

SAP User Experience –  
Accessibility  
SAP INFORMATION



# Accessibility Guidelines for ABAP Dynpro

**Version 3.0**

September 18, 2007



# Copyright

© 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.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, 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, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation 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.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

## Table of Content

<b>1</b>	<b>Disclaimer .....</b>	<b>4</b>
<b>2</b>	<b>General .....</b>	<b>5</b>
2.1	Prerequisites.....	5
2.2	Terminology .....	5
2.3	General Rules for Programming in ABAP.....	6
2.4	Accessibility Mode .....	6
2.5	Creating Tooltips.....	7
2.6	Creating Labels.....	10
<b>3</b>	<b>UI Elements and ABAP Language Elements .....</b>	<b>15</b>
3.1	Icon .....	15
3.2	Pushbutton.....	16
3.3	Checkbox.....	16
3.4	Radio Button .....	17
3.5	Input Field .....	19
3.6	Combo Box and Dropdown Box .....	20
3.7	Menu.....	21
3.8	Graphics, Charts, Maps, and Bitmaps .....	23
3.9	Table and Tree (Table Control, ALV Grid, ABAP List, ALV Tree, and Tree Control) .....	25
3.10	Special Recommendations for ALV Grid .....	27
3.11	Special Recommendations for Table Control .....	29
3.12	Special Recommendations for Tree Controls .....	29
3.13	Special Recommendations for ALV Tree Control .....	31
3.14	ABAP List.....	32
3.15	Step Loop .....	33
3.16	HTML Control .....	33
3.17	Calendar Control.....	33
3.18	Tabstrip .....	35
3.19	Group.....	36
3.20	Selection Screen.....	37
<b>4</b>	<b>How to Integrate Third Party Products .....</b>	<b>41</b>
4.1	Embedded Components .....	41
4.2	Launching External Programs .....	41
<b>5</b>	<b>Testing ABAP Dynpro .....</b>	<b>42</b>
5.1	Prerequisites for the Test Environment .....	42
5.2	General Rules for Testing ABAP Applications.....	42
5.3	Testing UI Elements .....	42
<b>6</b>	<b>Useful Links .....</b>	<b>50</b>

# 1 Disclaimer

These guidelines do not represent any promise or obligation on SAP's part to make any aspect of the software accessible, nor any application that is evaluated against the Accessibility checklist accessible for ABAP Developers.

This document is for informational purposes only. Its content is subject to change without notice, and SAP does not warrant that it is error-free. **SAP MAKES NO WARRANTIES, EXPRESS OR IMPLIED, OR OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.**

The information contained in this document represents SAP's current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP specifically disclaims any liability with respect to this document and no contractual obligations or commitments are formed either directly or indirectly by this document. This document is for internal use only and may not be circulated or distributed outside your organization without SAP's prior written authorization. © 2006 SAP AG

## 2 General

### 2.1 Prerequisites

For information about the prerequisites, see the newest version of the document "Front-End Requirements and Infrastructure for Accessibility" (settings in SAP GUI for Windows, information about screen reader and SAP GUI Screen Reader Extension). You can access this document at [SAP Design Guild Edition Accessibility](#) under "Resources" → "Frontend Requirements and Infrastructure for Accessibility".

### 2.2 Terminology

Term	Description
Caption	Textual description on or beside a user interface element. Static part of the dialog. Language-dependent.
Context	Situation inside an application.
Label	Textual description on or beside a user interface element. Static part of the dialog. Language-dependent. Typically aligned left of a succeeding user interface element.
Title	A visually attractive title text above a subordinate screen area. Language-dependent.
Tool Tip & Quick Information	Additional information about a user interface element and its range of use. Example: A label "To" can have a tool tip "Recipient the mail is being sent to".
Tutor Message	An extract of information about how to manipulate a focused user interface element. Further information is available in the online documentation. Example: On a button, the tutor message could be "Press [Space] to activate".  Tutor Messages cannot be defined by application development, since this information is provided by the screen reader extensions which are installed locally on the client computer.
Screen Reader	Assistive software for blind users. The software extracts the screen content, formats it, and sends it to an output device. The output device can be a speech synthesizer or a refreshable braille display. Both devices can also be used together. The output of the screen reader can also be displayed on the screen.
Value	Textual data (of any format) inside data input fields.

You can find more terms and their descriptions in the [Accessibility Glossary](#) of the SAP Design Guild.

## 2.3 General Rules for Programming in ABAP

We recommend that you observe these rules when programming with UI elements in ABAP:

- Use ALV Grid instead of ABAP lists, if possible.
- Use ALV Grid or table controls instead of step loops, if possible.
- To create forms, use *SAP Interactive Forms by Adobe* instead of SAP Smart Forms or SAP-script.
- Avoid using HTML controls. When you use HTML controls, you must ensure that accessible content is displayed in the control.
- Create a solution that can be used by all users and that looks the same in accessible mode and in non-accessible mode. (More information on this topic is available in chapter 3.8.)
- Avoid abbreviations. Abbreviations are difficult to understand when read by a screen reader.
- Create titles for complex elements such as ALV Grid and table controls to help the user understand what they are for.

## 2.4 Accessibility Mode

You can use the function module `GET_ACCESSIBILITY_MODE` in an application to query whether the user has set accessibility mode. This query is generally only necessary when an application has both an accessibility mode and a non-accessibility mode. If this is the case, you can use this function module to offer the accessible solution directly to users who have activated the accessibility mode.

You can call this function module as follows:

```
TYPE-POOLS ABAP.  
DATA l_mode TYPE abap_bool.  
CALL FUNCTION 'GET_ACCESSIBILITY_MODE'  
  IMPORTING  
    accessibility      = l_mode  
  EXCEPTIONS  
    its_not_available = 1  
    OTHERS           = 2.
```

*Listing 2.1: Code Example for Calling: GET\_ACCESSIBILITY\_MODE*

We recommend that you query this function module as rarely as possible. Instead, create a solution that looks the same whether in accessibility mode or not. Remember that accessibility mode is also used by users without visual impairments. Users with motor impairments, for example, switch on accessibility mode for the additional support it gives when using the keyboard.

If you use the function module `GET_ACCESSIBILITY_MODE` to replace graphics with tables in accessibility mode, users with motor impairments then have no way of viewing these graphics in accessibility mode. We can assume that this group of users would prefer to have the information displayed in a graphic and not a table.

We recommend that you design your solution so that it offers both the graphic and a text alternative, such as a table. You can also implement a pushbutton that allows the user to toggle between the graphic and the text alternative. Chapter 3.8 discusses the various options in more detail.

Another alternative is to let the user decide whether to display the information as a graphic or as a table by making a personalized setting in the application. This requires you to integrate an appropriate dialog into the application.

## 2.5 Creating Tooltips

A tooltip displays information about SAP GUI for Windows elements in a short description that appears when the mouse pointer hovers over the element. You can also display the text of a *focused* element by using the key combination [Ctrl]+[q].



*Figure 2.1: Pushbutton with Tooltip*

**Figure 2.1** shows a pushbutton *Details* with the tooltip *Flight Details*.

We refer to the usage of tooltips several times over the next few sections. Here is some advance information about how to create tooltips for the following UI elements:

- Input field
- Pushbutton
- Radio button
- Checkbox
- Combo box and dropdown box

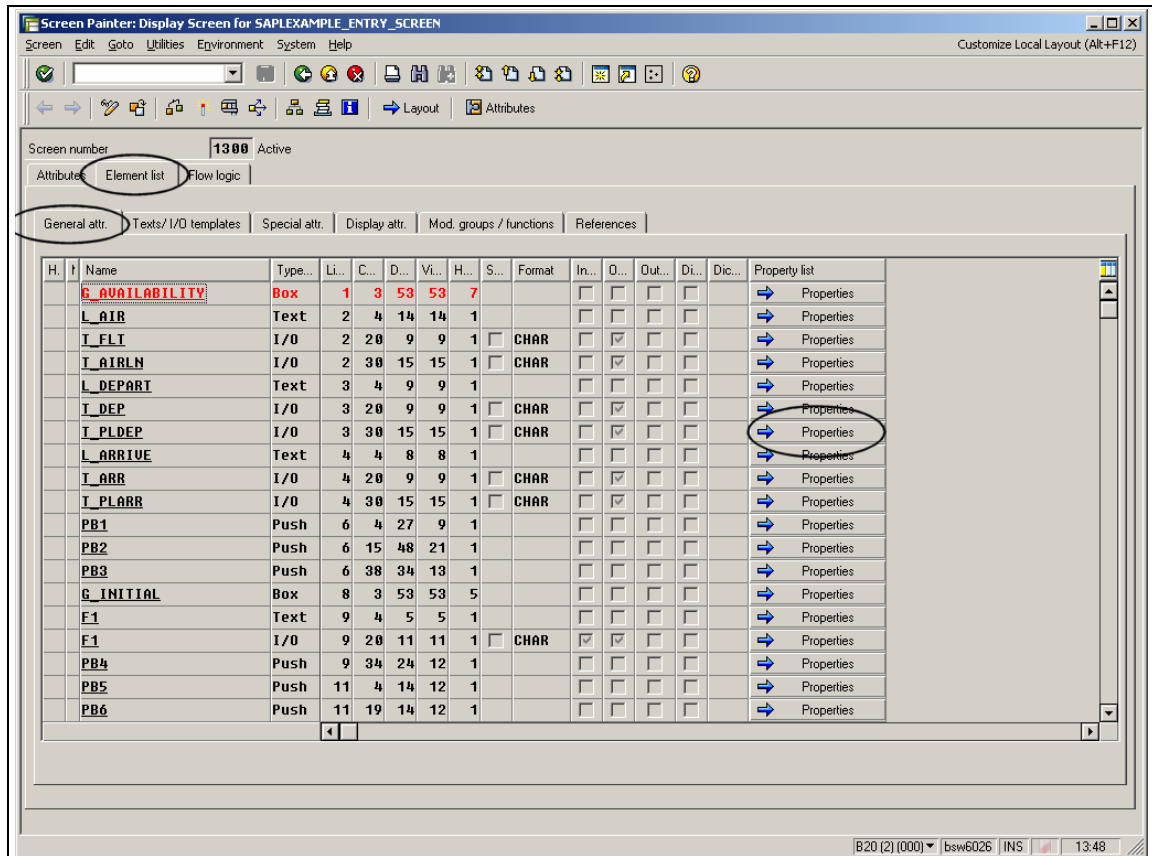


Figure 2.2: Screen Painter – General Attributes Tab Page

To enter a text for a tooltip, proceed as follows:

1. In *Screen Painter* (transaction code SE51), choose the *Element list* tab page.
2. Choose the *General attributes* tab page, as shown in **Figure 2.2**.
3. Choose a screen element and then choose its *Properties* pushbutton. The *Properties* dialog box is displayed, as shown in **Figure 2.3**.
4. On the properties dialog shown in **Figure 2.3**, choose the *Text* pushbutton in the *Tooltip-Text* row and specify the text element or variable that contains the tooltip.

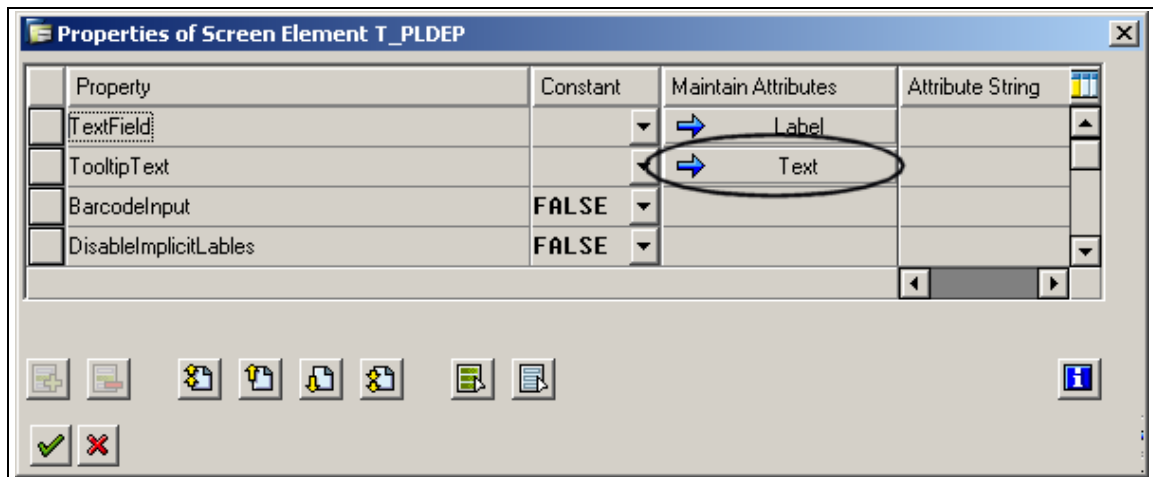


Figure 2.3: Screen Painter – Properties Dialog

Alternatively, you can call the attribute window with the relevant pushbutton from the application toolbar in *Graphical Screen Painter*.

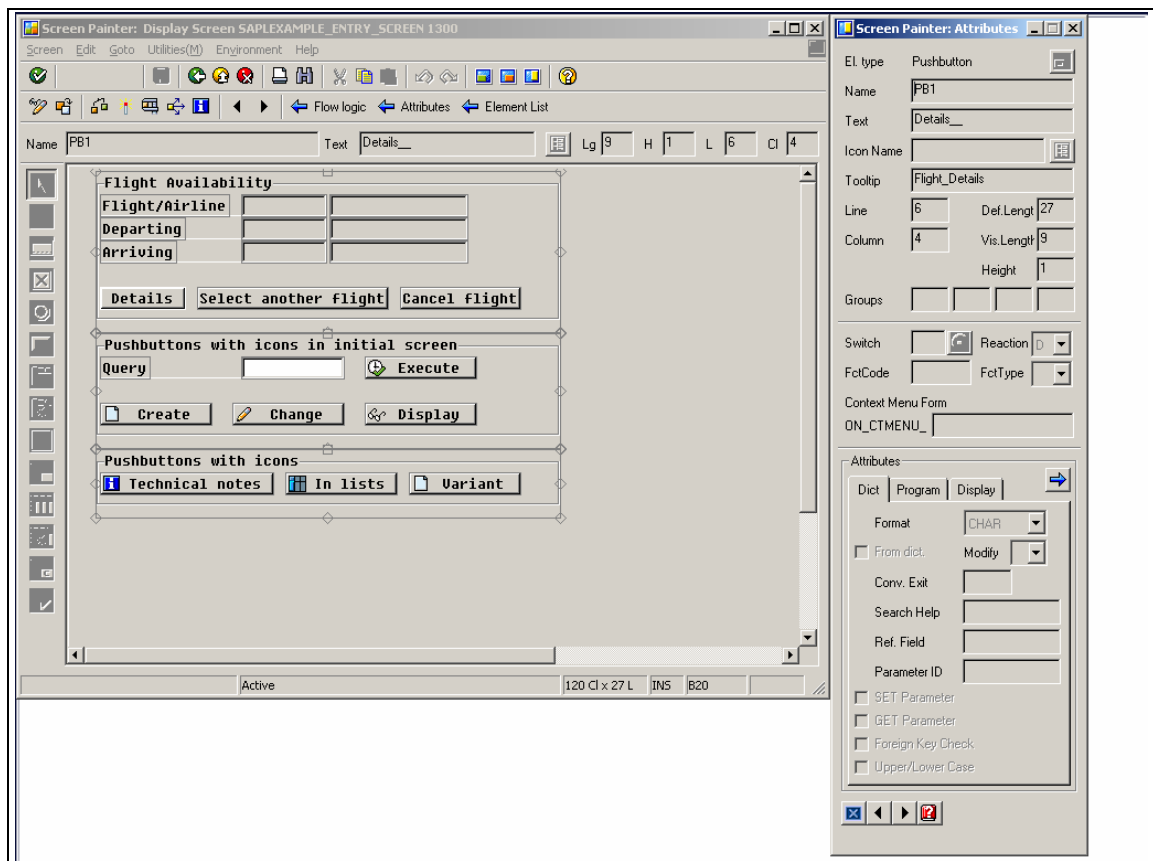


Figure 2.4: Graphical Screen Painter – Attributes Window

When you choose the *Layout* pushbutton in the *Attributes* window, as shown in **Figure 2.4**, the *Properties* dialog is displayed here as well. To specify the text element or variable that contains the tooltip, choose the *Text* pushbutton in the *TooltipText* row on the *Properties* dialog.

**Notes:**

- The functions described in this section are available in SAP R/3 Enterprise or Basis Support Package 44 (Release 4.6C) or higher.

- From Basis Release 700, you can use the method `set_accdescription` to create a tooltip for each front-end control derived from the class `CL_GUI_CONTROL`.

## 2.6 Creating Labels

A label is a name given to a UI element that can be accessed by the user. A control element consists of a label and an interaction element. A checkbox, for example, consists of a checkable interaction element and the accompanying label text. The interaction element is the part of the control element that is relevant for the display of the data, either as text or as a graphical indicator.



*Figure 2.5: Checkboxes with the Labels “Direct flights” and “Cheap flights”*

The label text must be meaningful and unique. The individual UI element determines whether the label is placed on the left or right of the interaction element. Generally speaking, the labels of checkboxes and radio buttons are placed on the right of the interaction element, and the labels of input fields, combo boxes, and dropdown boxes are placed on the left. However, there are some exceptions and special cases. The following describes the rules for labels and the relevant exceptions. It goes on to discuss special cases, such as *As label on right* for the short description of an input field.

### Dos for Labels

- Use meaningful labels.
- Place the label of a checkbox on the right of the square checkable element, as shown in **Figure 2.5**. In exceptional cases, you can place the label on the left of the element, for layout reasons. One example of such an exception is when you have arranged input fields vertically, and placed a checkbox underneath them. Labels for input fields are placed on the left of the fields, which is why in this case, the checkbox can also have its label on the left.
- Place the label of a radio button on the right of the round element.
- Place the label of an input field on the left of the input area.
- Place the label of a combo box or a dropdown box on the left of the combo box or dropdown box.

We refer to the usage of labels several times over the next few sections. Here is some advance information about how to create labels for the following UI elements:

- Input field
- Radio button
- Checkbox
- Combo box and dropdown box

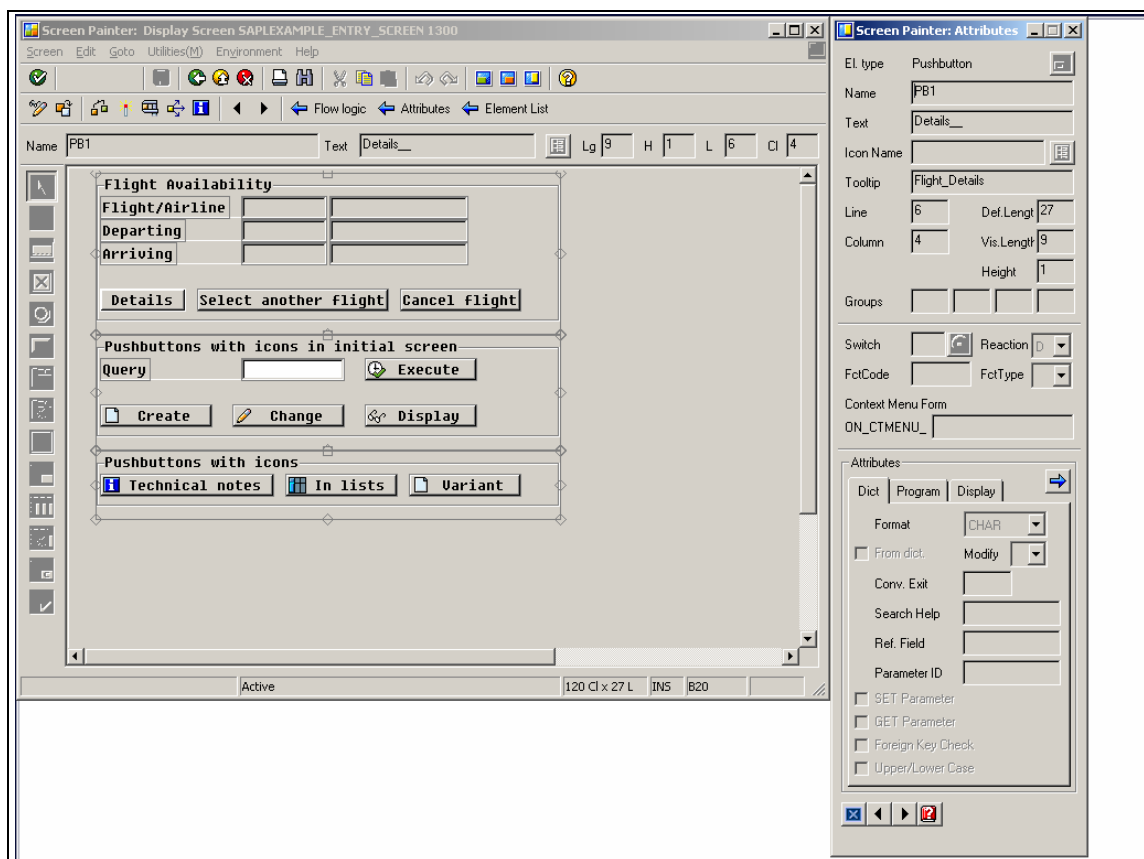


Figure 2.6: Graphical Screen Painter – Attributes Window

In the application toolbar of the *Graphical Screen Painter*, choose the pushbutton that displays the attribute window.

If the label is on the left of the interaction element, go to the *Display* tab page and activate the *As label on left* attribute for the field that contains the label, as shown in **Figure 2.7**.

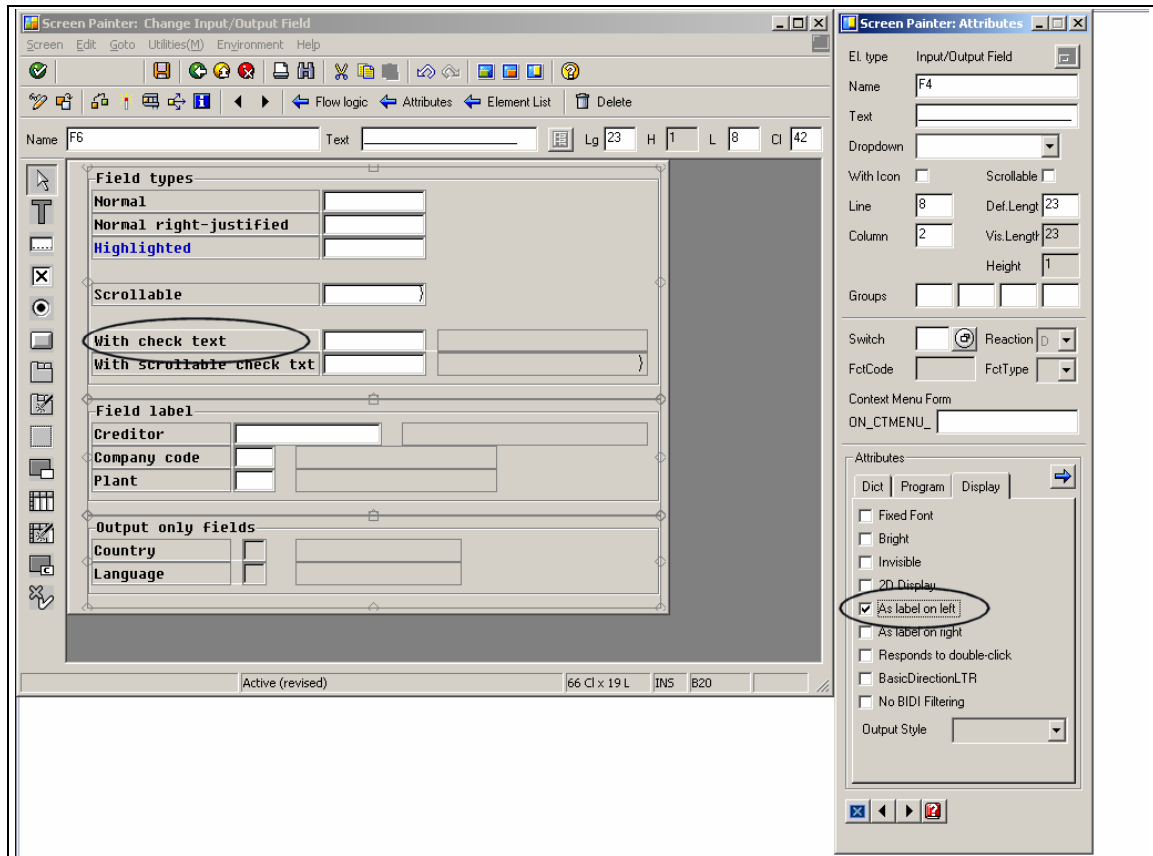


Figure 2.7: Graphical Screen Painter – “As label on left” Attribute

If the label is on the right of the interaction element, activate the *As label on right* attribute, as shown in **Figure 2.8**.

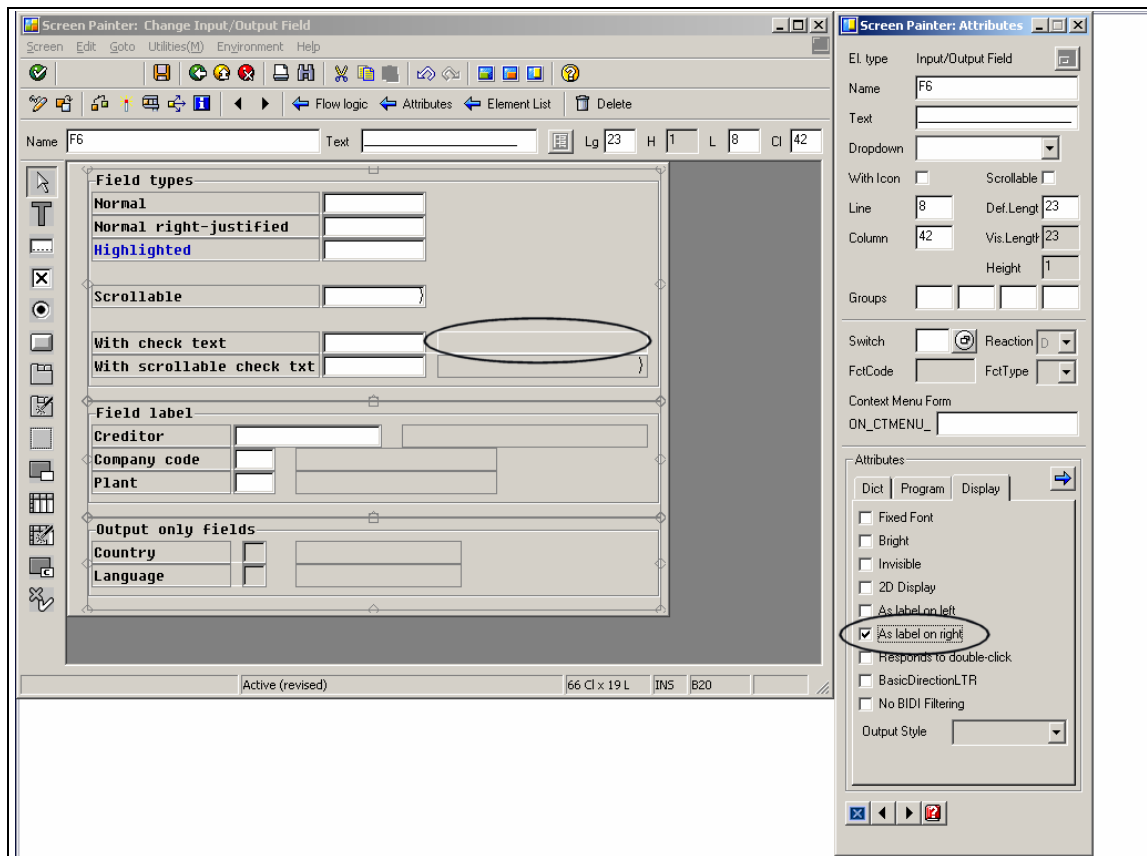


Figure 2.8: Graphical Screen Painter – “As label on right” Attribute

Figure 2.9 shows an example in which you need both attributes, *As label on left* and *As label on right*. The input field with the value **0001** has both a label, here *Company Code*, and a field with a short description that refers to the value in the input field.



Figure 2.9: Example of an Input Field with a Short Description

If you have set both attributes correctly, a screen reader can read the relationship between the label, the input field, and the field with the short description correctly.

When you navigate with the tab key, the focus moves to the input field. It does not move to the label *Company Code* or the short description *Helsinki*. In this way, the user can access all relevant data with a single use of the tab key.

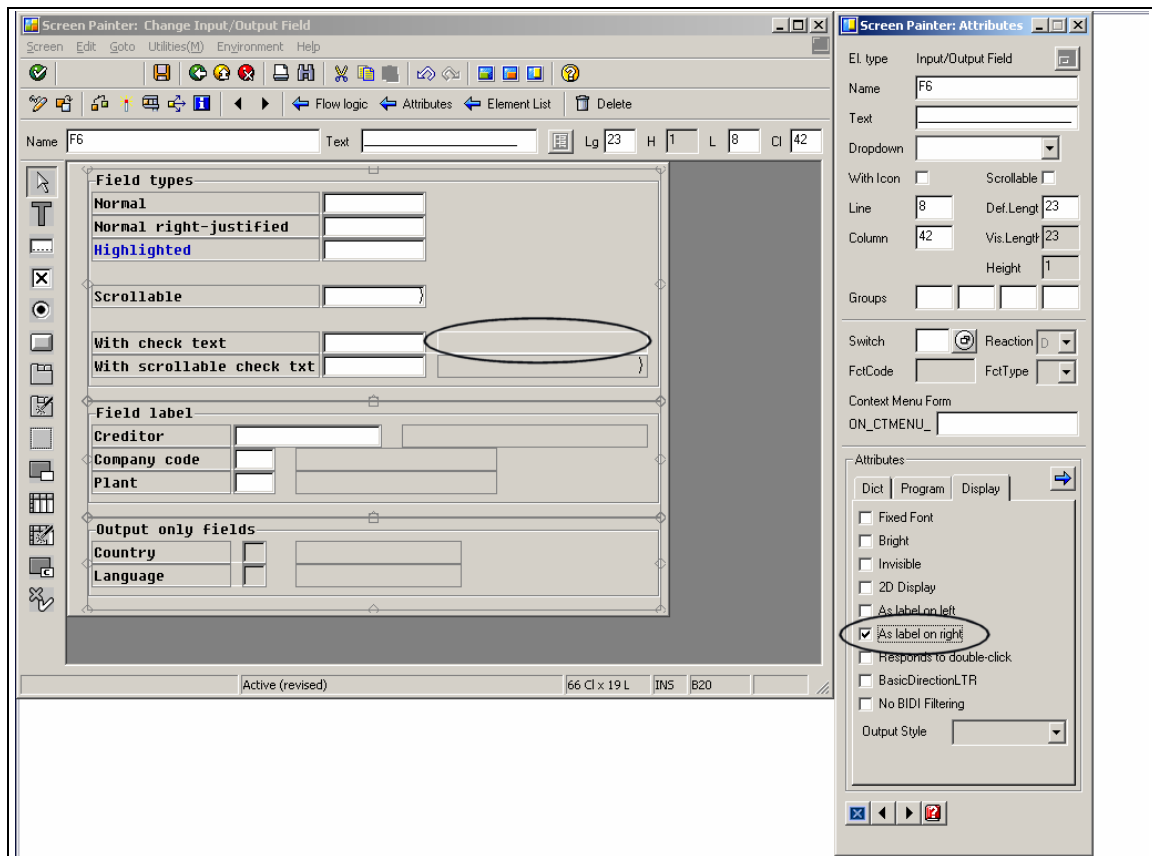


Figure 2.10: Graphical Screen Painter – “As label on right” Attribute for an Input Field’s Short Description

The following tells you how to set the attributes *As label on left* and *As label on right* for the previous example.

Go to the *Attributes* window in the *Graphical Screen Painter* and set the following indicators on the *Display* tab page:

- *As label on left* for the label of the input field
- *As label on right* for the field with the short description

**Figure 2.10** shows how the *As label on right* attribute is set for a short description.

**Note:**

The functions described in this section are available in SAP R/3 Enterprise or Basis Support Package 44 (Release 4.6C) and higher.

## 3 UI Elements and ABAP Language Elements

### 3.1 Icon

An icon is used as the graphical image of an object or a function.



Figure 3.1: Icons highlight important input fields

SAP provides you with approximately 1200 predefined icons in the SAP TYPE-POOL *ICON*. If you want to display all available icons, call transaction **ICON** or run the ABAP program *SHOWICON*.

You can use icons to provide a visual enhancement for important screen content, as shown in **Figure 3.1**. In exceptional cases, you can also use an icon as a label for an input field.

#### Dos for Icons

- Before you use an icon, read its description and make sure you are using it in the correct context. Create a meaningful tooltip for the icon, relevant to the context.
- If your chosen icon indicates information about a particular status, explain this status in its tooltip. For example, the tooltip of a red traffic light icon could be “Transmission failed” and the tooltip for a green light could be “Transmission successful”.
- Make sure that you use the same icons for the same functions or objects within a screen or application.
- Do not use too many icons on screens or in dialogs. We recommend that you use no more than 15 icons on each screen or in each dialog.

You can create your own tooltips for icons or use the tooltip provided.

If you choose to use your own tooltip, make sure it is meaningful and relevant to the context of the icon. Fill the parameter *INFO* when you call the function *ICON\_CREATE*. If you choose to use the description provided by SAP, use *INFO* = ' ' and *ADD\_STDINF* = 'X'.

```

type-pools: icon.
DATA: icon_descr(64) TYPE c.

CALL FUNCTION 'ICON_CREATE'
  EXPORTING
    name           = icon_okay
    TEXT           = ' '
    INFO           = text-010
    ADD_STDINF     = ' '
  IMPORTING
    RESULT        = icon_descr
  EXCEPTIONS
    ICON_NOT_FOUND = 1
    OUTPUTFIELD_TOO_SHORT = 2
    OTHERS        = 3.

```

Listing 3.1: Code Example for Parameter *INFO* when Calling *ICON\_CREATE*

## 3.2 Pushbutton

A pushbutton is a UI element that you can use to execute the associated function. Pushbuttons can contain both text and graphical icons.



Figure 3.2: Pushbutton with Tooltip

### Dos for Pushbuttons

- Make sure that the meaning of a pushbutton is obvious to the user.
- When you label pushbuttons, use verbs such as *Display* or *Print*, if possible. In the example above, *Display Flight Details* would be a better text.
- If the meaning of a pushbutton is not immediately obvious to the user, for example because the text is an abbreviation or single word, then create a tooltip. **Figure 3.2** shows a pushbutton called *Details* and its tooltip *Flight Details*. In the example above, *Display Flight Details* would be a better tooltip.
- Create a tooltip for a pushbutton if the pushbutton only has a graphical icon and no text.
- If you have placed both a graphical icon and a text on your pushbutton, and the graphical icon contains additional information not included in the text, provide this information in the tooltip.
- If a keyboard command exists for your pushbutton, such as [Ctrl]+[s] for a *Save* pushbutton, we recommend that you display this keyboard command in the tooltip.
- If you have specified a pushbutton as the default pushbutton, a user can choose [Enter] to operate this pushbutton even if it does not have the focus. Include this information, for example "Enter", in the tooltip of any pushbuttons you classify as default pushbuttons.
- For better usability, important pushbuttons should be given a graphical icon as well as a text. Make these important pushbuttons large enough to include both these elements. This makes these pushbuttons easier to see and access with the mouse for users without visual impairments as well.
- Arrange your pushbuttons so that the most important button is the farthest left.

Chapter 2.5 tells you how to give a pushbutton a tooltip.

## 3.3 Checkbox

You can use checkboxes to select or deselect an option. A checkbox consists of a square checkmark interaction element and a descriptive label. Make sure that the description in the label is meaningful.



Figure 3.3: Checkboxes

### Dos for Checkboxes

- In most cases, place the label on the right of the checkable interaction element. In exceptional cases, you can place the label on the left of the element, for layout reasons.
- Use meaningful labels.
- If you want to create an associated input field on the right of a checkbox, note the following rule: if the flag in the checkbox is not set, then the input field is displayed in an inactive status. Note that the concept *inactive* has a different meaning in ABAP, where it is used in the sense of an *invisible status*.

A user who uses a screen reader, and navigates through the screen using the tab key, hears the status of the field from the screen reader. The user is not led to believe that entering data is an option.

If the checkbox is flagged (selected), however, the input field becomes active. The same applies to radio buttons.

Chapter 2.6 tells you how to give a checkbox a label. Chapter 2.5 tells you how to give a checkbox a tooltip.

## 3.4 Radio Button

A radio button is a user interface element you can use to choose a single item from a list of items. A radio button consists of a round interaction element and a descriptive text.

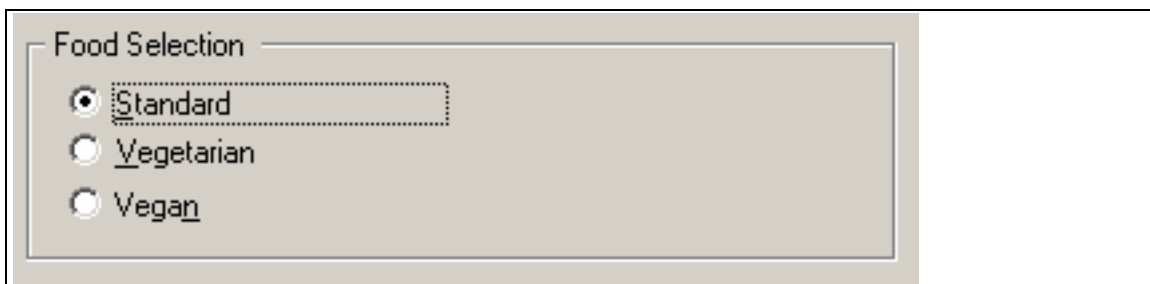


Figure 3.4: Radio Button Group

Each radio button has precisely one label, placed to the right of the round selection element. Make sure that the description in the label is meaningful.

### Dos for Radio Buttons

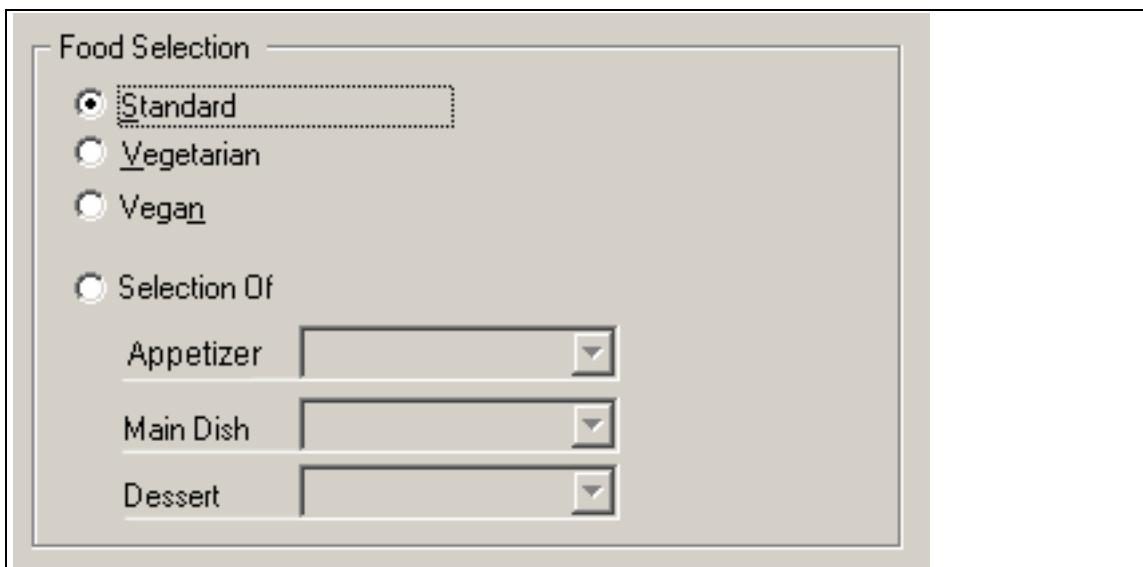
- Always use more than one radio button. Radio buttons are only used to select a single item from a list of items and cannot be used in isolation.
- Each radio button has its own label, placed to the right of the round interaction element.
- If the meaning of a radio button is not immediately obvious, for example because its text is an abbreviation, then create a tooltip.
- If you want to create selection screens with radio buttons, observe the rules described in Chapter 3.20.
- Use only one label for each radio button. Do not split the label into multiple labels. It is only technically possible for one label to be assigned to a radio button. If you split a label into two or more labels, a screen reader may not be able to recognize and read the relationship between the second (or subsequent labels) and the radio button.

- If you place an associated input field on the right of a radio button, you do not need to create a separate label for this field. In this case, associate the label of the radio button with the input field.
- Create a meaningful title for a group of radio buttons, as shown in **Figure 3.4**.

Chapter 2.6 tells you how to give a radio button a label. Chapter 2.5 tells you how to give a radio button a tooltip.

In **Figure 3.5** the radio button with the label *Standard* is selected. This makes the dropdown boxes under *Selection Of* inactive.

When using the tab key to navigate through the screen, users are not given the false impression that they can select a starter for their meals.



*Figure 3.5: Radio Button Group with Inactive Dropdown Boxes*

As soon as the user selects the *Selection Of* radio button, as shown in **Figure 3.6**, the associated dropdown boxes become active. The user can now select a starter, a main dish, and a dessert.

In this case, you must move the input focus to the selected radio button and not let it be moved to the first input field automatically by the system. The user can use the tab key to navigate to this field.

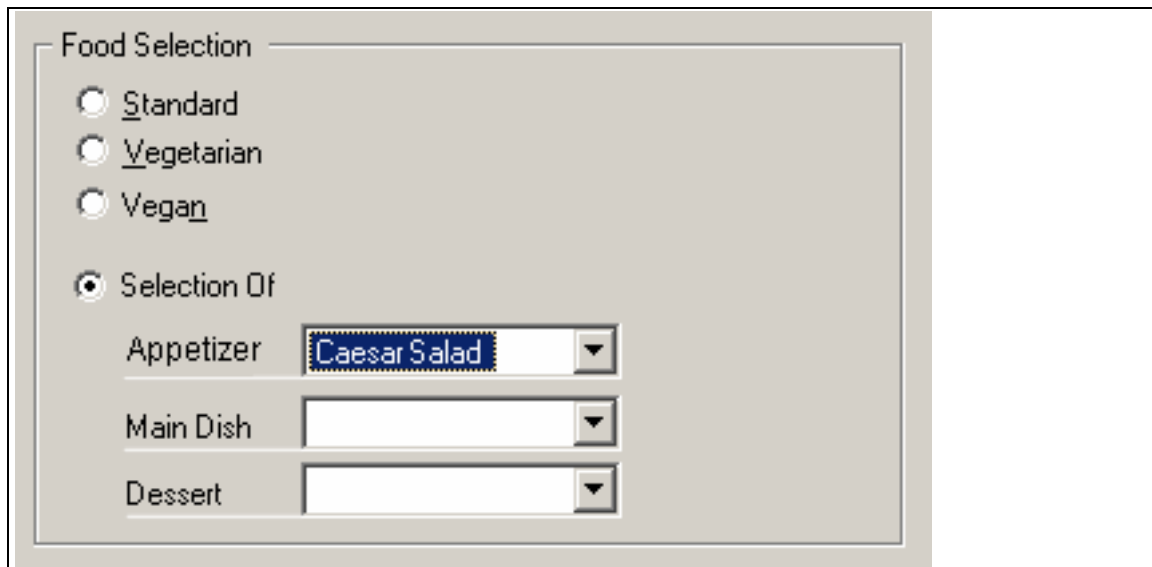


Figure 3.6: Radio Button Group with Active Dropdown Boxes

### 3.5 Input Field

You can use an input field to enter and edit text. It consists of a label and an input area.

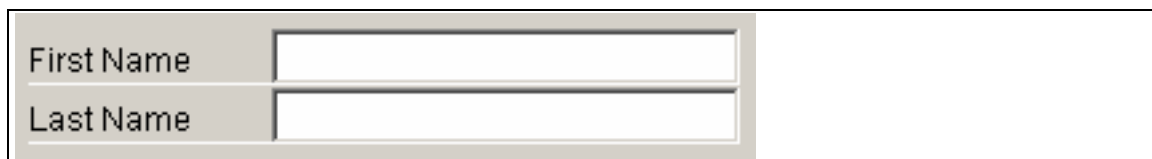


Figure 3.7: Active Input Fields

Generally speaking, the label is placed on the left of the input area. In accessibility mode, a light-colored underscore is used to indicate that the label element belongs to the input area.

#### Dos for Input Fields

- Give every input field a label. Place the label of an input field on the left of the input area. A light-colored underscore is used to indicate that the label belongs to the input area. If for layout reasons, you do not want to use a label, create a tooltip instead.
- If the input field also has a field with a short description, create a frame around this field to make the layout clearer. To do this, call the *Attributes* window with the relevant pushbutton from the application toolbar in Screen Painter. For the short description field, set the *Output Field* attribute (but not the *Output Only* attribute) on the *Program* tab page in the *Attributes* window.
- Figure 3.8 shows the short descriptions **Frankfurt** and **Los Angeles** surrounded by an appropriate frame.
- In Screen Painter, set the attributes as described in Chapter 2.6 under *As label on right for an input field with short description*. If you set these attributes, a screen reader can read the relationship between the label, the input field, and the field with the short description correctly.
- If a single input value is spread across a row of multiple input fields for usability reasons, you only need to create a label for the first of these fields. (A good example of this is when a

single credit card number is entered in several blocks of numbers.) Associate the visible label with the first input field.

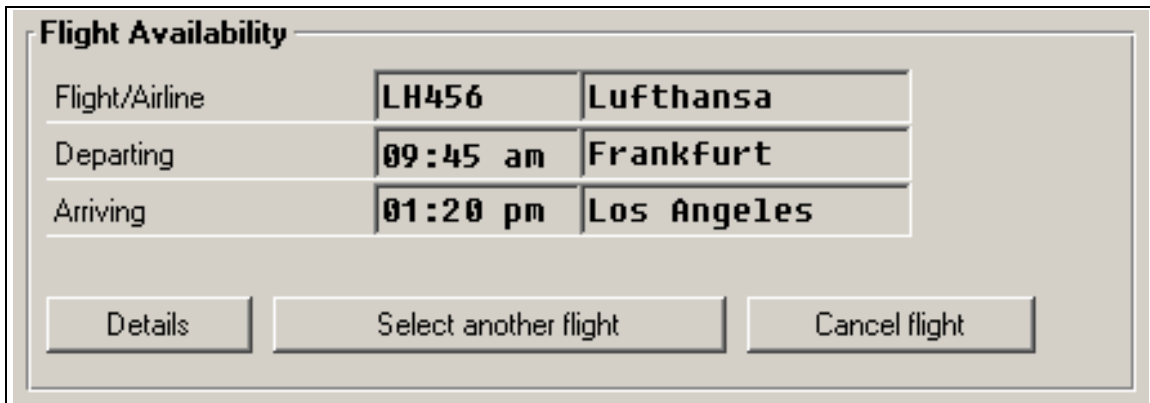


Figure 3.8: Inactive Input Fields in a Group

Chapter 2.6 tells you how to give an input field a label. Chapter 2.5 tells you how to give an input field a tooltip.

### 3.6 Combo Box and Dropdown Box

A combo box consists of an input field and a dropdown box. You can use a combo box to either enter data directly or select an item from the dropdown box. The dropdown box is not editable. You can use it to select an item from a list of items, making it easier to enter predefined data.

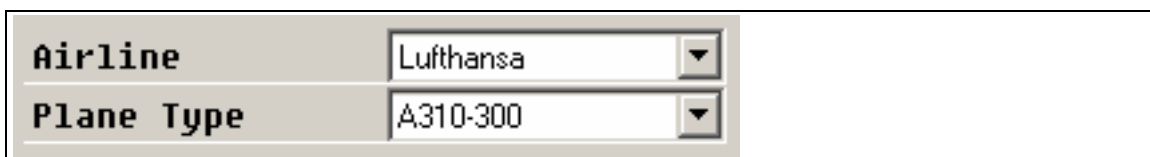


Figure 3.9: Combo Box and Dropdown Box

Create meaningful labels for combo boxes and dropdown boxes. Generally speaking, the label is placed on the left of the interaction element. These interface elements can also be used as preselection fields (see the example below).

#### Dos for Combo Boxes and Dropdown Boxes

- Create labels for combo boxes and dropdown boxes. If this is not possible for layout reasons, create a tooltip for the combo box or dropdown box. Also create a tooltip if the meaning of the label is not immediately obvious, for example because it is an abbreviation.
- Place the label on the left of the combo box or the dropdown box.

Chapter 2.6 tells you how to give a combo box or a dropdown box a label. Chapter 2.5 tells you how to give a combo box or a dropdown box a tooltip.

#### Preselection Field

A preselection field (also known as a label shuffler) is a dropdown box in which the user selects an object or element before entering a specific value for this object or element in a second field.

A leading preselection field should be prefixed with a label that applies to all subsequent fields, or each field should have a tooltip. In the example in **Figure 3.10**, the leading preselection field

(containing *Change*), is not prefixed with a label. This field therefore needs a tooltip telling the user that he or she can select a function in the first dropdown box. A possible tooltip would be: *First select a function here.*

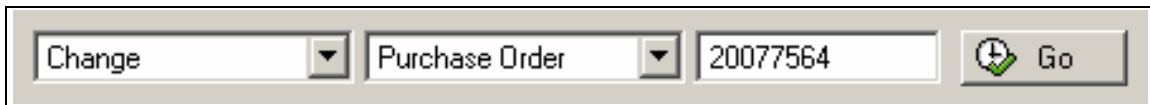


Figure 3.10: Leading Preselection Field Not Prefixed with a Label

The second preselection field (containing *Purchase Order*) needs a tooltip telling the user that he or she can select an object class here. The values in the second dropdown box change according to the value selected in the first dropdown box. This assists the user in entering correct combinations of data.

A possible tooltip for the second dropdown box would be: *Now select the type of transaction.*

The next input field contains the object instance. It needs a tooltip telling the user that the key of the object is entered here and that this field belongs to the preselection field. A possible tooltip for the input field would be: *Now enter the object key.*

### 3.7 Menu

A menu is a UI element that contains application-specific or object-specific functions. Menus can be menu bars such as the main menu bar in an SAP application, or detached menus such as context menus.

**Figure 3.11** shows the *Edit* menu in the *SAP Easy Access* screen menu bar, including the keyboard commands for each menu item. **Figure 3.12** shows the context menu for the favorites list on the *SAP Easy Access* screen.

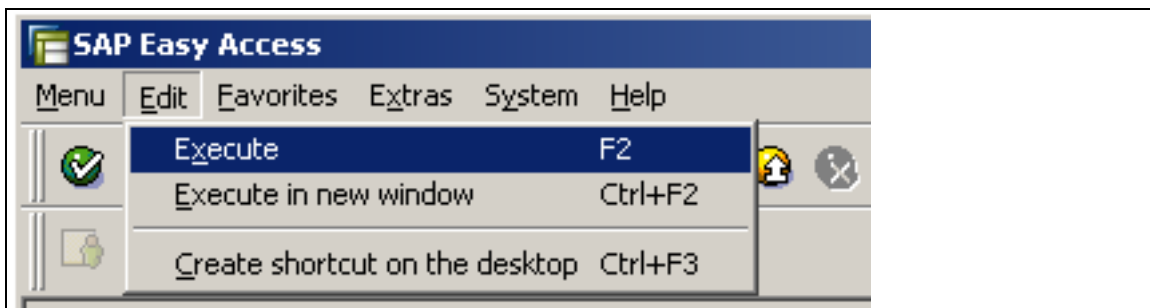


Figure 3.11: SAP Easy Access Menu Bar

When you select an executable object on the *SAP Easy Access* screen, you can activate the menu item *Execute* as follows:

- Choosing the *Execute* menu item in the *Edit* menu
 

You can access this menu item either with the mouse (click *Edit*, then click *Execute*) or by using shortcut keys on the keyboard. The shortcut key is identified by the underlined letters in the menu texts. To open the menu, you also need to use the Alt key. To activate the menu item *Execute* from the keyboard, first press the key combination [Alt]+[e] and then [x].
- With the function key [F2] on the keyboard
 

This is the quickest way to use this function, since you do not need to open the menu in a separate step.
- Using the context menu for the selected object (as described in **Figure 3.12**).

You can access the context menu as follows:

- with the secondary mouse button
- with the context menu key on the keyboard
- with the key combination [Shift]+[F10]

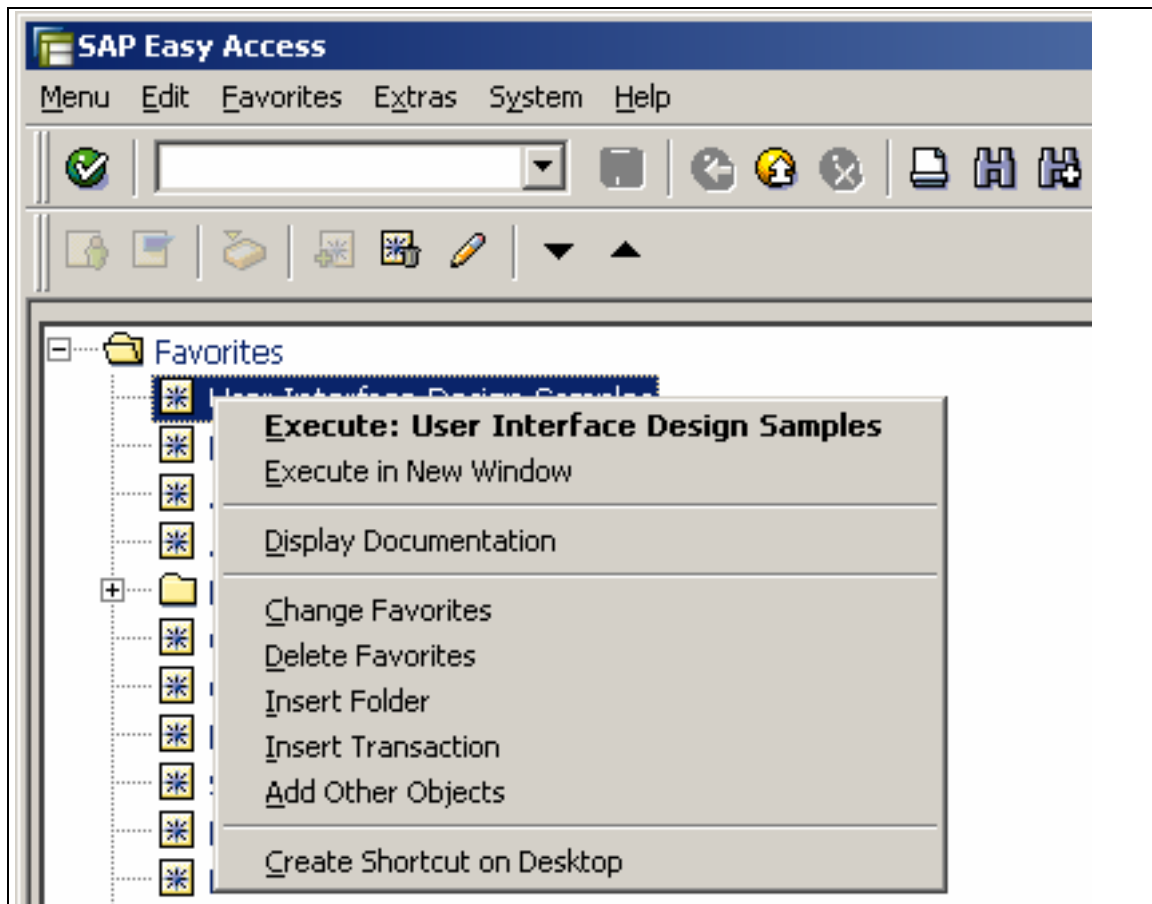


Figure 3.12: Context Menu for the Favorites List on the SAP Easy Access Screen

### Dos for Menus

- We recommend that you offer users keyboard commands, such as [Ctrl]+[s] for the *Save* function. This makes working with the system much easier for users who use the keyboard a lot or use it exclusively.
- The system displays these keyboard commands in the individual items of the main menu and in the corresponding context menu. This tells the user immediately that these keyboard combinations exist, without having to consult the documentation.
- To improve the usability of functions from the keyboard, provide the functions of the application in the context menu. Place the object-specific functions at the top of the context menu, and functions used across the application further down.
- Making the functions available in the application toolbar, as well as elsewhere, is a great help to mouse users. The text of the item in the context menu and the labelling of the pushbutton in the application toolbar should be identical. If the function is combined with a key combination, this should also be identical so that the user can identify it as the same function.

- To improve the usability of functions from the keyboard, you should make all functions in the context menu available in the main menu bar as well, under *Edit*, unless any technical limitations make this impossible.

## 3.8 Graphics, Charts, Maps, and Bitmaps

A graphic or chart is a UI element that contains graphical objects.

For any non-textual elements, you must provide an accessible text equivalent that demonstrates the same range of functions.

### Dos for Graphical Objects

- For any non-textual elements, provide an accessible text equivalent that demonstrates at least the same range of functions. Explain the interactions of the elements portrayed by a graphic or chart.
- For each graphic, chart, or bitmap, decide whether a short and precise description is enough, or whether you want to provide the information in a different way, such as a table. A short description will probably not be enough for graphics that contain a lot of information or that are highly complex.
- If your graphic is followed immediately by a text with the same content or the same meaning, you do not need to provide a tooltip for the image.
- If your graphic, chart, or map offers interaction, you must provide this in the accessible text equivalent as well, for example in a context menu item. The interaction could be the action that occurs when a user clicks an object.
- Create a solution that can be used by all users and that looks the same in accessible and non-accessible mode.
- We recommend that you design your solution so that it offers both the graphic and a text alternative, such as a table, and that the two options are placed alongside each other. You can also implement a pushbutton that allows the user to toggle between the graphic and the text alternative. Here are some more detailed solutions.

### Graphical and Textual Display on One Screen

**Figure 3.13** shows a graphic and its text alternative alongside each other on the same screen.

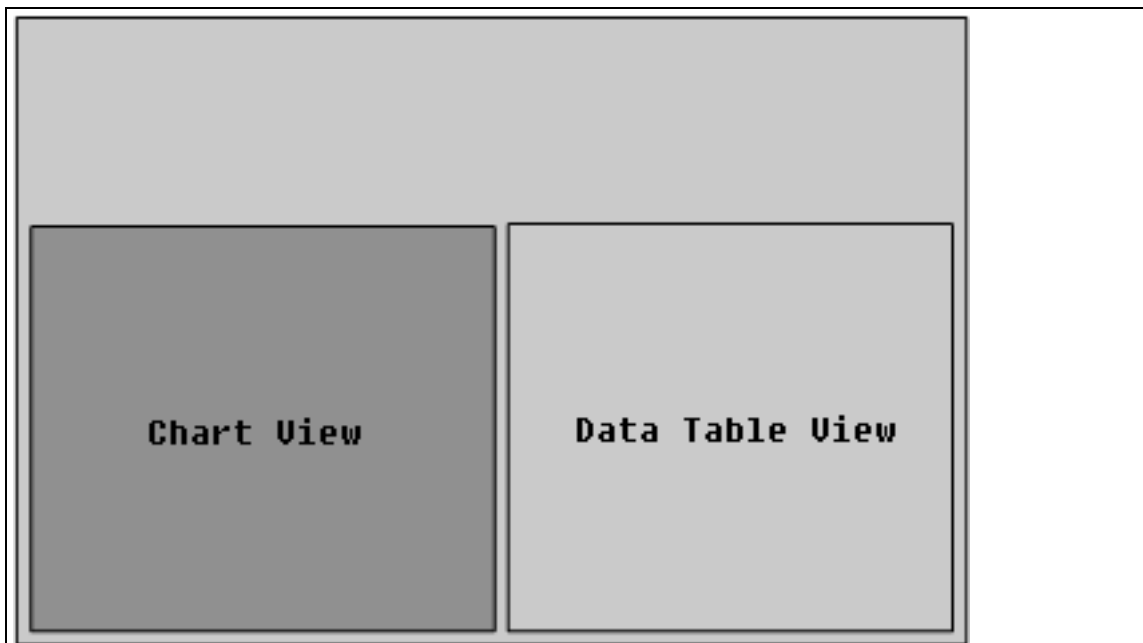


Figure 3.13: Graphical View and Alternative View Side by Side

### Graphical and Textual Display on Different Tab Pages

If the two alternatives cannot be displayed side by side for layout reasons, you can place them on separate tab pages as shown in **Figure 3.14**.

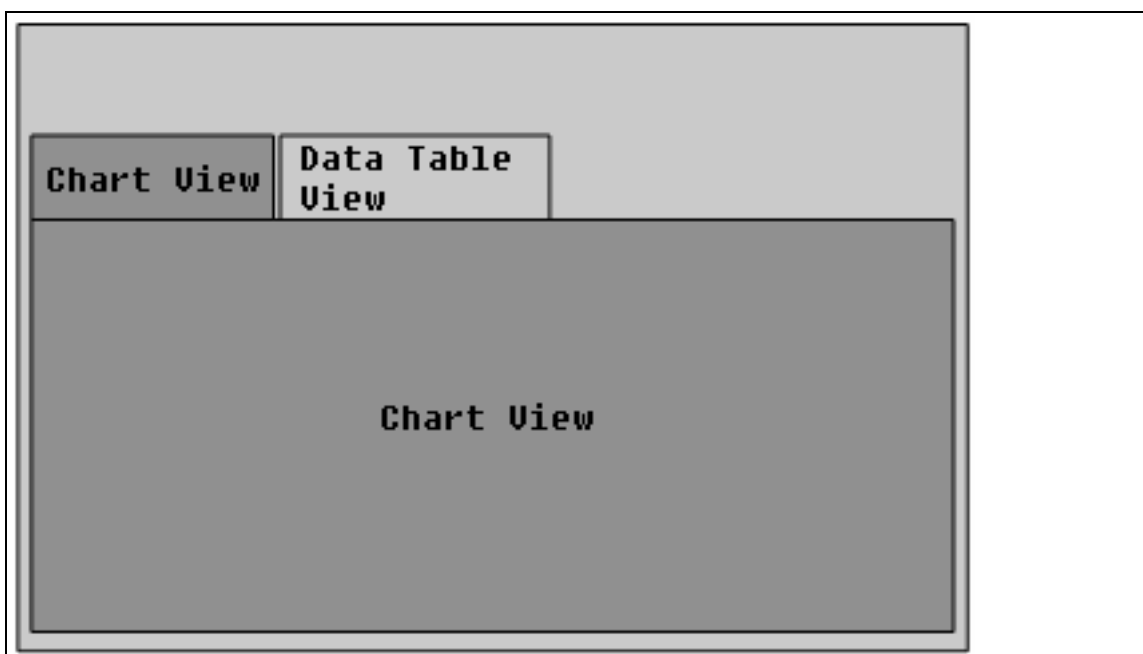


Figure 3.14: Graphical View and Alternative View on Different Tab Pages

### Toggleing Between Graphical Display and Textual Display

**Figure 3.15** shows a dialog where the user can use a pushbutton to toggle between the graphical display and its text alternative.



Figure 3.15: Pushbutton for Toggling Graphical View and Alternative View

When the user chooses the *Data Table* pushbutton to switch to the alternative display, the label of this pushbutton changes to *Chart View*. The last display chosen by the user should be saved so that he or she sees this display again when next logged on.

### 3.9 Table and Tree (Table Control, ALV Grid, ABAP List, ALV Tree, and Tree Control)

You can display structured information in data tables or trees. SAP GUI for Windows offers various components that enable you to visualize this information.

#### Tables

ALV Grid, table controls, and ABAP lists all display information in tabular form.

- **ALV Grid** (as shown in Figure 3.16)
  - You can use ALV Grid to display and edit table data. A main feature of ALV Grid is that it contains a standard set of functions. These are more comprehensive than the functions of a table control.
  - Keyboard navigation is also different in ALV Grid. ALV Grid offers better usability than table controls.
- **Table controls**
  - You can use a table control to display and edit data in tables.
- **ABAP lists**
  - ABAP lists are list views in SAP GUI for Windows that display formatted text using defined character width fonts. More information about ABAP lists is available in Chapter 3.14.

ID	No.	Flight Date	Airfare	Curr.	Plane Type	Capacity	Occupied
AA	17	14.03.2005	422	JPY	747-400	385	375
AA	17	16.03.2005	422,94	USD	747-400	385	375
AA	17	13.04.2005	422,94	USD	747-400	385	372
AA	17	11.05.2005	422,94	USD	747-400	385	374
AA	17	08.06.2005	422,94	USD	747-400	385	364
AA	17	06.07.2005	422,94	USD	747-400	385	364

Figure 3.16: ALV Grid

### Trees

A tree is a hierarchical arrangement of elements. Top level elements are called roots and the subordinate elements are called nodes. Tree controls and ALV tree controls both display data in tree-like hierarchies.

- **Tree controls**

Three different tree controls are available:

- Simple trees consist only of nodes with simple labels and no additional information
- Item trees have nodes with various types of additional information
- Column trees have columns. Each column contains the same category of information.

- **ALV tree controls** (as shown in **Figure 3.17**)

- ALV tree controls are wrappers for column trees. An ALV tree control can have a toolbar and a header.

Carry/Nr/Flight Date	Price	Currency	Pl.type	Capacity	Occupied
AA				0	0
AZ				0	0
0555				0	0
06.09.2006	185.00	EUR	A310-300	280	263
15.11.2006	185.00	EUR	A310-300	280	271
24.01.2007	185.00	EUR	A310-300	280	262
04.04.2007	185.00	EUR	A310-300	280	271
13.06.2007	185.00	EUR	A310-300	280	138
22.08.2007	185.00	EUR	A310-300	280	54
31.10.2007	185.00	EUR	A310-300	280	4
0789				0	0

Figure 3.17: ALV Tree Control

### Do's for Tables and Trees

- Use ALV Grid instead of table controls. ALV Grid includes many standard functions that are not available in table controls.
- Give every table control, ALV Grid, and ABAP list a title. (More information is available in the relevant chapters).
- Give each column a column heading.

- If you use abbreviations in column headings or data entries, the screen reader will try to read these as actual words. Abbreviations can be very difficult to understand in speech synthesis. You must accompany every abbreviation with a meaningful tooltip (in ALV Grid, table controls, ABAP lists, ALV trees, and tree controls).
- If you use icons or symbols, each of these images must also be accompanied by a tooltip. Information about how to give an icon a tooltip is available in Chapter 3.1.
- Create either a label or an icon and tooltip for each pushbutton used in ALV Grid or in table controls.
- You do not need to create a label for any input fields, radio buttons, checkboxes, combo boxes, or dropdown boxes that you use in table cells. The column header is used instead of a label.

**Figure 3.18** shows you a table where the currency column does not have a header. In this case, you should create “Currency” as the header label.

ID	No.	Flight Date	Airfare		Plane Type	Capacity	Occupied
AA	17	14.03.2005	422	JPY	747-400	385	375
AA	17	16.03.2005	422,94	USD	747-400	385	375
AA	17	13.04.2005	422,94	USD	747-400	385	372
AA	17	11.05.2005	422,94	USD	747-400	385	374
AA	17	08.06.2005	422,94	USD	747-400	385	364
AA	17	06.07.2005	422,94	USD	747-400	385	364

*Figure 3.18: Don't - Table Without Column Header*

We recommend that you give every table control, ALV Grid, and ABAP list a visible title.

A title is optional, but still recommended, in the following cases:

- If you use only one complex element in your application, such as a table, a title is optional. The window title is also the table's title in this case. However, the window title must be meaningful.
- If you use a complex element in a group box, a title is optional. The group title is also the complex element's title in this case. However, the group title must be meaningful.
- If you use tables with master and detail sections, you do not have to create a separate title for the details section that is displayed for a selected object, but we recommend that you do so for design reasons.

**Note:**

From Basis Release 700 you can use the method `set_accdescription` to create a tooltip for each front-end control derived from the class `CL_GUI_CONTROL`. For technical reasons, you cannot provide a tooltip instead of a table control title.

### 3.10 Special Recommendations for ALV Grid

Chapter 3.9 discussed our general recommendations for table controls, ALV Grid, ALV trees, and tree controls.

This chapter discusses additional recommendations that apply only to ALV Grid. All information refers to the class `CL_SALV_TABLE`.

### Creating a Title in ALV Grid

The example program `SALV_DEMO_TABLE_FORM_EVENTS` shows you how to create a title in the header area. You can run this program in ABAP Editor (transaction code SE38).

From Basis Release 700 you can create a tooltip for an ALV Grid using the method `set_accdescription`.

**Figure 3.19** shows an ALV Grid with a title.

ID	No.	Flight Date	Airfare	Curr.	Plane Type	Capacity	Occupied
AA	17	14.03.2005	422	JPY	747-400	385	375
AA	17	16.03.2005	422,94	USD	747-400	385	375
AA	17	13.04.2005	422,94	USD	747-400	385	372
AA	17	11.05.2005	422,94	USD	747-400	385	374
AA	17	08.06.2005	422,94	USD	747-400	385	364
AA	17	06.07.2005	422,94	USD	747-400	385	364

*Figure 3.19: ALV Grid with Title*

### Using Semantic Colors in ALV Grid

The following code example shows you how to accompany a semantic color with a text. This text can then be read by screen readers and displayed on refreshable braille displays. It does not, however, appear in a tooltip on the screen.

```

Form SET_COLOR_TOOLTIPS
  data: settings type
        ref to CL_SALV_FUNCTIONAL_SETTINGS,
        tooltips type
        ref to CL_SALV_TOOLTIPS.
  settings = gr_table->get_functional_settings( ).
  tooltips = settings->get_tooltips( ).
  try.
    tooltips->add_tooltip(
      type = CL_SALV_TOOLTIP=>C_TYPE_COLOR
      value = '6'
      tooltip = 'Delayed delivery' ).
    tooltips->add_tooltip(
      type = CL_SALV_TOOLTIP=>C_TYPE_COLOR
      value = '5'
      tooltip = 'Delivered in time' ).
  catch cx_salv_existing.
endtry.
Endform.

```

*Listing 3.2: Code Example for Providing Text for Semantic Colors*

#### Note:

The functions described in this section are available in Basis Release 640 and higher.

## 3.11 Special Recommendations for Table Control

Chapter 3.9 discussed our general recommendations for table controls, ALV Grid, ALV trees, and tree controls.

This chapter discusses additional recommendations that apply only to table controls.

### Creating Table Titles for Table Controls

We recommend that you create a title for every table. To create a table title in a table control, proceed as follows in Screen Painter:

1. If you have created a table and the focus is in the table, call the *Attributes* window with the relevant pushbutton in the application toolbar.
2. Select the *with Title* flag in the *Attributes* window.
3. Create a new field called "TITLE" in the table title area. You can create either a text field or an input/output field.
4. If you want to use an input/output field as a title, activate the *Output Only* attribute in the *Attributes* window of the field. You can assign a field from ABAP Dictionary to this field or use an application program to fill the title field at runtime.
5. If you use a text field, create the table title in the *Text* field in the *Attributes* window of the field.

#### Note:

For technical reasons, you cannot provide a tooltip instead of a table control title.

### Creating Tooltips for Column Headers in Table Controls

Column headers that contain abbreviations are one example of texts that require tooltips.

To create a column header with a tooltip in a table control, activate the *Attributes* window for the field that contains the column header. Enter the text for the tooltip in the *Tooltip* field.

### No Semantic Colors in Table Controls

You cannot use semantic colors in a table control. You can, however, flag the content of a column as *highlighted*.

## 3.12 Special Recommendations for Tree Controls

You can use a tree control to create a single-column or multiple column tree display. Column 1 contains the actual hierarchy.

Chapter 3.9 discussed our general recommendations for table controls, ALV Grid, ALV trees, and tree controls.

This chapter discusses additional recommendations that apply only to tree controls.

If you use multiple columns, use `ITEM_SELECTION = "X"` to activate element-by-element navigation instead of navigation by row. This enables the screen reader to read each tree node and tree element individually.

## Creating a Title in a Tree Control

Create a meaningful label for the root node. If the tree takes up the whole screen, you can use a meaningful window title instead of a label for the root node.

### Note:

From Basis Release 700 you can create a tooltip for a tree control using the method `set_accdescription`.

## Using Colors in Tree Controls

The following list shows you the color type values of the interface `if_salv_c_tree_style` and also which values you can use to set colors:

- DEFAULT
- INTENSIFIED
- INACTIVE
- INTENSIFIED\_CRITICAL
- EMPHASIZED\_NEGATIVE
- EMPHASIZED\_POSITIVE
- EMPHASIZED
- EMPHASIZED\_A
- EMPHASIZED\_B
- EMPHASIZED\_C

The following code example shows you how to give a node the style `inactive`:

```
node-node_key = 'Root'.
node-isfolder = 'X'.
node-text = 'This is my node'.
node-style = cl_gui_simple_tree=>style_inactive.
```

### Listing 3.3: Giving a Node the Style “inactive” (Part I)

Use an additional description text to specify the style being used:

```
call method mytree->ADD_STYLE_INFO
exporting
    style = cl_gui_simple_tree=>style_inactive
    info = 'Inactive'.
```

### Listing 3.4: Giving a Node the Style “inactive” (Part II)

### Note:

The functions described in this section are available in Basis Release 700 and higher.

### 3.13 Special Recommendations for ALV Tree Control

An ALV tree control is a wrapper for a column tree (a type of tree control).

Chapter 3.9 discussed our general recommendations for table controls, ALV Grid, ALV trees, and tree controls.

This chapter discusses additional recommendations that apply only to ALV tree controls.

If you use a column tree, use `ITEM_SELECTION = "X"`, to activate element-by-element navigation instead of navigation by row. This enables the screen reader to read each tree node and tree element individually.

You can use icons in an ALV tree control, but no symbols.

#### Creating a Title in an ALV Tree Control

The following code example shows you how to create a title for an ALV tree control. `gr_tree` is a reference to an instance of an ALV tree control:

```
DATA: settings TYPE REF TO cl_salv_tree_settings.
DATA: title TYPE salv_de_tree_text.

settings = gr_tree->get_tree_settings( ).
title = 'mein Titel'.
settings->set_header( title ).
```

*Listing 3.5: Creating a Title for an ALV Tree Control*

#### Note:

The functions described in this section are available in Basis Release 640 and higher. From Basis Release 700 you can create a tooltip for an ALV tree control using the method `set_accdescription`.

#### Using Semantic Colors in an ALV Tree Control

The following list shows you the color type values of the interface `if_salv_c_tree_style` and also which values you can use to set colors:

- DEFAULT
- INTENSIFIED
- INACTIVE
- INTENSIFIED\_CRITICAL
- EMPHASIZED\_NEGATIVE
- EMPHASIZED\_POSITIVE
- EMPHASIZED
- EMPHASIZED\_A
- EMPHASIZED\_B
- EMPHASIZED\_C

The following code example shows you how to create a text in an ALV tree control that can then be read by a screen reader:

```

method SET_TREE_COLOR_TOOLTIPS.
  data: settings type ref to
          CL_SALV_TREE_SETTINGS,
        tooltips type ref to CL_SALV_TOOLTIPS,
        value TYPE lvc_value.
  settings = gr_tree->get_tree_settings( ).
  tooltips = settings->get_tooltips( ).
  value = if_salv_c_tree_style=>emphasized_negative.
  try.
    tooltips->add_tooltip(
      type      = CL_SALV_TOOLTIP=>c_type_TREE_STYLE
      value     = value
      tooltip   = 'Delayed delivery' ).
  catch cx_salv_existing.
  endtry.
  value = if_salv_c_tree_style=>emphasized_positive.
  try.
    tooltips->add_tooltip(
      type      = CL_SALV_TOOLTIP=>c_type_TREE_STYLE
      value     = value
      tooltip   = 'Delivered in time' ).
  catch cx_salv_existing.
  endtry.
Endmethod.

```

*Listing 3.6: Creating a Text for a Color in an ALV Tree Control*

**Note:**

The functions described in this section are available in Basis Release 700 and higher.

### 3.14 ABAP List

ABAP lists are list views in SAP GUI for Windows that display formatted text using defined character width fonts. The texts displayed in an ABAP list are generally unstructured texts, table content, or hierarchical data.

You can create a list display with the ABAP command WRITE.

**Dos for ABAP Lists**

- Use ALV Grid instead of tabular ABAP lists, if possible. Also migrate any existing ABAP lists. Keep the ABAP list alongside the new solution for printing purposes if this is a customer requirement.

**Note:**

When migrating ABAP lists to ALV Grid, read the information about batch input in SAP Note 311440.

If you need to display the ABAP list, for example for printing purposes, switch the “native” ABAP list programming to ALV Classic (use function module REUSE\_ALV\_LIST\_DISPLAY).

## 3.15 Step Loop

A step loop is the multiple display of a group of elements on a single screen in a table-like form.

Avoid using step loops. Use ALV Grid (Chapter 3.9 and 3.10) or table controls (Chapter 3.9 and 3.11).

### Dos for Step Loops

- Use ALV Grid or a table control instead of step loops.
- If you must use a step loop, create a column header for every column. There are two different ways of creating a column header:
- Use the label of the column's data field in ABAP Directory as the column header, prefixed by an asterisk (\*).
- Alternatively, you can create column headers by using the "Text Field" attribute of the column's data field to refer to an existing text field or output field.
- There can be no more than one empty row between the column headers and the first row of data.
- Make sure that the space created by blanks is the same in the column headers and the column content so that they are aligned correctly.

## 3.16 HTML Control

An HTML control displays HTML or XML content. HTML controls are one of the UI elements that you should avoid using in SAP GUI for Windows. If you must use HTML controls, you need to ensure that accessible content is displayed in the control.

A user who is working in SAP GUI for Windows and then encounters an HTML control must switch to the different keyboard access to UI elements implemented in Microsoft Internet Explorer.

### Dos for HTML Controls

- Avoid using HTML controls.
- If you must use HTML controls, you need to ensure that accessible content is displayed in the control.
- Code examples are available in "**HTML Techniques for Web Content Accessibility Guidelines 1.0**" (at <http://www.w3.org/TR/WCAG10-HTML-TECHS/>).
- If you link from an HTML document to other HTML documents, make sure that the linked HTML documents are accessible.
- Use the method *set\_accdescription* to create a tooltip for the HTML control. This function is available from Basis Release 700.

## 3.17 Calendar Control

Calendar control (as shown in **Figure 3.20**) is an input help that assists you when you enter dates. It also gives you a good overview of your appointments.

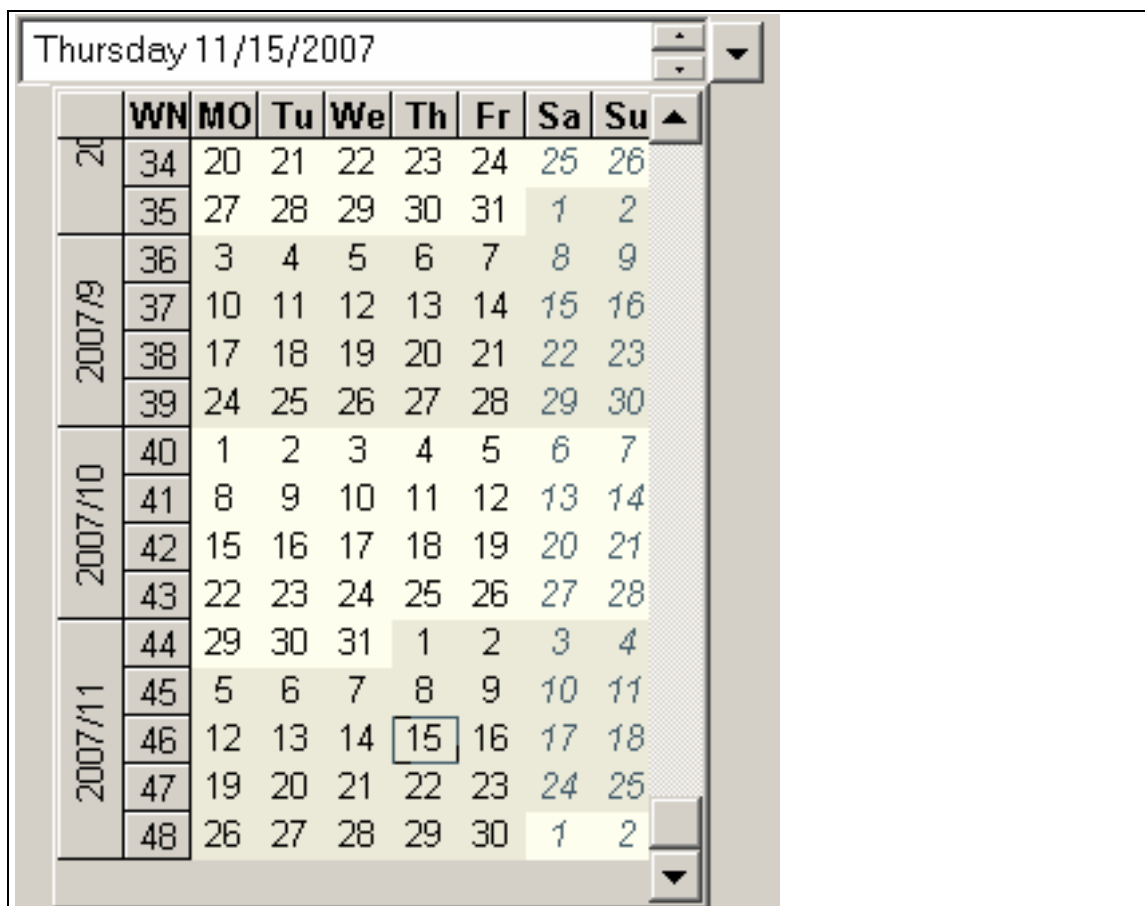


Figure 3.20: Calendar Control

### Using Colors in Calendar Controls

You can choose to display a day in a specific color in calendar controls. You can specify the date, color type of the day, and a description in the table `CNCA_DAY_INFO`. You can also create a color information text in an additional row in this table. You must leave the date empty in this row.

The following table lists the permitted color types and the color assigned to them:

Color Type ID	Meaning
Cnca_color_null	Displays the day in the default background color of the calendar control.
Cnca_color_1 Cnca_color_2 ... Cnca_color_8	Displays the day in the specified color type (1 through 8).

Table 3.1: Color Types in Calendar Control

The following code example shows you how you can convey the semantic meaning of colors in a calendar control.

```

TYPE-POOLS CNCA.
DATA dayinfo TYPE cnca_day_info,
      dayinfos TYPE cnca_itab_day_info.
DATA mydatenavigator TYPE REF TO CL_GUI_CALENDAR.
* Setting color information for Color 1
dayinfo-text = 'Delivery expected'.
dayinfo-color = cnca_color_1.
APPEND dayinfo TO dayinfos.

* Set two days to Color 1
dayinfo-date = sy-datum.
dayinfo-color = cnca_color_1.
APPEND dayinfo TO dayinfos.

dayinfo-date = sy-datum + 1.
dayinfo-color = cnca_color_1.
APPEND dayinfo TO dayinfos.

CALL METHOD mydatenavigator->SET_DAY_INFO
      EXPORTING
          day_info = dayinfos.

```

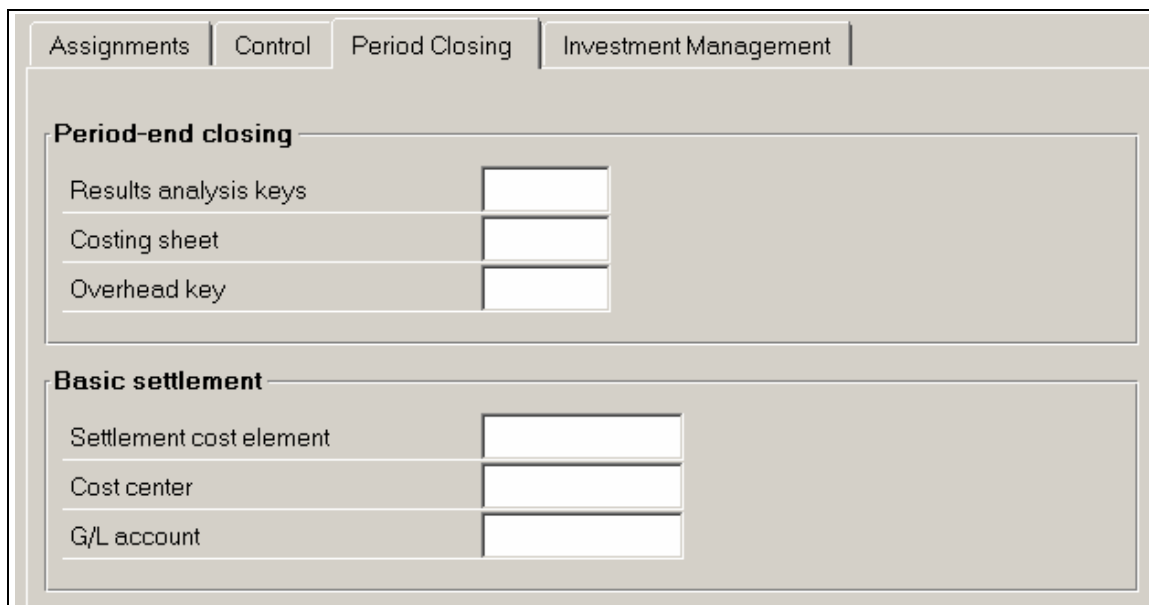
*Listing 3.7: Code Example for Calendar Control*

**Note:**

This function is available from Basis Release 700.

### 3.18 Tabstrip

A tabstrip is a grouping element that can display different application elements on a single screen (as shown in **Figure 3.21**). A tabstrip consists of multiple pages and each page has a title. The title of a tab page is on the tab index. This index must contain a meaningful and unique name for each of the tab pages.



*Figure 3.21: Tabstrip with Four Tab Pages*

### Dos for Tabstrips

- Create a meaningful title for each tab page, such as: *Addresses, Invoices, Logistics*. Avoid using tab page titles such as *Miscellaneous, More, Details, or Extras*.
- Use at least two tab pages in a tabstrip. Do not use too many tab pages. This keeps your application uncluttered and easy to use. We recommend a minimum of two and a maximum of seven.
- Arrange your tab pages so that the most important page is the farthest left.
- Do not include any mandatory input fields on tab pages. Mandatory fields should always be placed above the tab pages.
- Do not use abbreviations in the titles if possible. Screen readers will try to read abbreviations as actual words, which can be very difficult to understand in speech synthesis. If you do need to use abbreviations or if the meaning of a title is not immediately obvious, create a tooltip.

### Don'ts for Tabstrips

- Do not use tabstrips within groups (as discussed in Chapter 3.19) since they already group elements.
- Do not nest tabstrips within each other. This keeps your application easy to use. Instead, we recommend that you use filter mechanisms, such as preselection fields (as discussed in Chapter 3.6), or redesign your screen or application.
- Do not use tabstrips if the content of one tab page depends on the information on a preceding tab page. This could be the case if mandatory input fields are used. Split the user actions into steps instead. If there is a small number of steps (two to seven), we recommend that you implement a wizard. If more steps are involved, place the data that governs the rest of the data above the tabstrip.

### Creating a Tooltip for a Tabstrip

To create a tooltip for a tabstrip, move the focus to the tab index in Graphical Screen Painter and choose the relevant pushbutton in the application toolbar to activate the Attributes window. Enter the tooltip text in the *Tooltip* field. Leave the *Icon* field empty.

## 3.19 Group

You can use groups to emphasize the relationships between elements that have the same context. A group always contains multiple UI elements and a group title.

**Flight Availability**

Flight/Airline	LH456	Lufthansa
Departing	09:45 am	Frankfurt
Arriving	01:20 pm	Los Angeles

Figure 3.22: Group

### Dos for Groups

- Give each group a title. For technical reasons, you cannot create tooltips for groups.
- Only create a group if it contains more than one element.
- Place semantically connected elements on a screen in appropriate groups. Do not locate these elements in unconnected parts of the screen.

### Don'ts for Groups

- Do not use tabstrips within groups (as discussed in Chapter 3.18) since they already group elements.

## 3.20 Selection Screen

A selection screen is a special type of screen that can be defined in the global declaration part of executable programs, function groups, and module pools. You define it with the statements `SELECTION-SCREEN`, `PARAMETERS`, or `SELECT-OPTIONS`. You cannot use Screen Painter.

Document Types			
<input checked="" type="radio"/>	Documents		
<input type="radio"/>	Recurring Entry Documents		
<input type="radio"/>	Sample Documents		
<input type="radio"/>	Parked Documents		
<input type="radio"/>	Documents with Parking		
<input type="radio"/>	External Documents		


General Selections			
Company Code	<input type="text"/>	to	<input type="text"/> 
Document Number	<input type="text"/>	to	<input type="text"/>
Fiscal Year	<input type="text"/>	to	<input type="text"/>
Receiving Company Code	<input type="text"/>	to	<input type="text"/>
Changed On	<input type="text" value="10/09/2006"/>	to	<input type="text"/>
Changed By	<input type="text"/>	to	<input type="text"/>

Figure 3.23: Selection Screen

### Dos for Selection Screens

- If an input field is located next to a selected radio button and its label, you must show this field as being ready for input, as shown in **Figure 3.24**.
  - If an input field is located next to a non-selected radio button and its label, you must show this field as inactive. Chapter 3.4 discusses why this is necessary.
- Do not place more than one element on the right of the label. If more than one element is placed after the label, then screen readers cannot read the following relationships correctly:
  - Relationship between a checkbox and a label
  - Relationship between a radio button and a label
  - Relationship between an input field and a label

### Example: selection screen with input field next to radio button

**Figure 3.24** shows a selection screen on which an additional element is placed on the right of each checkbox's or radio button's label. The code example below is designed so that the relationship between the checkbox and label and the relationship with the associated input field can be read correctly by a screen reader.

The code example also enables the corresponding relationship between the radio buttons, labels, and input fields to be read correctly.

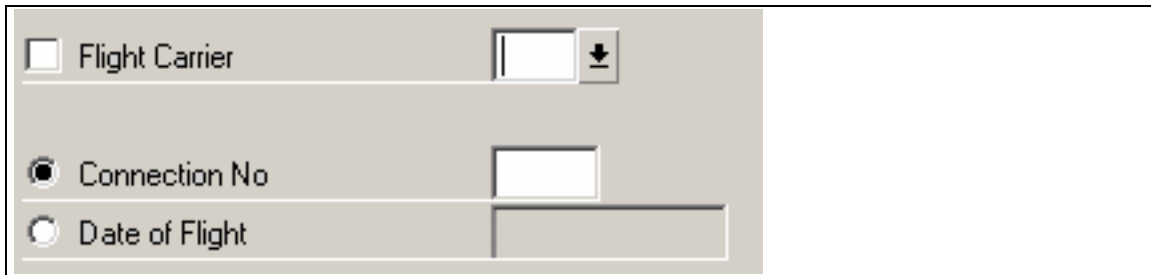


Figure 3.24: Selection Screen with Input Field and Checkbox

You must create the language-dependent selection texts in *ABAP Workbench*. More information is available in the SAP documentation under the key words `SELECT-OPTIONS`, `PARAMETERS`, and `SELECTION-SCREEN`.

```
selection-screen begin of line.
parameter testcb type c as checkbox.
selection-screen comment (20) for field test.
selection-screen position 23.
parameter test type sflight-carrid.
selection-screen end of line.
```

```
selection-screen skip.
```

```
selection-screen begin of line.
parameter testA type c radiobutton group A1.
selection-screen comment (20) for field test1.
selection-screen position 23.
parameter test1 type sflight-connid.
selection-screen end of line.
```

```
selection-screen begin of line.
parameter testB type c radiobutton group A1.
selection-screen comment (20) for field test2.
selection-screen position 23.
parameter test2 type sflight-fldate.
selection-screen end of line.
```

Listing 3.8: Selection Screen

### Input Field with Status “Not Ready for Input”

The next code example shows you how to assign the status *Not ready for input* to input fields.

```
tables usr01.
select-options s_test for usr01-bname modif id S00.
at selection-screen output.
loop at screen.
  if screen-group1 = 'S00'.
    screen-active = 0.
    modify screen.
  endif.
endloop.
```

Listing 3.9: Giving Input Fields the Status “Not ready for input”



A SELECT-OPTION or a parameter on a selection screen can consist of multiple fields, which is why we recommend that you use SCREEN-GROUP to group these fields. This makes it easier to hide an entire SELECT-OPTION. Otherwise, you would have to use the names of each of the fields (SCREEN-NAME): you might not know the names of the labels that were generated.

## 4 How to Integrate Third Party Products

### 4.1 Embedded Components

Embedded components always run as a part of a screen inside an SAP application window. Embedded components can be, for example, ActiveX Controls (OLE Controls).

The component must therefore conform to the following standards:

- Tab In  
The focus moves from the SAP application window to the component and the user can navigate through the objects/items inside the component.
- Tab Out  
The component loses the keyboard focus and should save the last focus position for when the user goes back to the component later. (This is highly recommended to avoid extra navigation overhead). The focus must return to the application window.
- Content must be navigable (for example, by tab keystroke or arrow keys).
- Focus must be clearly visualized.

Furthermore, the component must comply with the standards of US Section 508 and the standards of WCAG 1.0 (Priority 1 and 2).

### 4.2 Launching External Programs

Launched programs usually appear in their own application window, on top of the started application window. The keyboard focus should automatically move to the new program window.

After the external program is closed, the focus should return to the application window that started the external program.

Furthermore, the component must comply with the standards of US Section 508 and the standards of WCAG 1.0 (Priority 1 and 2).

## 5 Testing ABAP Dynpro

### 5.1 Prerequisites for the Test Environment

For information about the prerequisites for the test environment, see the newest version of the document "Front-End Requirements and Infrastructure for Accessibility" (settings in SAP GUI for Windows, information about screen reader and SAP GUI Screen Reader Extension). You can access this document at [SAP Design Guild Edition Accessibility](#) under "Resources" → "Frontend Requirements and Infrastructure for Accessibility".

### 5.2 General Rules for Testing ABAP Applications

This chapter describes the general test criteria that apply to the UI elements in ABAP applications listed in Chapter 5.3, if not stated otherwise for a particular UI element. To keep the checklists as clear and concise as possible for the reader, we have not repeated these general rules for each UI element.

- **Navigation**
  - Each UI element must be accessible using the tab key, with [Tab] used for forward navigation and [Shift] + [Tab] used for backward navigation.
  - You can use the key combination [Ctrl] + [Tab] to skip whole groups of elements (such as toolbars). You can use the key combination [Shift] + [Ctrl] + [Tab] for backward navigation through groups.
- **Abbreviations**
  - If you have to use abbreviations, you must provide the full text in a tooltip. Abbreviations are difficult to understand when read by a screen reader.
- **Screen Reader**
  - For simple UI elements, a screen reader can read out the label, object type, state, and value (if any). If a tooltip exists, a screen reader reads this out as well.
  - You can test UI elements with a screen reader. Extra aspects to watch out for in certain UI elements are described in Chapter 5.3 below.

**Note:**

Note that certain functions only become available in a particular Basis release, as described in Chapter 2 and Chapter 3. Further information on each of the UI elements is also available in Chapter 3.

### 5.3 Testing UI Elements

This chapter describes the special test criteria that apply to the individual UI elements. For each of these UI elements we have listed the associated requirements and the methods you can use to test them.

#### 5.3.1 Icon

**Requirement**

An icon (as shown in Figure 5.1) must have a meaningful and context-dependent tooltip.

## Test

- Use the keyboard to navigate to the icon that you want to test.
- Check that a tooltip exists and that it meets the requirements.

You can use the mouse to display a tooltip by moving the cursor over the icon. You can also use the keyboard to display the tooltip by moving the keyboard focus to the icon and pressing the key combination [Ctrl] + [q].



Figure 5.1: Icons highlight important input fields

## 5.3.2 Pushbutton

### Requirements

- A pushbutton (as shown in **Figure 5.2**) must have a visible label.
- If a pushbutton only has a graphical symbol and no label, it must have a tooltip.
- If a pushbutton has both a graphical symbol and a label, and the graphical symbol contains additional information not included in the label, this information must be provided in the tooltip.

### Test

- Use the keyboard to navigate to the pushbutton that you want to test.
- Check that the pushbutton contains a visible label. Also check that a tooltip exists, if necessary, and that it meets the requirements.
- You can use the mouse to display a tooltip by moving the cursor over the pushbutton. You can also use the keyboard to display the tooltip by moving the keyboard focus to the pushbutton and pressing the key combination [Ctrl] + [q].

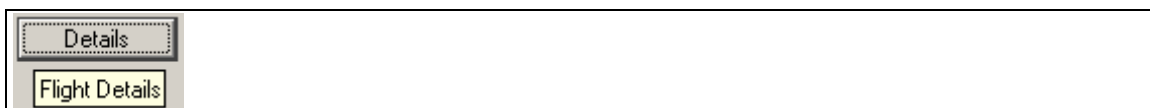


Figure 5.2: Pushbutton with Tooltip

## 5.3.3 Checkbox

### Requirements

- A checkbox (as shown in **Figure 5.3**) must have a visible label placed, in most cases, on the right of the interaction element.
- The label and the interaction element must be associated with each other.

### Test

- Use the keyboard to navigate to the checkbox that you want to test.
- Check that the checkbox has a visible label and that the focus moves to both the label and the interaction element.

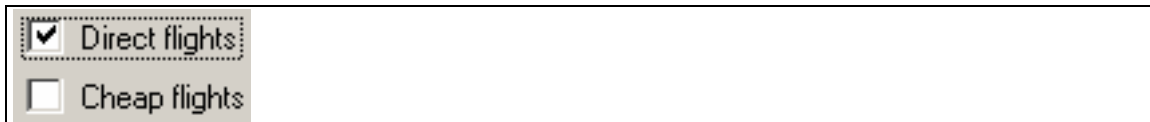


Figure 5.3: Checkboxes

### 5.3.4 Radio Button

#### Requirements

- Associated radio buttons must be in a group with a visible title (as shown in **Figure 5.4**).
- A radio button must have a visible label placed, in most cases, on the right of the round interaction element.
- The visible label and the round interaction element must be associated with each other.
- If an associated input field is placed on the right of a radio button, this field does not need a separate label. In this case, the label of the radio button must be associated with the input field. This input field must be ready for input only when the radio button is selected. More information is available in Chapter 3.4.

#### Test

- Use the keyboard to navigate to the radio button group and check that it has a visible title.
- You can use the tab key to navigate to the selected radio button in the radio button group. You can use the key combination [Ctrl] + [Arrow Up] or [Ctrl] + [Arrow Down] to navigate to the other radio buttons without selecting them.
- Check that the radio buttons have visible labels and that the focus moves to both the label and the round interaction element when you navigate.
- You can use the mouse to display a tooltip (if available) by moving the cursor over the radio button. You can also use the keyboard to display the tooltip by moving the keyboard focus to the radio button and pressing the key combination [Ctrl] + [q].
- If a radio button has an input field, check that the input field is displayed in accordance with the requirement above.

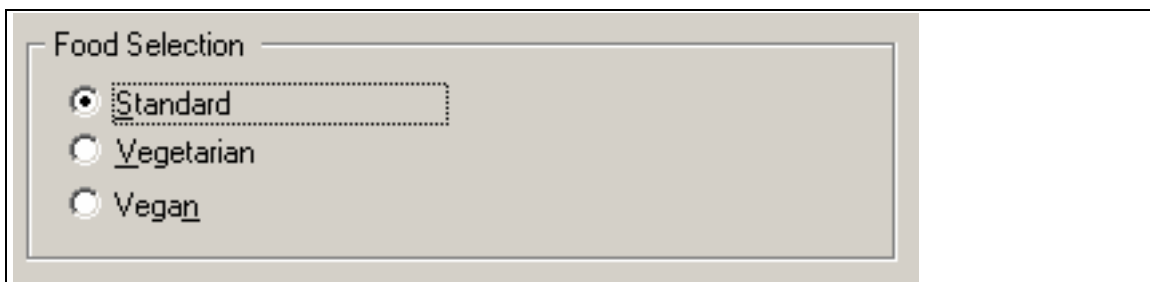


Figure 5.4: Radio Button Group

### 5.3.5 Input Field

#### Requirements

- An input field (as shown in **Figure 5.5**) must have a visible label placed, in most cases, on the left of the input area.
- The visible label and the input area must be associated with each other. A light-colored underscore is used to indicate that the label belongs to the input area.

- If a field with a short description is also associated with the input field, this short description must also be associated with the input area.
- If two or more input fields are arranged side by side, only the first input area needs a label. The visible label and the first input area must be associated with each other.

#### Test

- Use the keyboard to navigate to the input field that you want to test.
- The visible label and the input area must be associated with each other. This must be indicated by a light-colored underscore. Only the input area of the input field must be included in the tab chain. The label of the input field must not be included.
- If a field with a short description is also associated with the input field, this description is *not* included in the tab chain. Only the input area of the input field must be included in the tab chain. When the focus moves to the input area, the screen reader must read out the label, the input area, and the short description.

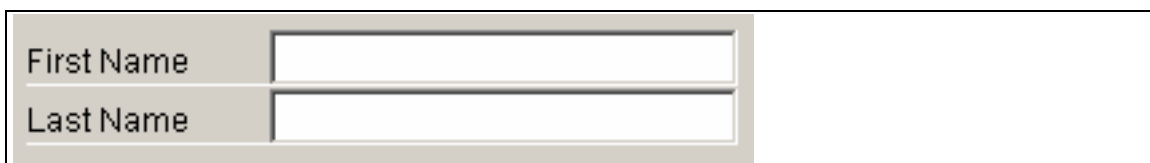


Figure 5.5: Input Fields Ready for Input

### 5.3.6 Combo Box and Dropdown Box

#### Requirements

- A combo box or dropdown box (as shown in **Figure 5.6**) must have a visible label placed, in most cases, on the left of the interaction element.
- The label and the interaction element must be associated with each other. A light-colored underscore is used to indicate that the label belongs to the interaction element.
- A leading preselection field (or label shuffler) must have a visible label placed on the left of the interaction element. More information about preselection fields is available in xxxChapter 3.6.
- The visible label and the interaction element of the preselection field must be associated with each other. A light-colored underscore is used to indicate that the label belongs to the interaction element. If no visible label exists, each element must have a tooltip.

#### Test

- Use the keyboard to navigate to the combo box or dropdown box that you want to test.
- The visible label and the interaction element must be associated with each other and this must be indicated by a light-colored underscore. Only the interaction area must be included in the tab chain and not the label of the combo box or dropdown box.
- Check that all elements of the preselection fields meet the requirements above.

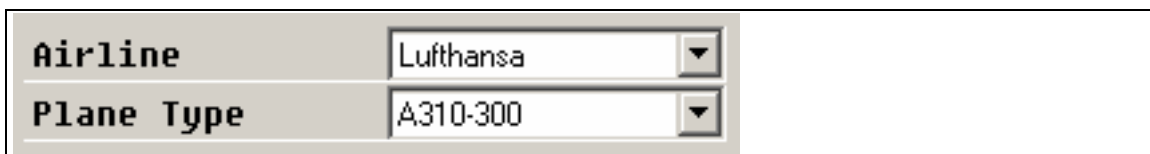


Figure 5.6: Combo Box and Dropdown Box

### 5.3.7 Graphics, Charts, Maps, and Bitmaps

#### Requirements

- Each non-textual element must have an accessible text equivalent.
- The form of the text equivalent must be appropriate to the complexity of the non-textual element. A short description in a tooltip is enough for simple elements, but will most likely not be enough for graphics that contain a lot of information or that are highly complex.
- Any functions in non-textual elements must also be available as an accessible alternative.

#### Test

- Check that the requirements above are met for all non-textual elements.
- You can use the mouse to display a tooltip (if available) by moving the cursor over the graphic. You can also use the keyboard to display the tooltip by moving the keyboard focus to the graphic and pressing the key combination [Ctrl] + [q].

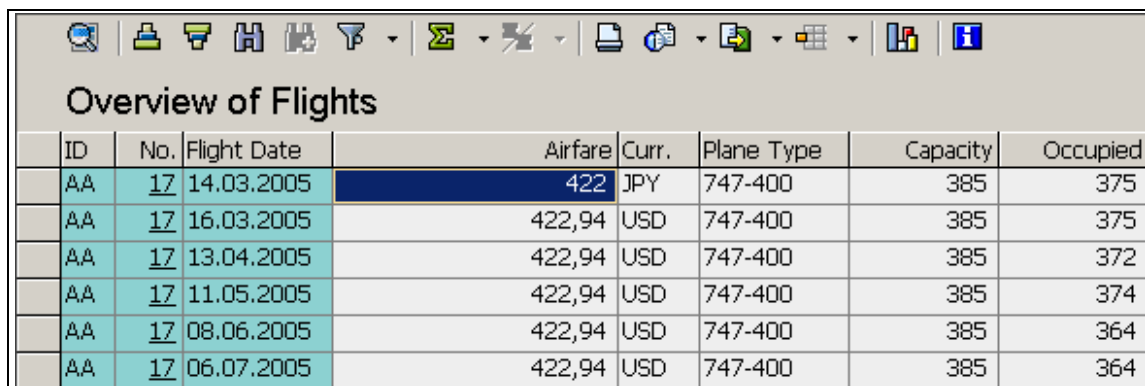
### 5.3.8 ALV Grid

#### Requirements

- Each ALV Grid (as shown in **Figure 5.7**) must have a meaningful title or a meaningful tooltip. Titles are optional in certain cases, as described in Chapter 3.9.
- Each column in an ALV Grid must have a meaningful and visible column header. Labels are not necessary for any input fields, radio buttons, checkboxes, combo boxes, or dropdown boxes used in table cells. The column header is used instead of a label.
- Any semantic colors that are used must be associated with an explanation of their meaning.

#### Test

- Determine whether the application contains any ALV Grid.
- Note: the screen reader *JAWS® for Windows* by Freedom Scientific® announces an ALV Grid as a “Grid View”.
- Check that the ALV Grid has a title. If it does not have a title, you can use the screen reader to check that it has a tooltip.
- Check that all columns have meaningful and visible column headers.
- If semantic colors are used, a screen reader is the only way of checking that the meaning of these colors is announced.



ID	No.	Flight Date	Airfare	Curr.	Plane Type	Capacity	Occupied
AA	17	14.03.2005	422	JPY	747-400	385	375
AA	17	16.03.2005	422,94	USD	747-400	385	375
AA	17	13.04.2005	422,94	USD	747-400	385	372
AA	17	11.05.2005	422,94	USD	747-400	385	374
AA	17	08.06.2005	422,94	USD	747-400	385	364
AA	17	06.07.2005	422,94	USD	747-400	385	364


Figure 5.7: ALV Grid

### 5.3.9 Table Control

#### Requirements

- Each table control must have a meaningful title. Titles are optional in certain cases, as described in Chapter 3.9.
- Each column in a table control must have a meaningful and visible column header. Labels are not necessary for any input fields, radio buttons, checkboxes, combo boxes, or drop-down boxes used in table cells. The column header is used instead of a label.

#### Test

- Determine whether the application contains any table controls.
- If a configurable button  appears in the upper right corner of the table, it is a table control. Not every table control has this configurable button, however.
- Note: the screen reader *JAWS® for Windows* by Freedom Scientific® announces a table control as a “table”.
- Check that the table control has a title.
- Check that all columns have meaningful and visible column headers.

### 5.3.10 Tree (Tree Control, ALV Tree)

#### Requirements

- The root node of a tree must have a meaningful and visible label. If the tree takes up the whole screen, a meaningful window title can be used instead of a label for the root node. In exceptional cases, a tooltip can be used instead of a label.
- Any semantic colors that are used must be associated with an explanation of their meaning.
- Each column in a column tree must have a column header.
- Each element in a column tree row must be accessible using element-by-element navigation and not just row-by-row navigation.

#### Test

- Determine whether the application contains any trees.
- Check that the tree has a label or tooltip.
- If semantic colors are used, a screen reader is the only way of checking that the meaning of these colors is announced.
- Check that all columns in column trees have meaningful and visible column headers.
- Check that you can navigate to each element in a column tree row using element-by-element navigation.

### 5.3.11 Step Loop

#### Requirements

- Step loops should be avoided.
- If step loops have to be implemented, the following rules must be observed:

Each column in a step loop must have a visible column header.

There can be no more than one empty row between the column headers and the first row of data.

The column headers must be created as described in Chapter 3.15.

#### Test

- Determine whether the application contains any step loops.
- Check that each column has a visible header.
- Check that there is no more than one empty row between the column headers and the first row with data. If this is not the case, the next test cannot be performed.
- A screen reader is the only way of checking that the column headers have been created correctly. Navigate to the data fields and check that the associated column headers are announced by the screen reader.

### 5.3.12 HTML Control

#### Requirements

- HTML controls should be avoided.
- If HTML controls have to be used, their content must be accessible in accordance with *HTML Techniques for Web Content Accessibility Guidelines 1.0* (available at <http://www.w3.org/TR/WCAG10-HTML-TECHS/>). More information about HTML controls is available in Chapter 3.16.

#### Test

- Determine whether the application contains any HTML controls.
- Note: the screen reader *JAWS® for Windows* by Freedom Scientific® announces an HTML control as an “HTML control container”.
- If you want to check that the content of the HTML control meets the requirements in *HTML Techniques for Web Content Accessibility Guidelines 1.0*, you must familiarize yourself with the different keyboard access to UI elements implemented in Microsoft Internet Explorer.

### 5.3.13 Calendar Control

#### Requirements

If a calendar day in a calendar control is not displayed in a standard background color, the meaning of the color must be explained in an associated description text. More information is available in Chapter 3.17.

#### Test

- Determine whether the application contains any calendar controls.
- Check that a description of the meanings of the colors exists. You can display the description by moving the cursor over the UI element.
- You can also use a screen reader to check that the meaning of the color is announced.

### 5.3.14 Tabstrip

#### Requirement

Each tab page must have a visible and meaningful title (as shown in **Figure 5.8**).

#### Test

- Use the keyboard to navigate to the tabstrip.
- If the focus is on a tab page title, you can press [Arrow Left] or [Arrow Right] to navigate to the other tab page titles. You can use the space bar to select a tab page.

Assignments | Control | **Period Closing** | Investment Management

**Period-end closing**

Results analysis keys	<input type="text"/>
Costing sheet	<input type="text"/>
Overhead key	<input type="text"/>

**Basic settlement**

Settlement cost element	<input type="text"/>
Cost center	<input type="text"/>
G/L account	<input type="text"/>

Figure 5.8: Tab Strip with Five Tab Pages

### 5.3.15 Group

#### Requirement

A group must have a visible and meaningful title (as shown in **Figure 5.9**).

#### Test

- Use the keyboard to navigate to the first element in the group.
- Check that the group has a visible title.
- You can also use a screen reader to check that the group title is announced when the focus moves to the first element in the group.

**Flight Availability**

Flight/Airline	LH456	Lufthansa
Departing	09:45 am	Frankfurt
Arriving	01:20 pm	Los Angeles

Details | Select another flight | Cancel flight

Figure 5.9: Group "Flight Availability"

## 6 Useful Links

For information about keyboard navigation for user interface elements, see [SAP Design Guild Edition Accessibility](#) under “Links & More” → “Related Topics” → “Keyboard Control of UI Elements” (English and German versions available).

The following list contains the links used in this document.

[SAP Design Guild Edition Accessibility](#)

<http://www.sapdesignguild.org/editions/edition9/acc.asp>

[Accessibility glossary of the SAP Design Guild](#)

[http://www1.sapdesignguild.org/editions/edition9/acc\\_glossary.asp](http://www1.sapdesignguild.org/editions/edition9/acc_glossary.asp)

Latest patches for SAP GUI for Windows 6.40 or latest Screen Reader Extension Setup for the integration of the screen reader JAWS® for Windows by Freedom Scientific® with SAP GUI for Windows on [SAP Service Marketplace](#)

[http://service.sap.com/~form/handler?\\_APP=00200682500000001943&\\_EVENT=DISPHIER&HEADER=Y&FUNCTIONBAR=N&EVENT=TREE&TMPL=01200615320200006164&V=MAINT&TA=ACTUAL&PAGE=SEARCH](http://service.sap.com/~form/handler?_APP=00200682500000001943&_EVENT=DISPHIER&HEADER=Y&FUNCTIONBAR=N&EVENT=TREE&TMPL=01200615320200006164&V=MAINT&TA=ACTUAL&PAGE=SEARCH)

SAP Note [740785](#): “Accessibility: SAP GUI for Windows and JAWS integration”.

[http://service.sap.com/~form/handler?\\_APP=01100107900000000342&\\_EVENT=REDIR&\\_NNUM=740785](http://service.sap.com/~form/handler?_APP=01100107900000000342&_EVENT=REDIR&_NNUM=740785)

SAP Note [855018](#): “Accessibility recommendations for SAP GUI for Windows”.

[http://service.sap.com/~form/handler?\\_APP=01100107900000000342&\\_EVENT=REDIR&\\_NNUM=855018](http://service.sap.com/~form/handler?_APP=01100107900000000342&_EVENT=REDIR&_NNUM=855018)

SAP Note [494173](#): “Screen: “Left label” and “Right label” attribute”

[http://service.sap.com/~form/handler?\\_APP=01100107900000000342&\\_EVENT=REDIR&\\_NNUM=494173](http://service.sap.com/~form/handler?_APP=01100107900000000342&_EVENT=REDIR&_NNUM=494173)

SAP Note [657563](#) “Dynamic documents: Accessibility enhancements”

[http://service.sap.com/~form/handler?\\_APP=01100107900000000342&\\_EVENT=REDIR&\\_NNUM=657563](http://service.sap.com/~form/handler?_APP=01100107900000000342&_EVENT=REDIR&_NNUM=657563)

[Standards of WCAG 1.0](#)

<http://www.w3.org/TR/WAI-WEBCONTENT/>

[Standards of US Section 508](#)

<http://www.section508.gov/index.cfm?FuseAction=Content&ID=12>

[HTML Techniques for Web Content Accessibility Guidelines 1.0](#)

<http://www.w3.org/TR/WCAG10-HTML-TECHS/>