the graphical Table Painter. The entries you make in the Table Painter are automatically copied to the alphanumeric overview and vice versa.
- You insert a new cell by vertically dragging the mouse pointer which has the shape of a pencil at the desired position while keeping the left mouse button pressed. Alternatively, you can divide the cell in which the mouse pointer is positioned using the context menu (right mouse button → *Split* → *Cell*).
- You insert a new line by horizontally dragging the mouse pointer at the desired position. Alternatively, you can use the context menu (right mouse button → *Insert* → *Line*.)
- You change the width of a cell by placing the mouse on a cell boundary and dragging it to the desired position while keeping the left mouse button pressed. (The mouse pointer assumes the shape of a double arrow.)
- You cannot set the height of individual line types because the height is determined dynamically at application program runtime, depending on the data that is output.



- You can define gridlines for the columns and lines of a table. To do this, you select from a number of table patterns. Choose *Select Pattern* on the *Table* tab. You can also set the line width on the *Table* tab.
- Decide whether you want the whole table to be framed or not (***Display Framed Pattern*** pushbutton) Select the pattern you want to use by clicking it with the mouse. You can choose whether the first, the last, or all columns are to be separated by vertical gridlines and/or whether the first, the last, or all line types are to be separated by horizontal gridlines.
- The selected pattern then appears on the *Table* tab.
- You cannot set separate patterns for different line types because the pattern is always applied to the entire table.

Contents in Table Lines

SAP

The screenshot shows a SAP interface for managing bookings. At the top, there's a navigation tree node labeled 'BOOKINGS Bookings'. Below it is a table containing flight information:

LH0400	11.17.2007	581.00	EU
Frankfurt - New York			
Airbus A310			
LH2407	11.17.2007	669.00	EUR
Berlin - Frankfurt			
Boeing 737			

Red arrows point from the table rows to corresponding nodes in the navigation tree on the right:

- LH0400 points to 'C_FLIGHT Flight'
- 11.17.2007 points to 'C_DATE Date'
- 581.00 points to 'C_PRICE Price'
- EU points to 'C_CITY Start/Dest.'
- LH2407 points to 'C_PLANE Airplane'

Below the table is a dialog box titled 'Output Options' with tabs for 'General Properties' and 'Output Options'. The 'Output Options' tab is selected, showing settings for 'Output Table':

- New Line (highlighted in yellow)
- New Cell
- Line Type: ITEM
- 0 Skip Cells

A dashed arrow points from the 'New Table Line' button in the dialog to the 'New Line' checkbox in the settings.

© SAP AG 2006

- To output content in tables, you need to create text nodes, addresses, or graphics as subnodes of the table. These table subnodes have special input fields on the *Output Options* tab page: Here, you specify the output in table lines:

Option *New line*:

This option allows you to select one of the line types for this table line that you defined on the *Table* tab page. If you do not select a line type, the system automatically uses the line type selected as the default type.

The new table line is displayed with gridlines, provided that you have selected a table pattern for the table.

If you select *New cell*, the content is output in the next cell of the line type. If the line type has no more cells to go to, an error message is issued during program execution. You can also skip several cells. If you select *New line*, the text is automatically output in the first cell of the line type selected unless you want to skip cells.

- It is also possible to create more than one node in a cell.
- A table line may extend over two pages at the most.
- To indicate in the navigation tree that specific nodes are in a line, create a folder for these nodes. See Unit 7 - *Process Control*.

The screenshot shows the SAP Fiori Events interface. On the left, under the 'Events' tab, there are two sections: 'Header' and 'Footer with Height'. The 'Header' section has a checkbox for 'Header' and a dropdown for 'Print Time' with options 'at Start of Table' (checked) and 'at Page Break'. The 'Footer with Height' section has a checkbox for 'Footer with Height' and a dropdown for 'Print Time' with options 'at Page Break' (checked) and 'at End of Table'. A height of '3.00 CM' is specified. Red arrows point from the 'Header' and 'Footer with Height' sections to the corresponding nodes in the navigation tree on the right.

Events

Header
Print Time
 at Start of Table
 at Page Break

Footer with Height
Print Time
 at Page Break
 at End of Table 3.00 CM

BOOKINGS

Header

- H_FLIGHT Flight
- H_DATE Date
- H_PRICE Price

Footer

- F_NEXT Next Page

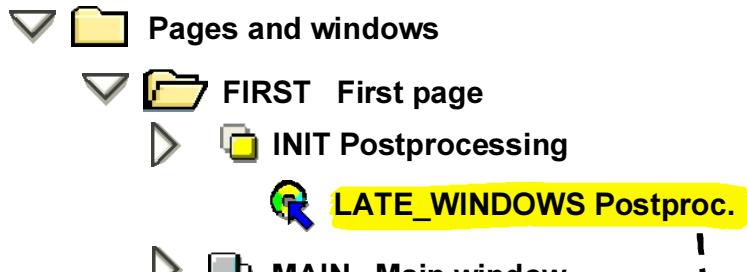
© SAP AG 2006

- You can use events to control the output of headers and footers in a table. To do this, you select a header and/or footer on the *Events* tab of the table node. The corresponding event node then appears in the navigation tree.
- You can output headers at the beginning of the table and/or after a page break. Similarly, footers can be output at the end of the table and/or before a page break. You must specify a height for the footer so that the form processor reserves sufficient space.
- You use headers for column headings, for example. To do this, create a text node and - if required - select an appropriate line type on its *Output options* tab.
- Footers are typically used to output subtotals since footers are always processed when the page break takes place. You calculate subtotals using nodes of the *Program lines* type. (See Unit 7 - *Flow Control*.)
- You cannot create footers and headers directly as nodes in the navigation tree. You must always follow the procedure described for the *Events* tab.
- Note that all lower-level nodes are lost if you deactivate a header or footer.

Postprocessing and Total Page Number

SAP

Fly & Smile 4 Truckee Way, 17 12345 Atlanta	
Ms. Martina Plum 15 Portobello Road 67845 Atlanta	99,00 29,00 29,00 Invoice: \$ 4,085.00 5,00
Dear Ms. Plum, Please settle the following invoice: Flight Price AA017 \$ 799,00 AA029 \$ 829,00	



```
DATA: l_subrc TYPE sy-subrc.  
PERFORM set_late_window  
    IN PROGRAM saplstxbc  
    USING 'LATE1'  
    CHANGING l_subrc.
```

© SAP AG 2006

- You may sometimes need to use a condition on the first page to evaluate the total number of pages of the form being processed (for example, for bar codes in machines for filling envelopes) or want to display a total on a page although this total is not calculated until the form is processed. The variable SFSY-FORMPAGES does indeed already exist, however, its value is not determined until the end of form processing, and it is subsequently inserted by the form processor in a second processing run. Fields for which a value has not yet been determined cannot be evaluated in a normal window.
- As of the SAP Web Application Server 6.10, there is a new window type called *Final window*. Windows of this type are processed only in a second repetition, that is, at a time when the values of all the fields are available, that are not filled until the last page.
- However, in SAP R/3 4.6C, the following procedure can be used to create windows that are first processed by the form processor in a second run:
Create the secondary window that is to be processed in the second run.
On the first form page that is processed, create a secondary window as the top subnode that can only contain a node of type *Program lines*. In this node, a form routine is called for every post processing window and the name of the window is passed to it (see code in the graphic above).
- For further information, see SAP Note 359009. Note that forms that use the described technology must be changed for SAP Web AS 6.10 (remove code and mark secondary window as final window).

- **Forms**
- **Texts**
- **Print programs**
- **Styles**

© SAP AG 2006



SAPscript

Forms

Problem: Different Concepts

- Interface between print program and form
 - Formats
 - SAPscript command
 - Loop logic
- Migrate with tool and adapt



© SAP AG 2002

- SAP delivers SAP Smart Forms for important business processes. You can also convert your existing (active) SAPscript forms into SAP Smart Forms. There are two tools that support this conversion process. Note, however, that SAPscript forms are integrated into programs in an entirely different way than SAP Smart Forms and that you might therefore need to manually adjust the form and the print program which may be very time-consuming if the form you want to convert is very complex.
- You can convert individual forms on the initial screen of the SAP Smart Forms transaction. Enter the name of the SAP Smart Form to be created in the *Form* field and choose *Utilities → Migrate SAPscript form*. Then choose the SAPscript form and the language you want to migrate. The program first tries to find the SAPscript form in the current client; if the form does not exist there, the system looks in client 000. If you choose *Enter*, the form is migrated. The system then takes you automatically to the Form Builder where you can make any necessary adjustments and save and activate the new form.
- When you migrate a form, general form settings and layout information including pages, windows, and their attributes and positions on the pages are copied unchanged. However, the definition of the paragraph and character formats is lost (since these are not saved in the form itself but centrally in styles in SAP Smart Forms). As far as text elements are concerned, their character formats are preserved but not their paragraph formats or SAPscript commands. Note that in SAP Smart Forms, in contrast to SAPscript, all fields must be defined explicitly.



SAPscript

Texts

Use as include texts

Caution:

- Commands
- Styles
- Fields
- Client dependency

Print programs

Adapt by hand

Styles

Migrate with tool

© SAP AG 2002

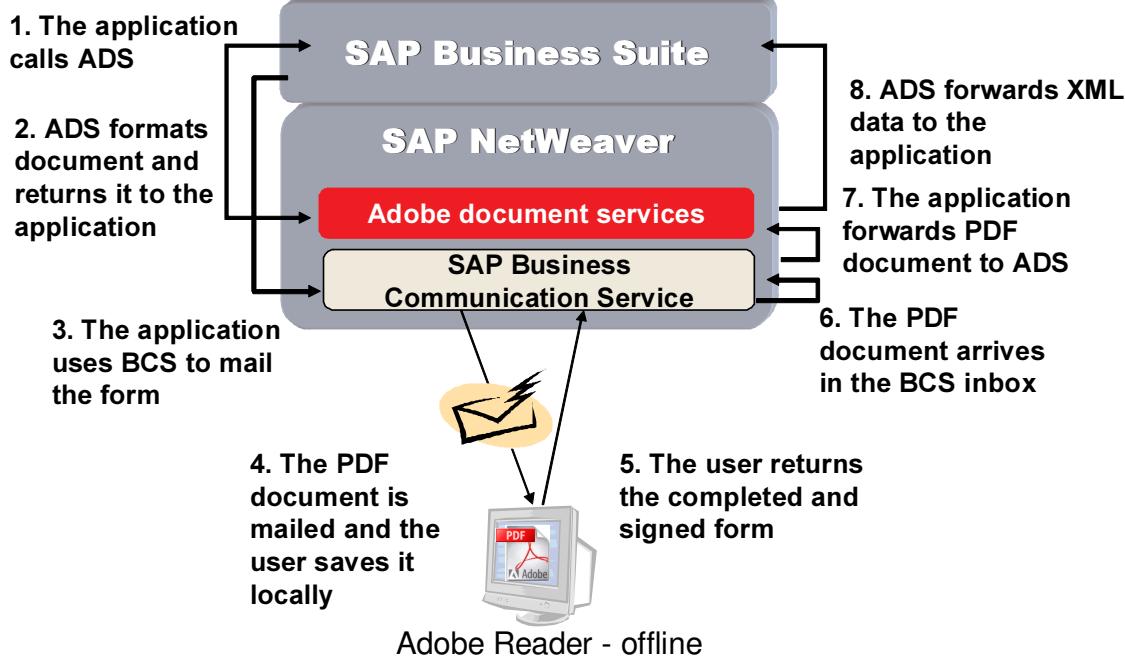
- SAPscript texts can be used directly in SAP Smart Forms. Note, however, that SAPscript commands are not executed within these texts and that SAPscript styles are generally ignored. Besides, all fields of the texts in the SAP Smart Form must be defined - otherwise, the generated function module terminates with an error message. Note that SAPscript texts are client-specific, while SAP Smart Forms are not. This is why you have to make sure that the texts are available in all production clients.
- Print programs must be adjusted manually. In particular, you must replace the function modules OPEN_FORM, CLOSE_FORM, START_FORM and END_FORM that were needed previously with the generated function module with the interface. The processing of the text elements with the function module WRITE_FORM, including the setting/deleting of headings, must be completely moved from the print program to the SAP Smart Form.
- SAPscript styles can be easily converted into Smart Styles. To do this, go to the initial screen of the Smart Styles maintenance transaction (SMARTSTYLES) and choose *Smart Styles* → Convert SAPscript style.

- Overview

© SAP AG 2006

SAP Interactive Forms: Offline Scenario

SAP

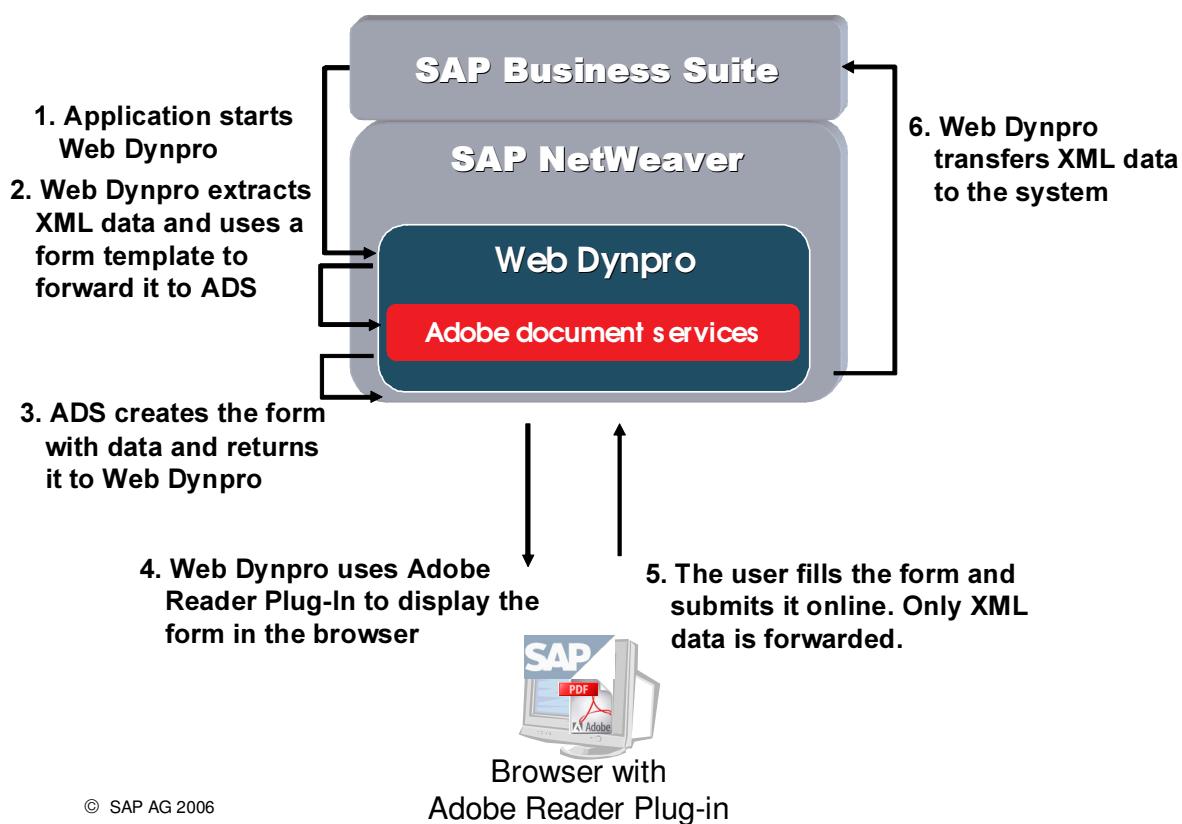


© SAP AG 2006

- Since SAP NetWeaver 2004, SAP Interactive Forms by Adobe provides another form tool that is used in some applications instead of or in addition to SAPscript/SAP Smart Forms.
- There are roughly three different scenarios:
 - In the **offline scenario**, a PDF document is created with reader rights that allow user entries. This completed form can subsequently be returned to the system at any time and the system will extract the data.
 - In the **online scenario**, the form that is ready for input is integrated into a Web application (for example, Web Dynpro). The user enters data and then sends it, which means that the data is transferred to the system.
 - The **print scenario** corresponds to the print scenario of SAP Smart Forms as regards its idea, but not its architecture.

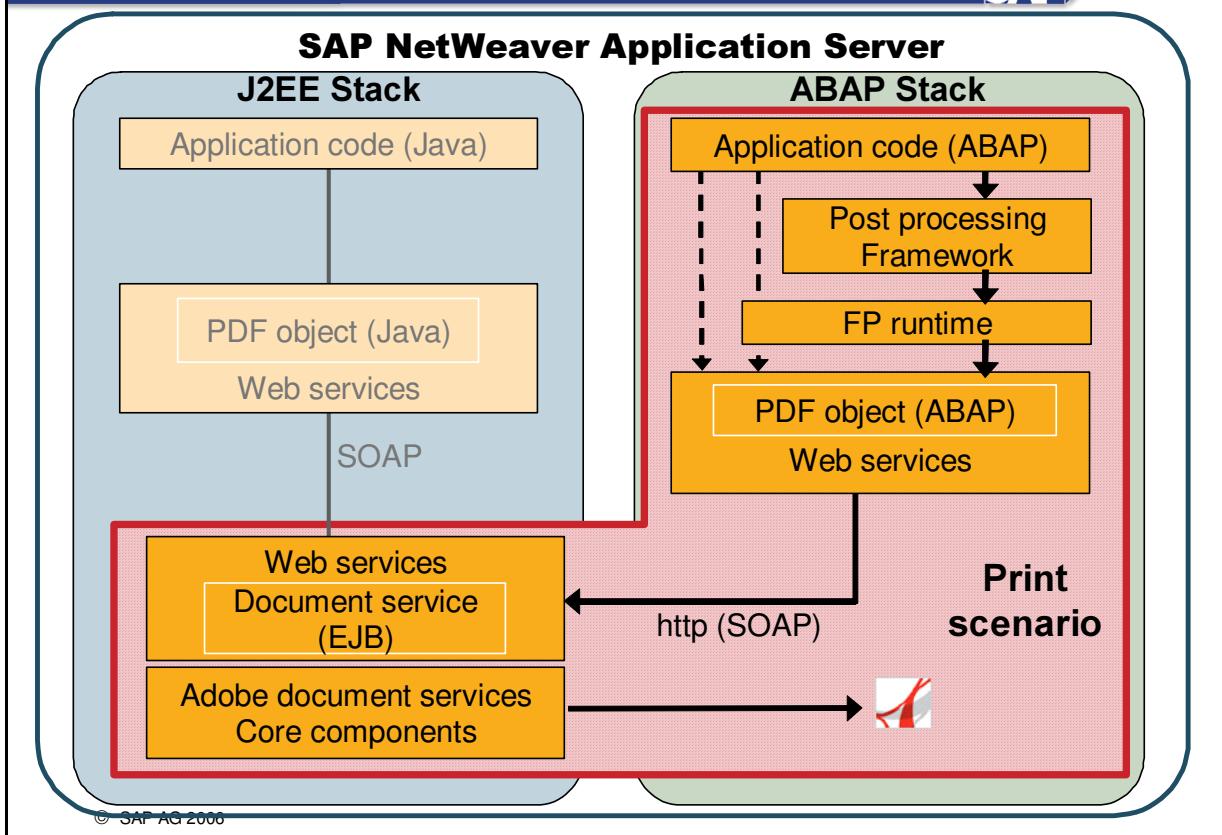
SAP Interactive Forms: Offline Scenario

SAP



© SAP AG 2006

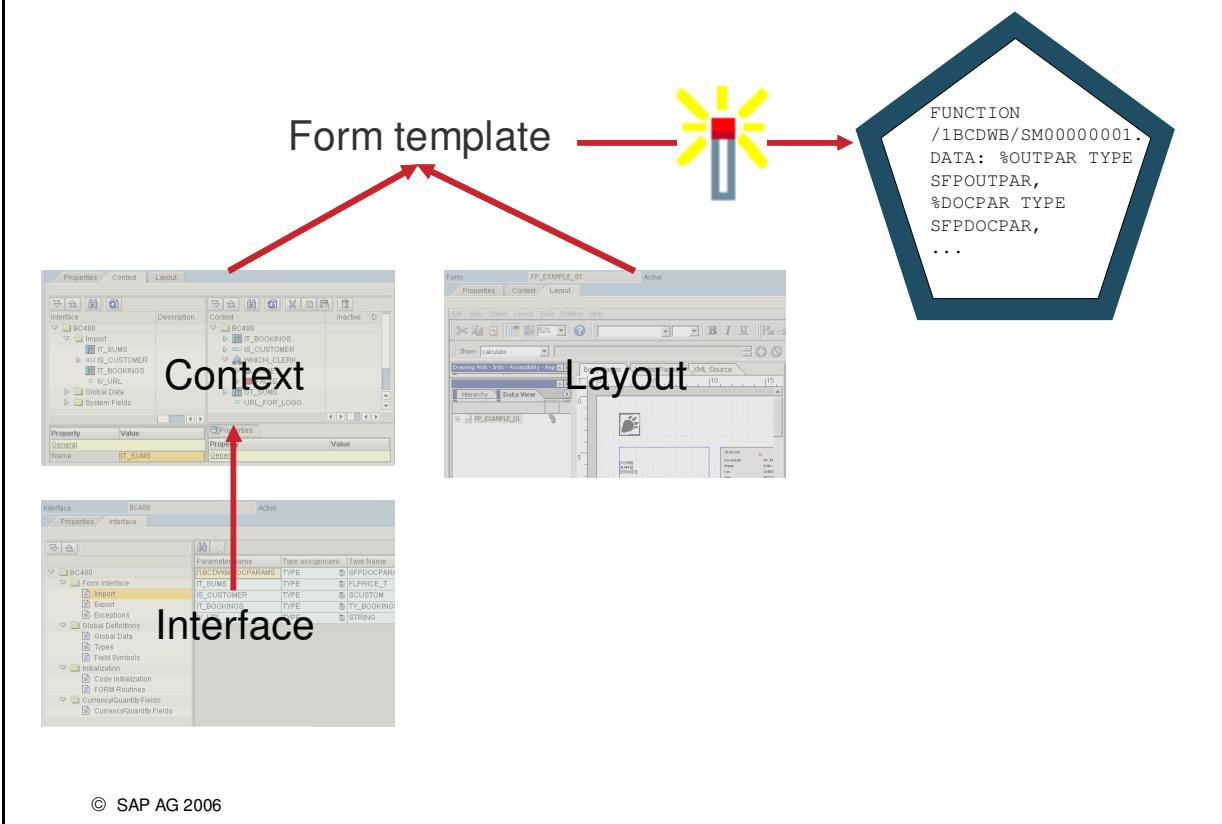
Architecture of a print scenario



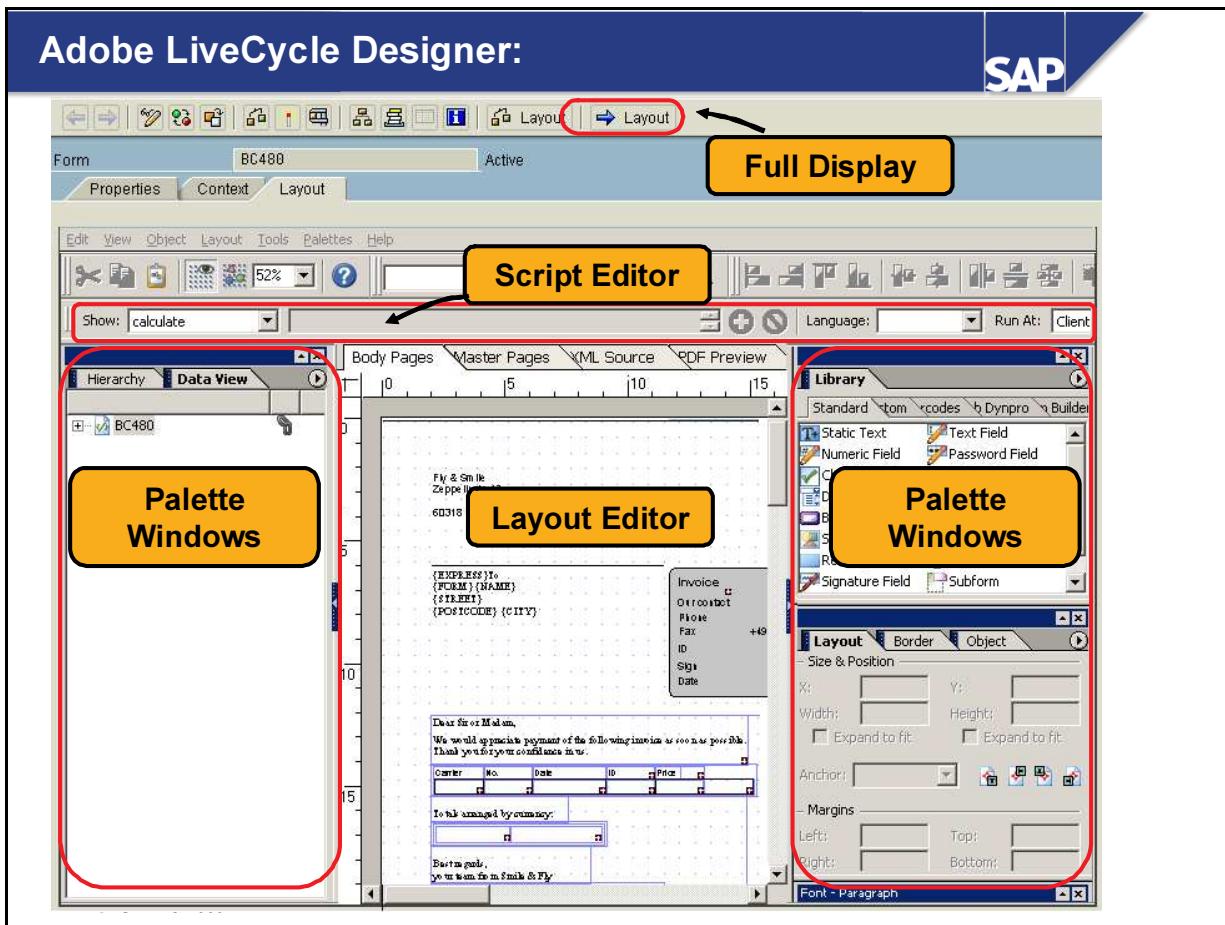
- A short overview of the runtime architecture of a print scenario
- At runtime, an ABAP application program uses the Post Processing Framework (PPF) to determine whether an output is required and, if so, which output. The Post Processing Framework provides SAP applications with a uniform interface for condition-dependent generation of actions (for example, printing delivery notes, faxing order confirmations, or triggering approval procedures). The actions are generated if specific conditions occur for an application document. They are then either processed immediately or at a later time. PPF is the successor to output control.
- The form processing runtime provides functions such as opening and closing a spool job for PDF-based forms. It will then call a Web service, which will communicate with Adobe document services via the XML standard Simple Object Access Protocol (SOAP) using the http protocol. Adobe document services are responsible for form rendering, including filling in fields, page breaks, or applying layout settings.
- Adobe document services must be deployed on the J2EE engine. You include the Web Service Adobe document services as well as other services that can be installed with the SAP NetWeaver Application Server.

Print scenario: Tools Involved (Design Time)

SAP



- To create a print form, you must carry out the following steps:
- Interface (transaction SE80 or SFP): The interface determines which data a program can pass on to a form. It also contains global data and initialization coding that can be used in a form.
- Form design (transaction SE80 or SFP): In general, a print form design consists of the context and the layout.
- In the context, you define which parts of the interface you actually want to use in that particular form. You can also add elements like text modules or images.
- The layout usually consists of static and dynamic elements. The layout is defined in Adobe LiveCycle Designer. It is displayed in a special XML format.
- Before a form template can be used, it must be activated. When a form template is activated, the system generates a function module. The generated function module encapsulates all properties of a form and is called whenever a program triggers form processing.



- Form processing takes place in transaction SFP or SE80. Adobe LiveCycle Designer is integrated there to determine the form design.
- The Designer workspace consists of four main areas.
- In the top area, the Script Editor can be displayed. It allows you to enter scripts for calculations. You can choose between JavaScript and Adobe's FormCalc.
- The subdivisions of the left and right areas are called pallet windows with further subdivisions of pallets. It is up to you to decide which pallet windows you want to display and in which size.