FUJITSU Software
Interstage Business Process Manager
V11.4.1

QuickForm UI Widgets Reference

Preface

Purpose of this Manual

This manual describes the UI widgets provided with QuickForm.

Structure of this Manual

The structure of this manual is as follows:

Chapter 1 Introduction

Provides an overview of UI widgets.

Chapter 2 Screen Widgets

Explains the widgets which are displayed on the screen.

Chapter 3 Additional Functional Widgets

Explains the additional widgets of the various functions.

Chapter 4 Notes

Gives notes on using UI widgets.

Abbreviations

The table below lists the product name abbreviations that are commonly used in this manual.

Note that the trademark (™) and registered trademark (®) symbols are omitted throughout this manual.

Abbreviation	Full Name
Internet Explorer	Windows(R) Internet Explorer(R) 9 and later

Export Permission

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Contents

Chapter 1 Introduction

This chapter gives an overview of the UI widgets.

1.1 UI Widgets

"UI widgets" is a generic name for a group of widgets defined in a QuickForm. There are the following two types:

- Screen Widgets

- Additional Functional Widgets

Screen Widgets

"Screen widgets" is a group of widgets which are displayed on the screen such as text input fields, tables, etc. Widgets are transformed into HTML elements at runtime in the Web browser if you specify widgets in the accepted format for HTML.

The following table shows all screen widgets:

Name	Type (This value is defined to type property)	Overview
Text	IBPMText	The widget to display text. This corresponds to text in HTML.
TextInput	IBPMTextInput	The widget to input and edit a single line of text. This corresponds to <input type="text" value="value"> and <input type="password" value="value"> in HTML.
CheckBox	IBPMCheckBox	The widget to display an "on/off" (selected or cleared) check box. This corresponds to <input type="checkbox" value="value "> in HTML.
RadioButton	IBPMRadioButton	The widget to display a radio button. This corresponds to <input type="radio" value="value"> in HTML.
TextArea	IBPMTextArea	The widget to input and edit one or more lines of text. This corresponds to <textarea> in HTML.
Select	IBPMSelect	The widget to display a list from which single or multiple items can be selected. This corresponds to <select> <option> in HTML.
ComboBox	IBPMComboBox	The widget to display a combo box (sometimes called "dropdown box"), which consists of an input field and a list of items from which a single item can be selected. It is also possible to enter text directly in the input field.
DateInput	IBPMDateInput	A type of TextInput widget. This widget to input and edit the date and time.
NumberInput	IBPMNumberInput	A type of TextInput widget. This widget to input and edit numerical values.
SelectList	IBPMSelectList	The widget to display a list from which single or multiple items can be selected.
CheckList	IBPMCheckList	The widget to display a list with check box items.
ViewContainer	IBPMViewContainer	The general container widget. This widget can contain HTML elements, and the whole can be handled as one widget. ViewContainer itself does not have a view component.

Name	Type (This value is defined to type property)	Overview
Panel	IBPMPanel	The widget to display a container with a title bar. This consists of title part and body part. This widget can contain HTML elements for the body part.
ViewStack	IBPMViewStack	The widget to switch between containers in the same position.
TabPanel	IBPMTabPanel	The widget to display containers on tabbed pages. Users can switch between pages by clicking the tabs.
FragmentContainer	IBPMFragmentContainer	The container widget to load and display contents from an external source after the page has been displayed.
TableView	IBPMTableView	The widget to display two-dimensional data in table format. Data can be sorted in ascending or descending order.
TableEdit	IBPMTableEdit	The widget to display two-dimensional data in table format and edit the data.
ViewColumn	IBPMViewColumn	The widget to define columns for TableView and TableEdit.
ButtonForTable	IBPMButtonForTable	The widget to display a button. This is used with TableEdit to add a new row or delete a selected row.
ViewColumnGroup	IBPMViewColumnGroup	The widget to group columns defined with ViewColumn.
Calendar	IBPMCalendar	The widget to display a calendar and select a date.
PopupCalendar	IBPMPopupCalendar	The widget to display a popup calendar window.
CalendarButton	IBPMCalendarButton	The button widget to display a popup calendar when pressed. This is used with PopupCalendar.

Additional Functional Widgets

Additional functional widgets add functions to screen widgets.

Name	Type (This value is defined to type property)	Overview
RadioButtonGroup	RadioButtonGroup	The widget to group radio buttons. Using this widget, you can group radio buttons together into a group.

1.2 Using UI Widgets

- For information on basic descriptions, refer to section 1.2.1 Describing UI Widgets.

- For information on style settings, refer to section 1.2.2 Style Settings.

1.2.1 Describing UI Widgets

UI widgets are described in HTML using the <div> tag or <span> tag.

Format for the <div> tag:

```
<div rcf:id="widget ID" rcf:type="widget type" rcf:property_name="value" ... ></div>
```

Format for the <span> tag:

```
<span rcf:id="widget ID" rcf:type="widget type" rcf:property_name="value" ... ></span>
```

Attribute	Description
rcf:id	Specifies the widget ID (optional). In Interstage BPM Studio, you can specify a User Defined Attribute (UDA) in this property to overwrite other specific property with the UDA value when the widget is initialized. And the value is saved to the UDA value. This id property's value format is uda_ plus <identifier of the UDA>. Refer to 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.
rcf:type	Specifies the widget type.
rcf:property_name	Specifies details such as the properties and style properties defined for each widget.

For example, to relate this widget to UDA (UDA id is Variable1) and to specify the maxLength property and the color property for TextInput, describe as follows:

```
<span rcf:id="uda_Variable1" rcf:type="IBPMTextInput" rcf:maxLength="10" rcf:color="blue">
</span>
```

Note that you can describe some screen widgets with either the <div> tag or the <span> tag, but you must describe certain screen widgets with the <div> tag.
For the widgets that you can describe with either <div> or <span>, note the following difference in display:

- When described with the <div> tag, line feed is inserted before and after the widget.

- When described with the <span> tag, line feed is not inserted before and after the widget.

For more details on each screen widget, refer to Chapter 2 Screen Widgets.

For additional functional widgets, there is no difference whether you describe the <div> tag or <span> tag.

P Point

Note the following points when describing UI widgets:

- End tags
  Always include an end tag, even when describing widgets that contain no child elements. If you describe such widgets as empty elements, subsequent widgets may not be displayed.

  - Correct example:

    ```
    <span rcf:id="id1" rcf:type="IBPMTextInput"></span>
    ```

  - Incorrect example:

    ```
    <span rcf:id="id2" rcf:type="IBPMTextInput"/>
    ```

- Character strings starting with "rcf-"
  Character strings that start with "rcf-" have a special meaning in QuickForm. For this reason, do not specify a character string starting with "rcf-" as the value of the following:

  - HTML id attributes

  - HTML style attributes

  - rcf:id attributes

- rcf:id attributes and HTML id attributes
  When you specify the rcf:id attribute, the ID value that you specify must satisfy the following conditions:

  - It must be unique in the window for which it is specified.
    For example, during delayed loading of a window (when only a part of the window is prepared separately and the window information is loaded in the background while the user operates the window), all of the widget IDs, including the window information loaded by delayed loading, must be unique.

- It must not duplicate any of the values of HTML id attributes for the window.
  If you specify an ID for an HTML element using the id attribute, it cannot be the same as any of the widget IDs for that window.
  If a value is duplicated, correct behavior cannot be guaranteed.

- Array elements
  With properties that have String or Number arrays as values, you can specify multiple array elements by separating them with semi-colons (;).
  Example: For the options property of SelectList

```
<div rcf:type="IBPMSelectList" rcf:options="item1;item2;item3"></div>
```

For empty array elements, specify an empty string as the attribute value.
You cannot specify other types of array (such as Object type) as the attribute value.

Note

The 'onChange' event and other properties are used in the internal operation of the form. Use of properties other than those specifically documented here, may conflict with the operation of the form, and result in unpredictable behavior.

1.2.2 Style Settings

There are two ways to specify the style of screen widgets:

- Settings Using Style Properties

- Settings Using CSS

Settings Using Style Properties

You can set the styles that you specify for screen widgets as properties (style properties).

For example, to change the background color, character color and font size of the TextInput widget, specify as follows:

```
<span rcf:type="IBPMTextInput"
 rcf:backgroundColor="#FFFFFF" rcf:color="#000000" rcf:fontSize="large"></span>
```

Note also that any specified style properties that cannot be specified for screen widgets are ignored.

The style property includes the style property to the entire parts and the style property to specific parts.

Please refer to section 2.6 Style Properties for the naming convention of the style property.

For details of style properties for each widget, please refer to the explanation of each widget.

Settings Using CSS

You can specify a class name for a screen widget using the styleClass property. Using the class name, you can describe styles in CSS.

The following is an example of a description for the SelectList widget.

```
<style type="text/css">
  .myClass .rcf-SelectList{
    width: 100px;
    height: 200px;
    color: blue;
    border-color: black;
    background-color: lightgrey;
  }
  .myClass .rcf-SelectList-optionSelected {
    color: red;
  }
</style>
```

```
<div rcf:type="IBPMSelectList" rcf:styleClass="myClass" rcf:options="foo;bar">
</div>
```

When you specify a class name for a screen widget using the styleClass property, you can describe in CSS the styles that can be specified in the style properties of that widget.

Note the following rules that affect describing styles in CSS:

- If the style property name is entirely in lower case (for example: width, height), the name can be specified as is in CSS:

- If the style property name contains uppercase characters (for example: backgroundColor, fontSize), each uppercase character must be changed to lower case and preceded by a dash (-) in CSS, as in the examples below.

  - Change "backgroundColor" to "background-color".

  - Change "fontSize to "font-size".

- If the style property name contains a prefix (also called a part name), such as titleHeight or bodyPadding, specify the style property name with a specific class name in CSS. (For example ".myClass" is the class name specified in styleClass.)

  For the titleHeight of the Panel:

```
.myClass .rcf-Panel-title {
   height: 30px;
}
```

  For the bodyBackgroundColor of the Panel:

```
.myClass .rcf-Panel-body {
   background-color: white;
}
```

  Note that, the result of defining styles in CSS that you cannot define in screen widget style properties has not been determined.

1.2.3 Attributes that can be used with UI Widget \<div\> tags and \<span\> tags

You can only use attributes that start with "rcf:" and the style attributes in UI widget \<div\> tags and \<span\> tags.

The CSS style properties that you can include in the style attribute are as follows:

- position

- top

- left

- z-index

You cannot use the style property with the PopupCalendar.

1.2.4 Number of Screen Widgets per Page

It is recommended to use no more than 50 screen widgets on a single page. This total includes widgets that are not initially displayed, but are displayed later using widgets such as ViewStack and FragmentContainer.

In the case of a large number of widgets, the page may take a long time to display and the messages may be displayed.

Note that, depending on the performance of the client machine, even if you include fewer than 50 screen widgets, the page may take a long time to display or an error message may be displayed if you use many complex widgets such as TableView, TableEdit and Calendar.

Chapter 2 Screen Widgets

This chapter describes the screen widgets.

2.1 Form Widgets

Form widgets are widgets mainly for allowing users to enter data, for example text boxes, radio buttons, etc.

This section explains how to use form widgets.

2.1.1 Text

Text is the widget to display text.

Sample Display

Text

Format

```
<div rcf:type="IBPMText" ... ></div>
```

Or

```
<span rcf:type="IBPMText" ... ></span>
```

Note

You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements for details.

Point

This widget is displayed in the following way:

- Using with <div> tag, line feed is inserted before and after the widget.

- Using with <span> tag, line feed is not inserted before and after the widget.

Properties

The meaning of the column headers is as follows:

- Required: Whether or not the property is a required setting.

- Default Value: The value assumed if the property is not set.

Name	Data Type	Description	Required	Default Value
value	String	Specifies text. If you specify a UDA to the id property, this property is overwritten by the UDA value when the widget is initialized.	No	""

Name	Data Type	Description	Required	Default Value
		Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.		
xpath	String	This property is used for only a custom data type UDA. Specifies xpath of the targeted data when you specify a custom data type UDA to the id properties. You can specify the xpath of the text node or the attribute node.	Yes	-
title	String	Specifies the text to be displayed in the tooltip.	No	""
labelProvider	-	Cannot be specified in Interstage BPM Studio.	-	-

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

📖 Note

. . . .

If you specified a long character string to the title property, it is automatically wrapped or cut out.

If an empty string is specified, nothing is displayed.

. . . .

Note on displaying text which includes special characters.

When you display text which includes special characters, you must replace special character by escape character.

The correspondence table are as follows:

special character	escape character
&	&
<	<
>	>
"	"
'	'
Line feed character (\r\n or \r or \n)	
Space (single-byte space or tab)	

Example: In the case of the character string <formatted"text">,you specify the text to property of UDA as follows:

```
[Before Conversion]
      <formatted"text">
    (1)        (2)  (2)(3)
```

```
[After Conversion]
      &lt;formatted&quot;text&quot;&gt;
      (1)              (2)        (2)  (3)
```

If you specify the special character to property by mistake,you remove the character.

Then if QuickForm doesn't work correctly, you shuld chack the property in the Source View.

Style Properties
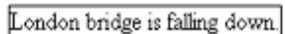
Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-Text	- Color - Font - Text (except for textIndent, textAlign, and whiteSpace) - Border - Binding

Refer to section 2.6 Style Properties for details.

Note

Notes on borders

When a border is specified in Text, it may be incorrectly displayed as in the illustration below.

This can be avoided by using one of the following ways:

- Adding an empty line under the Text widget

```
<div rcf:type="IBPMText" rcf:value="character string"></div>
<br/> 
```
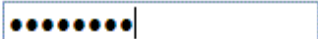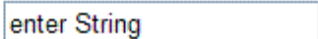
- Specifying the position:relative style attribute in <div> or <span> tag for Text widget

```
<div rcf:type="IBPMText" style="position:relative" rcf:value="character string"></div>
```

2.1.2 TextInput

TextInput is the widget to input and edit a single line of text.

Sample Display

You can undo using Ctrl+Z. Refer to section 4.1.6 Undo Using Ctrl+z in Text Input Widgets for details.

Format

```
<div rcf:type="IBPMTextInput" ... ></div>
```

Or

```
<span rcf:type="IBPMTextInput" ... ></span>
```

🉐 Note

You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements for details.

🅿 Point

This widget is displayed in the following way:

- Using with <div> tag, line feed is inserted before and after the widget.

- Using with <span> tag, line feed is not inserted before and after the widget.

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
value	String	Specifies text. If you specify a UDA to the id property, this property is overwritten by the UDA value when the widget is initialized. And the value entered in the input field is saved to the UDA value. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.	No	""
xpath	String	This property is used for only a custom data type UDA. Specifies xpath of the targeted data when you specify a custom data type UDA to the id properties. You can specify the xpath of the text node or the attribute node.	Yes	-
title	String	Specifies the text to be displayed in the tooltip.	No	""
labelProvider	-	Cannot be specified in Interstage BPM Studio.	-	-
enabled	Boolean	Specifies whether or not this widget is enabled. - true: Enabled - false: Disabled You cannot operate this widget.	No	true
readOnly	Boolean	Specifies whether or not this widget is read-only. - true: Read-only - false: Writable	No	false
password	Boolean	Specifies whether or not password input is required.	No	false

Name	Data Type	Description	Required	Default Value
		- true: A password is required. The input does not appear as text, but as asterisks or bullets, depending on the browser. - false: A password is not required.		
maxLength	Number	Specifies the maximum text length. Each double-byte character is counted as a single character. - 1 or higher: The specified number means the maximum number of characters allowed for this widget. (0 or lower is invalid.) - None: Browser limitations apply.	No	None (-1)
autoEscape	-	Cannot be specified in Interstage BPM Studio.	-	-
tabIndex	Number	Specifies the tab order when moving the focus with the Tab key. You can specify it in the same way as the tabindex attribute in HTML. The focus moves in ascending order if you specify integers greater than zero.	No	0
uppercase	Boolean	Specifies whether or not single-byte and double-byte alphanumeric characters are converted to uppercase. - true: Input characters are converted to uppercase. Characters are converted when they are initially displayed and when the input is determined (when the focus is lost). - false: Input characters are not converted to uppercase.	No	false
imeMode	-	Cannot be specified in Interstage BPM Studio	-	-
mandatory	Boolean	Specifies whether or not the entered value is required. - true: Required Error message is displayed in error field when the value is empty. - false: Not required	No	false

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

📒 **Note**
. . . .

If you specified a long character string to the title property, it is automatically wrapped or cut out.

If an empty string is specified, nothing is displayed.
. . . .

Style Properties

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-IBPMTextInput	- Size - Color - Font (except lineHeight) - Text (except textIndent and whiteSpace) - Border - Padding

Refer to section 2.6 Style Properties for details.

2.1.3 CheckBox

CheckBox is the widget to display a check box.

Sample Display

☐ CheckBox

Format

```
<div rcf:type="IBPMCheckBox" ... ></div>
```

Or

```
<span rcf:type="IBPMCheckBox" ... ></span>
```

📑 **Note**

. . . .

You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements for details.

. . . .

📘 **Point**

. . . .

This widget is displayed in the following way:

- Using with <div> tag, line feed is inserted before and after the widget.

- Using with <span> tag, line feed is not inserted before and after the widget.

. . . .

Properties

The meaning of the column headers is explained in the Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
value	String	Specifies the value of the check box.	No	""
checked	Boolean	Specifies whether or not the button is checked.	No	false

Name	Data Type	Description	Required	Default Value
		- true: Checked - false: Unchecked If you specify a UDA to the id property, this property is overwritten by the UDA value when the widget is initialized and the value of the check box is saved to the UDA value. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.		
title	String	Specifies the text to be displayed in the tooltip.	No	""
label	String	Specifies the label text for the check box.	No	""
enabled	Boolean	Specifies whether or not this widget is enabled. - true: Enabled - false: Disabled You cannot operate this widget.	No	true
tabIndex	Number	Specifies the tab order when moving the focus with the Tab key. You can specify it in the same way as the tabindex attribute in HTML. The focus moves in ascending order if you specify integers greater than zero.	No	0

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

📝 **Note**

. . . .

Notes on using the title property
If you specified a long character string to the title property, it is automatically wrapped or cut out.
If an empty string is specified, nothing is displayed.

Notes on using the label property
If you specified a long character string whose width is greater than the CheckBox width to the label property, it is wrapped by default.

. . . .

Style Properties

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-CheckBox	- Size - Color - Border

Part Name	Prefix	Class Name	Available Styles
			- Padding
Label	label	rcf-CheckBox-label	- Color
			- Font
			- Text (except textIndent, textAlign, and whiteSpace)

Figure 2.1 Structure of CheckBox Widget
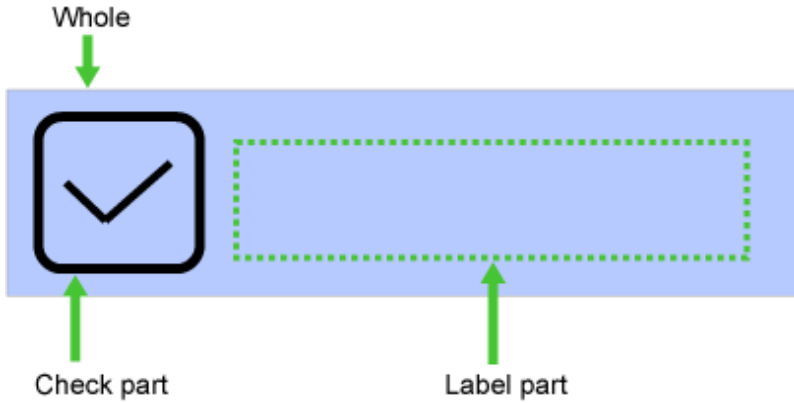
Refer to 2.6 Style Properties for details.

2.1.4 RadioButton

RadioButton is the widget for displaying a radio button.

P Point
....
You can use this with RadioButtonGroup to group radio buttons that are mutually exclusive.
....

Sample Display

Format

```
<div rcf:type="IBPMRadioButton" ... ></div>
```

Or

```
<span rcf:type="IBPMRadioButton" ... ></span>
```

Note
....
You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements for details.
....

P Point
....
This widget is displayed in the following way:

- Using with \<div\> tag, line feed is inserted before and after the widget.

- Using with \<span\> tag, line feed is not inserted before and after the widget.

· ·

Properties

The meaning of the column headers is explained in the Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
value	String	Specifies the value of the radio button.	No	""
checked	Boolean	Specifies whether or not the radio button is selected. - true: Selected - false: Not selected If you specify a UDA to the id property, this property is overwritten by the UDA value when the widget is initialized. And the value of the radio button is saved to the UDA value. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.	No	false
title	String	Specifies the text to be displayed in the tooltip.	No	""
label	String	Specifies the label text for the radio button.	No	""
enabled	Boolean	Specifies whether or not this widget is enabled. - true: Enabled - false: Disabled You cannot operate this widget.	No	true
tabIndex	Number	Specifies the tab order when moving the focus with the Tab key. You can specify it in the same way as the tabindex attribute in HTML. The focus moves in ascending order if you specify integers greater than zero.	No	0

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

Note

· ·

Notes on using the title property
If you specified a long character string to the title property, it is automatically wrapped or cut out.
If an empty string is specified, nothing is displayed.

Notes on using the label property
If you specified a long character string whose width is greater than the RadioButton width to the label property, it is wrapped by default.

Style Properties

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-RadioButton	- Size - Color - Border - Padding
Label	label	rcf-RadioButton-label	- Color - Font - Text (except textIndent, textAlign, and whiteSpace)

Figure 2.2 Structure of RadioButton Widget
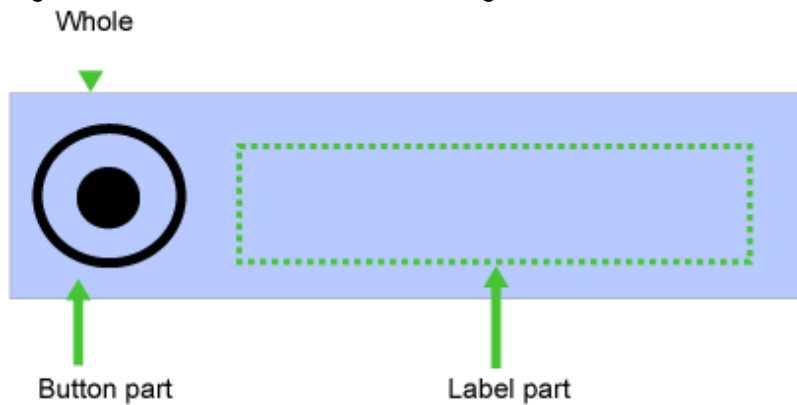
Refer to section 2.6 Style Properties for details.

🜲 Note

Sample code for RadioButton:

You cannot use RadioButtons of UI Widgets to relate multiple RadioButtons with the same UDA because all of the widget IDs must be unique. In such case, you can use RadioButtons of Basic category.
Example:
To relate two radio buttons with the same UDA (UDA identifier is Variable1)


```
<span rcf:type="IBPMText" rcf:value="Language:" ></span>
<input type="radio" NAME="uda_Variable1" value="English"
<%=(ibpmUDAprop.getProperty("uda_Variable1").equals("English"))?"checked":"" %>>
<span rcf:type="IBPMText" rcf:value="English" ></span>
<input type="radio" NAME="uda_Variable1" value="Japanese"
<%=(ibpmUDAprop.getProperty("uda_Variable1").equals("Japanese"))?"checked":"" %>>
<span rcf:type="IBPMText" rcf:value="Japanese" ></span>
```

2.1.5 TextArea

TextArea is the widget to display an input field for one or more lines of text.

🖙 **Note**

· ·

About line feed code

If the character string entered in TextArea contains a line feed, it is represented in the character string value of the value property as "\n"(LF, character code 10).

· ·

Sample Display

```
Enter a Text String
```

You can undo using Ctrl+Z. Refer to section 4.1.6 Undo Using Ctrl+z in Text Input Widgets for details.

Format

```
<div rcf:type="IBPMTextArea" ... ></div>
```

Or

```
<span rcf:type="IBPMTextArea" ... ></span>
```

🖙 **Note**

· ·

You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements for details.

· ·

🅿 **Point**

· ·

This widget is displayed in the following way:

- Using with <div> tag, line feed is inserted before and after the widget.

- Using with <span> tag, line feed is not inserted before and after the widget.

· ·

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
value	String	Specifies text. If you specify a UDA to the id property, this property is overwritten by the UDA value when the widget is initialized. And the value entered in the input field is saved to the UDA value.	No	""

Name	Data Type	Description	Required	Default Value
		Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.		
xpath	String	This property is used for only a custom data type UDA. Specifies xpath of the targeted data when you specify a custom data type UDA to the id properties. You can specify the xpath of the text node, the attribute node or the entire value("/").	No	/
title	String	Specifies the text to be displayed in the tooltip.	No	""
labelProvider	-	Cannot be specified in Interstage BPM Studio.	-	-
enabled	Boolean	Specifies whether or not this widget is enabled. - true: Enabled - false: Disabled You cannot operate this widget.	No	true
readOnly	Boolean	Specifies whether or not this widget is read-only. - true: Read-only - false: Writable	No	false
rows	Number	Specifies the number of rows. You can specify an integer greater than or equal to 1. If a widget height is specified in the height style property, that specification takes precedence.	No	2
cols	Number	Specifies the number of characters the TextArea can contain, taking into account the average character width. You can specify values of 1 or more. If a widget width is specified in the width style property, that specification takes precedence.	No	20
tabIndex	Number	Specifies the tab order when moving the focus with the Tab key. You can specify it in the same way as the tabindex attribute in HTML. The focus moves in ascending order if you specify integers greater than zero.	No	0

Name	Data Type	Description	Required	Default Value
imeMode	-	Cannot be specified in Interstage BPM Studio.	-	-
mandatory	Boolean	Specifies whether or not the entered value is required. - true: Required Error message is displayed in error field when the value is empty. - false: Not required	No	false

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

![Note icon] **Note**

. . . .

If you specified a long character string to the title property, it is automatically wrapped or cut out.

If an empty string is specified, nothing is displayed.

. . . .

Style Properties

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-TextArea	- Size - Color - Font - Text (except whiteSpace and textIndent) - Border - Padding

Refer to section 2.6 Style Properties for details.

2.1.6 Select

Select is the widget to display a list box from which single or multiple items can be selected.

Sample Display

Format

```
<div rcf:type="IBPMSelect" ... ></div>
```

Or

```
<span rcf:type="IBPMSelect" ... ></span>
```

Note

You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements for details.

Point

This widget is displayed in the following way:

- Using with <div> tag, line feed is inserted before and after the widget.

- Using with <span> tag, line feed is not inserted before and after the widget.

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
options	String Array	Specifies the available options in the list. You can specify Strings as an array. The label and value for each item have the same value. If an element of array is an empty string, an error occurs. If you specify a UDA to the prop_options property, this property is overwritten by the UDA value when the widget is initialized.	It is necessary to set the value of either the options property or the prop_options property.	[]
prop_options	String	Specifies a String type UDA in specific format. This UDA value has the available options which are separated by comma. The label and value for each item have the same value. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.		""
multiple	Boolean	Specifies whether or not multiple options can be selected. - true: Multiple selections - false: Single selection	No	false
selectedIndex	Number	In case of single selection, - Specifies the selected index (starting from zero) by default. - If -1 is specified, no item is selected. - If an invalid value is specified, an error occurs (RCF12604). - If you specify a UDA to the id property, this property is overwritten by the value depending on the UDA value when the widget is initialized;	No	-1

Name	Data Type	Description	Required	Default Value
		and the value depending on the selected index is saved to the UDA value. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA. In case of multiple selections, - Specifies the last selected index.		
selectedIndexes	Number Array	Specifies an array of selected indexes (starting from zero). Indexes are stored in the order in which items are selected. You can specify the initial selection. If an invalid value is specified, an error occurs (RCF12604). If you specify a UDA to the id property, this property is overwritten by the value depending on the UDA value when the widget is initialized; and values depending on selected indexes are saved to the UDA value in comma separated format. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.	No	[]
xpath	String	This property is used for only a custom data type UDA. Specifies xpath of the targeted data when you specify a custom data type UDA to the id properties. You can specify the xpath of the text node or the attribute node.	Yes	-

selectedValue	String	In case of single selection,
		
		  - Specifies the value of the selected item by default.
		
		  - If an empty string is specified, no item is selected.
		
		  - If an invalid value is specified, an error occurs (RCF12605).
		
		In case of multiple selections,
		
		  - Specifies the value of the last selected item. Values are stored in the order in which items are selected.
			No	""

selectedValues	String Array	Specifies a value array of the selected items. Values are stored in the order in which items are selected. You can specify the initial selection. If an invalid value is specified, an error occurs (RCF12605).	No	[]
size	Number	Specifies the height (the number of rows) of the selection list. You can specify an integer greater than 1.	No	(Set automatically)

Name	Data Type	Description	Required	Default Value
		If you specify a value less than 1, an error occurs.		In case of single selection: 1 In case of multiple selections: 4
title	String	Specifies the text to be displayed in the tooltip.	No	""
enabled	Boolean	Specifies whether or not this widget is enabled. - true: Enabled - false: Disabled You cannot operate this widget.	No	true
tabIndex	Number	Specifies the tab order when moving the focus with the Tab key. You can specify it in the same way as the tabindex attribute in HTML. The focus moves in ascending order if you specify integers greater than zero.	No	0
mandatory	Boolean	Specifies whether or not the entered value is required. - true: Required Error message is displayed in error field when the value is empty. - false: Not required	No	false

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

📝 **Note**

....

- If you specify the prop_options property to relate to a UDA, the available options cannot contain commas since commas are used to separate options.

- If you specified a long character string to the title property, it is automatically wrapped or cut out. If an empty string is specified, nothing is displayed.

....

Relations between the multiple property and the selected state properties

The behavior of the following properties is different depending on the value of the 'multiple' property:

- selectedIndex

- selectedIndexes

- selectedValue

- selectedValues

As a rule, you have to use these properties as follows:

- In case of single selection (multiple=false)

    - selectedIndex

    - selectedValue

If selectedIndex and selectedValue are specified together, selectedIndex takes precedence.

- In case of multiple selections (multiple=true)

    - selectedIndexes

    - selectedValues

If selectedIndexes and selectedValues are specified together, selectedIndexes takes precedence.

Style Properties

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-Select	- Width - Color - Font (except lineHeight)

Refer to section 2.6 Style Properties for details.

Additional Notes

- About multiple selections
  The operations for making multiple selections using the keyboard keys and the mouse are as follows by default:

    To add or cancel individual selections,

    - Click an item while pressing the Ctrl key
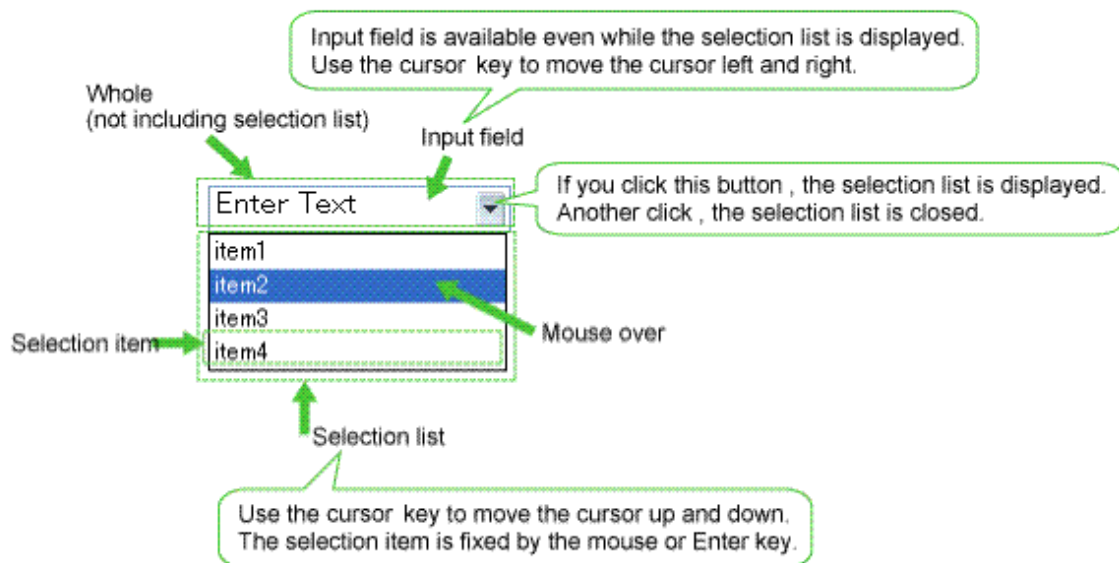
    To select an area,

    - Press the cursor key while pressing the Shift key

    - Drag the mouse

2.1.7 ComboBox

ComboBox is the widget for displaying a combo box, which consists of an input field and a list of items from which a single item can be selected.

Sample Display

You can undo using Ctrl+Z. Refer to section 4.1.6 Undo Using Ctrl+z in Text Input Widgets for details.

 Note

. . . .

In cases where you place a ComboBox at the bottom of a form, or you set a large number of selection items in the list without setting the height for the widget, the ComboBox may not fit inside the form boundaries.
If this happens, depending on the browser, the user cannot see the entire list and the only way for the user to make selections is by using the keyboard.

For this reason, always make sure that the ComboBox widget is set to fit completely inside the form boundaries.
. . . .

Format

```
<div rcf:type="IBPMComboBox" ... ></div>
```

 Note

. . . .

You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements for details.
. . . .

 Point

. . . .

Line feed is inserted before and after the widget.
. . . .

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
xpath	String	This property is used for only a custom data type UDA. Specifies xpath of the targeted data when you specify a custom data type UDA to the id properties. You can specify the xpath of the text node or the attribute node.	Yes	-
list	String Array	Specifies the values of items to display in the list. The contents of the array are Strings. If you specify a UDA to the prop_list property, this property is overwritten by the UDA value when the widget is initialized.	It is necessary to set the value of either the list property or the prop_list property.	[]
prop_list	String	Specifies a String type UDA in specific format. This UDA value has the available options which are separated by comma.		""

Name	Data Type	Description	Required	Default Value
		Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.		
buttonImage	-	Cannot be specified in Interstage BPM Studio.	-	-
fixButtonImageSize	Boolean	Specifies whether or not the size of the drop down button image is fixed. - true: Fixed - false: Resized depend on the widget size Refer to Example for fixButtonImageSize property for details.	No	false
buttonImage Size	-	Cannot be specified in Interstage BPM Studio.	-	-
mandatory	Boolean	Specifies whether or not the entered value is required. - true: Required Error message is displayed in error field when the value is empty. - false: Not required	No	false

You can also specify TextInput properties. Refer to Properties in section 2.1.2 TextInput for details.
You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

Note

- If you specify the prop_list property to relate to a UDA, the available options cannot contain commas since commas are used to separate options.

- If you specified a long character string to the title property, it is automatically wrapped or cut out. If an empty string is specified, nothing is displayed.

Point

- If the maxLength property is specified, character strings in the selection list must be no longer than the maximum number of characters specified in maxLength.

- If the value of the uppercase property is "true", letters of the Roman alphabet in the selection list are converted to uppercase for display.

Example for fixButtonImageSize property

The size of the drop down button image is different depending on the fixButtonImageSize property.

- fixButtonImageSize=false

  If you change the width style property and the height style property, the size of the dropdown button image (in pixels) is changed as follows:

  - Width: width/10(omit decimals) px

  - Height: height-2 px

- fixButtonImageSize=true

  If you change the width style property and the height style property, the size of the drop-down button image (in pixels) is not changed.

Style Properties

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-ComboBox	- Size - Border
Input field	input	rcf-ComboBox-input	- Color - Font - Text (except textIndent, textAlign and whiteSpace)
Selection list	list	rcf-ComboBox-list	- Color - Font - Text (except textIndent, textAlign and whiteSpace) - Border - height for size
Selection item without the cursor	item	rcf-ComboBox-item	- Border
Selection item with the cursor	itemHovered	rcf-ComboBox-itemHovered	- Color

Refer to section 2.6 Style Properties for details.

Additional Notes

- About combo box size:
  The height and width of the widget are specified in pixel units ("px"). If you use any other units for these settings, widgets may not be correctly displayed.

- Selecting value from a list:
  When you use the mouse to display the list, you cannot drag the mouse over items in the list in the same movement in order to select an item. After displaying the list, you must click the item once more to select it.

2.1.8 DateInput

DateInput is the widget to input and edit the date.

Sample Display

May, 11 2010 19:32:26

You can undo using Ctrl+Z. Refer to section 4.1.6 Undo Using Ctrl+z in Text Input Widgets for details.

Format

```
<div rcf:type="IBPMDateInput" ... ></div>
```

Or

```
<span rcf:type="IBPMDateInput" ... ></span>
```

Note

You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements for details.

Point

This widget is displayed in the following way:

- Using with <div> tag, line feed is inserted before and after the widget.

- Using with <span> tag, line feed is not inserted before and after the widget.

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
date	-	Cannot be specified in Interstage BPM Studio. You can specify a UDA in the id property to define the date and time. The value entered in the input field is saved to the UDA value. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.	-	-
xpath	String	This property is used for only a custom data type UDA. Specifies xpath of the targeted data when you specify a custom data type UDA to the id properties. You can specify the xpath of the text node or the attribute node.	Yes	-
dateFormat	String	This property is used for only a custom data type UDA.	Yes	-

Name	Data Type	Description	Required	Default Value
		Specifies date format of the targeted data when you specify a custom data type UDA to the id properties. You can specify any of the below date formats : 1. dd.MM.yyyy 2. dd-MM-yyyy 3. MM.dd.yyyy 4. MM/dd/yy 5. MM/dd/yyyy 6. yyyy-MM-dd 7. MM-dd-yyyy 8. MM dd yyyy 9. yyyy MM dd 10. YYYY/MM/dd 11. yyyy/MM/dd hh:mm:ss 12. dd/MM/yyyy 13. yyyy.MM.dd If a target UDA value is "03.21.2010",you specify the following format. specified format : MM.dd.yyyy.		
utc	-	Cannot be specified in Interstage BPM Studio.	-	-
converter	-	Cannot be specified in Interstage BPM Studio.	-	-
refocus	Boolean	Specifies whether or not the focus is returned to the widget when the conversion from the input text into the Date object fails. - true: Focus is returned - false: Focus is not returned	No	false
mandatory	Boolean	Specifies whether or not the entered value is required. - true: Required Error message is displayed in error field when the value is empty. - false: Not required	No	false

You can also specify TextInput properties. Refer to Properties in section 2.1.2 TextInput for details.

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

....

When you select the DateInput widget in the Palette View, and you drop this widget to the Ajax Page Editor, the following widgets are defined together.

- DateInput,

- PopupCalendar

- CalendarButton

May, 11 2010 19:32:26

If you want to relate the UDA to DateInput widget, you need to define "uda_"+<this UDA identifier> to the following properties.

- id property of DateInput.

- targetDateInputId property of PopupCalendar.

....

....

- If you specified a long character string to the title property, it is automatically wrapped or cut out. If an empty string is specified, nothing is displayed.

- Input text is only converted to a JavaScript Date object when the input field loses the focus and its value is different from the previous one.

- The value property cannot be specified in Interstage BPM Studio.

- The refocus property is to get back the focus when the conversion from the input text into the Date object fails. However, it may not work correctly in the following case:

    - When moving the focus to area outside a page (such as the browser address bar or a toolbar). For example, when clicking the address bar or displaying the address bar dropdown list, focus may be shifted away from DateInput.

....

....

If you are using DATE UDA and enter "January 01, 1970 00:00:00" as the default value while adding the UDA, the value of date displayed in the form will be empty.

If you keep DateInput control empty, the value stored in the mapped DATE type UDA will be "January 01, 1970 00:00:00".

....

Style Properties

The style properties for this widget are the same as for TextInput. Refer to Style Properties in section 2.1.2 TextInput for details. However, note that the class name is "rcf-IBPMDateInput". Refer to section 2.6 Style Properties for details.

2.1.9 NumberInput

NumberInput is the widget to display an input field for integers and real numbers.

Sample Display

123.456

You can undo using Ctrl+Z. Refer to section 4.1.6 Undo Using Ctrl+z in Text Input Widgets for details.

Format

```
<div rcf:type="IBPMNumberInput" ... ></div>
```

Or

```
<span rcf:type="IBPMNumberInput" ... ></span>
```

Note

. . . .

You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements for details.

. . . .

Point

. . . .

This widget is displayed in the following way:

- Using with <div> tag, line feed is inserted before and after the widget.

- Using with <span> tag, line feed is not inserted before and after the widget.

. . . .

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
number	Number	Specifies numerical value data. If you specify a UDA to the id property, this property is overwritten by the UDA value when the widget is initialized. And the value entered in the input field is saved to the UDA value. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.	No	NaN
xpath	String	This property is used for only a custom data type UDA. Specifies xpath of the targeted data when you specify a custom data type UDA to the id properties. You can specify the xpath of the text node or the attribute node.	Yes	-
converter	-	Cannot be specified in Interstage BPM Studio.	-	-
refocus	Boolean	Specifies whether or not the focus is returned to the widget when the conversion from the	No	false

Name	Data Type	Description	Required	Default Value
		input text into the Date object fails. - true: Focus is returned - false: Focus is not returned		
mandatory	Boolean	Specifies whether or not the entered value is required. - true: Required Error message is displayed in error field when the value is empty. - false: Not required	No	false
minimumValue	Number	Specifies the minimum value for the input field.	No	""
maximumValue	Number	Specifies the maximum value for the input field.	No	""

You can also specify TextInput properties. Refer to Properties in section 2.1.2 TextInput for details.

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

Note

・・・

- If you specified a long character string to the title property, it is automatically wrapped or cut out.

- If an empty string is specified, nothing is displayed.

- Input text is only converted to a JavaScript number object when the input field loses the focus and its value is different from the previous one.

- The value property cannot be specified in Interstage BPM Studio.

- The refocus property is to get back the focus when the conversion from the input text into the number object fails. However, it may not work correctly in the following case:
  When moving the focus to area outside a page (such as the browser address bar or a toolbar). For example, when clicking the address bar or displaying the address bar dropdown list, focus may be shifted away from NumberInput.

- In Studio, if you define the minimumValue property or maximumValue property, these values are not validated. Therefore:

  - Please enter only number.

  - If you define both the values, please set a value that is more than the minimumValue up to the maximumValue.

・・・

Style Properties

The style properties for this widget are the same as for TextInput. Refer to Style Properties in section 2.1.2 TextInput for details. However, note that the class name is "rcf-IBPMNumberInput".
Refer to section 2.6 Style Properties for details.

2.1.10 SelectList

SelectList is the widget to display a list from which single or multiple items can be selected.

Sample Display

Format

```
<div rcf:type="IBPMSelectList" ... ></div>
```

🈁 **Note**

. . . .

You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements for details.

. . . .

🅿 **Point**

. . . .

Line feed is inserted before and after the widget.

. . . .

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
options	Array	Specifies the available value options in the list. If you specify a UDA to the prop_options property, this property is overwritten by the UDA value when the widget is initialized.	It is necessary to set the value of either the options property or the prop_options property.	[]
prop_options		Specifies a String type UDA in specific format. This UDA value has the available value options which are separated by comma. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.		""
labelProvider	-	Cannot be specified in Interstage BPM Studio.	-	-
multiple	Boolean	Specifies whether or not multiple options can be selected. - true: Multiple selections - false: Single selection	No	false
selectedIndex	Number	In case of single selection, - Specifies the selected index (starting from zero) by default.	No	-1

Name	Data Type	Description	Required	Default Value
		- If -1 is specified, no item is selected. - If an invalid value is specified, an error occurs (RCF13302). - If you specify a UDA to the id property, this property is overwritten by the value depending on the UDA value when the widget is initialized; and the value depending on the selected index is saved to the UDA value. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA. In case of multiple selections, - Specifies the last selected index.		
selectedIndexes	Number Array	Specifies an array of selected indexes (starting from zero). Indexes are stored in the order in which items are selected. You can specify the initial selection. If an invalid value is specified, an error occurs (RCF13302). If you specify a UDA to the id property, this property is overwritten by the value depending on the UDA value when the widget is initialized. And values depending on selected indexes are saved to the UDA value in a comma-separated format. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.	No	[]
xpath	String	This property is used for only a custom data type UDA. Specifies xpath of the targeted data when you specify a custom data type UDA to the id properties. You can specify the xpath of the text node or the attribute node.	Yes	-
title	String	Specifies the text to be displayed in the tooltip.	No	""
enabled	Boolean	Specifies whether or not this widget is enabled. - true: Enabled - false: Disabled You cannot operate this widget.	No	true
tabIndex	Number	Specifies the tab order when moving the focus with the Tab key. You can specify it in the same way as the tabindex attribute in HTML. The focus	No	0

Name	Data Type	Description	Required	Default Value
		moves in ascending order if you specify integers greater than zero.		
renderer	-	Cannot be specified in Interstage BPM Studio.	-	-
mandatory	Boolean	Specifies whether or not the entered value is required. - true: Required Error message is displayed in error field when the value is empty. - false: Not required	No	false

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

🈁 Note

. . . .

- If you specify the prop_options property to relate to a UDA, the available options cannot contain commas since commas are used to separate options.

- If you specified a long character string to the title property, it is automatically wrapped or cut out. If an empty string is specified, nothing is displayed.

. . . .

🅿 Point

. . . .

- In case of single selection,

    - Only one item can be selected at any one time. You need to use the selectedIndex property. If no value is specified, the first item in the list is selected by default.

    - You can change the selected item by clicking on the item.

    - You can use the vertical cursor key to move the selection between items.

- In case of multiple selections,

    - You need to use the selectedIndexes property. If no value is specified, no item in the list is selected by default.

    - You can add selections by clicking on the item. If you click a selected item, you clear the selection.

    - You can use the vertical cursor key to move the focus between items (mouseover items) and the space key to select/deselect items.

. . . .

Style Properties

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-SelectList	- Size - Color - Font (except lineHeight) - Text (except textIndent, textAlign and whiteSpace) - Border - Padding

Refer to section 2.6 Style Properties for details.

Style of selected items

Prefix	Class	Style Properties	Description	Default
optionSelected (selected item)	rcf-SelectList-optionSelected	backgroundColor	Specifies the background color of the selected item.	#004E98
		color	Specifies the font color of the selected item.	#FFFFFF(white)
optionHovered (mouseover item)	rcf-SelectList-optionHovered	backgroundColor	Specifies the background color of the mouseover item.	#316AC5
		color	Specifies the font color of the mouseover item.	#FFFFFF(white)

Relations between display and style properties

The style properties for this widget are the same as for CheckList. However, no check boxes are displayed. Refer to Style Properties in section 2.1.11 CheckList for details.

2.1.11 CheckList

CheckList is the widget to display a list with check button items from which single or multiple items can be selected.
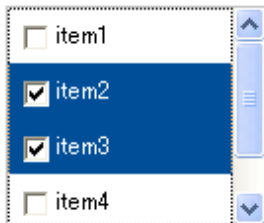
Sample Display

Figure 2.3 Single Selection Mode

In single selection mode, radio buttons are displayed.

Figure 2.4 Multiple Selections Mode

In multiple selections mode, check boxes are displayed.

Format

```
<div rcf:type="IBPMCheckList" ... ></div>
```

Note

You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements for details.

Point

Line feed is inserted before and after the widget.

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
options	Array	Specifies the available value options in the list. If you specify a UDA to the prop_options property, this property is overwritten by the UDA value when the widget is initialized.	It is necessary to set the value of either the options property or the prop_options property.	[]
prop_options		Specifies a String type UDA in specific format. This UDA value has the available value options which are separated by comma. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.		""
labelProvider	-	Cannot be specified in Interstage BPM Studio.	-	-
multiple	Boolean	Specifies whether or not multiple options can be selected. - true: Multiple selections - false: Single selection	No	false
selectedIndex	Number	In case of single selection, - Specifies the selected index (starting from zero) by default. - If -1 is specified, no item is selected. - If an invalid value is specified, an error occurs (RCF13402). - If you specify a UDA to the id property, this property is overwritten by the value depending on the UDA value when the widget is initialized; and the value depending on the selected index is saved to the UDA value. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA. In case of multiple selections,	No	-1

Name	Data Type	Description	Required	Default Value
		- Specifies the last selected index.		
selectedIndexes	Number Array	Specifies an array of selected indexes (starting from zero). Indexes are stored in the order in which items are selected. You can specify the initial selection. If an invalid value is specified, an error occurs (RCF13402). If you specify a UDA to the id property, this property is overwritten by the value depending on the UDA value when the widget is initialized. And values depending on selected indexes are saved to the UDA value in a comma-separated format. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.	No	[]
xpath	String	This property is used for only a custom data type UDA. Specifies xpath of the targeted data when you specify a custom data type UDA to the id properties. You can specify the xpath of the text node or the attribute node.	Yes	-
title	String	Specifies the text to be displayed in the tooltip.	No	""
enabled	Boolean	Specifies whether or not this widget is enabled. - true: Enabled - false: Disabled You cannot operate this widget.	No	true
tabIndex	Number	Specifies the tab order when moving the focus with the Tab key. You can specify it in the same way as the tabindex attribute in HTML. The focus moves in ascending order if you specify integers greater than zero.	No	0
mandatory	Boolean	Specifies whether or not the entered value is required. - true: Required Error message is displayed in error field when the value is empty. - false: Not required	No	false

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

🖐 **Note**
....

- If you specify the prop_options property to relate to a UDA, the available options cannot contain commas since commas are used to separate options.

- If you specified a long character string to the title property, it is automatically wrapped or cut out. If an empty string is specified, nothing is displayed.

P Point

- You can specify the initial selections to the selectedIndex property. If no value is specified, no item in the list is selected by default.

- In case of single selection, you can use the vertical cursor key to move the selection between items.

- In case of multiple selections, you can use the vertical cursor key to move the focus between items.

Style Properties

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-CheckList	- Size - Color - Font (except lineHeight) - Text (except textIndent, textAlign and whiteSpace) - Border - Padding

Refer to section 2.6 Style Properties for details.

Style of selected items

Prefix	Class	Style Properties	Description	default
optionSelected (selected item)	rcf-CheckList-optionSelected	backgroundColor	Specifies the background color of the selected item.	#004E98
		color	Specifies the font color of the selected item.	#FFFFFF (white)
optionHovered (mouseover item)	rcf-CheckList-optionHovered	backgroundColor	Specifies the background color of the mouseover item.	#316AC5
		color	Specifies the font color of the mouseover item.	#FFFFFF (white)

Figure 2.5 Relation of Display and Style Properties
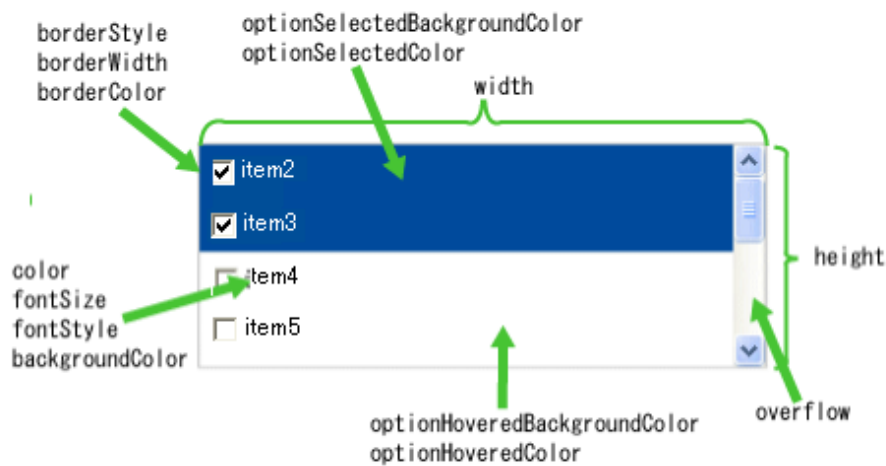
If the size of the items is too large for the display area, the widget is displayed in the following way:

- If the width of item is greater than the width of the display area, item beyond the right margin is hidden. The horizontal scrollbar is not displayed.

- If the height of items is greater than the height of the display area, the vertical scrollbar is displayed to allow users to view all items.

Additional Notes

- Moving the focus with the Tab key
  In case of single selection, the focus is moved to the radio button within the widget when you press the Tab key to move the focus. To move the focus to the next widget, press the Tab key again.

2.2 Container Widgets

Container widgets are widgets for placing multiple screen widgets.

This section explains how to use container widgets.

2.2.1 ViewContainer

ViewContainer is the widget for specifying screen information (container).

ViewContainer can contain HTML elements in child elements, and the whole can be handled as one widget. ViewContainer is used in ViewStack and TabPanel to specify views.

Format

```
<div rcf:type="IBPMViewContainer" ... >
    HTML element
</div>
```

P Point

You can specify HTML elements and UI widgets in child elements.

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
label	String	Specifies the tab name when used as a child element of the tab panel.	No	""

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

Style Properties

| Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-ViewContainer	- Size - Color - Border - Padding - Overflow

Refer to section 2.6 Style Properties for details.

Additional Notes

- ViewContainer is used with ViewStack or TabPanel.

- If ViewContainer is used without ViewStack or TabPanel, it is drawn in the same way as the <div> element.

- You can add only a widget of Advanced Category to a ViewContainer.

- You cannot add a widget of Basic Category to a ViewContainer.

2.2.2 Panel

Panel is the widget to display a container with a title bar. This consists of title part and body part.

Sample Display
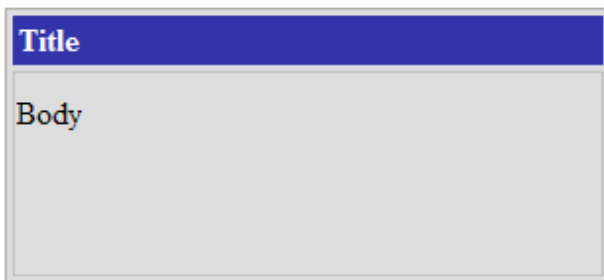
Format

```
<div rcf:type="IBPMPanel" ... >
    HTML element
</div>
```

🅿 Point

You can specify HTML elements and UI widgets in child elements. These widgets are used as contents of the body part.

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
label	String	Specifies the label text for the title part. If you specify a UDA to the id property, this property is overwritten by the UDA value when the widget is initialized. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.	No	""

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

Style Properties

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-Panel	- Size (*) - Color - Border - Padding
title part	title	rcf-Panel-title	- Size (except width) - Color - Font - Text (except white-space) - Border - Padding - Margin
body part	body	rcf-Panel-body	- Color - Font - Text - Border - Padding - Overflow

(*) The default width is 300px.

Refer to section 2.6 Style Properties for details.

Additional Notes

About Panel size

- The width and height is specified in the style properties. It may not be displayed correctly if the width and height are specified in CSS.

- The width and height are specified in pixel units ("px"). If you use any other units for these settings, widgets may not be correctly displayed.
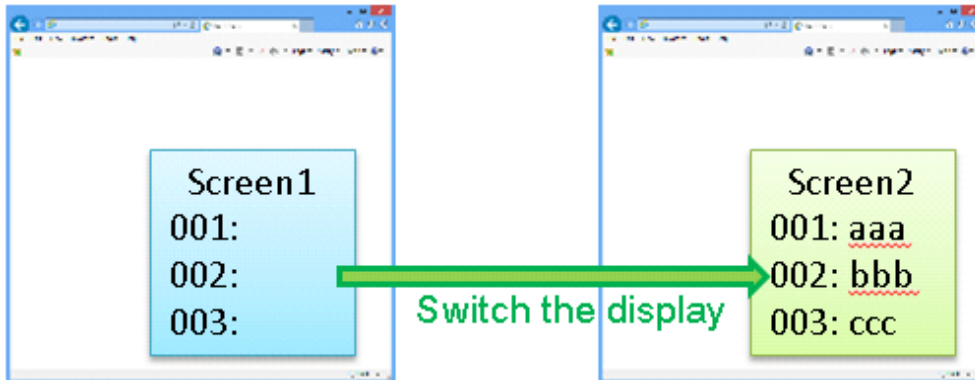
- The minimum size for elements in the body part is 100px x 100px. If you attempt to specify a smaller size, 100px x 100px is set.

2.2.3 ViewStack

ViewStack is the widget for grouping interchangeable screen information (container). ViewStack is used to switch between containers in the same location.

You can use ViewContainer or FragmentContainer for the container. Describe the information (HTML content) of each screen as child elements of the container.

Figure 2.6 Behavior of ViewStack

Format

```
<div rcf:type="IBPMViewStack" ... >
  <div rcf:type="IBPMViewContainer">
    HTML element
  </div>
  <div rcf:type="IBPMFragmentContainer" rcf:src="about:blank">
  </div>
  ...
</div>
```

 Point
. . . .

- You have to define at least one child element.

- You can define ViewContainer or FragmentContainer in the child element.

- If you define anything other than ViewContainer or FragmentContainer in the child element, it cannot be displayed.
. . . .

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
selectedIndex	Number	Specifies the selected index by default. The index increments automatically by 1, starting at 0, in the order in which the containers defined as child elements appear. If you specify an index that does not exist, all containers are	No	0

Name	Data Type	Description	Required	Default Value
		invisible (visibility:hidden; display:none). If you specify a UDA to the id property, this property is overwritten by the UDA value when the widget is initialized. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.		
selectedIndexLock	-	Cannot be specified in Interstage BPM Studio.	-	-

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

Style Properties

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-ViewStack	- Size - Color - Border - Padding - Overflow

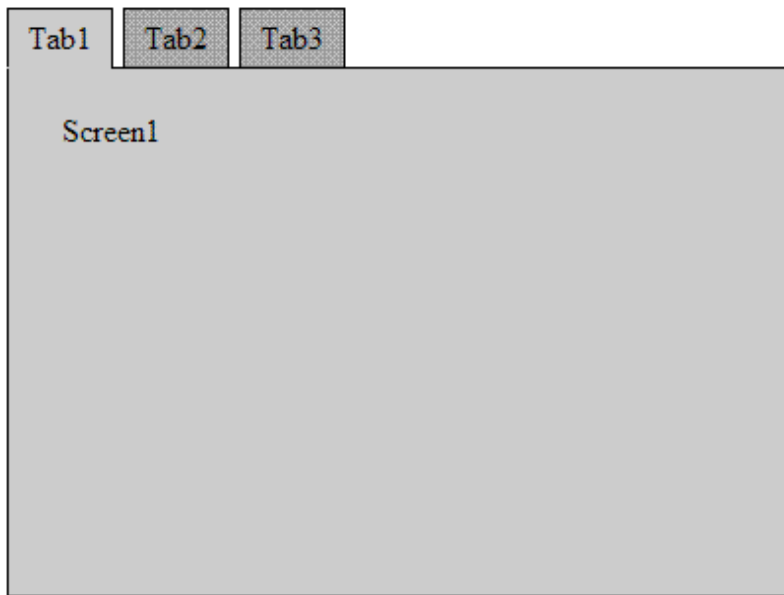Refer to section 2.6 Style Properties for details.

Additional Notes

If the size of the content to display in the ViewStack widget (including the title part) exceeds the size of the whole widget, the widget may not be displayed correctly. In such a case, set the size of the whole widget so that it is large enough for all content.

2.2.4 TabPanel

TabPanel is the widget for grouping tab-switched screen information (container). TabPanel is used to switch the display by clicking tabs.

When the tab or a form widget in this widget is focused, you can switch tabs by keyboard operation. You can also customize keys for tab switching.

Sample Display

Format

```
<div rcf:type="IBPMTabPanel" ... >
  <div rcf:type="IBPMViewContainer" rcf:label="tab 1">
    HTML element
  </div>
  <div rcf:type="IBPMFragmentContainer" rcf:label="tab 2" ... >
  </div>
  ...
</div>
```

P Point

. . . .

- You have to define at least one child element.

- You can define ViewContainer or FragmentContainer in the child element.

- If you define anything other than ViewContainer or FragmentContainer in the child element, it cannot be displayed.

. . . .

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
selectedIndex	Number	Specifies the selected index by default. The index increments automatically by 1, starting at 0, in the order in which the containers defined as child elements appear. If you specify an index that does not exist, all containers are invisible (visibility:hidden; display:none). If you specify a UDA to the id property, this property is overwritten by the UDA	No	0

| Name	Data Type	Description	Required	Default Value
		value when the widget is initialized. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.		
tabPosition	String	Specifies the tab location.	No	TOP_LEFT
nextKey	String	Specifies the key for switching to the next tab. Refer to section Specifying keys in 2.2.4 TabPanel for details.	No	Ctrl+right cursor key ("39+CTRL")
previousKey	String	Specifies the key for switching to the previous tab. Refer to section Specifying keys in 2.2.4 TabPanel for details.	No	Ctrl+left cursor key ("37+CTRL")
tabSeparator	Boolean	Specifies whether or not to leave a gap between tabs. - true: Gapped - false: Not gapped.	No	true
tabRenderer	-	Cannot be specified in Interstage BPM Studio.	-	-
tabIndex	Number	Specifies the tab order when moving the focus with the Tab key. You can specify it in the same way as the tabindex attribute in HTML. The focus moves in ascending order if you specify integers greater than zero.	No	0
selectedIndexLock	Boolean	Specifies whether or not the switching view is locked. - true: Locked You cannot switch the view. - false: Not locked	No	false

You can also specify ViewContainer properties. Refer to Properties in section 2.2.1 ViewContainer.

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

Specifying basic tab position

Using the tabPosition property, you can specify the location of the first tab and the direction in which subsequent tabs are located.

- This property is not case-sensitive.

- In the diagram below, the blue dotted lines represent the direction in which subsequent tabs are located.

- Regardless of the direction of tabs, the text in the tab label always runs from left to right.
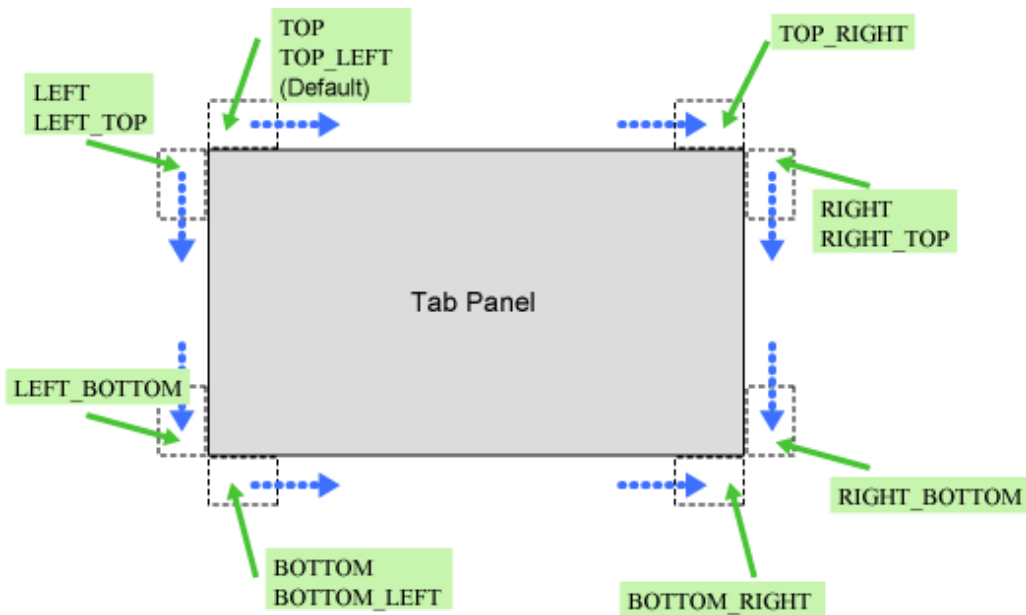
Specifying keys

Using the nextKey property and previousKey property, you can specify a key to switch the selected tab.
You can specify the following keys separated by a plus sign (+):

- Number or alias: Key code

- ALT: Alt key

- CTRL: Ctrl key

- SHIFT: Shift key

The key code is required, and must be specified first.

The table below contains examples.

Example	Description
13	Enter key
13 + SHIFT	Enter key + Shift key
13 + CTRL + SHIFT	Enter key + Ctrl key + Shift key
SHIFT	Invalid (no key code)
CTRL+13	Invalid (key code is not specified first)

You can specify the key code number or the alias listed in the table below. The resulting behavior is the same.
For example, "ENTER+SHIFT" is handled in the same way as "13+SHIFT".

Alias	Label on Key	Actual Key Code
BACKSPACE	Backspace key	8
TAB	Tab key	9
ENTER	Enter key	13
ESC	Esc key	27
SPACE	Space key	32
PAGEUP	Page Up key	33
PAGEDOWN	Page Down key	34

Alias	Label on Key	Actual Key Code
END	End key	35
HOME	Home key	36
LEFT	Left cursor key	37
UP	Up cursor key	38
RIGHT	Right cursor key	39
DOWN	Down cursor key	40
INSERT	Insert key	45
DELETE	Delete key	46
F1	F1 key	112
F2	F2 key	113
F3	F3 key	114
F4	F4 key	115
F5	F5 key	116
F6	F6 key	117
F7	F7 key	118
F8	F8 key	119
F9	F9 key	120
F10	F10 key	121
F11	F11 key	122
F12	F12 key	123

Style Properties

Width and height

Specify the TabPanel width and height using the height and width properties. Each property includes the tab height (width).
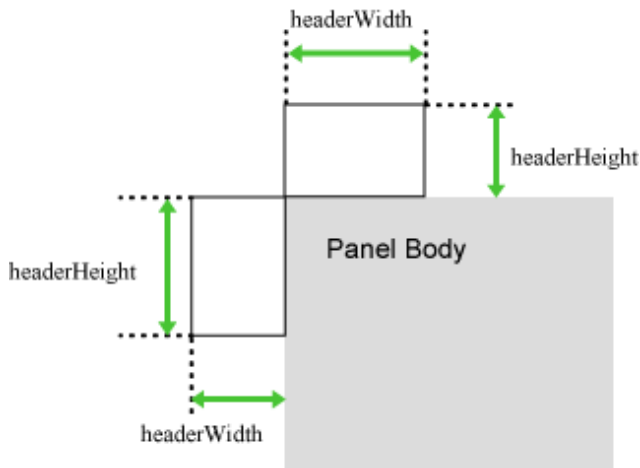
By default, the width is 100% and the height is auto.
When the height is auto, the tab panel height is changed to match the height of ViewContainer or FragmentContainer displayed in the panel.
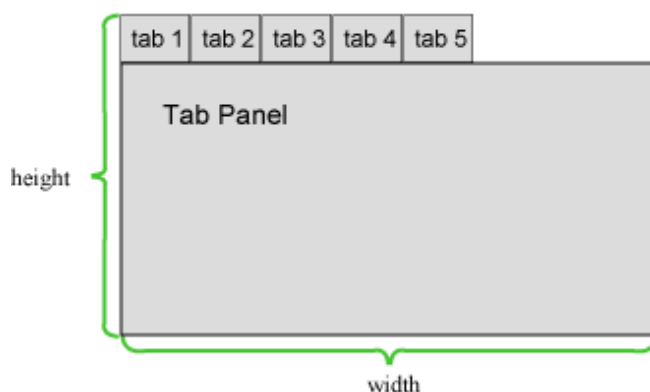

Specifying tab size

Regardless of the tab location, the default value of the headerWidth and headerHeight properties is auto.

If tabs are placed on the left or right side, the tab width depends on the length of the longest tab label.

If tabs are placed at the top or bottom, the tab width is individually different depending on the length of each tab label.

Tab ornamentation properties
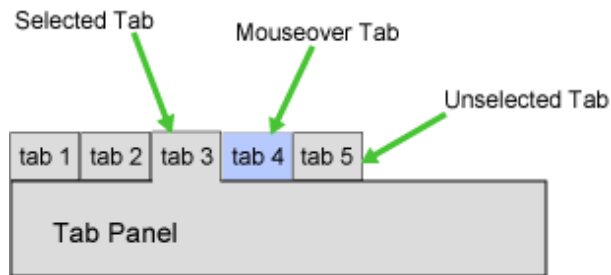
Style Properties

Part Name	Prefix	Class Name	Style Classification	Description
Whole widget	none	rcf-TabPanel	- Size	Size of whole widget
Header	header	rcf-TabPanel-header	- Size	Size of tab part Tab height and width can be specified, depending on tab location
Body	body	rcf-TabPanel-body	- Color - Border - Padding	Body part
Tab	tab	rcf-TabPanel-tab	- Color - Font - Text - Border - Padding	Individual tab style
Selected tab	tabSelected	rcf-TabPanel-tabSelected	- Color - Font	Selected tab style

Part Name	Prefix	Class Name	Style Classification	Description
			- Text - Border - Padding	
Mouseover tab	tabHovered	rcf-TabPanel-tabHovered	- Color - Font - Text - Border - Padding	Mouseover tab style
Separator	tabSeparator	rcf-TabPanel-tabSeparator	- Size - Border	Distance between tabs TOP/BOTTOM: width LEFT/RIGHT: height
Blank	tabBlank	rcf-TabPanel-tabBlank	- Border - Padding	Tabbed side style except tab header

Refer to section 2.6 Style Properties for details.

When selected tab and mouseover tab overlap

When the selected tab and mouseover tab overlap, the order in the tab style class is tabSelected, tabHovered.

Styles only available in CSS

tab, tabSelected, tabHovered, tabSeparator, and tabBlank can only be specified in CSS.

Notes on customizing

The default style settings of the body and tab are different depending on whether tabs are placed. The default style settings are as follows:

- Body

    - Tabbed side
      border-style: none

    - Untabbed sides
      border-style: solid

- Tab

    - All sides
      border-color: #000
      However, if the tab is selected and the mouse is not over the tab, the border color of the tab on the side adjoining the body is the same as the tab background-color.
      This behavior is the same if you customize the background color.

Additional Notes

About TabPanel size

- The size properties are specified in pixel units ("px"). This is also the same for border width, margins and padding. If you use any other units for these settings, widgets may not be correctly displayed.

- If the size of the content to display in the TabPanel widget (including the title part) exceeds the size of the whole widget, the widget may not be displayed correctly. In such a case, set the size of the whole widget so that it is large enough for all content.

About focus

When the TabPanel is focused using the Tab key, the focus shifts to the header part. The behavior is the same if tabs are switched using keys specified in the nextKey and previousKey properties.

When the header part is focused, the whole header part is focused, not just the header part of an individual tab.

When a widget in the body is focused, it is possible to switch tabs using keys specified in the nextKey and previousKey properties, but when the focus shifts using the Tab key, not the TabPanel but the widget itself is targeted.

About tab switching keys (nextKey, previousKey) settings

- If you specify a global event control key as a tab switching key, tabs may not switch correctly. For this reason, do not specify global event control keys as tab switching keys.

- If you specify a browser shortcut key as a tab switching key, tab switching takes precedence and so the shortcut key process may be cancelled. For this reason, do not specify important shortcut keys as tab switching keys.
  Example:
  If you specify "66+CTRL(Ctrl + B) " as a tab switching key, and press Ctrl + B to execute a widget function. You cannot use Ctrl + B to open "Organize Favorites" in Internet Explorer.

2.2.5 FragmentContainer

FragmentContainer is the widget for specifying screen information (container) in a separate file.

Whereas ViewContainer describes HTML elements as child elements in the same file, FragmentContainer describes HTML elements in a separate file.

Data of HTML in FragmentContainer (referred to hereafter as Fragment HTML) is retrieved by asynchronous communication. Note that Fragment HTML does not conform to a DTD as simple HTML.
Retrieved Fragment HTML are inserted into HTML with FragmentContainer (referred to hereafter as Container HTML), and then the data is reflected.

Examples:

- To improve the speed of initialization
  Use with TabPanel or ViewStack to enable delay loading of container content.

- To disperse development
  Because development resources (files) are divided, development can be dispersed.

Format

```
<div rcf:type="IBPMFragmentContainer" rcf:src="about:blank" ... ></div>
```

Note

You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements for details.

Figure 2.7 FragmentContainer in Use

As with ordinary HTML, Fragment HTML can use UI widgets.

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
src	String	In Interstage BPM Studio, specify "about:blank". You have to specify a UDA to the id property to define the URL of Fragment HTML. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.	Yes	-
showStatus	Boolean	Specifies whether or not the FragmentContainer status before activate (such as Initialized, Loading, and Error) is displayed. - true: Displayed - false: Not displayed	No	true
statusIconImage	-	Cannot be specified in Interstage BPM Studio.	-	-
timeout	Number	Specifies the timeout period for retrieving Fragment HTML. The units are milliseconds. If you specify a value less than 1, an error occurs.	No	60000(60 seconds)

You can also specify ViewContainer properties. Refer to Properties in section 2.2.1 ViewContainer.
You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

Style Properties

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-FragmentContainer	- Size - Color - Border - Padding - Overflow

Refer to section 2.6 Style Properties for details.

Additional Notes

- FragmentContainer is used in ViewStack and TabPanel to specify views.

- If FragmentContainer is used without ViewStack or TabPanel, it is displayed in the same way as an ordinary <div> tag.

- Fragment HTML are added to the HTML body. For this reason, you cannot define HTML header elements in Fragment HTML.

- Before assigning the UDA value to FragmentContainer, the value is first encoded and then assigned.

- For FragmentContainer, you should not change the rcf:src="about:blank" during design time. If you change the value of src property, one of the following messages is displayed:

  - Object expected.

  - 'excArrayObj' is undefined.

Note

- You can add only a widget of Advanced Category to a Fragment HTML.

- You cannot add a widget of Basic Category to a Fragment HTML.

- If you use a Fragment HTML, you have to do the following operations.

  1. Create a file by the QuickForm Wizard Dialog (execute New > QuickForm command).

     -> An Ajax Page Editor is displayed.

  2. Delete lines other than the following lines in the Source View of the Ajax Page Editor.

     ```
     <script type="text/javascript" src="../rcf_config.js" charset="UTF-8"></script>
     <script type="text/javascript" src="../acf/file/rcf/rcf.js" charset="UTF-8"></script>
     <body>
     </body>
     ```

     * Please refer to 8.2 Ajax Page Editor of the Studio User's Guide for Path specified for the src parameter.

  3. Define a widget to this file.

  4. When you upload this application or export this application, you have to delete the following lines from this Fragment HTML in the Source View of the Ajax Page Editor.

     ```
     <script type="text/javascript" src="../rcf_config.js" charset="UTF-8"></script>
     <script type="text/javascript" src="../acf/file/rcf/rcf.js" charset="UTF-8"></script>
     <body>
     </body>
     ```

     * Please refer to 8.2 Ajax Page Editor of the Studio User's Guide for Path specified for the src parameter.

  5. If you update this Fragment HTML again, you have to add the lines deleted in Step 4.

2.3 Table Widgets

Table widgets are widgets for displaying and editing tables.
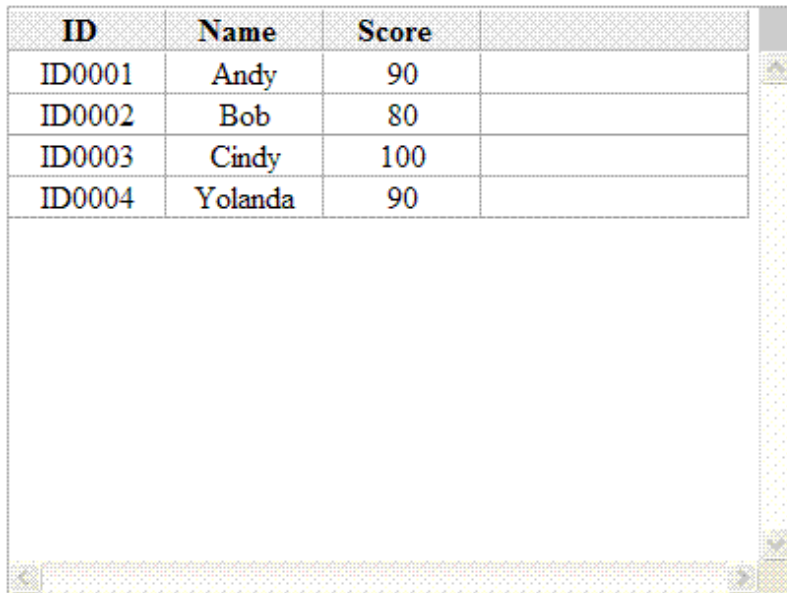
This section explains how to use table widgets.

🅿 Point

TableView and TableEdit are intended for comparatively simple table structures. TableView is used to display tables. TableEdit is used to edit tables. TableView and TableEdit use an XML type UDA and a custom data type UDA.

2.3.1 TableView

TableView is the widget for displaying two-dimensional data in table format.

Sample Display

ID	Name	Score	
ID0001	Andy	90	
ID0002	Bob	80	
ID0003	Cindy	100	
ID0004	Yolanda	90	

Format

```
<div rcf:type="IBPMTableView" rcf:id="uda_XMLVariable">
  <div rcf:type="IBPMViewColumn" rcf:name="ID" rcf:propertyName="id"></div>
  <div rcf:type="IBPMViewColumn" rcf:name="Name" rcf:propertyName="name"></div>
  <div rcf:type="IBPMViewColumn" rcf:name="Score" rcf:propertyName="score"></div>
</div>
```

🅿 Point

- Define ViewColumn in which table information is defined in child elements (optional).

- If ViewColumn is not defined it takes the tag names as column names and displays the table.

- Line feed is inserted before and after the widget.

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
data	-	Cannot be specified in Interstage BPM Studio. You can specify an XML type UDA or a cusom data type UDA in the id property to define the data displayed in the table. Refer to Specifying the id property and display data in 2.3.1 TableView for details. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.	-	-
xpath	String	This property is used for only a custom data type UDA. Specifies xpath of the targeted data when you specify a custom data type UDA to the id properties. You can specify the xpath of the non-leaf node or the entire value(/).	No	/
showColumn Header	Boolean	Specifies whether or not to show the column header. - true: Displayed - false: Not displayed You cannot resize the column width using the mouse. And you cannot click the column header to sort or select the column.	No	true
showRowHe ader	Boolean	Specifies whether or not to show the row header. - true: Displayed - false: Not displayed You cannot click the row header to select the row.	No	false
showDumm yColumn	Boolean	Specifies whether or not dummy columns may be displayed. This is used to fill the blank spaces with empty columns. true: Displayed false: Not displayed	No	true
columnWidt hResizable	Boolean	Specifies whether or not the column width should be resizable. - true: Resizable - false: Not resizable	No	true
defaultColu mnWidth	Number	Specifies the default column width in pixel. The minimum value is 10. If you specify a value less than 10, an error occurs.	No	80

Name	Data Type	Description	Required	Default Value
selectedRows	Array	Specifies the array of row indexes to be selected.	No	[]
multipleSelect	Boolean	Specifies whether or not multiple rows can be selected. - true: Multiple selections - false: Single selection	No	false
tabIndex	Number	Specifies the tab order when moving the focus with the Tab key. You can specify it in the same way as the tabindex attribute in HTML. The focus moves in ascending order if you specify integers greater than zero.	No	0

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

P Point

. . . .

- About sorting rows

    Rows and columns are usually displayed in index order. When you click the column header of a column, all of the display data is sorted using that column as they sort key. Also, when you sort rows, the content of UDA specified in the id property is also sorted.

    On the first click, data is sorted in ascending order, on the second click; data is sorted in descending order, alternating with any subsequent clicks. The focus status and selection status of the row are cleared when sorting.

- About dummy columns

    If the display area of the whole table is larger than the display area required to display the rows and columns that make up the table, you can display dummy columns to fill in the gaps.

    Dummy columns are displayed at the right side of the table.

. . . .

Specifying the id property and display data

Specify an XML type UDA in the id property to display data in the table. XML stored in UDA should be a two level deep XML. XML values only up to 2 levels are supported.

In a custom data type UDA case, you can specify the xpath property to target data. XML values only up to 2 levels from this xpath are supported.

When following XML is stored in UDA (XML type):

```xml
<scores>
    <score>
        <id>ID0001</id>
        <name>Andy</name>
        <score>90</score>
    </score>
    <score>
        <id>ID0002</id>
        <name>Bob</name>
        <score>80</score>
    </score>
    <score>
        <id>ID0003</id>
        <name>Cindy</name>
        <score>100</score>
    </score>
    <score>
        <id>ID0004</id>
        <name>Yolanda</name>
```

```
          <score>90</score>
        </score>
    </scores>
```

In the above example, TableView and three ViewColumn propertyName properties (id, name and score) are specified as follows:

```
<div rcf:type="IBPMTableView" rcf:id="uda_XMLVariable" rcf:selectedRows=""
rcf:multipleSelect="false">
  <div rcf:type="IBPMViewColumn" rcf:name="ID" rcf:propertyName="id"></div>
  <div rcf:type="IBPMViewColumn" rcf:name="Name" rcf:propertyName="name"></div>
  <div rcf:type="IBPMViewColumn" rcf:name="Score" rcf:propertyName="score"></div>
</div>
```

Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.

Each item in XML tag is set to cell as follows:

ID	Name	Score	
ID0001	Andy	90	
ID0002	Bob	80	
ID0003	Cindy	100	
ID0004	Yolanda	90	

When ViewColumn is not specified in a TableView child element

When ViewColumn is not specified in a TableView child element, the column not specified is not displayed.

Resizing column width

You can only use the mouse to resize column width by dragging within the bounds of the browser window.

If you release the mouse button after dragging outside the bounds of the browser window, it is assumed that you are still dragging TableView, and there is no change to the display.
To change the display in such a case, click inside the browser window. The column width is then resized.

About focus

When TableView is focused, the whole of TableView is focused, and you can use the keyboard to manipulate TableView. You can use the following methods to shift focus to TableView:

- By pressing the Tab key or Tab+Shift keys provided by the browser

- By directly clicking TableView with the mouse.

When TableView is focused, an outline is displayed around the TableView frame. Also, the focus shifts to the rows below and you can use the keyboard to shift the focus to other rows in TableView.

- When TableView is focused using the Tab key
  The last focused row in TableView is focused. If this is the first time that TableView is focused, focus shifts to the first row.

- When TableView is focused by mouse click

    If a row is clicked, that row is focused. If anything other than a row was clicked, as above, the last focused row in TableView is focused. If this is the first time that TableView is focused, the focus shifts to the first row.

Note

- When you click the TableView scrollbar, TableView loses focus.

- If the TableView data is empty, focus does not shift to a row.

Mouse operations

Operation	Process
Clicking on a row	- Single Selection Mode: Selects the clicked row. Even if a cell in a dummy column is clicked, it is not selected. - Multiple Selection Mode: Clears any other selection status and selects only the clicked row. Even if a cell in a dummy column is clicked, it is not selected.
Clicking on a row + Shift	- Single Selection Mode: Selects the clicked row. Even if a cell in a dummy column is clicked, it is not selected. - Multiple Selection Mode: Selects all rows from the previously selected or deselected row to the clicked row. If no row was previously selected, selects only the clicked row. Even if a cell in a dummy column is clicked, it is not selected.
Clicking on a row + Ctrl	- Single Selection Mode: Selects the clicked row. Even if a cell in a dummy column is clicked, it is not selected. - Multiple Selection Mode: Toggles the selection status of the clicked row. Even if a cell in a dummy column is clicked, it is not selected.
Clicking on a column header	- Single Selection Mode: Sorts based on the clicked column. - Multiple Selection Mode: Clears any other selection status and sorts based on the clicked column.
Clicking on a column header + Shift	- Single Selection Mode: Sorts based on the clicked column. - Multiple Selection Mode: Clears any other selection status and sorts based on the clicked column.
Clicking on a column header + Ctrl	- Single Selection Mode: Sorts based on the clicked column. - Multiple Selection Mode: Clears any other selection status and sorts based on the clicked column.
Clicking on a row header	- Single Selection Mode: Selects the clicked row. - Multiple Selection Mode: Clears any other selection status and selects the clicked row.
Clicking on a row header + Shift	- Single Selection Mode: Selects the clicked row. - Multiple Selection Mode: Selects all rows from the previously selected row to the clicked row. If no row was previously selected, selects only the clicked row.
Clicking on a row header + Ctrl	- Single Selection Mode: Selects the clicked row. - Multiple Selection Mode: Toggles the selection status of the clicked row.

Keyboard operations

Keyboard operations are only active when TableView is focused.

Operation	Process
Space	Single and Multiple Selection Modes: Selects the focused row.
Space + Shift	- Single Selection Mode: Selects the focused row. - Multiple Selection Mode: Selects all rows from the previously selected or deselected row to the focused row.
Space + Ctrl	- Single Selection Mode: Selects the focused row. - Multiple Selection Mode: Toggles the selection status of the focused row.
Up cursor	Single and Multiple Selection Modes: Shifts the focus up one row.
Down cursor	Single and Multiple Selection Modes: Shifts the focus down one row.
Left cursor	Single and Multiple Selection Modes: If the scrollbar is displayed, scrolls to the left.
Right cursor	Single and Multiple Selection Modes: If the scrollbar is displayed, scrolls to the right.
Home	Single and Multiple Selection Modes: Shifts the focus to the first row.
End	Single and Multiple Selection Modes: Shifts the focus to the last row.
Page up	Single and Multiple Selection Modes: If the scrollbar is displayed, scrolls up.
Page down	Single and Multiple Selection Modes: If the scrollbar is displayed, scrolls down.

Style Properties

Table styles

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-IBPMTableView	- Size
Column header	ccell	rcf-IBPMTableView-ccell	- Font (except lineHeight) - Cell
Cell	cell	rcf-IBPMTableView-cell	- Color - Font (except lineHeight) - Border (except borderWidth) - Cell

Point

- About specifying table width and height

    - The width and height are specified in pixel units ("px"). If you use any other units for these settings, widgets may not be correctly displayed.

    - If you specify a small table width and height, the scrollbar may not be displayed and items may not be displayed correctly. For this reason, it is recommended to specify at least 100px for both the width and height.

    - If line feeds are inserted automatically in the column header cell content and it spans multiple rows, the column header height is automatically adjusted. As a result, the actual table height may be more than the specified height.

    - The default table width is 400px. The default table height is 300px.

- About cell content display

  By default cell content is displayed without line feeds. To automatically insert line feeds, specify normal in the ccellWhiteSpace style property (for the column header part) and the cellWhiteSpace style property (for cells). You can also specify it in CSS.

- About borders

  Note the following on using borders in column headers and cells:

  - Border width is fixed as 1px. You cannot customize it.

    For this reason, it may not be possible to distinguish border styles (for example, a double-lined border appears the same as a single-lined border).

  - To improve visibility, the default border color for left and right borders is white.

- About left and right edges of row headers and bottom edge of column headers

  You cannot customize the color of the left and right edges of row headers or the bottom edge of column headers (left of the bottom scrollbar).

Changing styles for operations

You can only specify styles for different operations using CSS.

The style when the mouse is over the column header cell is as follows:

Prefix	Class Name	Available Styles
ccellHovered	rcf-IBPMTableView-ccellHovered	Color

The style when a row is focused is as follows:

Prefix	Class Name	Available Styles
rowFocused	rcf-IBPMTableView-rowFocused	Color

The style when a row is selected is as follows:

Prefix	Class Name	Available Styles
rowSelected	rcf-IBPMTableView-rowSelected	Color

Refer to section 2.6 Style Properties for details.

2.3.2 TableEdit

TableEdit is the widget for displaying two-dimensional data in table format and editing the data.

📒 Note

- You cannot sort the table data in TableEdit.

- If you specify the id of the XML type UDA in the id property of this table,XML stored in UDA value should be a two-level deep XML value.XML values only up to 2 levels are supported.

- If you specify the id of the custom data type UDA in the id property of this table, you can specify the xpath property to target data. XML values only up to 2 levels from this xpath are supported.

- You save the UDA value including comments or attributes with TableEdit.And comments or attributes are removed.

Sample Display

	ID	Name	Score
1	ID0001	Andy	90
2	ID0002	Bob	80
3	ID0003	Cindy	100
4	ID0004	Yolanda	90

Format

```
<div rcf:type="IBPMTableEdit" rcf:id="uda_XMLVariable">
  <div rcf:type="IBPMViewColumn" rcf:name="ID" rcf:propertyName="id"></div>
  <div rcf:type="IBPMViewColumn" rcf:name="Name" rcf:propertyName="name"></div>
  <div rcf:type="IBPMViewColumn" rcf:name="Score" rcf:propertyName="score"></div>
</div>
```

Point

- Define ViewColumn in which table information is defined in child elements (mandatory).

- Line feed is inserted before and after the widget.

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
editable	Boolean	Specifies whether or not the cell content can be edited. - true: Editable - false: Not editable Regardless of the ViewColumn editable property value, no cells can be edited.	No	true
selectedCell	-	Cannot be specified in Interstage BPM Studio.	-	-
innerTabMove	Boolean	Specifies whether or not moving the focus between cells with Tab key is enabled.	No	true

Name	Data Type	Description	Required	Default Value
		- true: Enabled - false: Disabled	-	

You can also specify TableView properties. Refer to Properties in 2.3.1 TableView for details.
However, the default value of the following properties has been changed as follows.

- showRowHeader: true

- multipleSelect: true

- showDummyColumn: false

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

🅿 Point

When you select the TableEdit widget in the Palette View, and you drop this widget to the Ajax Page Editor, the following widgets are defined together.

- TableEdit,

- Add button: Add a new row to this table.

- Remove button: Remove selected row(s) from this table.
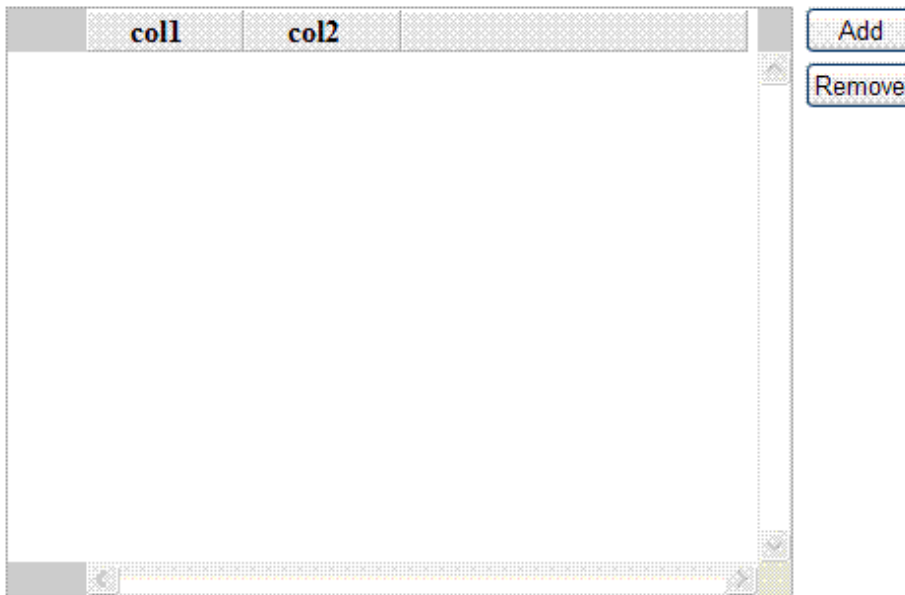
If you want to relate the UDA to the TableEdit widget and you want to use the Add/Remove button, you need to define "uda_"+<this UDA identifier> to the following properties.

- id property of the TableEdit.

- targetTableId property of the Add/Remove button.

Note

If the UDA value is empty (""), empty table will get displayed. In this scenario, even if you click the Add button, nothing will happen. When this table is not empty, you can add a row by clicking the Add button.

Point

- About entering values
  When you double-click a cell, or press the Enter key when the cell is focused, an input field is displayed, in which you can enter characters. When you press a key, the content of the selected cell is cleared and the character corresponding to the pressed key is entered.

- Initializing the table and saving the changed table data
  You can specify an XML type UDA or a custom data type UDA in the id property to define the data displayed in the table. Refer to Properties in 2.3.1 TableView for details. And the changed table data is saved to the UDA value.

When ViewColumn is not specified in a TableEdit child element

When ViewColumn is not specified in a TableEdit child element, the column not specified is not displayed.

Editing values

With TableEdit, you can edit the values. In order to edit, you must specify a TextInput widget for editing in ViewColumn.

Selecting rows

You can select a row by clicking its row header.

In order to select rows in this way, the row header must be displayed. Therefore, specify "true" for the showRowHeader property.

Resizing column width

Refer to Resizing column width in section 2.3.1 TableView.

About focus

When TableEdit is focused, you can use the keyboard to manipulate TableEdit. You can use the following methods to shift focus to TableEdit:

- By pressing the Tab key or Tab+Shift keys provided by the browser

- By directly clicking TableEdit with the mouse.

When TableEdit is focused, an outline is displayed around the TableEdit frame. Also, the focus shifts to the rows below and you can use the keyboard to shift focus to cells in TableEdit.

- When TableEdit is focused using the Tab key
  The last focused cell in TableEdit is focused. If this is the first time that TableEdit is focused, focus shifts to the first cell.

- When TableEdit is focused by mouse click
  If a cell was clicked, that cell is focused. If anything other than a cell was clicked, as above, the last focused cell in TableEdit is focused. If this is the first time that TableEdit is focused, focus shifts to the first cell.

Note

- When you click the TableEdit scrollbar, TableEdit loses focus.

- When the cell is in edit mode, the focus shifts away from TableEdit and the focus shifts to the input widget for editing.

- If the TableEdit data is empty, focus does not shift to a cell.

Mouse operations

Operation	Process
Clicking on a cell	Single and Multiple Selection Modes: Selects the clicked cell. You cannot select dummy column cells.
Double-clicking on a cell	Single and Multiple Selection Modes: Makes the cell available for editing. You cannot select dummy column cells.
Clicking on a row header	- Single Selection Mode: Selects the clicked row. - Multiple Selection Mode: Clears any other selection status and selects the clicked row.
Clicking on a row header + Shift	- Single Selection Mode: Selects the clicked row. - Multiple Selection Mode: Selects all rows from the previously selected row to the clicked row.
Clicking on a row header + Ctrl	- Single Selection Mode: Selects the clicked row. - Multiple Selection Mode: Toggles the selection status of the clicked row.

Keyboard operations

Keyboard operations are only active when TableEdit is focused.

Operation	Process (Single and Multiple Selection Modes)
Space	Selects the focused cell.
Up cursor	Shifts the focus up one cell.
Down cursor	Shifts the focus down one cell.
Left cursor	Shifts the focus left one cell.
Right cursor	Shifts the focus right one cell.
Enter	Toggles the editing status of the cell. - Non-edit mode: If the focused cell is editable, makes the cell available for editing. - Edit mode: Confirms the cell content and makes the cell unavailable for editing.
ESC	In edit mode, cancels the editing and makes the cell unavailable for editing.
Tab	Shifts the focus right one cell. If already in the right edge cell, shifts the focus to the left edge cell of the next row. If already in the right edge cell of the last row, the Tab focus shifting, which is provided by the browser, is enabled and the focus shifts away from TableEdit. Also, if the innerTabMove property is set to "false", does not affect TableEdit. The Tab focus shifting, which is provided by the browser, always enables and the focus shifts away from TableEdit.
Tab + Shift	Shifts the focus left one cell. If already in the left edge cell, shifts the focus to the right edge cell of the previous row. The Tab+Shift focus shifting, which is provided by the browser, is enabled and the focus shifts away from TableEdit. Also, if the innerTabMove property is set to "false", does not affect TableEdit. The Tab+Shift focus shifting, which is provided by the browser, always enables and the focus shifts away from TableEdit.
Home	Shifts the focus to the first cell.
End	Shifts the focus to the last cell.
PageUp	If the scrollbar is displayed, scrolls up.

Operation	Process (Single and Multiple Selection Modes)
PageDown	If the scrollbar is displayed, scrolls down.

Style Properties

Table styles

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-IBPMTableEdit	- Size
Column header	ccell	rcf-IBPMTableEdit-ccell	- Font (except lineHeight) - Cell
Cell	cell	rcf-IBPMTableEdit-cell	- Color - Font (except lineHeight) - Border (except borderWidth) - Cell

🅿 Point

- About specifying table width and height

  - The width and height are specified in pixel units ("px"). If you use any other units for these settings, widgets may not be correctly displayed.

  - If you specify a small table width and height, the scrollbar may not be displayed and items may not be displayed correctly. For this reason, it is recommended to specify at least 100px for both the width and height.

  - If line feeds are inserted automatically in the column header cell content and it spans multiple rows, the column header height is automatically adjusted. As a result, the actual table height may be more than the specified height.

  - The default table width is 400px. The default table height is 300px.

- About cell content display
  By default cell content is displayed without line feeds. To automatically insert line feeds, specify normal in the ccellWhiteSpace style property (for the column header part) and the cellWhiteSpace style property (for cells). You can also specify it in CSS.

- About borders
  Note the following on using borders in column headers and cells:

  - Border width is fixed as 1px. You cannot customize it.
    For this reason, it may not be possible to distinguish border styles (for example, a double-lined border appears the same as a single-lined border).

  - To improve visibility, the default border color for left and right borders is white.

- About left and right edges of row headers and bottom edge of column headers
  You cannot customize the color of the left and right edges of row headers or the bottom edge of column headers (left of the bottom scrollbar).

Changing styles for operations

You can only set styles for different operations using CSS.

The style when the row header is focused is as follows:

Prefix	Class Name	Available Styles
rowFocused	rcf-IBPMTableEdit-rowFocused	Color

The style when a row is selected using the row header is as follows:

Prefix	Class Name	Available Styles
rowSelected	rcf-IBPMTableEdit-rowSelected	Color

The style when a cell is focused is as follows:

Prefix	Class Name	Available Styles
cellFocused	rcf-IBPMTableEdit-cellFocused	Color

The style when a cell is selected is as follows:

Prefix	Class Name	Available Styles
cellSelected	rcf-IBPMTableEdit-cellSelected	Color

The style when a cell is being edited is as follows:

Prefix	Class Name	Available Styles
cellEditing	rcf-IBPMTableEdit-cellEditing	Color

Refer to section 2.6 Style Properties for details.

2.3.3 ViewColumn

ViewColumn is the widget for defining TableView and TableEdit table columns.

Format

```
<div rcf:type="IBPMTableEdit" ... >
  <div rcf:type="IBPMViewColumn" rcf:name="xxx" ... >
  </div>
  ...
</div>
```

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
renderer	-	Cannot be specified in Interstage BPM Studio.	-	-
name	String	Specifies the column name. The value must be unique in ViewColumn of TableView or TableEdit. If a non-unique value is specified, an error occurs.	Yes	-
property Name	String	Specifies the name of the tag to which ViewColumn refers in the XML data specified for TableView or TableEdit.	No	Value of name property
label	String	Specifies the character string to display in the column header.	No	Value of name property
columnW idth	Number	Specifies the column width in pixel. The minimum value is 10. The minimum value is 10. If you specify a value less than	No	Not specified

Name	Data Type	Description	Required	Default Value
		10, an error occurs. If this property is not specified, the column width is the value of defaultColumnWidth property in TableView or TableEdit.		
editable	Boolean	Specifies whether or not the column cell content can be edited. Ignored in TableView. - true: Editable - false: Not editable	No	true
sortable	Boolean	Specifies whether or not sorting can be based on the column cell content. Ignored in TableEdit. - true: Sortable - false: Not sortable	No	true
comparator	-	Cannot be specified in Interstage BPM Studio.	-	-
focusable	Boolean	Specifies whether or not the column cell can be focused. Ignored in TableView. - true: Focusable - false: Not focusable	No	true
selectable	Boolean	Specifies whether or not the column cell can be selected. Ignored in TableView. - true: Selectable - false: Not selectable	No	true

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

Style Properties
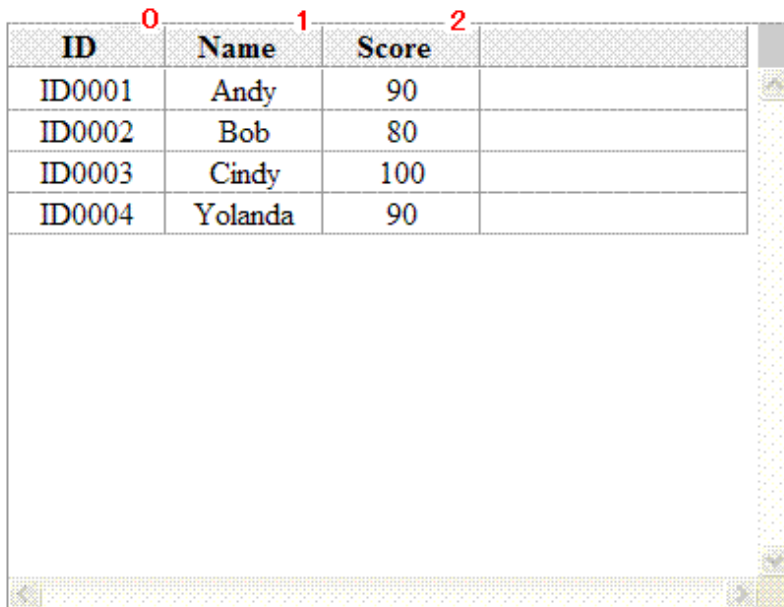
Part Name	Prefix	Class Name	Available Styles
Column header	ccell	- In TableView rcf-IBPMTableView-ccell* - In TableEdit rcf-IBPMTableEdit-ccell* The asterisk (*) represents the column number (starting from 0).	- Font (except lineHeight) - Cell
Cell	cell	- In TableView rcf-IBPMTableView-cell* - In TableEdit rcf-IBPMTableEdit-cell* The asterisk (*) represents the column number (starting from 0).	- Color - Font (except lineHeight) - Border (except borderWidth) - Cell

Refer to section 2.6 Style Properties for details.

Column numbers in Class name
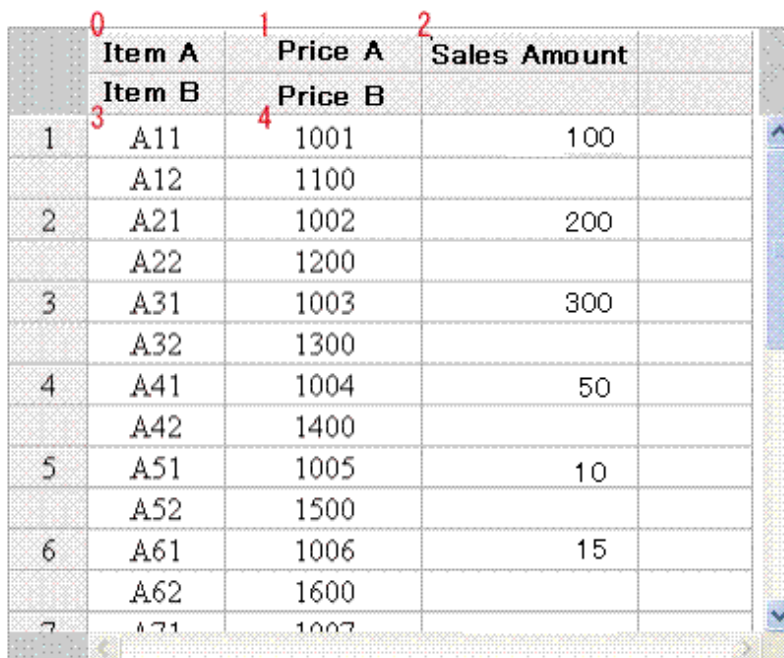
While setting styles using CSS add the column number at the end of the class name.
The column number is defined as follows:

- When columns are not grouped in ViewColumnGroup
  Columns are numbered from left, starting at 0, as in the illustration below.

- When columns are grouped in ViewColumnGroup
  Columns are numbered in the order defined in ViewColumn, as in the illustration below.

Notes on defining column styles in CSS

When setting styles for the entire TableView (or TableEdit) and for ViewColumn columns in CSS, make the settings in the following order:

1. Specify styles for column headers in TableView (or TableEdit) and entire cells.

2. Specify styles for columns headers in ViewColumn and cells of an individual column.

Example of cell styles:

```
.myStyle .rcf-IBPMTableView-cell {          <- Describe overall cell
   background-color: yellow;                style first
}

.myStyle .rcf-IBPMTableView-cell0 {
  background-color: blue;
}

.myStyle .rcf-IBPMTableView-cell2 {

}

 ...
```

If this order is reversed, the overall cell style may take precedence over other styles.

Order of style priority

The order of priority for ViewColumn and TableView (or TableEdit) styles is as follows:

For column headers

1. ViewColumn column header style in style properties

2. TableView (or TableEdit) column header style in style properties

3. ViewColumn column header style in CSS

4. TableView (or TableEdit) column header style in CSS

For cells

1. ViewColumn cell style in style properties

2. TableView (or TableEdit) cell style in style properties

3. ViewColumn cell style set in CSS

4. TableView (or TableEdit) cell style in CSS

2.3.4 ButtonForTable

ButtonForTable is the widget to display a button. ButtonForTable is used with TableEdit to add a new row and delete a selected row.

Sample Display
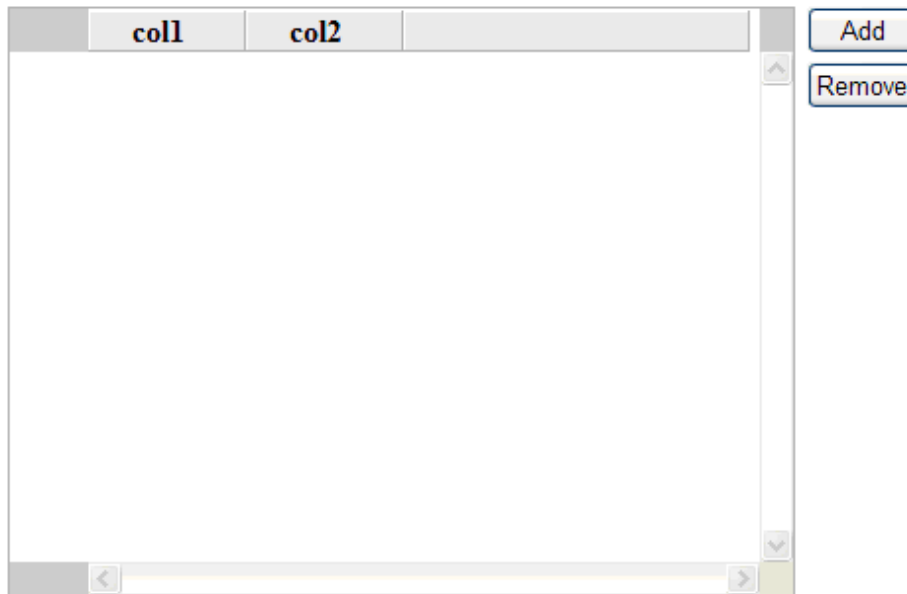
Format

```
<div rcf:id="uda_tableData" rcf:type="IBPMTableEdit" >
        <div rcf:type="IBPMViewColumn" rcf:name="col1"></div>
        <div rcf:type="IBPMViewColumn" rcf:name="col2"></div>
</div>
<div rcf:type="IBPMButtonForTable" rcf:label="Add" rcf:targetTableId="uda_tableData"
rcf:removeActionFlag="false"></div>
<div rcf:type="IBPMButtonForTable" rcf:label="Remove" rcf:targetTableId="uda_tableData"
rcf:removeActionFlag="true"></div>
```

P Point

. . . .

- Line feed is not inserted before and after the widget.

- This widget contains block elements or inline elements as child elements.
  You cannot specify the following as child elements:

    - <a> tags

    - <form> tags

    - <input> tags

    - <select> tags

    - <textarea> tags

    - <label> tags

    - <button> tags

    - <isindex> tags

    - <fieldset> tags

    - <iframe> tags

    - UI widgets

. . . .

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
targetTableId	String	Specifies the ID of the corresponding TableEdit.	Yes	-
removeActionFlag	Boolean	Specifies whether or not this widget is used for delete. - true: Used to delete a selected row - false: Used to add a new row	Yes	-
title	String	Specifies the text to be displayed in the tooltip.	No	""
enabled	Boolean	Specifies whether or not this widget is enabled. - true: Enabled - false: Disabled 　You cannot operate this widget.	No	true
tabIndex	Number	Specifies the tab order when moving the focus with the Tab key. You can specify it in the same way as the tabindex attribute in HTML. The focus moves in ascending order if you specify integers greater than zero.	No	0
label	String	Specifies the label text for the button. This property overwrites the content specified to be displayed on the button as a child element of this widget.	No	""

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

📒 **Note**

· ·

If you specified a long character string to the title property, it is automatically wrapped or cut out.

If an empty string is specified, nothing is displayed.

· ·

Style Properties

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-Button	- Color - Size - Text (except textIndent, textDecoration, and whiteSpace) - Font (except lineHeight) - Border - Padding

Part Name	Prefix	Class Name	Available Styles
			- verticalAlign

Refer to section 2.6 Style Properties for details.

2.3.5 **ViewColumnGroup**

ViewColumnGroup is the widget for creating groups of widgets such as ViewColumn. It is used to display table headers and data in multiple rows.

Point
....
About displaying dummy cells

When two or more ViewColumnGroups are defined and the number of ViewColumns defined in each ViewColumnGroup is not the same,

- Dummy ViewColumns are automatically added to one or more ViewColumnGroups so that all ViewColumnGroups have the same number of ViewColumns (equal to the highest number of ViewColumns in the ViewColumnGroups).

- These dummy cells are displayed.
....

Format

```
<div rcf:type="IBPMTableEdit" ... >
  <div rcf:type="IBPMViewColumnGroup">
    <div rcf:type="IBPMViewColumn" ... ></div>
    ...
  </div>
  ...
</div>
```

Point
....
- You can specify ViewColumns in child elements.

- ViewColumns in TableEdit or TableView are handled as if enclosed in an invisible ViewColumnGroup.
....

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
dispType	-	Cannot be specified in Interstage BPM Studio.	-	-

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

The usage example is shown as follows.

```
<div rcf:type="IBPMTableEdit" rcf:id="uda_XMLVariable">
  <div rcf:type="IBPMViewColumnGroup">
    <div rcf:type="IBPMViewColumn" rcf:name="ID" rcf:propertyName="id"></div>
    <div rcf:type="IBPMViewColumn" rcf:name="Name" rcf:propertyName="name"></div>
  </div>
  <div rcf:type="IBPMViewColumnGroup">
    <div rcf:type="IBPMViewColumn" rcf:name="Score" rcf:propertyName="score"></div>
  </div>
</div>
```

Style Properties

There are no style properties specific to this widget.

2.4 Calendar Widgets

Calendar widgets are widgets for displaying calendars. You can display a calendar directly in the form or display a pop-up calendar.

This section explains how to use calendar widgets.

2.4.1 Calendar

Calendar is the widget for displaying a calendar. Users can select a date in the displayed calendar.

Sample Display

Format

```
<div rcf:type="IBPMCalendar" ... ></div>
```

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
year	Number	Specifies the year to display in the calendar (1 or more).	No	Current year of the system date
month	Number	Specifies the month to display in the calendar (0-11).	No	Current month of system date
firstDayOfWeek	Number	Specifies the first day of the week to display in the first day column. The range of values is from 0 (Sunday) to 6 (Saturday).	No	0 (Sunday)

Name	Data Type	Description	Required	Default Value
selectedDates	-	Cannot be specified in Interstage BPM Studio. You can specify a UDA in the id property to specify the selected date. And the value selected in the calendar is saved to the UDA value. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.	-	-
xpath	String	This property is used for only a custom data type UDA. Specifies xpath of the targeted data when you specify a custom data type UDA to the id properties. You can specify the xpath of the text node or the attribute node.	Yes	-
dateFormat	String	This property is used for only a custom data type UDA. Specifies date format of the targeted data when you specify a custom data type UDA to the id properties. You can specify any of the below date formats : 1. dd.MM.yyyy 2. dd-MM-yyyy 3. MM.dd.yyyy 4. MM/dd/yy 5. MM/dd/yyyy 6. yyyy-MM-dd 7. MM-dd-yyyy 8. MM dd yyyy 9. yyyy MM dd 10. YYYY/MM/dd 11. yyyy/MM/dd hh:mm:ss 12. dd/MM/yyyy 13. yyyy.MM.dd If a target UDA value is "03.21.2010",you specify the following format. specified format : MM.dd.yyyy.	Yes	-
daysOfWeek	String array	Specifies the days of the week listed across the top of the Calendar (from Sunday to Saturday). You can specify a sequence of seven day-of-the-week character strings separated by semi-colon starting from	No	["Su.", "Mo.", "Tu.", "We.", "Th.", "F

Name	Data Type	Description	Required	Default Value
		Sunday. If you specify more or less than seven day-of-the-week character strings, an error occurs. For example: "S;M;T;W;T;F;S"		r.", "Sa."]
holidays	-	Cannot be specified in Interstage BPM Studio.	-	-
specialDates	-	Cannot be specified in Interstage BPM Studio.	-	-
dateToolTips	-	Cannot be specified in Interstage BPM Studio.	-	-
naviType	String	Specifies how to display the navigator. - ALL: Displays Previous Month button, Next Month button and title. - TITLE: Displays title only. - NONE: Does not display the navigator.	No	ALL
naviButtonRenderer	-	Cannot be specified in Interstage BPM Studio.	-	-
naviTitleLabelProvider	-	Cannot be specified in Interstage BPM Studio.	-	-
dayOfWeekCellRenderer	-	Cannot be specified in Interstage BPM Studio.	-	-
dateCellRenderer	-	Cannot be specified in Interstage BPM Studio.	-	-
tabIndex	Number	Specifies the tab order when moving the focus with the Tab key. You can specify it in the same way as the tabindex attribute in HTML. The focus moves in ascending order if you specify integers greater than zero.	No	0
selectable	Boolean	Specifies whether or not the date can be selected. - true: Selectable - false: Not selectable	No	true
selectMode	-	Cannot be specified in Interstage BPM Studio. Users can only select one date.	-	-
utc	-	Cannot be specified in Interstage BPM Studio.	-	-

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

Mouse operations

Operation	Process
Clicking on Previous Month button	Displays the previous month.
Clicking on Next Month button	Displays the next month.
Clicking on a date	Toggles the selection status of the clicked date.
Double-clicking on a date	Selects the clicked date.

Operation	Process
Hovering over a date	The date over which the mouse is hovering gets the focus.

Keyboard operations

Operation		Process
Space		Toggles the selection status of the focused date.
Left cursor	-	Shifts the focus to the previous day (within the displayed range).
	+ SHIFT	Displays the previous month.
Right cursor	-	Shifts the focus to the next day (within the displayed range).
	+ SHIFT	Displays the next month.
Up cursor		Shifts the focus back one week (within the displayed range).
Down cursor		Shifts the focus forward one week (within the displayed range).
Home		Shifts the focus to the first day of the displayed month.
End		Shifts the focus to the last day of the displayed month.
PageUp		Displays the previous month.
PageDown		Displays the next month.

Style Properties

Part Name	Prefix	Class Name	Available Styles
Whole widget (external frame)	None	rcf-IBPMCalendar	- Size - Border
Cell (common to all)	cell	rcf-IBPMCalendar-cell	- Size - Color - Font - Border - Cell
Navigation title	naviTitle	rcf-IBPMCalendar-naviTitle	- Size (except width) - Color - Font - Border - Cell
Navigation button	naviButton	rcf-IBPMCalendar-naviButton	- Size (except width) - Color - Font - Border - Cell
Day of the week header (common)	dayOfWeek	rcf-IBPMCalendar-dayOfWeek	- Size (except width) - Color - Font

Part Name	Prefix	Class Name	Available Styles
			- Border
			- Cell
Day of the week header (individual)	dayOfWeek0 - dayOfWeek6	rcf-IBPMCalendar-dayOfWeek0 - rcf-IBPMCalendar-dayOfWeek6	- Color
			- Font
			- Border
			- Cell
Date (common within the current month)	date	rcf-IBPMCalendar-date	- Color
			- Font
			- Border
			- Cell
Date (per day of the week within the current month)	date0 - date6	rcf-IBPMCalendar-date0 - rcf-IBPMCalendar-date6	- Color
			- Font
			- Border
			- Cell
Date (common except within the current month)	otherMonth	rcf-IBPMCalendar-otherMonth	- Color
			- Font
			- Border
			- Cell
Date (per day of the week except within the current month)	otherMonth0 - otherMonth6	rcf-IBPMCalendar-otherMonth0 - rcf-IBPMCalendar-otherMonth6	- Color
			- Font
			- Border
			- Cell
Focused date	focus	rcf-IBPMCalendar-focus	- Color
			- Font
			- Border
			- Cell
Selected date	selected	rcf-IBPMCalendar-selected	- Color
			- Font
			- Border
			- Cell
Holiday	holiday	rcf-IBPMCalendar-holiday	- Color
			- Font
			- Border
			- Cell
Special day	special0 - special9	rcf-IBPMCalendar-special0 - rcf-IBPMCalendar-special9	- Color
			- Font
			- Border

Part Name	Prefix	Class Name	Available Styles
			- Cell

Refer to section 2.6 Style Properties for details.

🈁 **Note**

. . . .

- The calendar is a table with TD elements for each day.

- You can also specify custom styles for days of the week in Date (within the current month), Date (except within the current month) and Day of the week header.
  In such a case, add a numerical value denoting the day of the week after the prefix and class name.
  (For example: Sunday0...Saturday6)

- The default order of priority for date cell styles is as follows:

    1. selected

    2. focus

    3. special0..9

    4. holiday

    5. date0..6 / otherMonth0..6

    6. date / otherMonth

    7. cell

- Apart from whole widget (external frame), you can only set styles using CSS. When setting styles in CSS, specify the styles in reverse order of priority.

- Styles set in CSS take precedence over default styles. If you only define styles of low priority (such as cell or date) in CSS, default styles of higher priority (such as selected or focus) may be ignored. To avoid this, define all required styles in the correct order of priority.

. . . .

2.4.2 PopupCalendar

PopupCalendar is the widget for displaying a pop-up calendar. PopupCalendar is used with CalendarButton to create a pop-up window in which users can select a date.

Sample Display
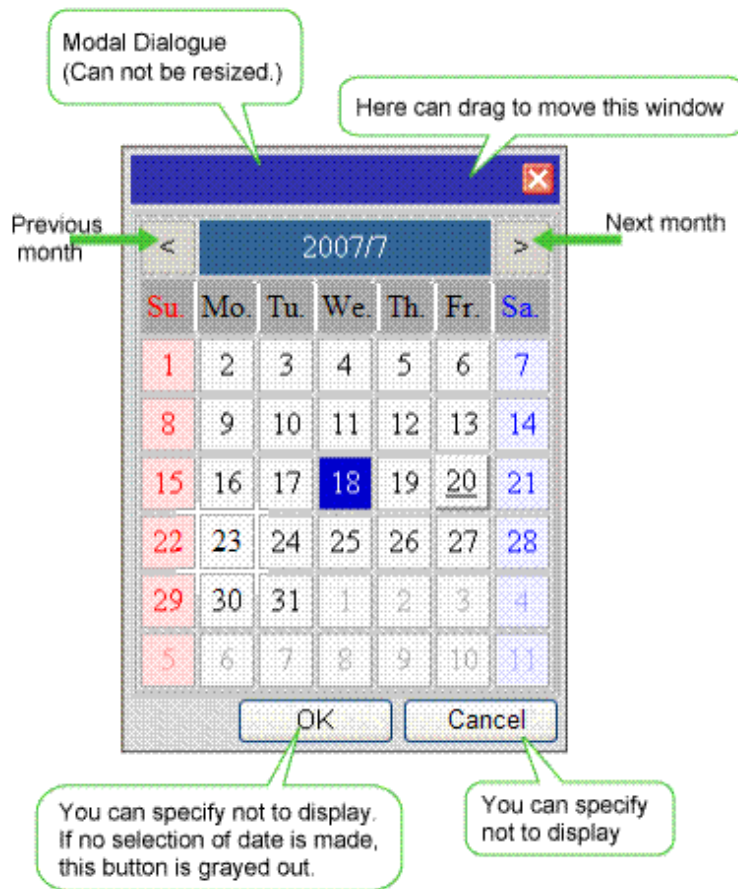
Format

```
<span rcf:id="uda_var1" rcf:type="IBPMDateInput" ...></span>
<div rcf:id="IBPMPopupCalendar_1" rcf:type="IBPMPopupCalendar"
rcf:targetDateInputId="uda_var1" ...></div>
<span rcf:id="CalendarButton_1" rcf:type="IBPMCalendarButton" rcf:target="IBPMPopupCalendar_1" ...></
span>
```

🛈 Note

You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements that cannot contain child elements for details.

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
selectedDate	-	Cannot be specified in Interstage BPM Studio. You can specify a UDA in the id property to specify the selected date. And the value	-	-

Name	Data Type	Description	Required	Default Value
		selected in the calendar is saved to the UDA value. Refer to section 2.5 Common Properties for Screen Widgets for information on how to specify a UDA.		
xpath	String	This property is used for only a custom data type UDA. Specifies xpath of the targeted data when you specify a custom data type UDA to the id properties. You can specify the xpath of the text node or the attribute node.	Yes	-
dateFormat	String	This property is used for only a custom data type UDA. Specifies date format of the targeted data when you specify a custom data type UDA to the id properties. You can specify any of the below date formats : 1. dd.MM.yyyy 2. dd-MM-yyyy 3. MM.dd.yyyy 4. MM/dd/yy 5. MM/dd/yyyy 6. yyyy-MM-dd 7. MM-dd-yyyy 8. MM dd yyyy 9. yyyy MM dd 10. YYYY/MM/dd 11. yyyy/MM/dd hh:mm:ss 12. dd/MM/yyyy 13. yyyy.MM.dd If a target UDA value is "03.21.2010",you specify the following format. specified format : MM.dd.yyyy.	Yes	-
labelOK	String	Specifies the label text on the OK button. If you specify an empty string (""), the button is not displayed. (*)	No	"OK"
labelCancel	String	Specifies the label text on the cancel button. If you specify an empty string (""), the button is not displayed. (*)	No	"Cancel"
targetDateInputId	String	Specifies the ID of the DateInput to display the date selected in the calendar.	No	""

(*)If you specify an empty string for both labelOK and labelCancel, the footer part (where the OK button and cancel button are displayed) is not displayed.

Note

. . . .

You can specify the following properties of Calendar:

-firstDayOfWeek

-daysOfWeek

. . . .

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

Style Properties

Part Name	Prefix	Class Name	Available Styles
OK/Cancel button	button	rcf-PopupCalendar-button	- Color - Font (except lineHeight)

You can also specify Calendar style properties. Refer to Style Properties in section 2.4.1 Calendar for details. However, note that the prefix and class name of the style property for the whole widget (external frame) is as follows:

- Prefix: calendar

- Class name: rcf-IBPMCalendar-calendar

Refer to section 2.6 Style Properties for details.

Additional Notes

- You cannot use <div> tag style attributes (such as position, top and left) with this widget. Refer to 1.2.3 Attributes that can be used with UI Widget <div> tags and <span> tags for details.

- Specify the top and left in style properties. Values specified in CSS are ignored.

2.4.3 CalendarButton

CalendarButton is the widget to display a pop-up calendar. CalendarButton is used with PopupCalendar.

PopupCalendar is displayed when CalendarButton is clicked or when the Enter key or space key is pressed with focused CalendarButton.

Sample Display

Format

```
<div rcf:type="IBPMCalendarButton" rcf:target="popupCalendar1" ... ></div>
```

Or

```
<span rcf:type="IBPMCalendarButton" rcf:target="popupCalendar1" ... ></span>
```

Note

. . . .

You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements for details.

. . . .

This widget is displayed as follows:

- Using with <div> tag, line feed is inserted before and after the widget.

- Using with <span> tag, line feed is not inserted before and after the widget.

Properties

The signification of the column headers is explained in the Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
alt	String	Specifies alternative text for the image.	No	Calendar Button
enabled	Boolean	Specifies whether or not the button is active (enabled). - true: Active - false: Inactive	No	true
target	String	Specifies the ID of the displayed PopupCalendar.	Yes	-
tabIndex	Number	Specifies the tab order when moving the focus with the Tab key. You can specify it in the same way as the tabindex attribute in HTML. The focus moves in ascending order if you specify integers greater than zero.	No	0

You can also specify common properties for widgets. Refer to section 2.5 Common Properties for Screen Widgets for details.

Example

A sample display and corresponding specification of PopupCalendar, DateInput and Text (for displaying an error message) are as follows:

```
 ...
<span rcf:id="uda_var1" rcf:type="IBPMDateInput" ...></span>
<div rcf:id="IBPMPopupCalendar_1" rcf:type="IBPMPopupCalendar"
rcf:targetDateInputId="uda_var1" ...></div>
<span rcf:id="CalendarButton_1" rcf:type="IBPMCalendarButton"
rcf:target="IBPMPopupCalendar_1" ...></span>
...
```

The input order is as follows:

1. Click CalendarButton.
   PopupCalendar is displayed.

2. Select a date in PopupCalendar.
   The date is input in DateInput through the UDA var1.

Style Properties

Part Name	Prefix	Class Name	Available Styles
Whole widget	None	rcf-CalendarButton	- Border - verticalAlign

Details of the verticalAlign style are as follows:

Name	Data Type	Description	Default Value
verticalAlign	String	Specifies the vertical alignment of the button. You can specify the CSS vertical-align property value.	baseline

Refer to section 2.6 Style Properties for details.

2.5 Common Properties for Screen Widgets

Note

· ·

The 'onChange' event (property) and other properties are used in the internal operation of the form. Use of properties other than those specifically documented here, may conflict with the operation of the form, and result in unpredictable behavior.

· ·

This section lists and describes the common properties for screen widgets.

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
id	String	Specifies the widget ID. (*) The specified character string must satisfy the following conditions: - The first character must be an alphanumeric character. - The second and subsequent characters must be alphanumeric characters, numbers, underscores, hyphens, colons or periods. Leading and trailing whitespace is ignored. If any other characters are specified, an error occurs. You cannot specify an empty string. In Interstage BPM Studio, you can specify a UDA in this property to overwrite other specific property with the UDA value when the widget is initialized. And the value is saved to the UDA value. However, note that the selectedIndex or selectedIndexes property is overwritten by the index depending on the specified UDA value when Select, SelectList or CheckList is initialized. And the value depending on the selected index is saved to the UDA value. Refer to How to specify a UDA in the property for details.	No	Widget name _Serial number
type	String	Specifies the widget type.	Yes	None

Name	Data Type	Description	Required	Default Value
styleClass	String	Specifies the style class name. You cannot specify this in an additional functional widget.	No	""

(*)This is different to the HTML id attribute. Not a rcf:id attribute but an id attribute must be specified to specify the HTML id attribute.

Properties related with UDA specified in the id property

The following table lists widgets whose id property is used to relate it with UDA:

Widget	Property	Type of UDA which can be specified in the id property
Text	value	STRING, BOOLEAN, INTEGER, LONG, FLOAT, BIGDECIMAL, XML, DATE, CUSTOM
TextInput	value	STRING, BOOLEAN, INTEGER, LONG, FLOAT, BIGDECIMAL, DATE, CUSTOM
CheckBox	value	STRING, BOOLEAN
RadioButton	value	STRING, BOOLEAN
TextArea	value	STRING, BOOLEAN, INTEGER, LONG, FLOAT, BIGDECIMAL, XML, DATE, CUSTOM
Select	selectedIndex, selectedIndexes	STRING, INTEGER, LONG, FLOAT, BIGDECIMAL, CUSTOM (*)
ComboBox	value	STRING, INTEGER, LONG, FLOAT, BIGDECIMAL, CUSTOM
DateInput	date	DATE, CUSTOM
NumberInput	number	INTEGER, LONG, FLOAT, BIGDECIMAL, CUSTOM
SelectList	selectedIndex, selectedIndexes	STRING, INTEGER, LONG, FLOAT, BIGDECIMAL, CUSTOM (*)
CheckList	selectedIndex, selectedIndexes	STRING, INTEGER, LONG, FLOAT, BIGDECIMAL, CUSTOM (*)
Panel	label	STRING, BOOLEAN, DATE, INTEGER, LONG, FLOAT, BIGDECIMAL, XML
ViewStack	selectedIndex	INTEGER
TabPanel	selectedIndex	INTEGER
FragmentContainer	src	STRING
TableView	data	XML,CUSTOM
TableEdit	data	XML,CUSTOM
Calendar	selectedDates	DATE, LONG, CUSTOM
PopupCalendar	selectedDate	DATE, LONG, CUSTOM

(*) In case of multiple selections, you can specify only String type and the values are separated by comma.

How to specify a UDA in the property

In Interstage BPM Studio, you can relate the widget with UDA using some properties.

In such case, the format of the property is as follows:

```
rcf:<property name>="uda_<uda identifier>"
```

Example: If you use the value of UDA Value1 and Options with Select widget, the properties are described as follows:

```
<div rcf:type="IBPMSelect" rcf:id="uda_Value1" rcf:prop_options="uda_Options"...></div>
```

In the above example, note that:

- Not the index but the value of the selected option is stored to UDA Value1.

- In the case of multiple selections, all selected values are stored to UDA Value1 in comma separated format.

If you relate a custom data type UDA to some widgets, you define it as follows:

```
rcf:id = "uda_<UDA's identifier>.unique string"
```

For example. If you relate a custom data type UDA to widgets ,you define it as follows:

UDA's identifier : myAddress

```
value   :<address>
                <street>44 Deerfallen Way</street>
                <city>Albuquerque</city>
                <state>NM</state>
         </address>
```

```
<div rcf:id="uda_myAddress.01" rcf:type="IBPMTextInput" rcf:xpath="/address/street/text() "...>
<div rcf:id="uda_myAddress.02" rcf:type="IBPMTextInput" rcf:xpath="/address/city/text() "...>
```
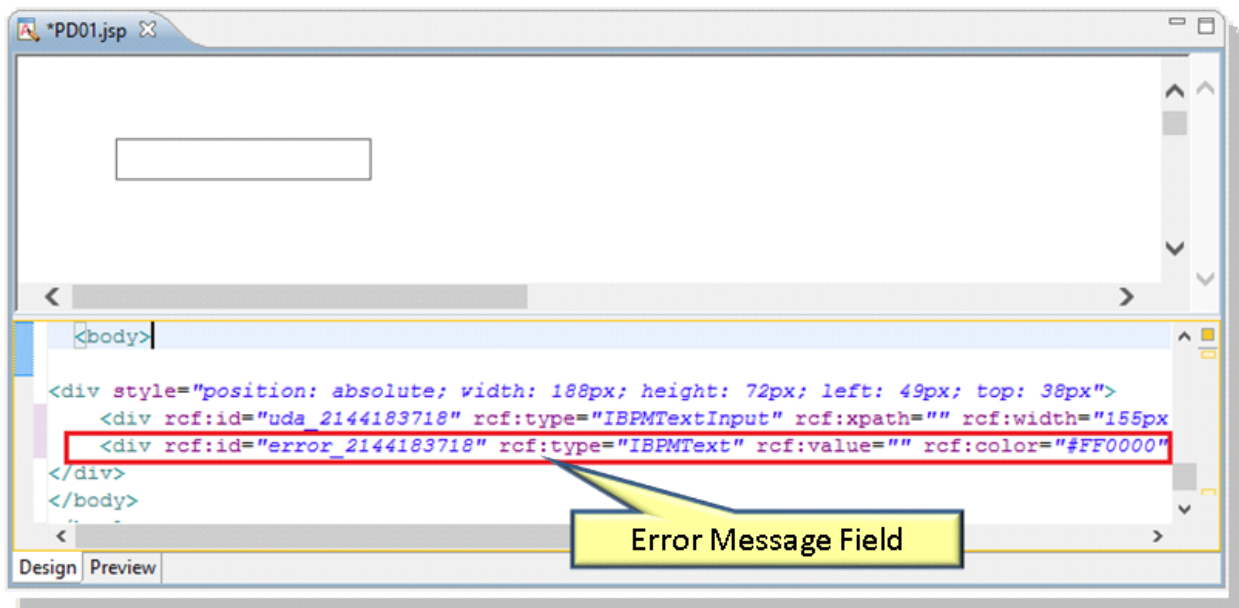
📒 Note

- Use commas to separate values in the prop_options property (used in Select, SelectList, and CheckList widgets) and the prop_list property (used in ComboBox widget).

- Use semicolon to separate values in the options property (used in Select, SelectList, and CheckList widgets) and the list property (used in ComboBox widget).

- Note the following behavior for spaces in prop_options and prop_list:

  1. If no data is entered after comma, this leads to an empty option being created.

  2. If value is empty or contains only spaces then an empty option will be displayed.

  3. The Select widget throws an error in case of duplicate options, option containing only spaces, or empty option.

  4. If spaces or empty value is the selected option, then value stored in server will be empty.

- About the mapping of UI widgets and UDA:

  - If invalid UDA (not existing, not supported type) is mapped to the UI widget, each property (value etc.) is not overwritten.

  - If invalid UDA is mapped to DateInput, the empty string is displayed as initial value.

  - Nothing is displayed if the value of INTEGER UDA mapped to ViewStack or TabPanel is beyond the limits of the index that can be used.

  - When the error occurs by processing FragmentContainer added to TabPanel, other Tab of TabPanel cannot be used.

- About UDA value:

  - All the UDA values that have leading or trailing spaces are displayed as untrimmed values (with spaces) during initial form load.

  - If you update this value and it contains any leading and trailing space, the value saved to the server will be trimmed.

  - In case of multiple selections (Select, SelectList, CheckList), leading and trailing spaces are trimmed for the joint value and not to each individual value.

About Error Message Field

When you select the following UI widgets in the Palette View and add these UI widgets to the Ajax Page Editor, the Error Message Field is added together:

- TextInput

- TextArea

- Select

- ComboBox

- DateInput

- NumberInput

- SelectList

- CheckList

- TableView

- TableEdit

- Calendar

- PopupCalendar

Error Message is displayed to this Error Message Field, when you execute the following operations in Console:

- You define "true" to this mandatory property, but the user does not enter this field in Console.

- You related a UDA which does not support type with this UI Widget.

- You specify an invalid path or the path that does not exist as the xpath value. etc.:

```
rcf:id = "error_<uda identifier>"
```

Note

If you define a Select widget and define "true" to the multiple property, this widget and this Error Message Field are overlapped. You have to move this Error Message Field.

2.6 Style Properties

This section explains style properties for screen widgets.
Refer to section 1.2.2 Style Settings if you want to know about specifying style properties.

Also, if the style properties are omitted, the browser's default values are used.

![Note icon] Note

. . . .

If an invalid value is specified in style properties, the setting is ignored.

. . . .

![Point icon] Point

. . . .

Style property names for the whole widget

For example, names of style properties for the whole widget are as follows:

- color: Specification of color

- backgroundColor: Specification of background color

- fontSize: Specification of font size.

Style property names for specific widget parts

A style property name for a specific screen widget part consists of a prefix followed by the style property name with no space in between, as follows:

```
[Prefix][Property name start with an uppercase]
```

For example, for the "title" part:

- titleColor: Specification of title part color

- titleBackgroundColor: Specification of title part background color

- titleFontSize: Specification of title part font size.

. . . .

Size

Name	Description
width	Specifies the widget width. You can specify the CSS width property value. If a numerical value is specified, the unit of measurement must also be supplied.
height	Specifies the widget height. You can specify the CSS height property value. If a numerical value is specified, the unit of measurement must also be supplied.

Color

Name	Description
color	Specifies the character color. You can specify the CSS color property value.
backgroundColor	Specifies the background color. You can specify the CSS background-color property value.

Font

Name	Description
fontFamily	Specifies the character display font. You can specify the CSS font-family property value.
fontSize	Specifies the character size. You can specify the CSS font-size property value. If a numerical value is specified, the unit of measurement must also be supplied.
fontStyle	Specifies the font style like Italic. You can specify the CSS font-style property value.
fontWeight	Specifies the font weight. You can specify the CSS font-weight property value.
lineHeight	Specify the amount of vertical space between lines of text. You can specify the CSS line-height property value. If a numerical value is specified, the unit of measurement must also be supplied.

Text

Name	Description
textIndent	Specifies the width to indent the first line. You can specify the CSS text-indent property value. If a numerical value is specified, the unit of measurement must also be supplied.
textAlign	Specifies the text alignment. You can specify the CSS text-align property value.
textDecoration	Specifies the text decoration. You can specify the CSS text-decoration property value.
letterSpacing	Specifies the amount of space between letters. You can specify the CSS letter-spacing property value. If a numerical value is specified, the unit of measurement must also be supplied.
wordSpacing	Specifies the amount of space between words. You can specify the CSS word-spacing property value. If a numerical value is specified, the unit of measurement must also be supplied.
whiteSpace	Specifies the way in which spaces are handled. You can specify the CSS white-space property value.

Border

Name	Description
borderWidth	Specifies the border width. You can specify the CSS border-width property value. If a numerical value is specified, the unit of measurement must also be supplied.
borderStyle	Specifies the border style. You can specify the CSS border-style property value.
borderColor	Specifies the border color. You can specify the CSS border-color property value.

Padding

Name	Description
padding	Specifies the padding. You can specify the CSS padding property value. If a numerical value is specified, the unit of measurement must also be supplied.

Margins

Name	Description
margin	Specifies the margin. You can specify the CSS margin property value. If a numerical value is specified, the unit of measurement must also be supplied.

Position

Name	Description
position	Specifies how an element is positioned. You can specify the CSS position property value.
top	Specifies how far from the top of the parent element. You can specify the CSS top property value. If a numerical value is specified, the unit of measurement must also be supplied.
right	Specifies how far from the right of the parent element. You can specify the CSS right property value. If a numerical value is specified, the unit of measurement must also be supplied.
bottom	Specifies how far from the bottom of the parent element. You can specify the CSS bottom property value. If a numerical value is specified, the unit of measurement must also be supplied.
left	Specifies how far from the left of the parent element. You can specify the CSS left property value. If a numerical value is specified, the unit of measurement must also be supplied.

Overflow

Name	Description
overflow	Specifies the behavior on overflow. You can specify the CSS overflow property value.

Cell

Name	Description
textAlign	Specifies the text alignment. You can specify the CSS text-align property value.
verticalAlign	Specifies the vertical alignment of content inside table cells. You can specify the CSS vertical-align property value.
whiteSpace	Specifies how whitespace characters should be handled. You can specify the CSS white-space property value.

Chapter 3 Additional Functional Widgets

This chapter describes additional functional widgets.

🛈 Note

- You cannot set multiple additional functional widgets of the same type for a single screen widget. If you set multiple additional functional widgets, your performance cannot be guaranteed.

- If you set an additional functional widget for a widget that is not a target of that additional functional widget, the additional functional widget is ignored.

3.1 Grouping Functions

This section explains the content of grouping function settings and describes how to make the settings.

3.1.1 RadioButtonGroup

RadioButtonGroup is an additional functional widget to create the mutually exclusive RadioButtons group.

Grouping radio buttons, RadioButtons belonging to the RadioButtonGroup behave as well as the radio button group used in HTML.

- There is a single focus for the whole group

- You can change the selection status of radio buttons in the group using the cursor keys

Format

```
<div rcf:type="RadioButtonGroup" rcf:targets="xxx1; xxx2; ... "></div>
```

Or

```
<span rcf:type="RadioButtonGroup" rcf:targets="xxx1; xxx2; ... "></span>
```

🛈 Note

- You cannot specify child elements. Refer to section 4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements for details.

- There is no difference if it is specified in the <div> tag or the <span> tag because this widget is not displayed on-screen.

Properties

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
targets	Array	Specifies the list of RadioButton IDs included in the group. Note that the order of items in the targets property is the order in which the focus shifts. An error occurs if the value of this property is one of the following: - If no widget with the specified ID exists	Yes	-

Name	Data Type	Description	Required	Default Value
		- If the widget with the specified ID is not a RadioButton - If the value of the value property for the specified RadioButton is an empty string - If multiple RadioButtons with the same value exist in the group		
selected Value	String	Specifies the selected RadioButton value. If you specify an empty string, the last RadioButton in the order specified in the targets property whose checked property is true is selected. If a value that does not match any value in the group is specified, an error occurs.	No	""

📙 Note

For creating a RadioButtonGroup, value property for each radio button of the RadioButtonGroup needs to be provided. As Interstage BPM Console does not use this value for updating UDA, you can provide any value to radio button value property.

You can also specify common properties for screen widgets and additional functional widgets. For more details, refer to section 2.5 Common Properties for Screen Widgets and section 3.2 Common Properties for Additional Functional Widgets.

Creating a radio button group

To create a radio button group, specify the IDs of the RadioButton widgets to include in the group in the targets property.

An example is as follows:

```
...
<span rcf:id="radioButton1" rcf:type="IBPMRadioButton" rcf:value="val1" rcf:label="Radio Button
1"></span>
<span rcf:id="radioButton2" rcf:type="IBPMRadioButton" rcf:value="val2" rcf:label="Radio Button
2"></span>
 ...
<span rcf:id="group1" rcf:type="RadioButtonGroup" rcf:targets="radioButton1; radioButton2" ... ></
span>
 ...
```

3.2 Common Properties for Additional Functional Widgets

This section describes common properties for additional functional widgets.

The meaning of the column headers is explained in Properties topic of the section 2.1.1 Text.

Name	Data Type	Description	Required	Default Value
enabled	Boolean	Specifies whether or not the function is enabled. true: Enabled. false: Disabled.	No	true

Chapter 4 Notes

This chapter describes notes on using UI widgets.

4.1 Notes for Screen Widgets

This section describes notes on screen widgets.

4.1.1 General Notes for Screen Widgets

Screen widgets are created using HTML, JavaScript, and CSS.

Because of this, differences in the functionality of the browser in which the widget is used can lead to differences in appearance and performance of the widget.

4.1.2 Errors for Displaying Screen Widgets

This section lists some representative errors that may occur when a screen widget is displayed.
When such an error occurs, display of the screen widget is interrupted.

- The ID specified in rcf:id is duplicated (RCF11000)

- A value other than "true" or "false" is specified for an attribute that specifies Boolean (RCF11001)

- A value that cannot be interpreted as a numerical value is specified for an attribute that specifies Number (RCF11001)

- A required property of a widget is not specified (RCF11003)

4.1.3 Behavior when child elements are specified for widgets that cannot contain child elements

Do not describe child elements for widgets that cannot contain child elements. If you do, the widget behavior is affected in the following:

- If the child element contains UI widgets described in <div> tags or <span> tags:
  Widgets described in the child element are ignored.

- If the child element contains any other generally used HTML tags:
  The behavior is unstable.

4.1.4 The width and height of Screen Widget

Describe screen widgets in <div> tags or <span> tags.
Width and height of Screen widgets do not include the size of border area and padding area. The description of width and height in <div> tags is shown below:

Figure 4.1 Width and Height in <div> tag
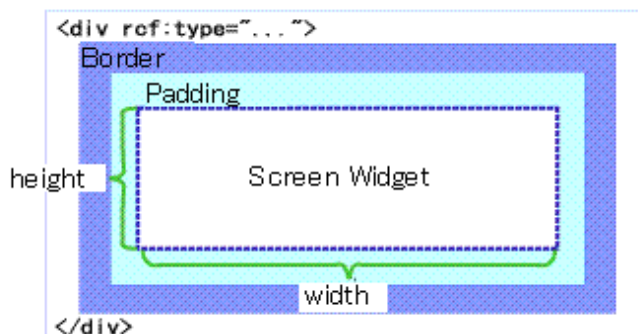
However, width and height of the following widgets include the size of border area:

- ButtonForTable

- Calendar

4.1.5 Layout of Screen Widgets

Screen widgets are designed taking into account the default Windows layout.
If you change the default Windows layout settings (for example in the Display Properties window), screen widgets may not be displayed correctly.

Screen widgets may also not be displayed correctly if you dynamically change the browser text size.

4.1.6 Undo Using Ctrl+z in Text Input Widgets

If you press Ctrl+z in a text input widget, the widget reverts to its status when it was focused.

Text input widgets are as follows:

- TextInput

- TextArea

- DateInput

- NumberInput

- ComboBox

4.1.7 Number of Screen Widgets per Page

Refer to section 1.2.4 Number of Screen Widgets per Page.

4.1.8 Notes for zoom in and zoom out a Page Size

Zoom in and zoom out functions are not available. If you do, the screen widgets may not be displayed correctly.

4.2 Common Notes for All Widgets

This section provides notes common to all widgets.

4.2.1 Notes for Focus Shifting in Internet Explorer

When a UI widget is focused and a scrollbar in another frame is clicked, keyboard input remains available for that widget. However, if keyboard input is made in this status, there is a possibility that the input data may be lost or incorrectly set. For this reason, it is recommended not to input data using the keyboard in this status.

When a part other than a scrollbar is clicked, the relation between focus and keyboard input returns to normal.

It is also recommended to consider whether or not to allow the display of scrollbars in frames when designing screens.

4.3 Error Messages

This section explains messages displayed on the Ajax Page Editor.

RCF11000

Duplicate component id. id=%1

Meaning of parameter

%1: Duplicated ID

Action

Try not to duplicate IDs for UI Widgets.

RCF11001

Invalid attribute value. name=rcf:%1 value=%2

Meaning of parameter

%1: Property name to which an illegal value is specified
%2: Specified value

Action

Try to specify an initial value of the correct data type.

RCF11003

The property %1 is undefined.

Meaning of parameter

%1: Undefined property name

Action

Confirm whether the required property is defined.

RCF12604

Specified index of "%1" property(%2) is out of range.

Meaning of parameter

%1: Property name to which an illegal index is specified
%2: Illegal index

Action

Confirm whether the value specified to the property is in valid range.

RCF12605

Specified value of "%1" property(%2) is not found in values of "options" property.

Meaning of parameter

%1: Property name to which an illegal value is specified
%2: Specified value

Action

Confirm whether the value specified to the property is included in values of the options property.

RCF13302

Specified index of "%1" property(%2) is out of range.

Meaning of parameter

%1: Property name to which an illegal index is specified
%2: Illegal index

Action:

Confirm whether the value specified to the property is in valid range.

RCF13402

Specified index of "%1" property(%2) is out of range.

Meaning of parameter

%1: Property name to which an illegal index is specified
%2: Illegal index

Action

Confirm whether the value specified to the property is in valid range.

Index