

Keyboard Access for Web Dynpro for ABAP / for Java



Release SAP NW 7.0



Copyright

© Copyright 2008 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, Informix, i5/OS, POWER, 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.

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.

Icons in Body Text

SAP - Important Disclaimers and Legal Information

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.

Coding samples






Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Internet hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint where to find supplementary documentation. SAP does not warrant the availability and correctness of such supplementary documentation or the ability to serve for a particular purpose. SAP shall not be liable for any damages caused by the use of such documentation unless such damages have been caused by SAP's gross negligence or willful misconduct.

Accessibility

The information contained in the SAP Library documentation 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.

| Icon | Meaning |
|---|----------------|
|  | Caution |
|  | Example |
|  | Note |
|  | Recommendation |
|  | Syntax |

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help → General Information Classes and Information Classes for Business Information Warehouse* on the first page of any version of *SAP Library*.

Typographic Conventions

| Type Style | Description |
|---------------------|--|
| <i>Example text</i> | Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation. |
| Example text | Emphasized words or phrases in body text, graphic titles, and table titles. |
| EXAMPLE TEXT | Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE. |
| Example text | Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools. |
| Example text | Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation. |
| <Example text> | Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system. |
| EXAMPLE TEXT | Keys on the keyboard, for example, F2 or ENTER. |

| | |
|---|----|
| Keyboard Access for UI Elements in Web Dynpro | 6 |
| BreadCrumb | 13 |
| Button | 13 |
| CheckBox | 13 |
| CheckBoxGroup | 14 |
| DateNavigator..... | 14 |
| DropDownByIndex | 15 |
| DropDownByKey | 16 |
| Group..... | 17 |
| Image..... | 18 |
| InputField..... | 18 |
| ItemListBox..... | 18 |
| LinkToAction..... | 19 |
| LinkToURL..... | 19 |
| MenuBar | 19 |
| PhaseIndicator | 20 |
| ProgressIndicator | 20 |
| RadioButton..... | 20 |
| RadioButtonGroupByIndex..... | 21 |
| RadioButtonGroupByKey | 21 |
| RoadMap | 22 |
| ScrollContainer | 22 |
| Table..... | 22 |
| TabStrip | 29 |
| TextEdit | 29 |
| TextView..... | 30 |
| Toggle Button | 30 |
| ToggleLink | 30 |
| Tray | 30 |
| Tree | 31 |
| TriStateCheckBox | 33 |



Keyboard Access for UI Elements in Web Dynpro

This document describes keyboard access for UI elements in applications running in the HTML client and based on Web Dynpro for ABAP or Web Dynpro for Java. This information is relevant to you if you use the keyboard to navigate around your application's UI and use its functions.

Prerequisites

This document is valid for the following releases only:

- SAP NetWeaver, Web Dynpro for ABAP 7.0, from Support Package Stack (SPS) 14
- SAP NetWeaver, Web Dynpro for Java 7.0, from SPS 14
- SAP NetWeaver, Web Dynpro for Java 7.01

To use the keyboard commands described here, you must [enable accessibility mode \[External\]](#). Many of the commands also work when accessibility mode is disabled, but others, such as group navigation or navigation of inactive UI elements, only work when it is enabled.

Keyboard Commands

The table below shows you the actions performed by each UI element and the keyboard commands associated with these actions.



The table shows you the original name of each UI element. The link incorporated into the name of the UI element takes you to its documentation in Web Dynpro for ABAP. (To keep the size of this file to a minimum, technical information about user interface elements that is not relevant for keyboard operations has been omitted from this file.)

In this document, the phrase *Moves the focus forwards* indicates that (within the tab order) you are able to move the focus rectangle forwards to a UI element and then further forwards to the next UI element. The phrase *Moves the focus backwards* indicates that (within the tab order) you are able to move the focus rectangle backwards to a UI element and then further back to the previous UI element.



Example: You can use the `Tab` key to move the focus rectangle forwards to the *Button* UI element. Once the focus is on the *Button* element, you can trigger the associated function by pressing the `Enter` key or the `Space` key. You can use the key combination `Shift + Tab` to move the focus rectangle backwards from the *Button* element.

Key Access in Web Dynpro: Overview

| UI Element | Action | Keyboard Command |
|--------------------------------------|---|------------------------|
| BreadCrumb [Page 13] | Moves the focus forwards The focus is then on the subelement currently selected. | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Moves between the subelements of the element | Arrow right/Arrow left |
| | Triggers the focused subelement | Space |

| UI Element | Action | Keyboard Command |
|--|--|----------------------------------|
| Button [Page 13] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Triggers the focused element | Enter or Space |
| | Triggers the element that is activated by default [External] (the element is triggered even when it does not have the focus) | Enter |
| CheckBox [Page 13] CheckBoxGroup [Page 14] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Moves between the checkbox elements | Tab |
| | Activates the focused checkbox (sets or removes the flag) | Space |
| DateNavigator [Page 14] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Activates the focused element | Space |
| | Moves between the subelements (days) of the element | Arrow keys |
| | Selects the focused subelement | Space |
| DropDownByIndex [Page 15] DropDownByKey [Page 16] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Moves between the items in the list box | Arrow up/Arrow down |
| | Moves to the first item in the list | Ctrl + Home |
| | Moves to the last item in the list | Ctrl + End |
| | Selects the selected item in the list | Enter |
| | Opens/closes the list box | Alt + Arrow down/Arrow up, or F4 |
| Group [Page 17] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Forwards group navigation | F6 |
| Image [Page 18] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| InputField [Page 18] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |

| UI Element | Action | Keyboard Command |
|------------|--|--|
| | Navigates within the input field | Arrow right/Arrow left |
| | Places the text cursor (also known as the caret) before the first character in the input field | Home |
| | Places the text cursor after the last character in the input field | End |
| | Moves to the next word in the input field | Ctrl + Arrow right |
| | Moves to the previous word in the input field | Ctrl + Arrow left |
| | Selects the character immediately after the text cursor | Shift + Arrow right |
| | Selects the character immediately before the text cursor | Shift + Arrow left |
| | Selects the word immediately after the text cursor | Ctrl + Shift + Arrow right |
| | Selects the word immediately before the text cursor | Ctrl + Shift + Arrow left |
| | Selects the area between the text cursor and the end of the input field | Shift + End |
| | Selects the area between the text cursor and the start of the input field | Shift + Home |
| | Removes the selection | Arrow right/Arrow left |
| | Copies the selected text to the clipboard | Ctrl + c (or Ctrl + Enter on the numeric keypad) |
| | Cuts the selected text and copies it to the clipboard | Ctrl + x (or Shift + Del on the numeric keypad) |
| | Pastes from the clipboard | Ctrl + v (or Shift + Ins on the numeric keypad) |
| | Opens or closes value help/F4 help | Alt + Arrow down/Arrow up or F4 |
| | Starts editing | Any valid character |
| | Stops editing | Tab |
| | Deletes the character immediately after the text cursor | Delete |
| | Deletes the character immediately before the text cursor | Backspace |
| | Undoes the last entry | Ctrl + z |

| UI Element | Action | Keyboard Command |
|---|--|------------------------|
| | Redoes the last entry | Ctrl + y |
| ItemListBox [Page 18] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Moves between the items in the list box | Arrow up/Arrow down |
| | Moves to the first item in the list box | Home |
| LinkToAction [Page 19] LinkToUrl [Page 19] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Activates the focused link | Enter |
| MenuBar [Page 19] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Moves between the items in the menu bar | Arrow right/Arrow left |
| | Moves to the first item in the menu bar | Home |
| | Moves to the last item in the menu bar | End |
| | Opens the menu of an item in the menu bar | Arrow down |
| | Navigates in the open menu | Arrow keys |
| | Selects the focused menu item | Enter |
| | Closes the current level of the menu | Escape |
| PhaseIndicator [Page 20] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Moves between the phases | Arrow right/Arrow left |
| | Displays the focused phase | Enter |
| ProgressIndicator [Page 20] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| RadioButton [Page 20] RadioButtonGroupByIndex [Page 21] RadioButtonGroupByKey [Page 21] | Moves the focus forwards (the focus first moves to the RadioButtonGroup (from outside the group) and then to the selected RadioButton) | Tab |
| | Moves the focus backwards (when moving backwards to the element, the focus first moves to the selected RadioButton element in the RadioButtonGroup (from outside the group)) | Shift + Tab |
| | Moves between the | Arrow keys |

| UI Element | Action | Keyboard Command |
|---|---|--|
| | RadioButton elements within a RadioButtonGroup, moving the selection at the same time | |
| | Moves between the RadioButton elements within a RadioButtonGroup, without moving the selection at the same time | Shift + Arrow keys |
| | If the selection is not moved: Sets the selection on the focused RadioButton element, not the selected RadioButton element | Space |
| RoadMap [Page 22] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Moves between the roadmap steps | Arrow right/Arrow left |
| | Activates the focused roadmap step | Space |
| ScrollContainer [Page 22] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Scrolls up one page in the focused element (if it is not an input element) | Page Up |
| | Scrolls down one page in the focused element (if it is not an input element) | Page Down |
| Table [Page 22] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Moves from cell to cell in the table (forwards, row by row) | Tab |
| | Moves from cell to cell in the table (backwards, row by row) | Shift + Tab |
| | Moves between cells in the table | Arrow keys, if the focus is not on an editable cell (since character-by-character navigation within the cell is used in this case) |
| | Moves between the characters in an editable cell | Arrow keys |
| | Move forwards out of the table (group navigation) | F6 |
| TabStrip [Page 29] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Activates the focused tab index and displays the related | Space |

| UI Element | Action | Keyboard Command |
|------------------------------------|--|--|
| | tab page | |
| | Moves between the tab indexes | Arrow right/Arrow left |
| | If tab pages are hidden: Displays the next tab page (pages forwards) | Ctrl + Page Down |
| | If tab pages are hidden: Displays the previous tab page (pages backwards) | Ctrl + Page Up |
| TextEdit [Page 29] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Moves the text cursor (also known as the caret) between the characters in a line | Arrow right/Arrow left |
| | Places the text cursor before the first character in the line | Home |
| | Places the text cursor after the last character in the line | End |
| | Moves vertically between the lines | Arrow up/Arrow down |
| | Deletes the character immediately before the text cursor | Backspace |
| | Deletes the character immediately after the text cursor | Delete |
| | Inserts a line break in the text | Enter |
| | Copies the selected characters from the text to the clipboard | Ctrl + c (or Ctrl + Enter on the numeric keypad) |
| | Cuts the selected characters from the text and copies them to the clipboard | Ctrl + x (or Shift + Del on the numeric keypad) |
| | Pastes characters from the clipboard into the text | Ctrl + v (or Shift + Ins on the numeric keypad) |
| | Undoes the last action in the text | Ctrl + z (or Alt + Backspace) |
| | Redoes the last action in the text | Ctrl + y |
| | Selects the character immediately after the text cursor | Shift + Arrow right |
| | Selects the character immediately before the text cursor | Shift + Arrow left |
| | Selects the word immediately | Ctrl + Shift + Arrow |

| UI Element | Action | Keyboard Command |
|--|--|---------------------------------|
| | after the text cursor | right |
| | Selects the word immediately before the text cursor | Ctrl + Shift + Arrow left |
| | Selects the area between the text cursor and the end of the line | Shift + End |
| | Selects the area between the text cursor and the start of the line | Shift + Home |
| | Removes the selection | Arrow keys |
| | Scrolls up one page | Page Up |
| | Scrolls down one page | Page Down |
| | Moves directly to the first page | Home |
| | Moves directly to the last page | End |
| TextView [Page 30] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| ToggleButton [Page 30] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Enables or disables the focused element | Space or Enter |
| ToggleLink [Page 30] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Opens or closes the focused link | Space or Enter |
| Tray [Page 30] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Expands the focused element | Plus key on the numeric keypad |
| | Collapses the focused element | Minus key on the numeric keypad |
| Tree [Page 31] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Moves down one node | Arrow down |
| | Moves up one node | Arrow up |
| | Expands a closed node | Arrow right |
| | Collapses an open node | Arrow left |
| TriStateCheckbox [Page 33] | Moves the focus forwards | Tab |
| | Moves the focus backwards | Shift + Tab |
| | Toggles between the three states of the focused element | Space |



BreadCrumb

A BreadCrumb displays the current page in the context of a navigation path. You can, for example, display a history of the pages last visited or the structure of the information provided. A BreadCrumb consists of individual links, or in its entirety represents one link.

You can insert two different types of BreadCrumb steps into one BreadCrumb.

- [BreadCrumbStep \[External\]](#)
- [MultipleBreadCrumbStep \[External\]](#)

BreadCrumbSteps are bound to individual context attributes. In this way, the number of displayed steps is defined during runtime. In comparison, a MultipleBreadCrumbStep is bound to a context node. This allows the number of displayed steps to be dynamically adjusted at runtime.

Example of the Visual Display

[Step 1](#) > [Step 2](#) > [Step 3](#) > [MultiStep1](#) > [MultiStep2](#) > MultiStep3



To enable the development of [accessible \[External\]](#) applications, the *tooltip* property is checked during the syntax check.



Button

The *Button* UI element represents the pushbutton on the screen. The user can execute statements and actions by clicking the pushbutton.



To enable the development of [accessible \[External\]](#) applications, the *tooltip* property is checked during the syntax check if the *text* property is not set.

A tooltip does not necessarily have to be set for this UI element, but it could make sense to set it if it contains detailed semantic information.

Examples of the Display

 Button text

 Button text



CheckBox

With a CheckBox you can implement a single on/off switch. A checkbox enables the user to select a Boolean value (TRUE/FALSE). The UI element consists of a graphic with text. The checkmark in the box indicates that the option is selected and the value is set to TRUE.



To enable the development of [accessible \[External\]](#) applications, the *tooltip* property is checked during the syntax check if the properties *text* and *label* are not set.

A tooltip does not necessarily have to be set for this UI element, but it could make sense to set it if it contains detailed semantic information.

Example of the Visual Display

CheckBox



CheckBoxGroup

The CheckBoxGroup allows users to select an element from a set of predefined options using the checkbox. The UI element CheckBoxGroup arranges the individual [CheckBoxes \[Page 13\]](#) in single-column or two-column tables.

To display a selected CheckBox in a CheckBoxGroup use the method SET_SELECTED from IF_WD_CONTEXT_ELEMENT for the relevant context element. Note that the selection of checkboxes in a CheckBoxGroup is not carried out using a bound context attribute.



To enable the development of [accessible \[External\]](#) applications, the *accessibilityDescription* property is checked during the syntax check if the *tooltip* property is not set.

Examples of the Display

Single Column

CheckBox 1
 CheckBox 2
 CheckBox 3
 CheckBox 4
 CheckBox 5
 CheckBox 6
 CheckBox 7

Two Column

CheckBox 1 CheckBox 2
 CheckBox 3 CheckBox 4
 CheckBox 5 CheckBox 6
 CheckBox 7

Three Column

CheckBox 1 CheckBox 2 CheckBox 3
 CheckBox 4 CheckBox 5 CheckBox 6
 CheckBox 7



DateNavigator

The *DateNavigator* UI element enables users to display and enter dates. Its features include the ability to navigate in a calendar and choose a day, month, year, or range of dates. Primarily, this element should be used to help users to input dates in an appropriate format.

You can use the [DateNavigatorLegend \[External\]](#) and the [DateNavigatorMarking Element \[External\]](#) to add a legend for the description of selected data to the DateNavigator UI element. This enables you to provide the user with information on specific data that is assigned to a category. For example, events in the calendar can be highlighted in a different color and each event can be described with agenda, time, and location.

Example of the Visual Display

February 2005

| | Mo | Tu | We | Th | Fr | Sa | Su |
|----|----|----|----|----|----|----|----|
| 5 | 31 | 1 | 2 | 3 | 4 | 5 | 6 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 7 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 8 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 9 | 28 | 1 | 2 | 3 | 4 | 5 | 6 |
| 10 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

Today Team Meeting
 Vacation



To enable the development of [accessible \[External\]](#) applications, the `accessibilityDescription` property is checked during the syntax check if the `tooltip` property is not set.

The property `accessibilityDescription` applies to the entire `DateNavigator` and is read out by the screen readers if the `DateNavigator` itself is focused on. The tooltip is displayed for each day, except if this day has a marking and this contains a filled tooltip.



DropDownByIndex

A `DropDownByIndex` UI element provides the user with a dropdown list box. You cannot select more than one entry from the selection list. The UI element consists of a text field, a button, and a selection list. Any list item already selected is displayed in the text field. When selecting the button, a list with all possible values is displayed.

Example of Use

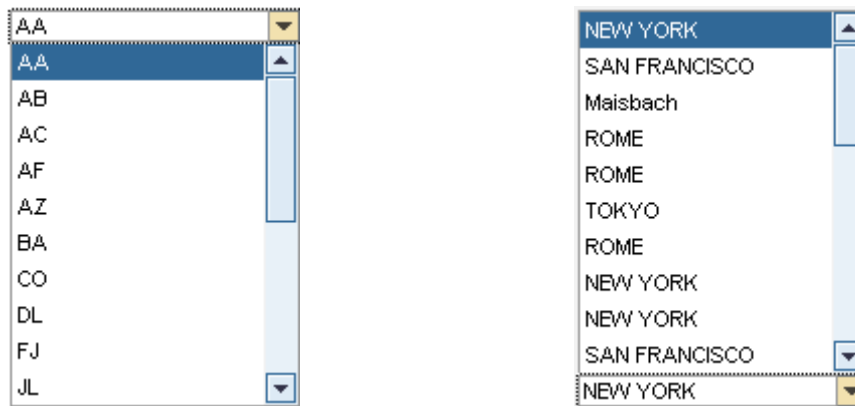
If you want to create a table with dropdown lists, whereby each table row can have different values in its dropdown list, you should use `DropDownByIndex`.



To enable the development of [accessible \[External\]](#) applications, the `label` property is checked during the syntax check.

If no label has been set, and no descriptive text has been specified for the appropriate bound context element in the ABAP Dictionary, the `tooltip` property is checked.

Examples of the Display



The dropdown list box UI elements, `DropDownByIndex` and [DropDownByKey \[Page 16\]](#), do not differ from each other when displayed on the screen. However, the data binding model for the `DropDownByKey` UI element has a completely different concept.



DropDownByKey

A `DropDownByIndex` UI element provides the user with a dropdown list box. You cannot select more than one entry from the selection list. The UI element consists of a text field, a button, and a selection list. Any list item already selected is displayed in the text field. When the user chooses the pushbutton, a list of all possible values is displayed.

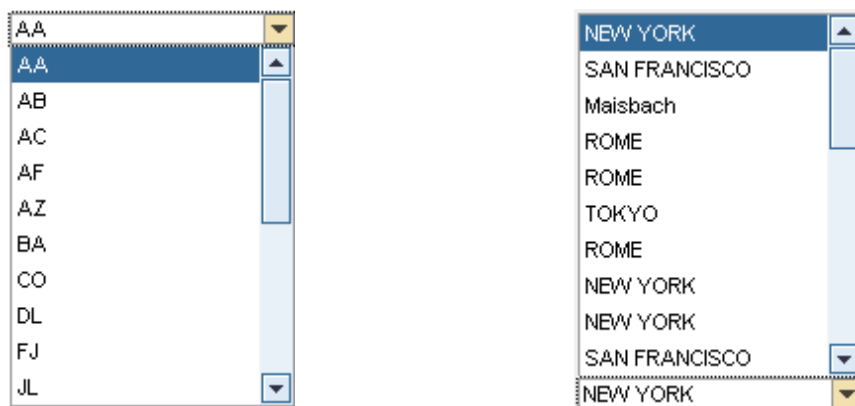
If you want to create a DropDown list in which some entries are available in multiple, use `DropDownByKey`.



To enable the development of [accessible \[External\]](#) applications the *label* property is checked during the syntax check.

If no label has been set, and no descriptive text has been specified for the appropriate bound context element in the ABAP Dictionary, the *tooltip* property is checked.

Examples of the Display



The dropdown list box UI elements, `DropDownByKey` and [DropDownByIndex \[Page 15\]](#), do not differ from each other when displayed on the screen. However, the data binding model for the `DropDownByKey` UI element has a completely different concept.



Group

The *Group* UI element is a UI element container and can be used to group multiple UI elements under one common title. The appearance of this UI element looks like a display panel with a colored background.

The enabled property has no effect on the UI element children you inserted into the [UI Element Container \[External\]](#). If, for example, you set the enabled property to false in the group UI element, an input field inserted in it is not automatically deactivated. If the UI element children in this group UI element are also to be deactivated, you must set the relevant property for each UI element separately.

Examples of the Display

Group with Primary Color

Organization

Organizational Unit: Division:

Sales Organization: Sales Office:

Group with Secondary Color and Tool Bar

Organization

Organizational Unit: Division:

Sales Organization: Sales Office:

Group mit SAP Color

Filter By

Org Unit

Cost Center

Position

Employee



To enable the development of [accessible \[External\]](#) applications, the *accessibilityDescription* property is checked during the syntax check if the *caption* property is not set.

The *tooltip* property is not checked.

A tooltip does not necessarily have to be set for this UI element, but it could make sense to set it if it contains detailed semantic information.



Image

The UI element *Image* enables you to integrate graphics into the Web application in a format that is processed by the Web Server – for example, GIF, JPG, and PNG format. Height and width of the graphic can be specified using the properties *height* and *width*. The graphic is displayed without borders.



To enable the development of [accessible \[External\]](#) applications, the *tooltip* property is checked during the syntax check if the properties *label* and *isDecorative* are not set..



InputField

The *InputField* UI element allows the user to edit or display a single-line text. With an *InputField*, you can edit any simple data type you like. The conversion from the internal display format and, when the user has input a value, back into the internal display format is done automatically. If, after the user has input a value into the field, an error occurs during the conversion to the internal format, the value is not put back into the context, but remains in the data container. When it is next displayed, the *InputField* with the erroneous value is bordered in red and displayed with an error message.

The input is not checked until the roundtrip is triggered.



To enable the development of [accessible \[External\]](#) applications, the *label* property is checked during the syntax check.

If no label has been set, and no descriptive text has been specified for the appropriate bound context element in the ABAP Dictionary, the *tooltip* property is checked.

Example of the Display with a Label



ItemListBox

This UI element is similar to the classical GUI concept of a selection list with simple and multiple selection (list box). A list of text entries is displayed in a box of a fixed size, which if necessary you can scroll through. One or two columns are displayed for the values and possibly one column for symbols is displayed before the value column(s).

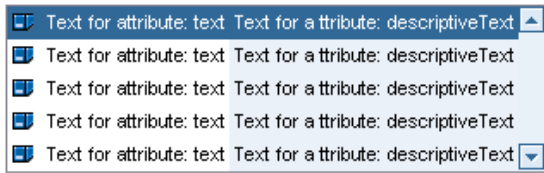
In the case of a simple selection the selected entry is determined by the lead selection of the *dataSource*. If the user changes the selection the property *selectionChangeBehaviour* is adhered to. With multiple selection the selected entries are determined by the *dataSource* selection only; *selectionChangeBehaviour* is not considered.



To enable the development of [accessible \[External\]](#) applications the property *label* is checked during the syntax check.

If no label has been set, and no descriptive text has been specified for the appropriate bound context element in the ABAP Dictionary, the property *tooltip* is checked.

Example of the Visual Display



LinkToAction

The *LinkToAction* UI element is a hypertext link. The navigation to this link triggers a Web Dynpro action.



To enable the development of [accessible \[External\]](#) applications, the *tooltip* property is checked during the syntax check if the property *text* is not set.

A tooltip does not necessarily have to be set for this UI element, but it could make sense to set it if it contains detailed semantic information.

Examples of the Display

| Simple LinkToAction | LinkToAction with Menu Aggregation |
|------------------------------|------------------------------------|
| LinkToAction | LinkToAction ≡ |



LinkToURL

The *LinkToURL* UI element is a hypertext link. When you choose this link, you are directed to a user-defined Web resource (URL).

The LinkToUrl is used to exclusively to open URLs in a separate window. To leave the Web Dynpro application and display a new URL use an exit plug.



To enable the development of [accessible \[External\]](#) applications, the *tooltip* property is checked during the syntax check if the property *text* is not set.

A tooltip does not necessarily have to be set for this UI element, but it could make sense to set it if it contains detailed semantic information.

Examples of the Display

| Simple LinkToURL | LinkToURL with Menu Aggregation |
|---------------------------|---------------------------------|
| LinkToURL | LinkToURL ≡ |



MenuBar

A MenuBar is used to display actions in a structure. The MenuBar is a toolbar that can be organized in different blocks, the [Menus \[External\]](#). Under each block, you can organize individual menu items or other menus.

Example of the Visual Display

Text for Attribute: title Text for Attribute: title Text for Attribute: title Text for Attribute: title



PhaseIndicator

Similar to the [RoadMap \[Page 22\]](#) UI element, the *PhaseIndicator* UI element displays the steps in a wizard. Each step is represented by a separate [Phase \[External\]](#) object. As opposed to using the RoadMap UI element, the application development can display larger steps using the PhaseIndicator UI element which may require more time by the user.

Example of the Visual Display



To enable the development of [accessible \[External\]](#) applications, the *accessibilityDescription* property is checked during the syntax check if the *tooltip* property is not set.



ProgressIndicator

The ProgressIndicator UI element shows how much progress an activity has made in the form of a horizontal bar, along with the value that you have assigned to the *percentValue* property. You can use the *displayValue* property to display a text in the ProgressIndicator on the left side of the UI element. This makes it possible to provide descriptions with specific percentage values. You can hide the DisplayValue value using the *showValue* property. You can display the ProgressIndicator UI element in different colors using the *barColor* property. You can assign a popup menu to a ProgressIndicator.

You can use the ProgressIndicator UI element to display, for example, a project status in percent.

Example of the Display



To enable the development of [accessible \[External\]](#) applications the *tooltip* property is checked during the syntax check.



RadioButton

The *RadioButton* UI element is a button with two states (on/off) that enables users to select options. The *RadioButton* UI element allows you to spread the displayed radio buttons individually on the screen instead of grouping them in a table. You can toggle the radio button when you bind the UI elements to the same context attribute.

The radio button is selected if the context attribute to which the *selectedKey* property is bound, contains the value of the key that belongs to this radio button. The key is specified by the *keyToSelect* property.



To enable the development of [accessible \[External\]](#) applications, the *tooltip* property is checked during the syntax check if the properties *text* and *label* are not set.

A tooltip does not necessarily have to be set for this UI element, but it could make sense to set it if it contains detailed semantic information.



RadioButtonGroupByIndex

The `RadioButtonGroupByIndex` UI element represents a number of radio buttons groups in columns and rows. Unlike UI element `CheckBoxGroup`, this UI element allows user to select one element only.

Example of the Visual Display

- Radiobutton 1
- Radiobutton 2
- Radiobutton 3

The dropdown list box UI elements, `RadioButtonGroup` and [RadioButtonGroupByKey \[Page 21\]](#), do not differ from each other when displayed on the screen. However, the data binding model for the `RadioButtonGroupByKey` UI element has a completely different concept (see section *Data Binding*).



To enable the development of [accessible \[External\]](#) applications, the *accessibilityDescription* property is checked during the syntax check if the *tooltip* property is not set.



RadioButtonGroupByKey

The `RadioButtonGroupByKey` UI element groups multiple `RadioButtons` in a table. Unlike UI element `CheckBoxGroup`, this UI element allows user to select one element only.

Example of the Visual Display

- Radiobutton 1
- Radiobutton 2
- Radiobutton 3

The dropdown list box UI elements, `RadioButtonGroup` and [RadioButtonGroupByIndex \[Page 21\]](#), do not differ from each other when displayed on the screen. However, the data binding model for the `RadioButtonGroupByKey` UI element has a completely different concept (see section *Data Binding*).



To enable the development of [accessible \[External\]](#) applications, the *accessibilityDescription* property is checked during the syntax check if the *tooltip* property is not set.

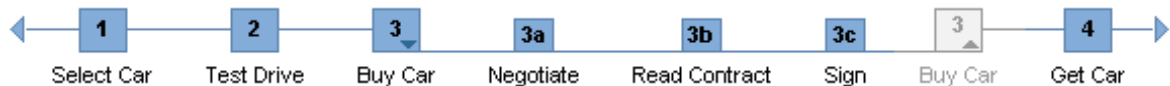


RoadMap

The *RoadMap* UI element displays the steps in a wizard. Each step is represented by a separate [RoadMapStep Object \[External\]](#) or [MultipleRoadMapStep \[External\]](#). You can use various symbols to mark the start points and the end points of this UI element. Assigning the value more to the property `startPointDesign` or `endPointDesign` indicates to the user that there are other steps before the first visible step, or after the last visible step.

The *RoadMap* UI element is used to display step by step workflows. This enables an application development team to visualize small steps of a clearly defined work process.

Example of the Visual Display



To enable the development of [accessible \[External\]](#) applications, the `accessibilityDescription` property is checked during the syntax check if the `tooltip` property is not set.



ScrollContainer

The *ScrollContainer* UI element enables you to use a vertical and horizontal scroll bar for the visible UI elements [Group \[Page 17\]](#) and [Tray \[Page 30\]](#).



To enable the development of [accessible \[External\]](#) applications, the properties `accessibilityDescription` and `tooltip` are checked during the syntax check.

A tooltip does not necessarily have to be set for this UI element, but it could make sense to set it if it contains detailed semantic information.



Table

In a Web Dynpro table, data is displayed two-dimensional in table cells arranged in rows and columns.

The Web Dynpro table consists of the higher-level UI element **Table** and several view elements for table columns, [TableColumn \[External\]](#). The table contains properties that apply to the entire table, such as one to determine whether entries can be made in the table (`readOnly=true`). In comparison, the `TableColumn` contains properties that are valid only for column headings, properties valid for the whole column, and also other properties valid for cells whose values can vary for each row binding.

The column heading is implemented by the aggregation [Header \[External\]](#) using a caption UI element. The second aggregation of `TableColumn` with the name [TableCellEditor \[External\]](#) contains a UI element that is used to display the cells in the column. If the relevant UI elements are bound to `readOnly` attributes, data cannot be entered in them in the table.

The `Table` represents the elements of the context node bound to `dataSource`. All the properties of a `TableCellEditor` that are bound to an attribute of `dataSource` or to context nodes within the `dataSource`, do not assume the value of the lead selection of the node as is usual, instead they have the value of the selected row.

Visualization

A Table consists of five horizontal areas:

- Table header
This aggregation (*Insert Header*) is optional.
- ToolBar
This aggregation (*Insert Toolbar*) is optional.
- Column headers/hierarchy of column headers (TableColumn header)
If the header is set to Caption.visibility=none for all columns (including hidden ones), the row with the column headers is hidden.
- Data rows with TableCellEditors (*Insert Cell Editor*)
The content of the data rows is determined by:
 - TableColumnns
 - Table.visibleRowCount
 - Content of Table.dataSource
 - TableCellEditors
- Scrollbars

Scroll bars can be displayed on the right-hand side of each table, provided you have set [Application Parameter \[External\]](#) WDTABLENAVIGATION to SCROLLBAR.

Example:

| List of Cd | | | | | |
|------------|--------------------|---------------------|---------------------|-------|-----------------|
| | Artist | Title | Label | Price | Salary currency |
| | Miles Davis | Kind of Blue | Col (Sony) | 5,99 | EUR |
| | Miles Davis | Birth of the Cool | Blue Note (EMI) | 4,99 | EUR |
| | Miles Davis | In a Silent Way | Col (Sony) | 9,99 | EUR |
| | United Jazz & Rock | na endlich! Live | mood records | 10,99 | EUR |
| | Bryan Adams | Best of me | A & M Reco (Univer: | 11,89 | EUR |
| | Robbie Williams | Swing When You're | Chrysalis (EMI) | 3,99 | EUR |
| | AC/DC | Highway To Hell | Epc (Sony) | 7,58 | EUR |
| | the Rolling Stones | Forty Licks - New V | Virgin UK (EMI) | 21,86 | EUR |

When you scroll through a table, a tooltip is displayed. You determine the texts for these scroll tips using the aggregation SCROLL_TIP_PROVIDER. If new data needs to be loaded, releasing the mouse triggers the round trip to the server.

- Footer (for accessible mode)
CL_WD_TREE_BY_NST_TABLE_COL=>E_CELL_DESIGN-GOODVALUE_DARK
Pushbuttons provided in the footer of the table allow navigation. The following scroll bar functions are available:



down down



Example of a simple table with header, column headings, data rows and scrollbar:

| List of Cd | | | | | |
|------------|--------------------|-------------------|-------------------|-------|-----------------|
| | Artist | Title | Label | Price | Salary currency |
| | Miles Davis | Kind of Blue | Col (Sony) | 5,99 | EUR |
| | Miles Davis | Birth of the Cool | Blue Note (EMI) | 4,99 | EUR |
| | Miles Davis | In a Silent Way | Col (Sony) | 9,99 | EUR |
| | United Jazz & Roc | na endlich! Live | mood records | 10,99 | EUR |
| | Bryan Adams | Best of me | A & M Reco (Unive | 11,89 | EUR |
| | Robbie Williams | Swing When You' | Chrysalis (EMI) | 3,99 | EUR |
| | AC/DC | Highway To Hell | Epc (Sony) | 7,58 | EUR |
| | the Rolling Stones | Forty Licks - New | Virgin UK (EMI) | 21,86 | EUR |

In addition to these standard functions, a table can also contain more complex features:

- Selection and selection column
- Selection key
- Master column hierarchies
- Popins (at cell and row levels)
- Header group
- Row grouping
- Column scrolling
- Variants
- Totaling

Note that aggregation is available only for the [SAP List Viewer Integration \[External\]](#).

Selection and selection column

In addition to the columns defined by TableColumns, the table can also contain a selection column. This is controlled by the property [Table.selectionMode \[External\]](#).



Example of a selection column in a table:

| Flight data | | | | | | |
|-------------|---------|---------------|------------|---------|------------------|------------|
| | Airline | Flight Number | Date | Airfare | Airline Currency | Plane Type |
| | AA | 0017 | 14.03.2005 | 4,22 | JPY | 747-400 |
| | | | 16.03.2005 | 422,94 | USD | 747-400 |
| | | | 13.04.2005 | 422,94 | USD | 747-400 |
| | | | 11.05.2005 | 422,94 | USD | 747-400 |
| | | | 08.06.2005 | 422,94 | USD | 747-400 |

In the case of readOnly tables (readOnly = true), you can select a row by clicking the row content. Note that the selection is not changed if you click a link, button, or the expand/collapse icons of the tree column.

Selection key

Tables can contain selection keys. There are two cases here:

- If selectionMode=auto and the dataSource are set, or if selectionMode=multi, multiple selection is possible:

The user can use the selection key to select everything (all rows) or to delete the selection.

You can find an example in the system in component WDR_TEST_TABLE, under selection.

- If the table has aggregation TableRowArrangement of type TableRowGrouping, the user can also expand and collapse everything, in addition to selecting and deleting everything:

You can find an example in the system in component WDR_TEST_TABLE, under RowGrouping.



Examples of selection key:

The screenshot displays two examples of selection keys in SAP Web Dynpro tables. The top example shows a table titled "Flight data" with columns: Airline, Flight Number, Date, Airfare, Airline Currency, and Plane Type. A context menu is open over the first row, showing "Select All" and "Delete Selection". The bottom example shows a table with a "Boolean" column. A context menu is open over the first row, showing "Select All", "Delete Selection", "Expand All", and "Collapse All".

Hierarchy

Tables can also represent a hierarchy using the aggregation master column (*Insert Master Column*).

There are two types of master column differentiated by the structure of the data source:

- [TreeByKeyTableColumn \[External\]](#)
The data source is a flat node, meaning that the hierarchy is displayed using the parent to key relationship.
- [TreeByNestingTableColumn \[External\]](#)
The data source is a recursive node whose subnodes contain lower-level elements.



Example of a master column in a table:

| MIME Browser | | | |
|--------------|---------|-----------|------|
| | Name | Mime Type | Size |
| ▼ | SAP | Folder | 0 |
| ▶ | BC | Folder | 0 |
| ▼ | PUBLIC | Folder | 0 |
| ▶ | SVER | Folder | 0 |
| ▶ | ALVTest | Folder | 0 |

Popin

[TablePopins \[External\]](#) display further information about a row or table cell directly beneath the row in question.



Example of a row popin in a table:

| Flight data | | | | | | | |
|--|-------------------------|---------|---------------|------------|----------------------------------|------------------|------------|
| | | Airline | Flight Number | Date | Airfare | Airline Currency | Plane Type |
| ▶ | Details | AA | 0017 | 14.03.2005 | 4,22 | JPY | 747-400 |
| ▼ | Details | AA | 0017 | 16.03.2005 | 422,94 | USD | 747-400 |
| Detailed Information ✕ | | | | | | | |
| Max. capacity econ.: | | | | | <input type="text" value="385"/> | | |
| Occupied econ.: | | | | | <input type="text" value="375"/> | | |
| ▶ | Details | AA | 0017 | 13.04.2005 | 422,94 | USD | 747-400 |
| ▶ | Details | AA | 0017 | 11.05.2005 | 422,94 | USD | 747-400 |
| ▶ | Details | AA | 0017 | 08.06.2005 | 422,94 | USD | 747-400 |

Header group

If several rows or columns have the same header, rows and columns can be combined using the aggregation *Insert Group Columns* and a hierarchy created. There are two ways of doing this:

- Group columns (TableColumnGroup)
- Table columns (TableColumn)

For this purpose the first three header lines are highlighted in color.



Example of a header grouping in a table:

| Group 1 | | | | Group 3 |
|----------|----------|----------|------------------|---------|
| Group 1 | | | Group 3 / 1 | |
| Gr1 H1 | Gr1 H2 | Header 2 | Header 3 / 1 / 1 | |
| Column 1 | Column 2 | Column 3 | Column 4 | |
| Column 1 | Column 2 | Column 3 | Column 4 | |
| Column 1 | Column 2 | Column 3 | Column 4 | |
| Column 1 | Column 2 | Column 3 | Column 4 | |
| Column 1 | Column 2 | Column 3 | Column 4 | |

Row grouping

To combine the fields belonging to one column, the value `TableColumn.groupingValue` must be bound to the `dataSource`. The consecutive fields that have the same values for `groupingValue` and `selectedCellVariant` are merged into one field.



Example of a row grouping in a table:

| Flight data | | | | | | |
|-------------|---------------|------------|---------|------------------|------------|--|
| Airline | Flight Number | Date | Airfare | Airline Currency | Plane Type | |
| AA | 0017 | 14.03.2005 | 4,22 | JPY | 747-400 | |
| | | 16.03.2005 | 422,94 | USD | 747-400 | |
| | | 13.04.2005 | 422,94 | USD | 747-400 | |
| | | 11.05.2005 | 422,94 | USD | 747-400 | |
| | | 08.06.2005 | 422,94 | USD | 747-400 | |

Column scrolling



We recommend that you do not insert too many columns into a table. If it is however still necessary to do so, horizontal scrolling in the browser window can be avoided by using column scrolling.

There are ways to implement column scrolling:

- Fixed number of visible columns

This mode is activated by specifying a fixed number of scrollable columns in the property `scrollableColumnCount` [External]. The maximum number of columns specified is always displayed and a paginator is provided in the footer for scrolling.

- Columns that fit in the table

So that this mode functions reliably, a range of prerequisites must be met:

- `scrollableColumnCount = -2` activates the mode.
- Furthermore `fixedTableLayout=true` should be activated so that columns are shown only as wide as specified.
- The width specifications of all columns and of the table must be defined, and in the same unit too (exception: the table width can be in a different unit if the column widths are given as a percentage).

As many columns as fit in the table will be displayed. If one of the columns is too wide, in the wrong unit of measure, or has no width specification, it is displayed separately in

the full table width. Note that with an existing selection column, this column is added to the table making the table wider than specified.

Variant

A variant is a list of alternative cells. It is created with the aggregation *Insert Cell Variants*. The *TableStandardCell* has its own *TableCellEditor*.



| Flight data | |
|-------------|------------|
| Name | Value |
| MANDT | 000 |
| CARRID | AA |
| CONNID | 0017 |
| FLDATE | 14.03.2005 |
| PRICE | 422 |



Note that from the three selection options only the *TableStandardCell* should be used for a table. The other two options are reserved for the [ALV \[External\]](#).

Totaling

With totaling you can create subtotals and totals.



Example of totaling in a table:

| Flight data | | | | | | | |
|-------------|---------|---------------|------------|-----|------------------|--------------|------------|
| | Airline | Flight Number | Date | | Airfare | Airline C... | Plane Type |
| | AA | 0064 | 02.09.2005 | | 422,94 | USD | A310-300 |
| | | | 30.09.2005 | | 422,94 | USD | A310-300 |
| | | | 28.10.2005 | | 422,94 | USD | A310-300 |
| | | | 25.11.2005 | | 422,94 | USD | A310-300 |
| | | | 23.12.2005 | | 422,94 | USD | A310-300 |
| | | | 20.01.2006 | | 422,94 | USD | A310-300 |
| | | | 17.02.2006 | | 422,94 | USD | A310-300 |
| | | | 17.03.2006 | | 422,94 | USD | A310-300 |
| | | | 06.06.2006 | | 422,94 | USD | A310-300 |
| | AA | ▲ 0064 | | ▪ | 6.767,04 | EUR | |
| | ▲ AA | | | ▪ ▪ | 12.692,42 | EUR | |
| | AZ | 0555 | 16.03.2005 | | 185,00 | EUR | A319 |
| | | | 13.04.2005 | | 185,00 | EUR | A319 |
| | | | 11.05.2005 | | 185,00 | EUR | A319 |
| | | | 08.06.2005 | | 185,00 | EUR | A319 |

Accessibility

To enable the development of [accessible \[External\]](#) applications, the *accessibilityDescription* property is checked during the syntax check if the caption property is not set.

In addition a check is made whether the aggregation [Header \[External\]](#) is set of columns and whether the aggregation header is visible.

The *tooltip* property is not checked.



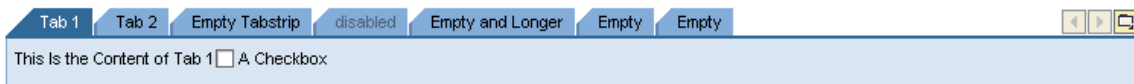
TabStrip

The TabStrip UI element allows the display of a tab. The user can toggle between several tab pages by selecting a specific [Tab \[External\]](#). The same window is shared by all tab pages and used for displaying the content. The user can display the content of a tab by selecting a tab title.

If no selectedTab is specified in the table, or the tab specified in the selectedTab is not visible, the first visible tab is displayed instead.

This does not trigger the onSelect event, nor is the selectedTab property changed.

Example of the Visual Display



To enable the development of [accessible \[External\]](#) applications, the *accessibilityDescription* property is checked during the syntax check if the *tooltip* property is not set.



TextEdit

UI element *TextEdit* makes it possible to enter and display multi-line text. The text in this UI element uses a uniform font, font size, and font style. The UI element is displayed with borders and the frame size is specified by the properties *col* and *row*. If the number of rows exceeds the value of the *row* property, a vertical scroll bar is displayed.

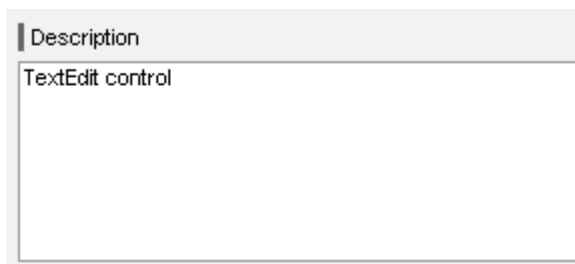
If the value of the *wrapping* property is off, the scroll bar is only be displayed if the text row length exceeds the value of the *col* property.



To enable the development of [accessible \[External\]](#) applications the property *label* is checked during the syntax check.

If no label has been set, and no descriptive text has been specified for the appropriate bound context element in the ABAP Dictionary, the property *tooltip* is checked.

Example of the Display of TextEdit with a Label





TextView

The TextView UI element enables text to be displayed.



When using a TextView UI element, you should always add a label to ensure accessibility.

To enable the development of [accessible \[External\]](#) applications the *tooltip* property is not checked during the syntax check.

A tooltip does not necessarily have to be set for this UI element, but it could be useful to set it if it contains detailed semantic information.

Examples of the Display

One Label: TextView text text text text text text text text text text text
text text text text text text




ToggleButton

The UI element ToggleButton represents the pushbutton on the screen. The user can execute statements and actions by clicking the ToggleButton.



To enable the development of [accessible \[External\]](#) applications, the *tooltip* property is checked during the syntax check if the property *text* is not set.

Example of the Display

▼  ToggleButton text



ToggleLink

The ToggleLink UI element is used to display a hypertext link for the extended search.

Example of the Visual Display

▶ [ToggleLink](#)



To enable the development of [accessible \[External\]](#) applications, the *tooltip* property is checked during the syntax check if the property *text* has not been set.



Tray

The *Tray* UI element (IWDTTray) is a UI element container like the [Group \[Page 17\]](#) UI element container and can be used to group a set of UI elements under one common title. Unlike the

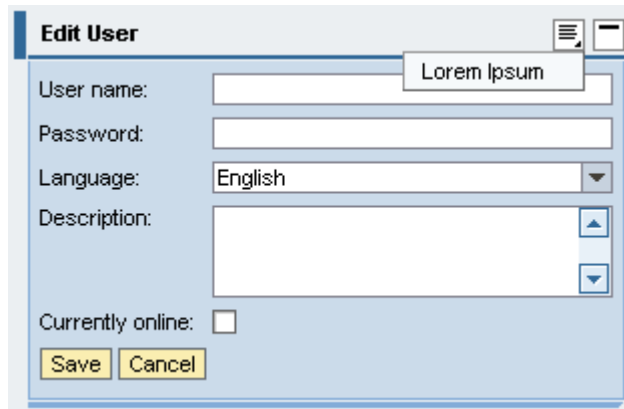
Group UI element it provides additional functions. For example, the Tray UI element can be displayed or hidden.

Examples of the Display

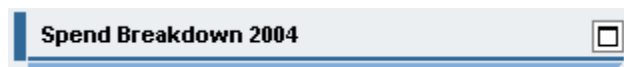
Tray with *Plain Design*



Tray with *Fill and Menu Design*



Collapsed Tray with *Transparent Design*



To enable the development of [accessible \[External\]](#) applications, the `accessibilityDescription` property is checked during the syntax check if the `caption` property is not set.

The `tooltip` property is not checked.

A tooltip does not necessarily have to be set for this UI element, but it could make sense to set it if it contains detailed semantic information.



Tree

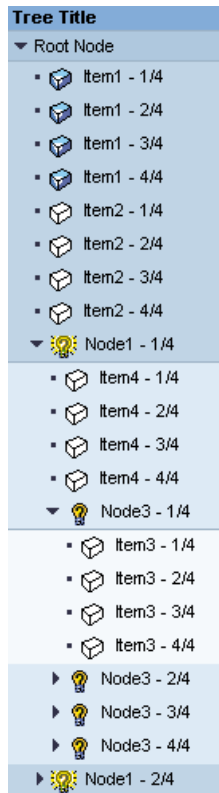
Hierarchies defined in the context can be visualized using the Tree UI element. The hierarchy to be displayed is defined in the context. You can describe this context structure in two ways:

- With a recursive node if the number of levels is not yet known at design time.
- With a non-recursive node if a certain number of levels can be specified at design time.

So a tree is used for navigation. By clicking on an entry the associated data is displayed in another place. If you want to display a tree structure in which data can be input and you want to provide a scroll bar, use a hierarchical table instead of a tree. More information:

[TreeByKeyTableColumn \[External\]](#) and [TreeByNestingTableColumn \[External\]](#). A tree does not enable the user to select entries, it can only interact with the LeadSelection. Note that although this looks like a selection, it does affect performance, which means that for most scenarios it would be a hindrance.

Example of the Visual Display



The Tree UI element is bound against the top-level context node to be displayed.

You use nodes ([TreeNodeType \[External\]](#) elements) or leaves ([TreeItemType \[External\]](#) elements) to specify which subnodes are to be displayed and which context attributes are to be displayed on these subnodes as a text or tooltip. The dataSource property of the TreeNodeType element or TreeItemType element is bound to the corresponding context node and the properties text, tooltip, and so on, are bound to the corresponding context attributes on this context node.

TreeItemType elements cannot have children. Therefore, they are always displayed as leaves. They are used when it is decided at design time that the corresponding node does not have children. When using TreeNodeType elements, the decision of whether to use children is dynamically made at runtime.



Hierarchy levels defined in the context cannot be left out when displaying the UI element. For example, a TreeNodeType element that is bound to the Orders must also exist to display the items for the hierarchy Customers → Orders → Items, which is defined in a context.

All nodes that are not directly below the context root node must be non-singleton nodes, because all elements should be displayed in a tree regardless of the lead selection.

You can also bind the tree so that the `dataSource` of the tree binds to a structured 1:1 node and the element nodes of the tree are located below them. This is necessary to map directory structures. With recursive trees you can display the recursions for this 1:1 node. This is then skipped with rendering.



To enable the development of [accessible \[External\]](#) applications, the `tooltip` property is checked during the syntax check.



TriStateCheckBox

This UI element is similar to a [CheckBox \[Page 13\]](#), with the difference that the clicked on status is variable:

- Option can be activated (selected)
- Option cannot be activated (not selected)
- Option is unspecified

Examples of the Display

Selected

TriStateCheckBox

Not Selected

TriStateCheckBox

Not Specified

TriStateCheckBox