Model's reference

July 3, 2022

Contents

1 AbstractBoolField Not-referenced

1.1 Diagram
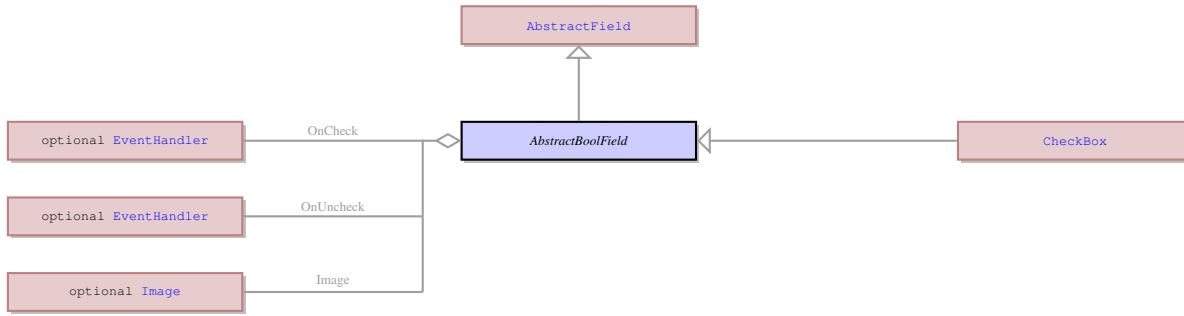
1.2 Description

Name: AbstractBoolField

It is an abstract UI element, which unites the concrete UI elements that can be in one of the two states: enabled (TRUE) or disabled (FALSE). The concrete UI elements that inherit their properties from the AbstractBoolField are CheckBox .

Parent: AbstractField - This UI element represents an abstract field from which all the form widgets inherit their properties. This abstract UI element unites all form fields - the form elements that can accept and display data - as opposed tp form containers - elements that determine the form layout.

It is an abstract UI element, which unites the concrete UI elements that can be in one of the two states: enabled (TRUE) or disabled (FALSE). The concrete UI elements that inherit their properties from the AbstractBoolField are CheckBox .

1.3 Children

- CheckBox - It is a concrete UI element that consists of a single check box and a label attached to it. It can be in only one of 2 states at a time - either checked or unchecked. Changing of the state can either change the value that will be written to the underlying variable, or trigger an event handler.

1.4 Fields

Name	Type	Description
OnCheck	optional EventHandler	The OnCheck field defines the event which will be triggered if the IsChecked field of the UI element is changed to TRUE.
OnUncheck	optional EventHandler	The OnUncheck field defines the event which will be triggered if the IsChecked field of the UI element is changed to FALSE.
Title	optional String	This is the inscription attached to the UI element. Usually this is the text of all sorts of labels.
Image	optional Image	It is an image that can be applied to other UI elements, e.g. to a button.
AllowNewlines	Bool	This property specifies whether the Enter key will be used to move to another form element at runtime (if the value is FALSE), or it will create a newline symbol inside the current field (if the value is TRUE). It is typically applied for the TextArea element.

2 AbstractComponent Not-referenced

2.1 Diagram
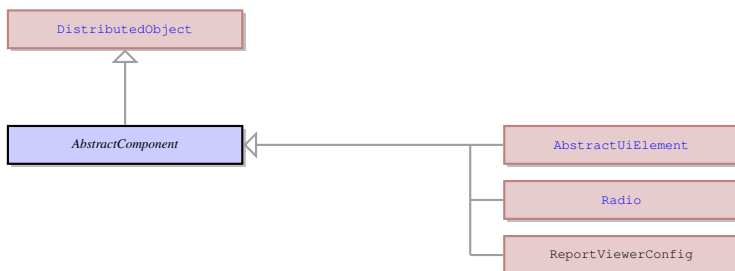
2.2   Description

Name: AbstractComponent
This is the common parent of all UI elements.
Parent: DistributedObject - This is the root of the UI element hierarchy.
This is the common parent of all UI elements.

2.3   Children

- AbstractUiElement - AbstractUiElement is the base class for UI widgets. It is a generic UI element that can accept user actions. Most of concrete UI elements must inherit the properties and action types from the AbstractUIElement.

- Radio - A Radio is a UI element that can only occur inside a RadioGroup . It can be in either of the two states at a time - checked or unchecked. The state of one Radio in a list influences and depends on the state of other items in the same list.

2.4   Fields

Name	Type	Description
Identifier	String	It is a unique name of a UI element by which it can be referenced.

3   AbstractContainer Not-referenced

3.1   Diagram

3.2   Description

Name: AbstractContainer
This UI element represents an abstract container from which all the form containers their properties. This abstract UI element unites all form containers - elements that determine the form layout.
Parent: AbstractUiElement - AbstractUiElement is the base class for UI widgets. It is a generic UI element that can accept user actions. Most of concrete UI elements must inherit the properties and action types from the AbstractUIElement.
This UI element represents an abstract container from which all the form containers their properties. This abstract UI element unites all form containers - elements that determine the form layout.

3.3   Children

- AbstractDataTable - This UI element is used to display and edit data in a customized two-dimensional table of cells. The data in the cell therefore can be retrieved by specifying the row and column identifier of that cell in the table. AbstractDataTable UI element manages the overall appearance and behavior of the table, but does not have direct influence on the columns and rows.

- ElementContainer - This UI element unites all the containers which can contain exactly one element. The containers that derive from ElementContainer UI element can be logically opposed to containers derived from ItemsContainer UI element that can contain any number of elements of any type. The elements that inherit their properties from ElementContainer can encompass such elements as ring menu area or any other container. They can also contain an element belonging to ui.AbstractFiled class, but only one such element.

- **ItemsContainer** - The containers that can contain any number of UI elements inherit their properties from the ItemsContainer UI element. These are the containers that can contain any number of form fields and other containers, as opposed to the containers belonging to ElementContainer class.

- **MenuBar** - This is the area for the top menu (is not applied to ring menus). It includes menu options and menu option groups.

- **Tab** - This is a special type of container which can contain any number of elements, but these elements can only be of TabPage . The Tab serves as the container for a stack of tab pages with only one page visible at a time. Other pages can be brought forward by clicking on their tabs.

- **Toolbar** - This is the container that incorporates toolbar buttons.

- **ToolbarGroup** - This is a set of toolbar buttons that are united into a single group. The group unites the toolbar buttons that have the same conditions for being displayed. It was designed to make the toolbar more dynamic - to display or hide the toolbar groups depending on what widgets are active and to combine different groups freely.

4 AbstractDataTable Not-referenced

4.1 Diagram
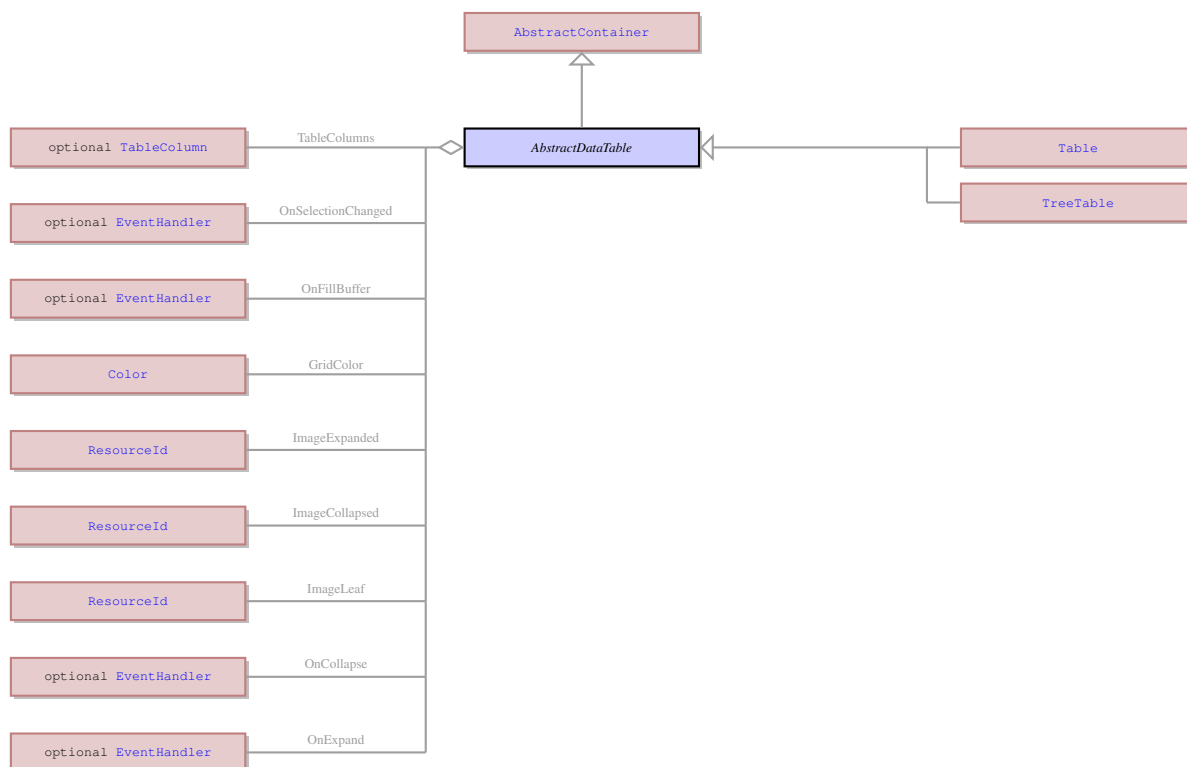
4.2 Description

Name: AbstractDataTable

This UI element is used to display and edit data in a customized two-dimensional table of cells. The data in the cell therefore can be retrieved by specifying the row and column identifier of that cell in the table. AbstractDataTable UI element manages the overall appearance and behavior of the table, but does not have direct influence on the columns and rows.

Parent: AbstractContainer - This UI element represents an abstract container from which all the form containers their properties. This abstract UI element unites all form containers - elements that determine the form layout.

This UI element is used to display and edit data in a customized two-dimensional table of cells. The data in the cell therefore can be retrieved by specifying the row and column identifier of that cell in the table. AbstractDataTable UI element manages the overall appearance and behavior of the table, but does not have direct influence on the columns and rows.

4.3 Children

- **Table** - This is a container that can only contain a specific type of element - TableColumn . It serves as the root container of a table with rows and columns of widgets used to display and input data.

- **TreeTable** - This is a special container that can contain only TableColumn elements. It is similar to a table, but arranges the items in a hierarchical order and allows to fold and unfold rows.

4.4 Fields

Name	Type	Description
TableColumns	list of TableColumn	A set of columns that belong to the same table.
RowHeight	optional String	It defines the default height of a table row in pixels.
IsMultiSelect	Bool	It enables or disables the possibility to select multiple rows of one table during DISPLAY ARRAY execution. The default value is FALSE - the multi-selection is turned off.
OnSelectionChanged	optional EventHandler	It defines an event which must be triggered if the current row is changed or if a new row is selected or deselected, if the multiselect mode is on.
GridColor	Color	The color of the grid lines that separate one table cell from the other cells.
Indent	optional Int	It specified how far should the tree elements in each sub-tree be offset to the right. It is used if the AutoIndent is set to false.
ImageExpanded	ResourceId	It specifies the icon to be shown next to an expanded tree element which has a sub-tree. Its priority is lower than that of the ImageColumn and it is ignored at runtime if both are used.
ImageCollapsed	ResourceId	It specifies the icon to be shown next to a collapsed tree element which has a sub-tree. Its priority is lower than that of the ImageColumn and it is ignored if both are used.
ImageLeaf	ResourceId	It specifies the global icon for the tree elements that do not have the nested elements / sub-trees. Its priority is lower than ImageColumn and is ignored at runtime if ImageColumn is also set.
OnCollapse	optional EventHandler	It is the event that is triggered when the tree or sub-tree received the command to collapse (the user clicked on the collapse button).
OnExpand	optional EventHandler	It is the event that is triggered when the tree or sub-tree received the command to unfold (the user clicked on the unfold button).
ColumnParentId	optional String	It specifies the identifier of the column that stores the id of the parent tree element which serves as the root of the sub-tree to which each row belongs. If a column's identifier is specified in here, the column becomes hidden.
ColumnId	optional String	It specifies the identifier of the column that stores the id of the row. If a column's identifier is specified in here, the column becomes hidden.
ColumnExpanded	optional String	It should be assigned to the column which indicates whether each tree element should be collapsed or expanded when the tree is first displayed at runtime. It is an optional column in the array that is used in the DISPLAY ARRAY for the tree container. In this column each row should have value 1, if the element on the row should be expanded, and 0 if it should be collapsed.
ColumnIsNode	optional String	It should be assigned to the column which indicates the tree items that have children. It is an optional column in the array that is used in the DISPLAY ARRAY for the tree container. In this column each row should have value 1, if the element on the row has children and 0 if it does not. For the rows where 1 is set, the icons indicating that the element includes a sub-tree will be shown next to the element at runtime even if it does not factually have any children. The elements for which 0 is set will look as if they have no children even if they actually do.
ColumnImage	optional String	It should be assigned to the column which contains individual images for each tree element. It is an optional column in the array that is used in the DISPLAY ARRAY for the tree container. In this column each row should contain a BYTE value which will be displayed next to the tree element at runtime.
ColumnEdit	optional String	It should be assigned to the column containing the labels for the tree items. By default is is the first column of the table.

5 AbstractField Not-referenced

5.1 Diagram
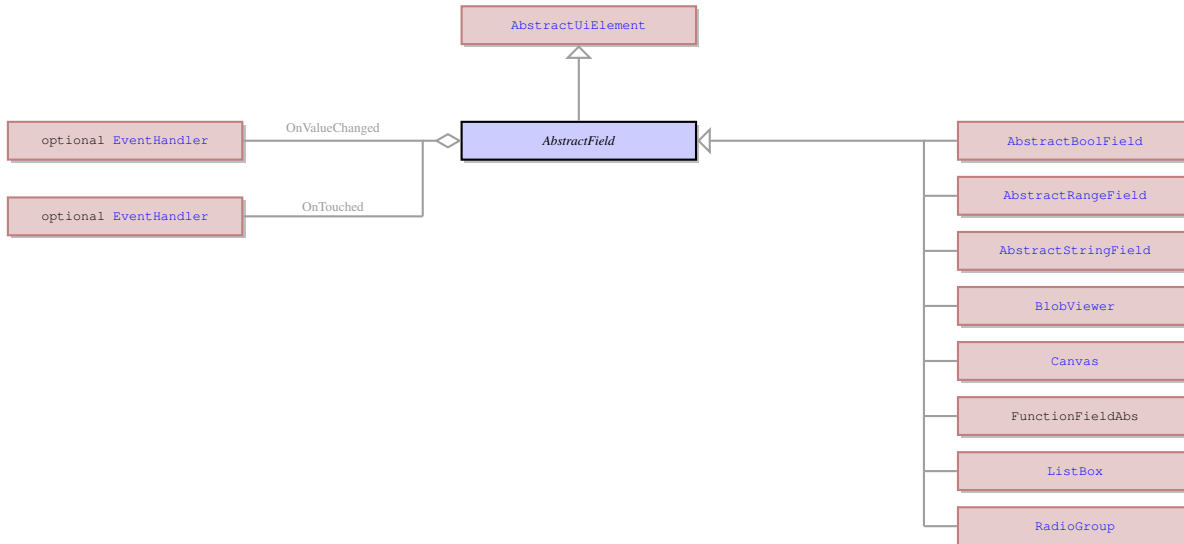
5.2 Description

Name: AbstractField

This UI element represents an abstract field from which all the form widgets inherit their properties. This abstract UI element unites all form fields - the form elements that can accept and display data - as opposed tp form containers - elements that determine the form layout.

Parent: AbstractUiElement - AbstractUiElement is the base class for UI widgets. It is a generic UI element that can accept user actions. Most of concrete UI elements must inherit the properties and action types from the AbstractUIElement.

This UI element represents an abstract field from which all the form widgets inherit their properties. This abstract UI element unites all form fields - the form elements that can accept and display data - as opposed tp form containers - elements that determine the form layout.

5.3 Children

- AbstractBoolField - It is an abstract UI element, which unites the concrete UI elements that can be in one of the two states: enabled (TRUE) or disabled (FALSE). The concrete UI elements that inherit their properties from the AbstractBoolField are CheckBox .

- AbstractRangeField - It is an abstract UI element, which unites the concrete UI elements which accept only the values included into the specified range. It is typically a range or numeric values, for example from 1 to 100. The concrete UI elements that inherit their properties from the AbstractRangeField are Slider , ProgressBar , Spinner , and ScrollBar .

- AbstractStringField - It is an abstract UI element, which unites the concrete UI elements that accept a character string as their value. Most of the concrete UI elements that are not containers inherit their properties from this element.

- BlobViewer - This UI element is used to display and edit BYTE or TEXT values e.g a text or a picture.

- Canvas - It is a concrete UI element that serves as a container for SVG images and allows interactions with such images.

- ListBox - It is a concrete UI element that has the form of a form field with a list of values inside available for selection. It does not accept values entered from the keyboard, but can participate in the input and records into the underlying variable the value that was selected from the list.

- RadioGroup - The Radio is a UI element - a form widget - that contains a set of Radio which are either in selected or deselected state. The user can select only one Radio belonging to the same RadioGroup at a time, selecting a new item from the set deselects the previously selected element.

5.4 Fields

Name	Type	Description
OnValueChanged	optional EventHandler	This event is triggered when the value of the UI element changes. The value of the element is the value which will be recorded to the underlying variable when the input finishes.
OnTouched	optional EventHandler	No information

6 AbstractRangeField Not-referenced

6.1 Diagram
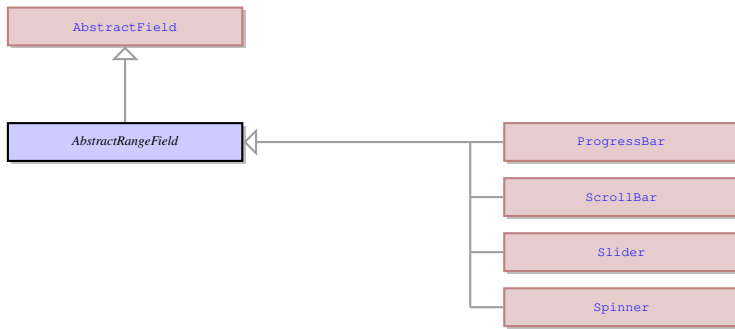
6.2 Description

Name: AbstractRangeField

It is an abstract UI element, which unites the concrete UI elements which accept only the values included into the specified range. It is typically a range or numeric values, for example from 1 to 100. The concrete UI elements that inherit their properties from the AbstractRangeField are Slider , ProgressBar , Spinner , and ScrollBar .

Parent: AbstractField - This UI element represents an abstract field from which all the form widgets inherit their properties. This abstract UI element unites all form fields - the form elements that can accept and display data - as opposed tp form containers - elements that determine the form layout.

It is an abstract UI element, which unites the concrete UI elements which accept only the values included into the specified range. It is typically a range or numeric values, for example from 1 to 100. The concrete UI elements that inherit their properties from the AbstractRangeField are Slider , ProgressBar , Spinner , and ScrollBar .

6.3 Children

- ProgressBar - This is a concrete UI element that has a form of a rectangular bar that can show the progress of the application execution by means of being filled with colour background gradually. For it to reflect the progress, the DISPLAY TO statement should be used to indicate the degree to which it must be filled after each stage. The progress bar should have the maximum value (when it is displayed to the progress bar it becomes 100 percent filled) and minimum value (when displayed makes the progress bar 0 percent filled).

- ScrollBar - It is a concrete UI element that is represented by a scrollbar. It as the maximum and minimum values and the slider can be moved by the user at runtime or by displaying values to the element.

- Slider - This is a concrete UI element that consists of a scale and a slider that can move across this scale. The slider widget has the minimum and maximum value which present the start and the end of the scale. It can be moved directly by the user during the input, or it can be moved if a value within its values range is displayed to it by the 4GL means.

- Spinner - This is a concrete UI element that has a form of a field available for inputting and displaying data that accepts only values inside the allowed range of values. It has the up and down arrows on the right that allow the user to scroll through the acceptable values and prevents the user from entering values from keyboard.

6.4 Fields

Name	Type	Description
MinValue	Int	The minimum value in the range of values accepted by a UI element.
MaxValue	Int	The maximum value in the range of values accepted by a UI element.

7 AbstractStringField Not-referenced

7.1 Diagram
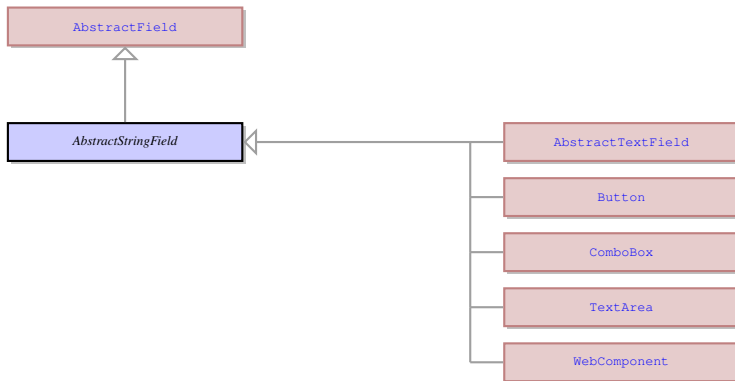
7.2 Description

Name: AbstractStringField

It is an abstract UI element, which unites the concrete UI elements that accept a character string as their value. Most of the concrete UI elements that are not containers inherit their properties from this element.

Parent: AbstractField - This UI element represents an abstract field from which all the form widgets inherit their properties. This abstract UI element unites all form fields - the form elements that can accept and display data - as opposed tp form containers - elements that determine the form layout.

It is an abstract UI element, which unites the concrete UI elements that accept a character string as their value. Most of the concrete UI elements that are not containers inherit their properties from this element.

7.3 Children

- AbstractTextField - It is an abstract UI element, which unites a subset of AbstractStringField elements with the exception of TextArea , ComboBox , and Button . Typically it includes the UI elements which allow entering values, like normal text fields, and usually are only one line wide.

- Button - It is a clickable concrete UI element in a form of a button that is typically used to trigger various events when it is pressed and/or released. It can have a text label or an image on it.

- ComboBox - It is a concrete UI element that has a form of a text field with a drop-down list. It can be restricted to accepting only values from this drop-down list, or it can be set to accept values from the list and the custom values entered by the user. Only one item from the drop-down combobox list can be selected at a time.

- TextArea - This is a concrete UI element that has the form of a text field and shares many features with TextField , but is designed for working with multiline text instead of single lines of text. It does not have some features of the text field that deal with the navigation between fields, but instead it had improved facilities for navigating inside the field.

- WebComponent - It is a concrete UI element that serves as a container for third party web components. It is basically just the space which is filled by the web component at runtime.

7.4 Fields

Name	Type	Description
Text	optional String	This is the value of the UI element, typically of a text field or a combo box which is recorded to the variable linked to it after the input or which is displayed to it.

8 AbstractTextField Not-referenced

8.1 Diagram
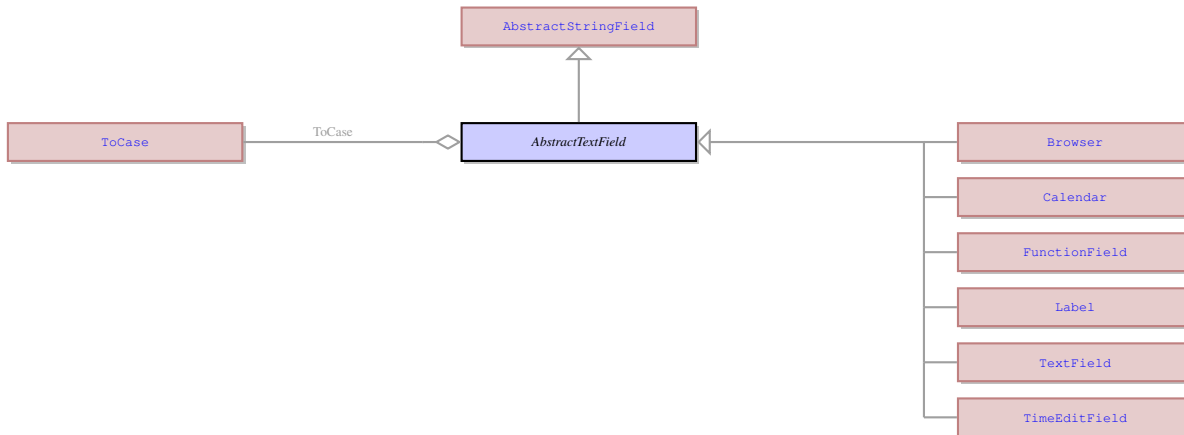
8.2 Description

Name: AbstractTextField

It is an abstract UI element, which unites a subset of AbstractStringField elements with the exception of TextArea , ComboBox , and Button . Typically it includes the UI elements which allow entering values, like normal text fields, and usually are only one line wide.

Parent: AbstractStringField - It is an abstract UI element, which unites the concrete UI elements that accept a character string as their value. Most of the concrete UI elements that are not containers inherit their properties from this element.

It is an abstract UI element, which unites a subset of AbstractStringField elements with the exception of TextArea , ComboBox , and Button . Typically it includes the UI elements which allow entering values, like normal text fields, and usually are only one line wide.

8.3 Children

- Browser - It is a concrete UI element that encompasses a built-in web browser with a somewhat limited functionality. It is used to display web pages, but can also work as a file explorer, display contents of files (e.g. text or image files), etc.

- Calendar - It is a concrete UI element that serves for displaying and inputting dates and has a drop-down lookup calendar for graphical date selection.

- FunctionField - This is a form widget in a form of a text field with a button attached to its right side.

- Label - It is a concrete UI element that has the form of a label with some text, image or both. The label is not an interactive widget and cannot be used for input, but the information displayed by it can be changed dynamically.

- TextField - This is a concrete UI element that is commonly used for input and displaying information. Normally it is used to process a single line of data.

- TimeEditField - This is a concrete UI element that accepts a limited range of time values. The value inside the field is formatted into hh:mm:ss format. It also has up and down arrows that can scroll the data in the field - whether hours, minutes or seconds are scrolled depends on there inside the field the cursor is located.

8.4 Fields

Name	Type	Description
IsPasswordMask	Bool	If enabled, it turns the entered value into a set of * signs to mask it. The value displayed to the field will also be masked with asterisks.
MaxLength	optional Int	It specifies the maximum length in bytes allowed for entering into the filed. Its value is normally taken from the data type and size of the variable linked to the field.
Format	optional String	It specifies the format pattern according to which the entered data should be formatted. Typically used for numeric values to specify the decimal point sign and location and the thousands separator.
ToCase	ToCase	This property specifies the case of a UI element. It can be applied to any UI element that allows entering text from keyboard. By default its value is None, meaning that the case of the letters does not change and remains as they were inputted.

TextPicture	optional String	It formats the entered value by specifying that only letters or only numbers or both can be entered and by supplying delimiters. It is typically used for character values. E.g. if picture is AA-XX, the value may be ab-3c.
Autonext	Bool	If enabled, moves the cursor to the next field during input automatically, when the MaxLength of the current field is met.
Editor	optional String	Specifies the program to be used for opening and editing the BYTE or TEXT value.
Required	Bool	No information

9 AbstractUiElement Not-referenced

9.1 Diagram
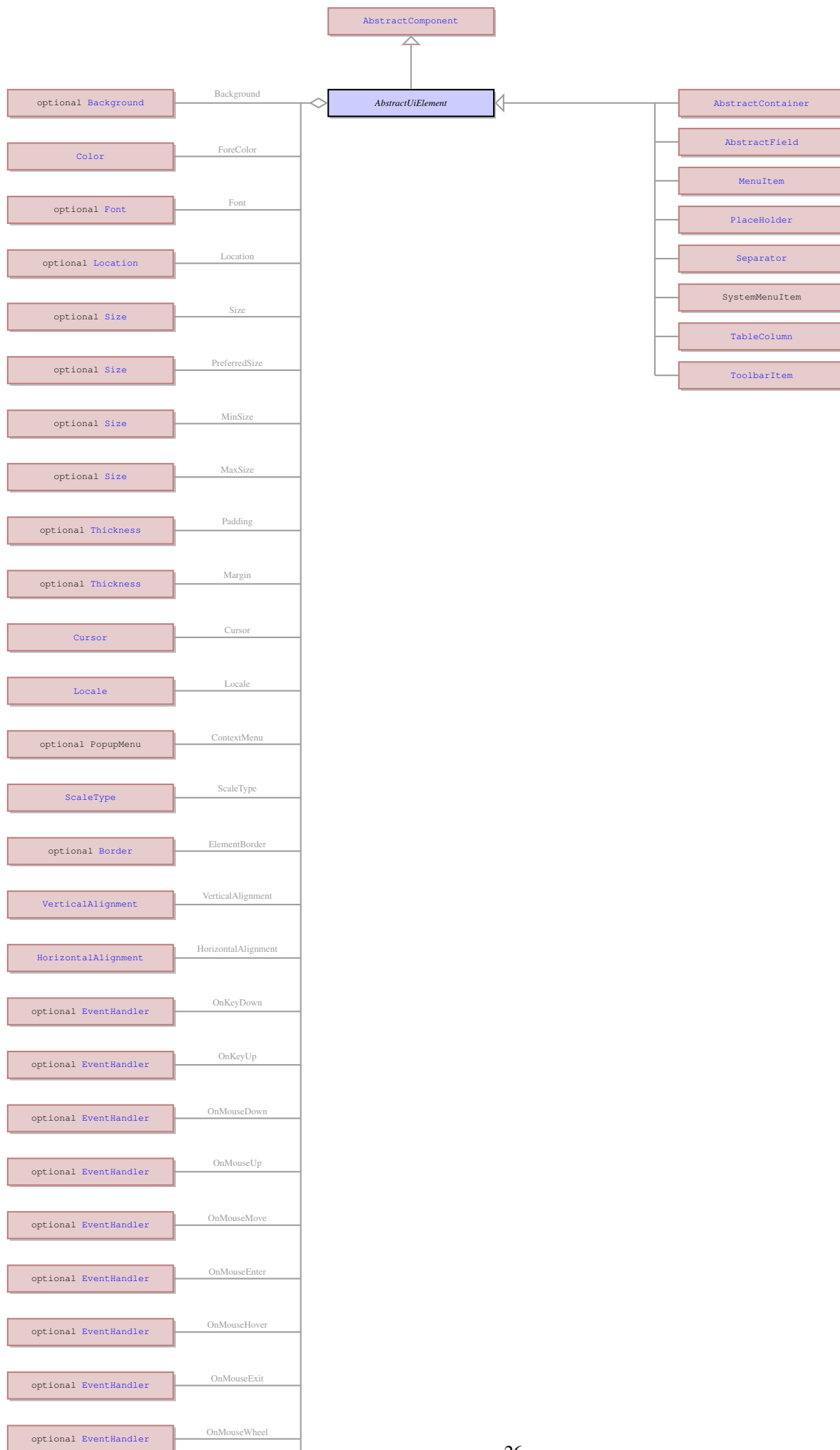
9.2   Description

Name: AbstractUiElement

AbstractUiElement is the base class for UI widgets. It is a generic UI element that can accept user actions. Most of concrete UI elements must inherit the properties and action types from the AbstractUIElement.

Parent: AbstractComponent - This is the common parent of all UI elements.

AbstractUiElement is the base class for UI widgets. It is a generic UI element that can accept user actions. Most of concrete UI elements must inherit the properties and action types from the AbstractUIElement.

9.3   Children

- AbstractContainer - This UI element represents an abstract container from which all the form containers their properties. This abstract UI element unites all form containers - elements that determine the form layout.

- AbstractField - This UI element represents an abstract field from which all the form widgets inherit their properties. This abstract UI element unites all form fields - the form elements that can accept and display data - as opposed tp form containers - elements that determine the form layout.

- MenuItem - This UI element serves as the base class for all menu items: menu commands, menu groups, and menu separators.

- PlaceHolder - No information

- Separator - Any kind of separator, e.g. the status bar separator.

- TableColumn - This is a container that can only be placed inside the Table container or TreeTable container. It can contain only one element belonging to the AbstractField class. Though only one element can be placed into a column, this element will be repeated till the bottom of the column, creating table row together with the elements in other columns, if any. All the duplicates of the element will have the same identifier and will be treated as a single element by th e form designer. The 4GL can differentiate between the instances of the element belonging to different rows by means of using the element identifier together with the number of the table row. The table row numbers start at number 1 at the top of the table.

- ToolbarItem - This is an abstract element that unites the toolbar buttons and toolbar separators.

9.4   Fields

Name	Type	Description
ClassNames	list of ClassName	The name of a class that is applied to the UI element. There can be a customly created class or one of the default classes. The default classes depend on the 4GL attributes applied to the element by means of the 4GL code or form file and usually specify the colour or intensity attribute.
Background	optional Background	Background - defines the background type, color and other parameters.
ForeColor	optional Color	ForeColor - foreground color of the control(used to draw text and/or control border)
Font	optional Font	The font to be used for the UI element.
Location	optional Location	The location of the UI element specified in pixels.
PreferredSize	optional Size	The size of the UI element in pixels that specified by the user that will override the size dynamically calculated at runtime.
MinSize	optional Size	The minimum size of the UI element smaller than which an element cannot shrink when resized.
MaxSize	optional Size	The maximum size of the UI element bigger than which an element cannot become when resized.
NotNull	Bool	If enabled, it forbids to save NULL values to the variable linked to the field.
Padding	optional Thickness	The space between the contents of the UI element (e.g. text in a text field) and the border of this element.
Margin	optional Thickness	The space between the border of the UI element and other UI elements surrounding it.
Cursor	Cursor	The type of the cursor that should be applied when the mouse cursor is hovering above the UI Element.
Locale	Locale	The custom locate of the UI element that may be different from the default locale of the application.
Visible	optional Bool	If enabled, the UI element is visible at runtime. If disabled, it is hidden. The default value is TRUE.
Collapsed	Bool	No information

Enable	optional Bool	If set to TRUE (the default value), the UI element can be interacted with (e.g. button can be pressed, text can be entered into the field). If a UI element is disabled, it is grayed and inaccessible.
ToolTip	optional String	It specifies the text of the tooltip to be visible when the mouse hovers over the element at runtime. If its value is empty, the element will have no tooltip.
TabIndex	optional Int	It specifies the order of the UI elements located on a single form. This order can be used during input for cursor navigation.
ZOrder	Int	It specifies whcih element should be on top if two or more elements overlap. It should be applied only to elements whose container is the coordinate panel.
EnableBorder	Bool	If set to TRUE (the default value), shows the default 1 pixel border around UI elements. If disabled, the element will have no default border.
ScaleType	ScaleType	It defines whether the element contents will be scaled, if the element is resized.
ElementBorder	optional Border	Sets the custom border for a UI element.
VerticalAlignment	VerticalAlignment	Specifies the vertical alignment of the UI element inside its container.
HorizontalAlignment	HorizontalAlignment	Specifies the horizontal alignment of the UI element inside its container.
OnKeyDown	optional EventHandler	The event specified will be triggered, when the cursor is in the given UI element and any key on the keyboard is pressed down.
OnKeyUp	optional EventHandler	The event specified will be triggered when the cursor is in the given UI element and the key on the keyboard previously pressed is released.
OnMouseDown	optional EventHandler	The event specified will be triggered when left mouse button is clicked on the UI element.
OnMouseUp	optional EventHandler	The event specified will be triggered when the left mouse button is released after it was clicked on the UI element.
OnMouseMove	optional EventHandler	The event specified will be triggered when the mouse cursor is moved inside the UI element area.
OnMouseEnter	optional EventHandler	The event specified will be triggered when the mouse cursor enters the UI element area.
OnMouseHover	optional EventHandler	The event specified will be triggered when the mouse cursor enters the UI element area and remains them for a second. Triggered only once while the cursor is inside the element.
OnMouseExit	optional EventHandler	The event specified will be triggered when the mouse cursor exits the UI element.
OnMouseWheel	optional EventHandler	The event specified will be triggered when the mouse wheel is rotated while the cursor hovers over the UI element.
OnMouseDoubleClick	optional EventHandler	The event specified will be triggered when the the user double-clicks on the UI element.
OnMouseClick	optional EventHandler	The event specified will be triggered when the the user left-clicks on the UI element.
OnMenuDetect	optional EventHandler	This event is triggered when the user right-clicks the UI element to invoke context menu.
OnDragStart	optional EventHandler	The event is triggered when the user clicks on an element, holds the mouse key and starts moving it away from its location.
OnDragEnter	optional EventHandler	The event is triggered when the mouse cursor with the dragged item enters the visual boundaries of the UI element to which the item may be dropped.
OnDragOver	optional EventHandler	The event is triggered when the mouse cursor with the item is dragged over a drop target. Typically invoked after OnDragEnter event.
OnDragFinished	optional EventHandler	Triggered after OnDragStart was invoked and then OnDrop executed successfully or the drag and drop action was terminated.
OnDrop	optional EventHandler	The event is triggered when the user releases the mouse button holding the dragged item over an area which allows the item to be dropped.
OnResize	optional EventHandler	The event is triggered when the size of a UI element is changed.
OnSelection	optional EventHandler	The event is triggered when a UI element is selected by mouse cursor.

OnFocusIn	optional EventHandler	The event is triggered when the UI element becomes the current element, e.g. is when the cursor enters the field or when an element is selected.
OnFocusOut	optional EventHandler	The event is triggered when the UI element stops being the current element, e.g. is when the cursor leaves the field or when an element is deselected.
TextAlignment	optional TextAlignment	It specifies the alignment of the text withing the UI element. E.g. the placement of the text inside the label area or in a text field.
IsProtected	Bool	If set to TRUE it prevents character strings displayed from 4GL to overlap with the UI elements. Such strings will be displayed below the UI elements where they are supposed to overlap.
BorderPanelItemLocation	BorderPanelItemLocation	It is applicable only if the UI element is located inside the BorderPanel container and indicates which part of the border panel the element occupies.
GridItemLocation	optional GridItemLocation	It is applicable only if the UI element is located inside the GridPanel container and indicates which cell of the grid panel the element occupies.
Comment	optional String	A character string with some sort of description.
FieldTable	String	No information

10 Action Not-referenced

10.1 Diagram
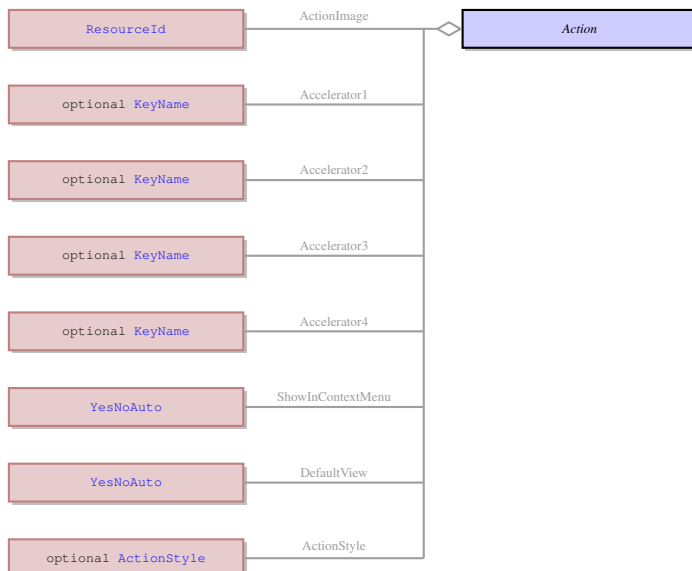
10.2 Description

Name: Action

This UI element is used for compatibility with Genero 4GL. It is used in Genero to define the default actions in a form. In Lycia 4GL they are converted into Toolbar elements via the form converter. The action defaults are necessary in the form model for the compatibility reasons, but it need not be possible to be added to form.

No parents.

This UI element is used for compatibility with Genero 4GL. It is used in Genero to define the default actions in a form. In Lycia 4GL they are converted into Toolbar elements via the form converter. The action defaults are necessary in the form model for the compatibility reasons, but it need not be possible to be added to form.

10.3 Fields

Name	Type	Description
Identifier	String	It is a unique name of a UI element by which it can be referenced.
Text	optional String	Specifies the text displayed to the trigger.
ActionImage	optional ResourceId	Specifies the image which is displayed to the trigger.
Comment	optional String	Specifies the text that is displayed at hover and usually describes the action.
Accelerator1	optional KeyName	Defines the accelerator key(s) that will trigger the action at runtime.

| Accelerator2	optional KeyName	Defines the accelerator key(s) that will trigger the action at runtime.
Accelerator3	optional KeyName	Defines the accelerator key(s) that will trigger the action at runtime.
Accelerator4	optional KeyName	Defines the accelerator key(s) that will trigger the action at runtime.
ShowInContextMenu	optional YesNoAuto	It defines whether an element should appear in application toolbar. This property is not applied on the client side, used only by application server.
Statical	optional Bool	Determines whether triggers are visible even if they are not defined.
Order	optional Int	Indicates in what order triggers appear on the toolbar.
DefaultView	optional YesNoAuto	Determines whether the trigger is visible on the toolbar.
Validate	optional Bool	Determines whether data validation is required for the action.
Place	optional String	It defines where a toolbar button will appear in the application toolbar. This property is applied on the client side. It has two predefined values - top and top-popup. The default value is top-popup.
ClassNames	list of ClassName	The name of a class that is applied to the UI element. There can be a customly created class or one of the default classes.
ActionStyle	optional ActionStyle	It specifies what should be shown in action view (e.g. toolbar button). I includes values "Icon", "Label" and "Icon Label". Default value is "Icon".

11 ActionEventHandler Not-referenced

11.1 Diagram
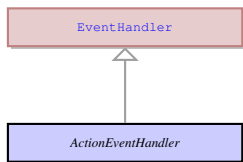
11.2 Description

Name: ActionEventHandler

This is an event handler that triggers the specified 4GL action. A 4GL action is identified by its name. This event handler can be assigned to any of the available events, like OnInvoke. When this event handler is invoked, the action is triggered and the 4GL code that references this action name is executed.

Parent: EventHandler - This is common class for all the specific event handler types.

This is an event handler that triggers the specified 4GL action. A 4GL action is identified by its name. This event handler can be assigned to any of the available events, like OnInvoke. When this event handler is invoked, the action is triggered and the 4GL code that references this action name is executed.

11.3 Fields

Name	Type	Description
ActionName	optional String	It is a string that contains the name of the 4GL action. It can consist of any printable symbols.

12 ActionStyle

12.1 Diagram
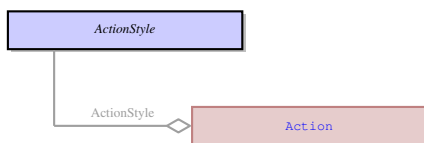
12.2 Description

Name: ActionStyle
No information
No parents.
No information

12.3 Options

Name	Description
Icon	Not described yet
Label	A label that can contain text or image; normally applied to Label UI element.
IconLabel	Not described yet

12.4 Referenced in

- ActionStyle field in optional Action - No information

13 Add Not-referenced

13.1 Diagram
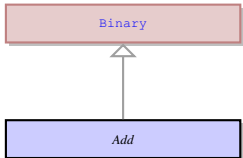
13.2 Description

Name: Add

It is a part of the conditional 4GL display attributes - +. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator +.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - +. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator +.

14 And Not-referenced

14.1 Diagram

14.2 Description

Name: And

It is a part of the conditional 4GL display attributes - AND. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of logical union AND.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - AND. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of logical union AND.

15 Background

15.1 Diagram
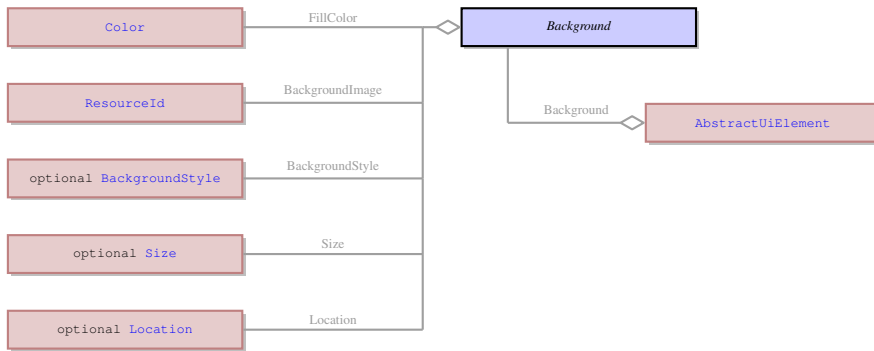
15.2 Description

Name: Background

This element determines the colour of the background of an element, the background image, if any, and its properties.

No parents.

This element determines the colour of the background of an element, the background image, if any, and its properties.

15.3 Fields

Name	Type	Description
FillColor	optional Color	The color of the background of an element.
BackgroundImage	optional ResourceId	A background image for the UI element.
BackgroundStyle	optional BackgroundStyle	The position of the background image of the UI element.
Size	optional Size	The size of the UI element in pixels that.
Location	optional Location	The location of the UI element specified in pixels.

15.4 Referenced in

- Background field in optional AbstractUiElement - This element determines the colour of the background of an element, the background image, if any, and its properties.

16 BackgroundStyle

16.1 Diagram
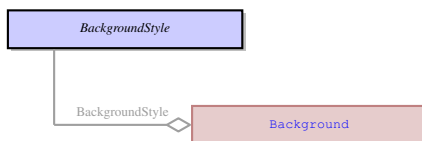
16.2 Description

Name: BackgroundStyle

This element determines the position and arrangement of the background image of the UI element. It is not applicable if the background of an element does not have a background image specified.

No parents.

This element determines the position and arrangement of the background image of the UI element. It is not applicable if the background of an element does not have a background image specified.

16.3 Options

Name	Description
Default	The window size is the size with which it was opened or which was set after opening by 4GL or graphical theme means.
Normal	The background image is not changed, it retains its size, unless Size is applied, and is placed in the top left colour, if the Location is not set.

Stretched	The background image is stretched to fill whole UI element without preserving the aspect ratio. Its size and location cannot be changed.
Tiled	The background image retains its original size, but it is multiplied and used to cover the whole UI element area in a form of tiles. The size and location of the image cannot be changed.
Centered	The background image retains its original size and is placed in the center of the UI element. Its size and location cannot be changed.
Uniform	The background image is stretched to fill whole UI element while preserving the aspect ratio. Some margin will be added to the image. Its size and location cannot be changed.
UniformToFill	The background image is stretched to fill whole UI element while preserving the aspect ratio. No margin will be added to the image. Its size and location cannot be changed.

16.4 Referenced in

- BackgroundStyle field in optional Background - This element determines the position and arrangement of the background image of the UI element. It is not applicable if the background of an element does not have a background image specified.

17 BatchEventHandler Not-referenced

17.1 Diagram

17.2 Description

Name: BatchEventHandler

This is an event handler which allows a UI element to have more than one event handler assigned to one event.

Parent: EventHandler - This is common class for all the specific event handler types.

This is an event handler which allows a UI element to have more than one event handler assigned to one event.

17.3 Fields

Name	Type	Description
Handlers	list of EventHandler	A set of event handlers assigned to a single event.

18 Between Not-referenced

18.1 Diagram
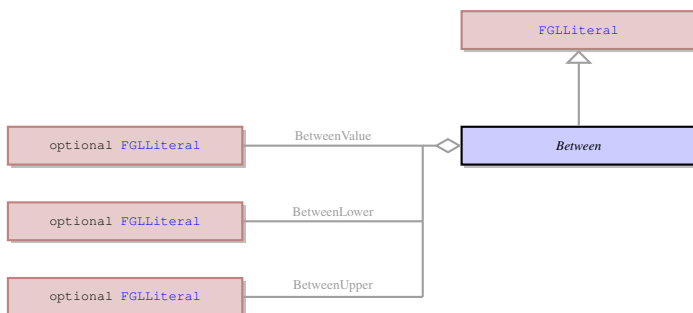
18.2 Description

Name: Between

It is a part of the conditional 4GL display attributes - BETWEEN...AND. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of membership, e.g.: field_tag BETWEEN 1 AND 100.

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

It is a part of the conditional 4GL display attributes - BETWEEN...AND. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of membership, e.g.: field_tag BETWEEN 1 AND 100.

18.3 Fields

Name	Type	Description
BetweenValue	optional FGLLiteral	This is the value which membership is tested by the operator.
BetweenLower	optional FGLLiteral	This is the value located before AND in "var BETWEEN val1 AND val2".
BetweenUpper	optional FGLLiteral	This is the value located after AND in "var BETWEEN val1 AND val2".

19   BevelBorder Not-referenced

19.1   Diagram
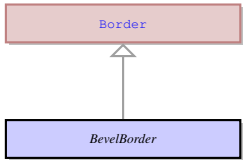
19.2   Description

Name: BevelBorder

This UI element is used to apply a custom bevel border to any concrete UI element. The border can be lowered or raised, its thickness or colour can be changed.

Parent: Border - It defines the properties of a custom border around a concrete UI element. The properties border can be applied to one of the three border types: BevelBorder , EtchedBorder , and LineBorder .

This UI element is used to apply a custom bevel border to any concrete UI element. The border can be lowered or raised, its thickness or colour can be changed.

19.3   Fields

Name	Type	Description
IsRaised	Bool	This property specifies whether custom the bevel or etched border should be raised or lowered.

20 Binary Not-referenced

20.1 Diagram
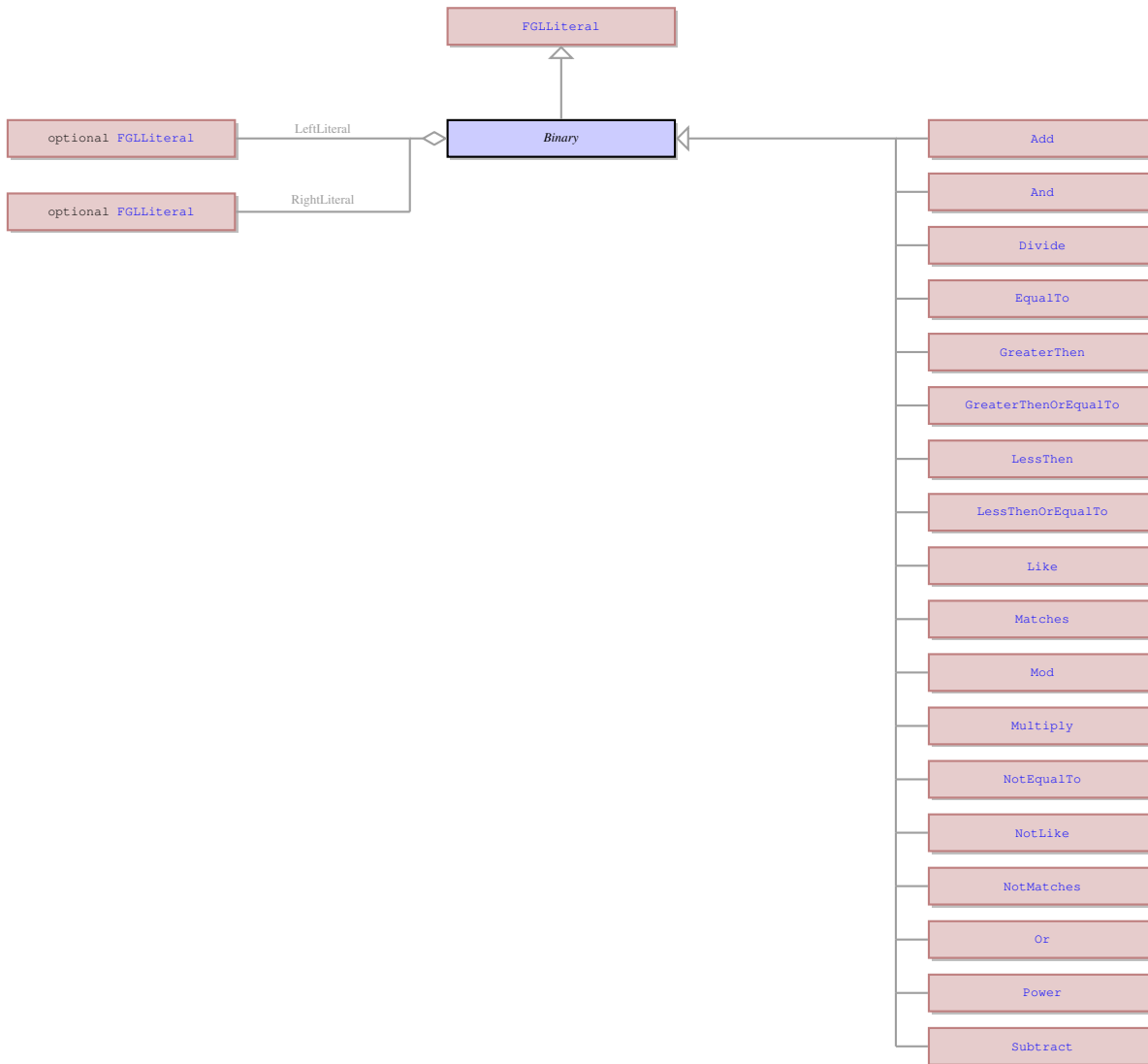
20.2 Description

Name: Binary

These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

20.3 Children

- Add - It is a part of the conditional 4GL display attributes - +. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator +.

- And - It is a part of the conditional 4GL display attributes - AND. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of logical union AND.

- Divide - It is a part of the conditional 4GL display attributes - /. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator /.

- EqualTo - It is a part of the conditional 4GL display attributes - =. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator =.

- GreaterThen - It is a part of the conditional 4GL display attributes - ¿. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator ¿.
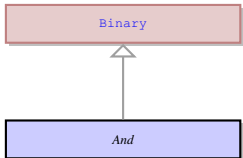
- GreaterThenOrEqualTo - It is a part of the conditional 4GL display attributes - ¿=. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator ¿=.

- LessThen - It is a part of the conditional 4GL display attributes - ¡. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator ¡.

- LessThenOrEqualTo - It is a part of the conditional 4GL display attributes - ¡=. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator ¡=.

- Like - It is a part of the conditional 4GL display attributes - LIKE. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of string comparison LIKE.

- Matches - It is a part of the conditional 4GL display attributes - MATCHES. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of string comparison MATCHES.

- Mod - It is a part of the conditional 4GL display attributes - MOD. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of module MOD.

- Multiply - It is a part of the conditional 4GL display attributes - *. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator *.

- NotEqualTo - It is a part of the conditional 4GL display attributes - !=. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator !=.

- NotLike - It is a part of the conditional 4GL display attributes - NOT LIKE. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of string comparison NOT LIKE.

- NotMatches - It is a part of the conditional 4GL display attributes - NOT MATCHES. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of string comparison NOT MATCHES.

- Or - It is a part of the conditional 4GL display attributes - OR. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of logical intersection OR.

- Power - It is a part of the conditional 4GL display attributes - **. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of exponentiation **.

- Subtract - It is a part of the conditional 4GL display attributes - -. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator -.

20.4   Fields

Name	Type	Description
LeftLiteral	optional FGLLiteral	This is the left operand of a binary operator.
RightLiteral	optional FGLLiteral	This is the right operand of a binary operator.

21   BlobViewer Not-referenced

21.1   Diagram
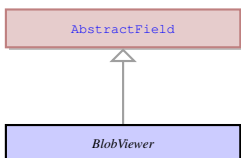
21.2   Description

Name: BlobViewer

This UI element is used to display and edit BYTE or TEXT values e.g a text or a picture.

Parent: AbstractField - This UI element represents an abstract field from which all the form widgets inherit their properties. This abstract UI element unites all form fields - the form elements that can accept and display data - as opposed tp form containers - elements that determine the form layout.

This UI element is used to display and edit BYTE or TEXT values e.g a text or a picture.

21.3 Fields

Name	Type	Description
Editor	optional String	Specifies the program to be used for opening and editing the BYTE or TEXT value.

22 BooleanLiteral Not-referenced

22.1 Diagram
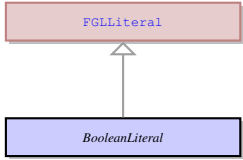
22.2 Description

Name: BooleanLiteral

It is a literal that can have values 'true' and 'false' only.

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

It is a literal that can have values 'true' and 'false' only.

22.3 Fields

Name	Type	Description
BooleanValue	Bool	This is a boolean value (either true or false).

23 Border Not-referenced

23.1 Diagram
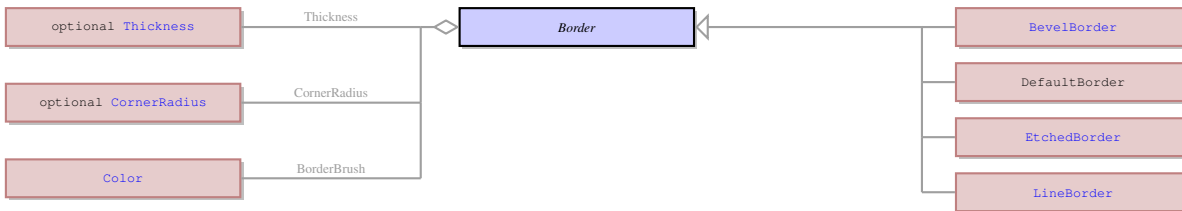
23.2 Description

Name: Border

It defines the properties of a custom border around a concrete UI element. The properties border can be applied to one of the three border types: BevelBorder , EtchedBorder , and LineBorder .

No parents.

It defines the properties of a custom border around a concrete UI element. The properties border can be applied to one of the three border types: BevelBorder , EtchedBorder , and LineBorder .

23.3 Children

- BevelBorder - This UI element is used to apply a custom bevel border to any concrete UI element. The border can be lowered or raised, its thickness or colour can be changed.

- EtchedBorder - It sets a custom etched border around the UI element. The border can be raised and lowered, its colour can be changed.

- LineBorder - This UI element is used to apply a custom line border to any concrete UI element. A line border is just a line of the defined thickness and colour that surrounds the element. The line border allows the CornerRadius to be set to round the corners.

23.4 Fields

Name	Type	Description
Thickness	optional Thickness	It defines the thickness of a border, or the space left empty for a margin or padding in pixels.
CornerRadius	optional CornerRadius	The radius of a corner of a custom border around the UI element. It is used to make the border corners rounded.
BorderBrush	optional Color	It specifies the colour of the border. Typically applied to LineBorder .

24 BorderPanel Not-referenced

24.1 Diagram
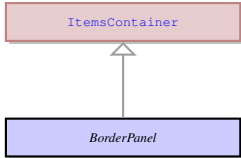
24.2 Description

Name: BorderPanel

It is a concrete UI element - a container for arranging the layout of other UI elements. Other UI elements can be located either alongside the panel borders or in its center, thus this panel can incorporate up to 5 elements - 1 for each side and 1 in the center. The elements are stretched by default, one element can take up more than one position cell. The position of an element inside the Border panel (that is which of the) is defined by the BorderPanelItemLocation property of this element.

Parent: ItemsContainer - The containers that can contain any number of UI elements inherit their properties from the ItemsContainer UI element. These are the containers that can contain any number of form fields and other containers, as opposed to the containers belonging to ElementContainer class.

It is a concrete UI element - a container for arranging the layout of other UI elements. Other UI elements can be located either alongside the panel borders or in its center, thus this panel can incorporate up to 5 elements - 1 for each side and 1 in the center. The elements are stretched by default, one element can take up more than one position cell. The position of an element inside the Border panel (that is which of the) is defined by the BorderPanelItemLocation property of this element.

25 BorderPanelItemLocation

25.1 Diagram
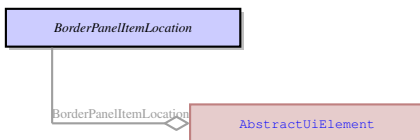
25.2 Description

Name: BorderPanelItemLocation

This property is applicable only if the UI element is located inside the BorderPanel container. It indicates which part of the border panel the element occupies. A Border panel can have 5 positions that elements can take. One element can take several adjacent positions at once. They cannot overlap.

No parents.

This property is applicable only if the UI element is located inside the BorderPanel container. It indicates which part of the border panel the element occupies. A Border panel can have 5 positions that elements can take. One element can take several adjacent positions at once. They cannot overlap.

25.3 Options

Name	Description
Center	The element is located without adjoining to any of the 4 borders of the container, in the space in the middle of the panel.
Left	The element is located adjoined to the left side of the border panel.
Right	The element is located adjoined to the right side of the border panel.
Top	The element is located adjoined to the top border of the border panel.
Bottom	The element is located adjoined to the bottom border of the border panel.

25.4 Referenced in

- BorderPanelItemLocation field in optional AbstractUiElement - This property is applicable only if the UI element is located inside the BorderPanel container. It indicates which part of the border panel the element occupies. A Border panel can have 5 positions that elements can take. One element can take several adjacent positions at once. They cannot overlap.

26 Browser Not-referenced

26.1 Diagram
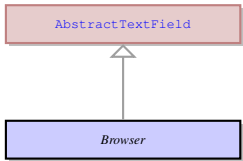
26.2 Description

Name: Browser

It is a concrete UI element that encompasses a built-in web browser with a somewhat limited functionality. It is used to display web pages, but can also work as a file explorer, display contents of files (e.g. text or image files), etc.

Parent: AbstractTextField - It is an abstract UI element, which unites a subset of AbstractStringField elements with the exception of TextArea , ComboBox , and Button . Typically it includes the UI elements which allow entering values, like normal text fields, and usually are only one line wide.

It is a concrete UI element that encompasses a built-in web browser with a somewhat limited functionality. It is used to display web pages, but can also work as a file explorer, display contents of files (e.g. text or image files), etc.

27 Button Not-referenced

27.1 Diagram
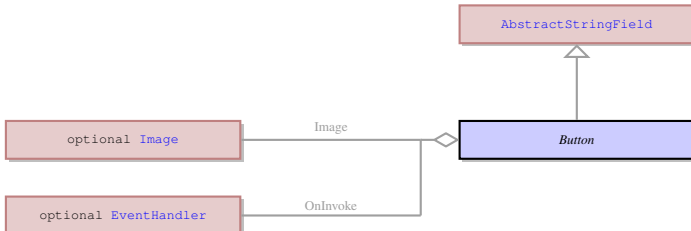
27.2 Description

Name: Button

It is a clickable concrete UI element in a form of a button that is typically used to trigger various events when it is pressed and/or released. It can have a text label or an image on it.

Parent: AbstractStringField - It is an abstract UI element, which unites the concrete UI elements that accept a character string as their value. Most of the concrete UI elements that are not containers inherit their properties from this element.

It is a clickable concrete UI element in a form of a button that is typically used to trigger various events when it is pressed and/or released. It can have a text label or an image on it.

27.3 Fields

Name	Type	Description
IsToggleButton	Bool	Determines that the button should be released automatically after it was pressed if set to FALSE (the default value). If set to TRUE - the button is treated as a toggle button which does not get released automatically. Once it was clicked it remains pressed and can only be released with another click.
Image	optional Image	It specifies the icon that should be displayed to the button instead of the inscription. The button is resized to the size of the icon applied.

OnInvoke	optional EventHandler	The event which is triggered when the UI element is invoked. It can be invoked by mouse click, by pressing Enter, or in some cases Space, when the cursor is in the element.
AllowNewlines	Bool	This property specifies whether the Enter key will be used to move to another form element at runtime (if the value is FALSE), or it will create a newline symbol inside the current field (if the value is TRUE). It is typically applied for the TextArea element.

28 Calendar Not-referenced

28.1 Diagram

28.2 Description

Name: Calendar

It is a concrete UI element that serves for displaying and inputting dates and has a drop-down lookup calendar for graphical date selection.

Parent: AbstractTextField - It is an abstract UI element, which unites a subset of AbstractStringField elements with the exception of TextArea , ComboBox , and Button . Typically it includes the UI elements which allow entering values, like normal text fields, and usually are only one line wide.

It is a concrete UI element that serves for displaying and inputting dates and has a drop-down lookup calendar for graphical date selection.

28.3 Fields

Name	Type	Description
LabelText	optional String	No information
HelperText	optional String	No information
PlaceholderText	optional String	No information

29 Canvas Not-referenced

29.1 Diagram

29.2 Description

Name: Canvas

It is a concrete UI element that serves as a container for SVG images and allows interactions with such images.

Parent: AbstractField - This UI element represents an abstract field from which all the form widgets inherit their properties. This abstract UI element unites all form fields - the form elements that can accept and display data - as opposed tp form containers - elements that determine the form layout.

It is a concrete UI element that serves as a container for SVG images and allows interactions with such images.

29.3 Fields

Name	Type	Description
Image	optional Image	It specifies the SVG image that should be displayed to the canvas area.

30 Century Not-referenced

30.1 Diagram

Century

30.2 Description

Name: Century

This enum defines how the year value in dated should be extended, if it has fewer than 4 numbers. It is applied to form elements of the DATE date type.

No parents.

This enum defines how the year value in dated should be extended, if it has fewer than 4 numbers. It is applied to form elements of the DATE date type.

30.3 Options

Name	Description
Closest	It means that a year value with fewer than 4 digits will be expanded with the digits from the century which is closest to the current date. E.g. 25 would be expanded as 2025, but 75 would be expanded as 1975.
Past	It means that a year value with fewer than 4 digits will be expanded with the digits from the century which is closest to the current date in the past. E.g. 25 would be expanded as 1925, but 75 would be expanded as 1975.
Future	It means that a year value with fewer than 4 digits will be expanded with the digits from the century which is closest to the current date in the future. E.g. 25 would be expanded as 2025, but 75 would be expanded as 2075.
Current	It means that a year value with fewer than 4 digits will be expanded with the digits from the current century. E.g. 25 would be expanded as 2025, but 75 would be expanded as 2075.

31 CheckBox Not-referenced

31.1 Diagram

AbstractBoolField
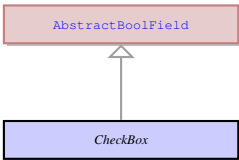
CheckBox

31.2 Description

Name: CheckBox

It is a concrete UI element that consists of a single check box and a label attached to it. It can be in only one of 2 states at a time - either checked or unchecked. Changing of the state can either change the value that will be written to the underlying variable, or trigger an event handler.

Parent: AbstractBoolField - It is an abstract UI element, which unites the concrete UI elements that can be in one of the two states: enabled (TRUE) or disabled (FALSE). The concrete UI elements that inherit their properties from the AbstractBoolField are CheckBox .

It is a concrete UI element that consists of a single check box and a label attached to it. It can be in only one of 2 states at a time - either checked or unchecked. Changing of the state can either change the value that will be written to the underlying variable, or trigger an event handler.

31.3 Fields

Name	Type	Description
Required	Bool	No information

32 ClientSideExecEventHandler Not-referenced

32.1 Diagram

```
EventHandler
     △
     |
ClientSideExecEventHandler
```
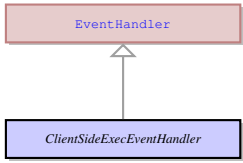
32.2 Description

Name: ClientSideExecEventHandler
    No information
    Parent: EventHandler - This is common class for all the specific event handler types.
    No information

32.3 Fields

Name	Type	Description
ExecCommand	optional String	No information
ExecParam	optional String	No information

33 Color Not-referenced

33.1 Diagram

```
Color  ◁───────  CustomizedColor
                  DefaultColor
                  SystemColor
```

33.2 Description

Name: Color
    It is the root element to all color properties that can be applied to any UI element.
    No parents.
    It is the root element to all color properties that can be applied to any UI element.

33.3 Children

- CustomizedColor - This enum defines a custom color in the RGB encoding plus the transparency.

- SystemColor - The system color defines a list of preset colours that can be applied to widgets, as opposed to the custom colour where the user needs to specify RGB of the color.

34 ComboBox Not-referenced

34.1 Diagram

```
                        AbstractStringField
                               △
                               |
optional ComboBoxItem  ──ComboBoxItems──◇  ComboBox
ToCase  ────────────ToCase────────────
```
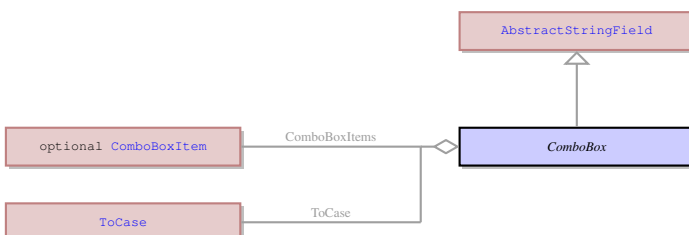
34.2   Description

Name: ComboBox

It is a concrete UI element that has a form of a text field with a drop-down list. It can be restricted to accepting only values from this drop-down list, or it can be set to accept values from the list and the custom values entered by the user. Only one item from the drop-down combobox list can be selected at a time.

Parent: AbstractStringField - It is an abstract UI element, which unites the concrete UI elements that accept a character string as their value. Most of the concrete UI elements that are not containers inherit their properties from this element.

It is a concrete UI element that has a form of a text field with a drop-down list. It can be restricted to accepting only values from this drop-down list, or it can be set to accept values from the list and the custom values entered by the user. Only one item from the drop-down combobox list can be selected at a time.

34.3   Fields

Name	Type	Description
ComboBoxItems	list of ComboBoxItem	The set values that should be present in the drop-down list of a combo box.
Editable	optional Bool	It indicates that the combo box accepts values that are not in its drop-down list.
ToCase	ToCase	This property specifies the case of a UI element. It can be applied to any UI element that allows entering text from keyboard. By default its value is None, meaning that the case of the letters does not change and remains as they were inputted.
MaxLength	optional Int	It specifies the maximum length in bytes allowed for entering into the filed. Its value is normally taken from the data type and size of the variable linked to the field.
Autonext	Bool	If enabled, moves the cursor to the next field during input automatically, when the MaxLength of the current field is met.
Required	Bool	No information
LabelText	optional String	No information
HelperText	optional String	No information

35   ComboBoxItem Not-referenced

35.1   Diagram
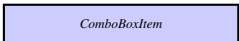
ComboBoxItem

35.2   Description

Name: ComboBoxItem

It is single item in a combo box drop-down list. If it is selected during input, its value is recorded into the variable linked to the combo box.

No parents.

It is single item in a combo box drop-down list. If it is selected during input, its value is recorded into the variable linked to the combo box.

35.3   Fields

Name	Type	Description
Text	optional String	This is the value of the combobox item, which is recorded to the variable linked to it after the input.

36   ComponentProperty Not-referenced

36.1   Diagram

ComponentProperty

36.2 Description

Name: ComponentProperty

This is the property of a WebComponent UI element. Each property is defined by the HTML file that describes the web component. No parents.

This is the property of a WebComponent UI element. Each property is defined by the HTML file that describes the web component.

36.3 Fields

Name	Type	Description
PName	optional String	It specifies the name of a web component property.
PValue	optional String	It specifies the value of a web component property.

37 ContentFilter Not-referenced

37.1 Diagram

ContentFilter

37.2 Description

Name: ContentFilter

No information

No parents.

No information

37.3 Fields

Name	Type	Description
ContentFilterTable	optional String	No information
ContentFilterField	optional String	No information

38 CoordPanel Not-referenced

38.1 Diagram

ItemsContainer
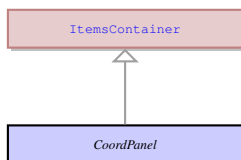
CoordPanel

38.2 Description

Name: CoordPanel

This is a container the location of the elements inside which is determined by the coordinates of the component. The coordinates are stored in pixels and specify the Location on the coord panel where the top left corner of the child element is placed.

Parent: ItemsContainer - The containers that can contain any number of UI elements inherit their properties from the ItemsContainer UI element. These are the containers that can contain any number of form fields and other containers, as opposed to the containers belonging to ElementContainer class.

This is a container the location of the elements inside which is determined by the coordinates of the component. The coordinates are stored in pixels and specify the Location on the coord panel where the top left corner of the child element is placed.

39 CornerRadius

39.1 Diagram

CornerRadius

CornerRadius — Border

CornerRadius — Separator
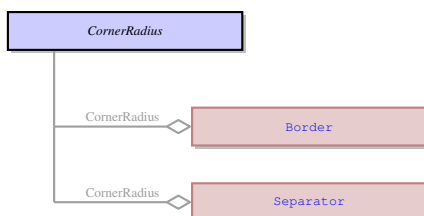
39.2 Description

Name: CornerRadius

This enum specifies the radius of a corner of a custom border around the UI element. It is used to make the border corners rounded. It can be applied only to LineBorder border type. All four corners can have different corner radius.

No parents.

This enum specifies the radius of a corner of a custom border around the UI element. It is used to make the border corners rounded. It can be applied only to LineBorder border type. All four corners can have different corner radius.

39.3 Fields

Name	Type	Description
BottomLeft	Float	The bottom left corner of the border frame.
BottomRight	Float	The bottom right corner of the border frame.
TopLeft	Float	The top left corner of the border frame.
TopRight	Float	The top right corner of the border frame.

39.4 Referenced in

- CornerRadius field in optional Border - This enum specifies the radius of a corner of a custom border around the UI element. It is used to make the border corners rounded. It can be applied only to LineBorder border type. All four corners can have different corner radius.

- CornerRadius field in optional Separator - This enum specifies the radius of a corner of a custom border around the UI element. It is used to make the border corners rounded. It can be applied only to LineBorder border type. All four corners can have different corner radius.

40 Current Not-referenced

40.1 Diagram
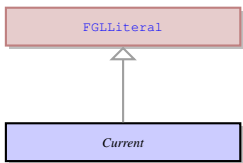
40.2 Description

Name: Current

It is a part of the conditional 4GL display attributes - CURRENT(). The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals.

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

It is a part of the conditional 4GL display attributes - CURRENT(). The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals.

41 CurrentTime Not-referenced

41.1 Diagram
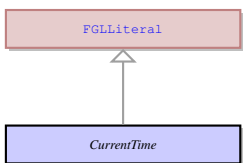
41.2 Description

Name: CurrentTime

It is a part of the conditional 4GL display attributes - TIME(). The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals.

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

It is a part of the conditional 4GL display attributes - TIME(). The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals.

42 Cursor

42.1 Diagram
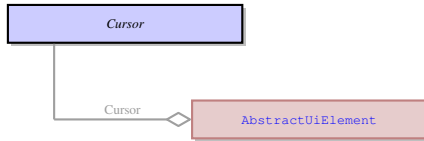
42.2 Description

Name: Cursor

It defines the animation the mouse cursor should have when hovering over the UI element for which this enum is specified. The cursor animation at runtime is selected on the basis of the cursors available for the system or for the browser, if the web client is used.

No parents.

It defines the animation the mouse cursor should have when hovering over the UI element for which this enum is specified. The cursor animation at runtime is selected on the basis of the cursors available for the system or for the browser, if the web client is used.

42.3 Options

Name	Description
Arrow	The default arrow cursor.
Cross	The cursor in a form of a cross.
IBeam	The cursor in a form of a vertical line.
SizeAll	The cursor in a form of a cross with arrows at all 4 ends.
SizeNESW	The cursor in a form of a diagonal line in direction from top right to bottom left with arrows on both ends .
SizeNS	The cursor in a form of a vertical line with arrows on both ends .
SizeNWSE	The cursor in a form of a diagonal line in direction from top left to bottom right with arrows on both ends .
SizeWE	The cursor in a form of a horizontal line with arrows on both ends .
UpArrow	The cursor in a form of a vertical line with an arrow pointing upwards .
WaitCursor	The default waiting cursor of the system (e.g. in Windows XP - glass clock, in Windows 7 - a blue ring).
Help	The default help cursor of the system (normally in a form of a question mark).
HSplit	The default cursor that appears when the mouse is positioned over a horizontal splitter bar.
VSplit	The default cursor that appears when the mouse is positioned over a vertical splitter bar.
Hand	The default hand cursor.

42.4 Referenced in

- Cursor field in optional AbstractUiElement - It defines the animation the mouse cursor should have when hovering over the UI element for which this enum is specified. The cursor animation at runtime is selected on the basis of the cursors available for the system or for the browser, if the web client is used.

43 CustomizedColor Not-referenced

43.1 Diagram
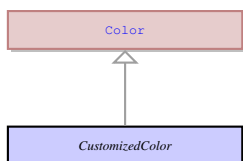
43.2 Description

Name: CustomizedColor

This enum defines a custom color in the RGB encoding plus the transparency.

Parent: Color - It is the root element to all color properties that can be applied to any UI element.

This enum defines a custom color in the RGB encoding plus the transparency.

43.3   Fields

Name	Type	Description
RedColor	Int	The value of the red colour in the RGB color model (0-255).
GreenColor	Int	The value of the green colour in the RGB color model (0-255).
BlueColor	Int	The value of the blue colour in the RGB color model (0-255).
Alpha	Int	The value of the transparency applied to the color. 0 - completely transparent. 255 - completely solid color.

44   DataType Not-referenced

44.1   Diagram
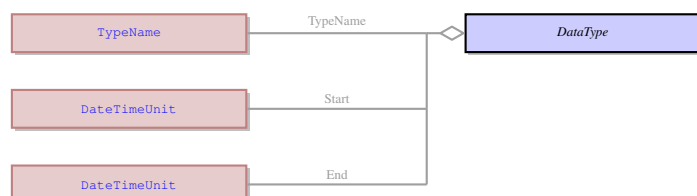
44.2   Description

Name: DataType

This is the data type of a form element. It contains all the required information about a datatype - its name, precision, scale, etc. No parents.

This is the data type of a form element. It contains all the required information about a datatype - its name, precision, scale, etc.

44.3   Fields

Name	Type	Description
TypeName	TypeName	The name of the datatype of the field.
Start	optional DateTimeUnit	This is the first part of the DATETIME or INTERVAL qualifier. It should have the form of a time unit - YEAR, DAY, HOUR, etc. - this is the first qualifier in a pair of qualifiers that usually look like this: YEAR (first qualifier) to SECOND. It must be a bigger or the same time unit than the End qualifier. In the INTERVAL data type there can be only two separate sets of qualifiers - an interval may only be from YEAR to MONTH (or a subset of these values, e.g. YEAR to YEAR), or DAY to FRACTION (or a subset of these values, e.g. HOUR to SECOND).
End	optional DateTimeUnit	This is the second part of the DATETIME or INTERVAL qualifier. It should have the form of a time unit - YEAR, DAY, HOUR, etc. - this is the first qualifier in a pair of qualifiers that usually look like this: YEAR to HOUR (second qualifier). It must be a smaller or the same time unit than the Start qualifier. In the INTERVAL data type there can be only two separate sets of qualifiers - an interval may only be from YEAR to MONTH (or a subset of these values, e.g. YEAR to YEAR), or DAY to FRACTION (or a subset of these values, e.g. HOUR to SECOND).
Scale	optional Int	It specifies the scale (the number of decimal places) for the fixed point decimal data types. It is used to specify the precision of the FRACTION time unit specicied as the End qualifier in the DATE-TIME datatype specification. It is also used as the precision of the End qualifier in the INTERVAL data type.
Precision	optional Int	It specifies the precision for the fixed point decimal data types. It is also used to specify the precision of the Start qualifier of the INTERVAL data type specification.

45 Database

45.1 Diagram
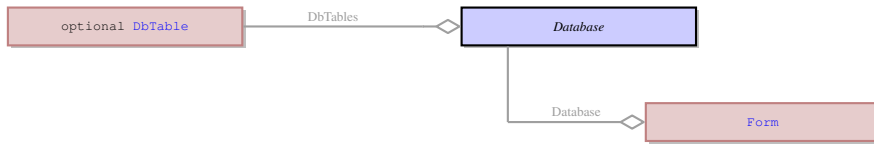
45.2 Description

Name: Database

It the database to which the current form is linked. The form only needs the information about the data types of the columns to set the data types of the fields linked to the columns. It also gets the NOT NULL attribute from the database column properties.

No parents.

It the database to which the current form is linked. The form only needs the information about the data types of the columns to set the data types of the fields linked to the columns. It also gets the NOT NULL attribute from the database column properties.

45.3 Fields

Name	Type	Description
Name	Name	This is the name of a database linked to a form.
Alias	optional String	This is an alias of a database linked to a form.
DbTables	list of DbTable	No information

45.4 Referenced in

- Database field in optional Form - It the database to which the current form is linked. The form only needs the information about the data types of the columns to set the data types of the fields linked to the columns. It also gets the NOT NULL attribute from the database column properties.

46 DateLiteral Not-referenced

46.1 Diagram
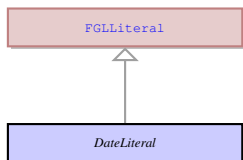
46.2 Description

Name: DateLiteral

This a 4GL literal whose value is date-formatted. It corresponds in the format to the requirements of the DATE 4GL data type. E.g. "12/23/2012".

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

This a 4GL literal whose value is date-formatted. It corresponds in the format to the requirements of the DATE 4GL data type. E.g. "12/23/2012".

46.3 Fields

Name	Type	Description
YearVal	Int	This value represents the year in the DATE value, e.g. 1975.
MonthVal	Int	This value represents the number of months in the DATE value. Shaould be integers from 1 to 12.
DayVal	Int	This value represents the number of days in the DATE value. Should be integers from 1 to 31.

47 DateTimeLiteral Not-referenced

47.1 Diagram
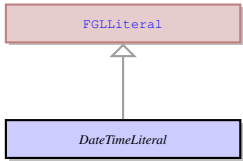
47.2 Description

Name: DateTimeLiteral

This a 4GL literal whose value presented in the format of date and time. It corresponds in the format to the requirements of the DATETIME 4GL data type. E.g. "DATETIME (2012-10-30 07:45:32.150) YEAR TO FRACTION(3)".

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

This a 4GL literal whose value presented in the format of date and time. It corresponds in the format to the requirements of the DATETIME 4GL data type. E.g. "DATETIME (2012-10-30 07:45:32.150) YEAR TO FRACTION(3)".

47.3 Fields

Name	Type	Description
YearValOpt	optional Int	This value represents the year in the DATETIME or INTERVAL VALUE. This value is optional and may be present or absent depending on the Start and End qualifiers.
MonthValOpt	optional Int	This value represents the number of months in the DATETIME or INTERVAL value. This value is optional and may be present or absent depending on the Start and End qualifiers. It includes integers from 1 to 12.
DayValOpt	optional Int	This value represents the number of days in the datetime or interval value. This value is optional and may be present or absent depending on the Start and End qualifiers. It includes integers from 1 to 31.
HourValOpt	optional Int	This value represents the number of hours in the DATETIME or INTERVAL value. This value is optional and may be present or absent depending on the Start and End qualifiers. It includes integers from 0 to 23.
MinuteValOpt	optional Int	This value represents the number of minutes in the DATETIME or INTERVAL value. This value is optional and may be present or absent depending on the Start and End qualifiers. It includes integers from 0 to 59.
SecondValOpt	optional Int	This value represents the number of seconds in the DATETIME or INTERVAL value. This value is optional and may be present or absent depending on the Start and End qualifiers. It includes integers from 0 to 59.
FractionValOpt	optional Int	This value represents the number of fractions of second in the DATETIME or INTERVAL value. This value is optional and may be present or absent depending on the Start and End qualifiers.
FromValOpt	optional String	It represents the first part of the qualifier.
ToValOpt	optional String	It represents the second part of the qualifier.
ScaleValOpt	optional Int	It represents scale of the End qualifier.

48 DateTimeUnit Not-referenced

48.1 Diagram

48.2 Description

Name: DateTimeUnit

This enum defines the units of time which are used in DATETIME and INTERVAL values. The largest time unit is a year, the smallest - a fraction of second.

No parents.

This enum defines the units of time which are used in DATETIME and INTERVAL values. The largest time unit is a year, the smallest - a fraction of second.

48.3 Options

Name	Description
Year	This is the YEAR time unit of a DATETIME or INTERVAL qualifier.
Month	This is the MONTH time unit of a DATETIME or INTERVAL qualifier.
Day	This is the DAY time unit of a DATETIME or INTERVAL qualifier.
Hour	This is the HOUR time unit of a DATETIME or INTERVAL qualifier.
Minute	This is the MINUTE time unit of a DATETIME or INTERVAL qualifier.
Second	This is the SECOND time unit of a DATETIME or INTERVAL qualifier.
Fraction	This is the FRACTION time unit of a DATETIME or INTERVAL qualifier.

49 DbTable Not-referenced

49.1 Diagram

49.2 Description

Name: DbTable

This is a valid name of a table in the database to which the form is linked.

No parents.

This is a valid name of a table in the database to which the form is linked.

49.3 Fields

Name	Type	Description
TableName	optional String	This is the name of the table in a database to which the form is linked.
Alias	optional String	This is a alias of a database column linked to a form.
Owner	optional String	No information

50 DecimalLiteral Not-referenced

50.1 Diagram
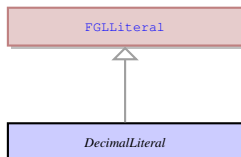
50.2 Description

Name: DecimalLiteral

These are decimal values with fixed decimal point. It corresponds in the format to the requirements of the DECIMAL and MONEY 4GL data types. E.g. 123.5436.

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

These are decimal values with fixed decimal point. It corresponds in the format to the requirements of the DECIMAL and MONEY 4GL data types. E.g. 123.5436.

50.3 Fields

Name	Type	Description
DecimalValue	Decimal	This is a value that consists of the integer part, decimal part and the decimal separator.

51 Direction

51.1 Diagram
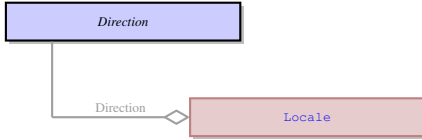
51.2 Description

Name: Direction

This enum defines the direction of the text: left to right or right to left.

No parents.

This enum defines the direction of the text: left to right or right to left.

51.3 Options

Name	Description
LTR	The text is written and displayed in the direction from left to right.
RTL	The text is written and displayed in the direction from right to left.

51.4 Referenced in

- Direction field in Locale - This enum defines the direction of the text: left to right or right to left.

52 DisplayMode Not-referenced

52.1 Diagram

52.2 Description

Name: DisplayMode

This is a set of conditional 4GL display attributes. One field can have several attributes, e.g. RED and GREEN. And the conditions will define that the value in the field will be red if it is ¡=100 and green when it is ¿=101. Typically the display mode consists of a display attribute and a logical expression the left operand of which is the tag of a form filed.

No parents.

This is a set of conditional 4GL display attributes. One field can have several attributes, e.g. RED and GREEN. And the conditions will define that the value in the field will be red if it is ¡=100 and green when it is ¿=101. Typically the display mode consists of a display attribute and a logical expression the left operand of which is the tag of a form filed.

52.3 Fields

Name	Type	Description
Condition	optional FGLLiteral	It is a boolean expression that represents the condition under which the attribute will be applied. E.g. f001 = 100 AND f003 ¿= 10. The condition typically consists of logical and membership operators, some literal values and field tags.
Appearance	String	It is the name of the 4GL display attribute which will be applied to the form field if the condition is met. E.g. it can be a color attribute (YELLOW, etc.), intensity attribute (BOLD, etc.) and others.

53 DistributedObject Not-referenced

53.1 Diagram
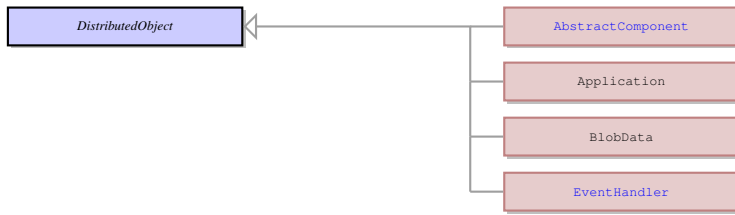
53.2 Description

Name: DistributedObject

This is the root of the UI element hierarchy.

No parents.

This is the root of the UI element hierarchy.

53.3 Children

- AbstractComponent - This is the common parent of all UI elements.

- EventHandler - This is common class for all the specific event handler types.

54 Divide Not-referenced

54.1 Diagram
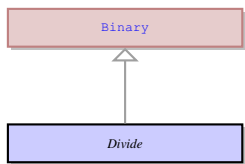
54.2 Description

Name: Divide

It is a part of the conditional 4GL display attributes - /. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator /.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - /. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator /.

55 ElementContainer Not-referenced

55.1 Diagram
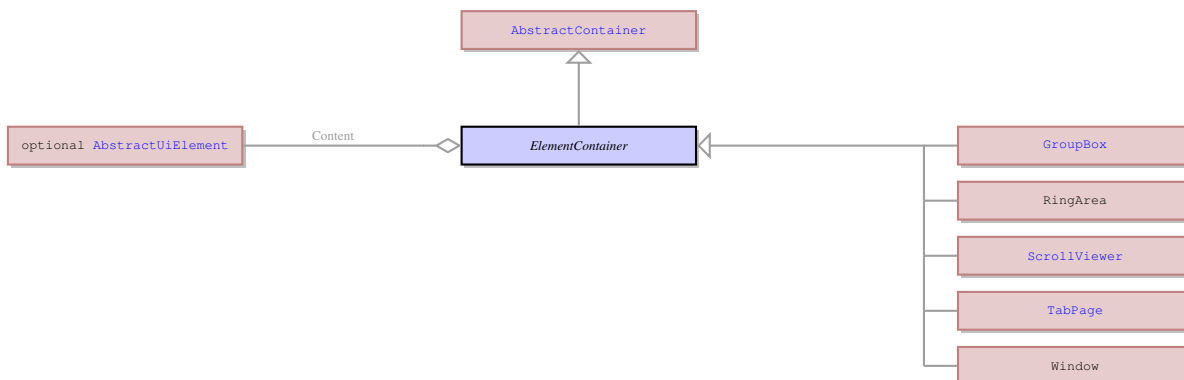
55.2 Description

Name: ElementContainer

This UI element unites all the containers which can contain exactly one element. The containers that derive from ElementContainer UI element can be logically opposed to containers derived from ItemsContainer UI element that can contain any number of elements of any type. The elements that inherit their properties from ElementContainer can encompass such elements as ring menu area or any other container. They can also contain an element belonging to ui.AbstractFiled class, but only one such element.

Parent: AbstractContainer - This UI element represents an abstract container from which all the form containers their properties. This abstract UI element unites all form containers - elements that determine the form layout.

This UI element unites all the containers which can contain exactly one element. The containers that derive from ElementContainer UI element can be logically opposed to containers derived from ItemsContainer UI element that can contain any number of elements of any type. The elements that inherit their properties from ElementContainer can encompass such elements as ring menu area or any other container. They can also contain an element belonging to ui.AbstractFiled class, but only one such element.

55.3 Children

- GroupBox - It is a container that groups the UI elements inside a visible border with an optional title at the top. It can contain only one other UI element. It can be another container or a form widget. Thus though it can encompass UI elements of the AbstractField group, having only one element of this group in a container makes little sense. So it should include one of the other containers first.

- ScrollViewer - It is a container the content of which can be bigger than the container. The scrollbars are used to view the content that does not fit. It can contain exactly one element. E.g. it can contain a stack panel container, the number of elements inside which can be bigger than fit the size of the Scroll Viewer.

- TabPage - This is a container that can only be placed inside the Tab container. A tab page can contain a single element of any type. Each tab page has a tab with the page title which is used to bring the page forward from the stack of other tab pages at runtime or during form modification.

55.4 Fields

Name	Type	Description
Content	AbstractUiElement	It specifies the UI element that is located inside the ElementContainer.

56 EqualTo Not-referenced

56.1 Diagram
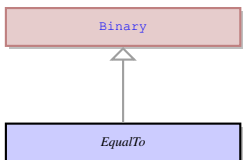
56.2 Description

Name: EqualTo

It is a part of the conditional 4GL display attributes - =. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator =.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - =. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator =.

57 EtchedBorder Not-referenced

57.1 Diagram
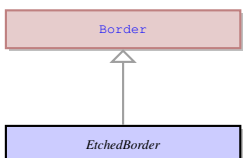
57.2 Description

Name: EtchedBorder

It sets a custom etched border around the UI element. The border can be raised and lowered, its colour can be changed.

Parent: Border - It defines the properties of a custom border around a concrete UI element. The properties border can be applied to one of the three border types: BevelBorder , EtchedBorder , and LineBorder .

It sets a custom etched border around the UI element. The border can be raised and lowered, its colour can be changed.

57.3 Fields

Name	Type	Description
IsRaised	Bool	This property specifies whether custom the bevel or etched border should be raised or lowered.

58 EventHandler Not-referenced

58.1 Diagram
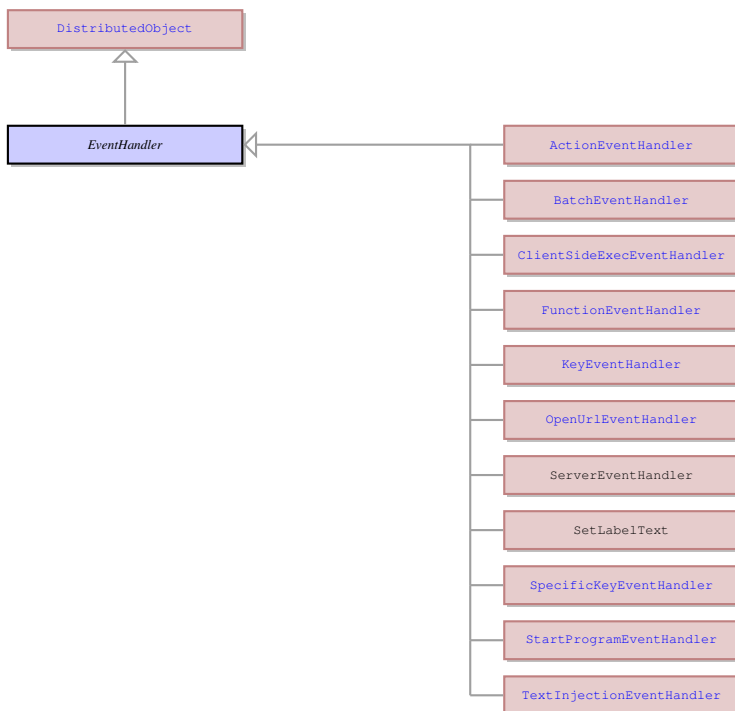
58.2 Description

Name: EventHandler

This is common class for all the specific event handler types.

Parent: DistributedObject - This is the root of the UI element hierarchy.

This is common class for all the specific event handler types.

58.3 Children

- **ActionEventHandler** - This is an event handler that triggers the specified 4GL action. A 4GL action is identified by its name. This event handler can be assigned to any of the available events, like OnInvoke. When this event handler is invoked, the action is triggered and the 4GL code that references this action name is executed.

- **BatchEventHandler** - This is an event handler which allows a UI element to have more than one event handler assigned to one event.

- **ClientSideExecEventHandler** - No information

- **FunctionEventHandler** - This event handler triggers the function specified in it, when the even to which it is assigned is invoked.

- **KeyEventHandler** - This is an event handler that is invoked when the KeyEvent is triggered.

- **OpenUrlEventHandler** - This is an event handler that can be assigned to any event. This handler opens the URL specified in the default system web browser.

- **SpecificKeyEventHandler** - This event handler specifies what event handler should be triggered when a specific key is pressed. It links the keypress with a 4GL event.

- **StartProgramEventHandler** - This event handler specifies the child 4GL program that should be launched and the parameters of this program. It is normally used for the MDI mode, but can be used in other cases.

- **TextInjectionEventHandler** - This event handler injects the text specified as its parameter into the current input widget. It can be assigned to any event.

59 EventInfo Not-referenced

59.1 Diagram
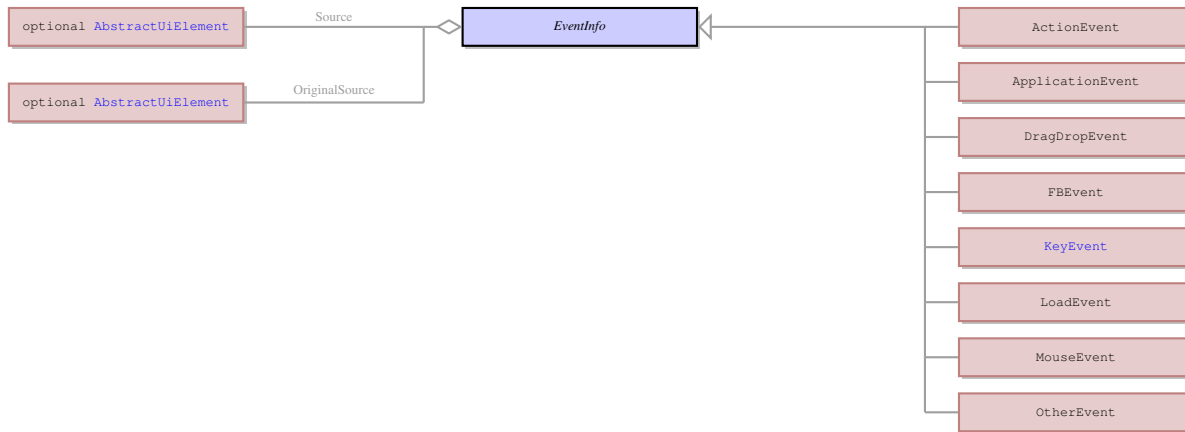
59.2 Description

Name: EventInfo

It is an abstract UI entity which is the root class for the KeyEvent . It is used to send the information to the server about the event triggered on the client side.

No parents.

It is an abstract UI entity which is the root class for the KeyEvent . It is used to send the information to the server about the event triggered on the client side.

59.3 Children

- **KeyEvent** - It is an event that is triggered when the specified key on the keyboard is pressed. This event is sent to the Application server on the keypress.

60 FGLLiteral Not-referenced

60.1 Diagram
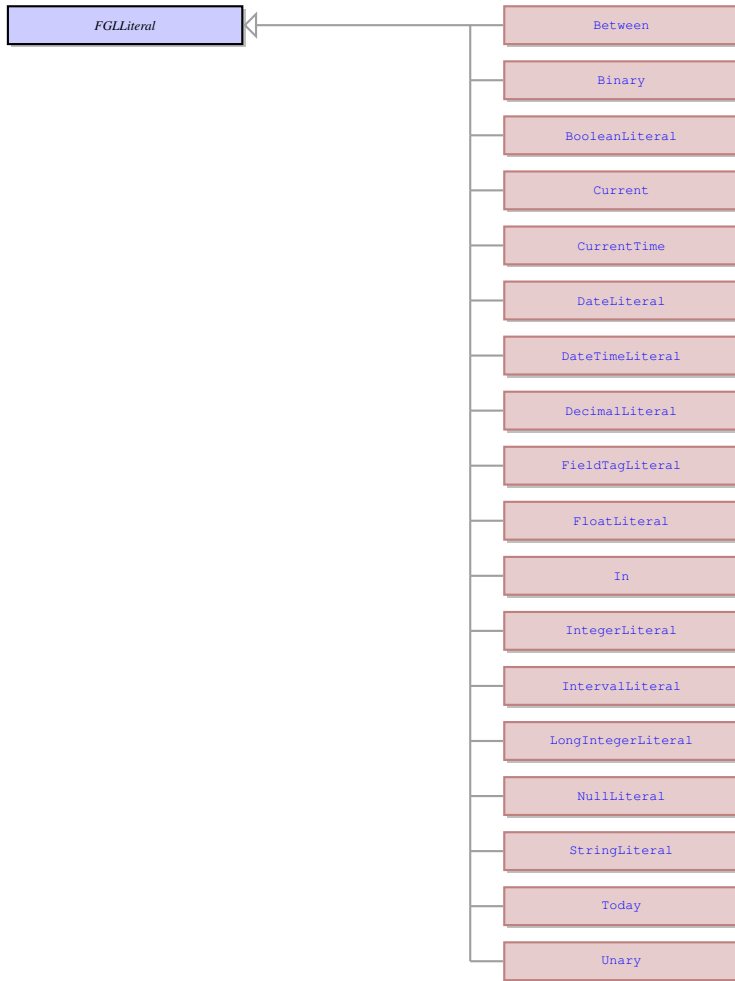
60.2 Description

Name: FGLLiteral

This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

No parents.

This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

60.3 Children

- Between - It is a part of the conditional 4GL display attributes - BETWEEN...AND. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of membership, e.g.: field_tag BETWEEN 1 AND 100.

- Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

- BooleanLiteral - It is a literal that can have values 'true' and 'false' only.

- Current - It is a part of the conditional 4GL display attributes - CURRENT(). The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals.

- CurrentTime - It is a part of the conditional 4GL display attributes - TIME(). The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals.

- DateLiteral - This a 4GL literal whose value is date-formatted. It corresponds in the format to the requirements of the DATE 4GL data type. E.g. "12/23/2012".

- DateTimeLiteral - This a 4GL literal whose value presented in the format of date and time. It corresponds in the format to the requirements of the DATETIME 4GL data type. E.g. "DATETIME (2012-10-30 07:45:32.150) YEAR TO FRACTION(3)".

- **DecimalLiteral** - These are decimal values with fixed decimal point. It corresponds in the format to the requirements of the DECIMAL and MONEY 4GL data types. E.g. 123.5436.

- **FieldTagLiteral** - This is a literal storing the tag of a field - normally in a form of a string. Field tags are introduced for the compatibility with the old text form format and are not actually used in the graphical forms.

- **FloatLiteral** - These are decimal values with floating decimal point taking up to 8 bytes maximum. It corresponds in the format to the requirements of the FLOAT and SMALLFLOAT 4GL data type. E.g. 123.5436.

- **In** - It is a part of the conditional 4GL display attributes - IN(). The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. Here is an example of the operator of membership IN(): filed_tag IN("mother", "father", "aunt").

- **IntegerLiteral** - These are integer values taking up to 4 bytes maximum. It corresponds in the format to the requirements of the INTEGER, SMALLINT and TINYINT 4GL data types. E.g. 1234.

- **IntervalLiteral** - These are interval values in different time units, e.g. INTERVAL (10-11) YEAR TO MONTH - means 10 years and 11 months. It corresponds in the format to the requirements of the INTERVAL 4GL data type.

- **LongIntegerLiteral** - These are integer values taking up to 8 bytes maximum. It corresponds in the format to the requirements of the INTEGER 4GL data type. E.g. 123456.

- **NullLiteral** - This is a 4GL literal that contains no value. This means the value of this literal is displayed as NULL or an empty space.

- **StringLiteral** - These are character strings enclosed in quotation marks. It corresponds in the format to the requirements of the STRING, CHAR, VARCHAR 4GL data types. E.g. "abcde".

- **Today** - It is a part of the conditional 4GL display attributes - TODAY(). The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals.

- **Unary** - These are binary operators. This class includes plus and minus operators that specify whether a number is positive or negative.

61   FieldTagLiteral Not-referenced

61.1   Diagram
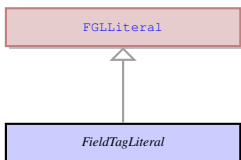
61.2   Description

Name: FieldTagLiteral

This is a literal storing the tag of a field - normally in a form of a string. Field tags are introduced for the compatibility with the old text form format and are not actually used in the graphical forms.

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

This is a literal storing the tag of a field - normally in a form of a string. Field tags are introduced for the compatibility with the old text form format and are not actually used in the graphical forms.

61.3   Fields

Name	Type	Description
FieldTagValue	optional String	It is a string that contains the text of the field tag. In text forms it could be different from the field identifier, in XML forms its value is typically the same as the element identifier.

62   FieldType Not-referenced

62.1   Diagram
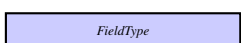
62.2 Description

Name: FieldType

This enum defines whether the field is linked to a database column. It defines the column name, if it is, and whether the name of the field should be tha same as the name of the column, or only the data type should be the same.

No parents.

This enum defines whether the field is linked to a database column. It defines the column name, if it is, and whether the name of the field should be tha same as the name of the column, or only the data type should be the same.

62.3 Options

Name	Description
FORM_ONLY	It means that the form element is not linked to a database column in any way, even if the form itself is linked to a database.
COLUMN_LIKE	It means that the form widget is linked to a table column, but it can have a name different from that of the column. It must only have the same data type as the linked column.
TABLE_COLUMN	It means that the form widget is linked to a table column, it has the same name as the database column and the same data type.
TABLE_ALIAS	It means that the form widget is linked to a table column, it has the same name as the alias assigned to the database column and the same data type.

63 FloatLiteral Not-referenced

63.1 Diagram
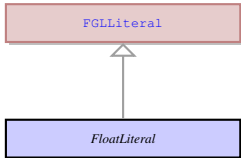
63.2 Description

Name: FloatLiteral

These are decimal values with floating decimal point taking up to 8 bytes maximum. It corresponds in the format to the requirements of the FLOAT and SMALLFLOAT 4GL data type. E.g. 123.5436.

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

These are decimal values with floating decimal point taking up to 8 bytes maximum. It corresponds in the format to the requirements of the FLOAT and SMALLFLOAT 4GL data type. E.g. 123.5436.

63.3 Fields

Name	Type	Description
FloatValue	Float	This is a value that consists of the integer part, decimal part and the floating decimal separator.

64 Font

64.1 Diagram
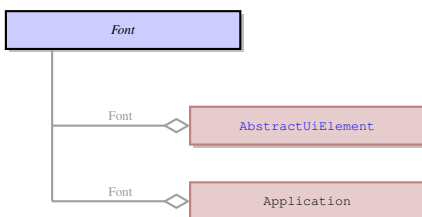
64.2 Description

Name: Font

The font to be used for any text that is a part of the UI element - either label ot inputted text.

No parents.

The font to be used for any text that is a part of the UI element - either label ot inputted text.

64.3 Fields

Name	Type	Description
Family	non-empty list of Name	This is the name of the font. E.g. Arial or Tahoma.
Bold	optional Bool	It indicates whether the text should be bold.
Italic	optional Bool	It indicates whether the text should be in italics.
Underline	optional Bool	It indicates whether the text should be underlined.
FontSize	optional Int	It specifies the font size.

64.4 Referenced in

- Font field in optional AbstractUiElement - The font to be used for any text that is a part of the UI element - either label ot inputted text.

- Font field in optional Application - The font to be used for any text that is a part of the UI element - either label ot inputted text.

65 Form Not-referenced

65.1 Diagram
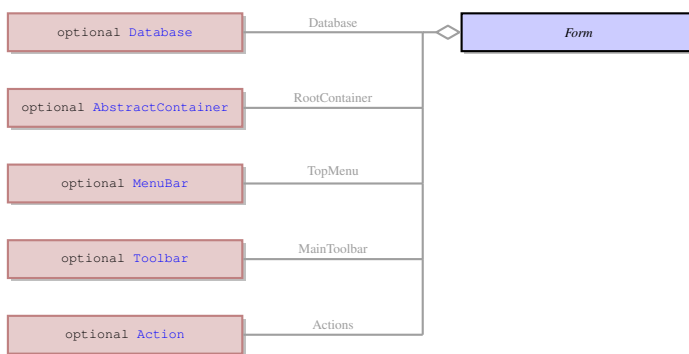
65.2 Description

Name: Form

This is an abstract element that serves as the root tag for the form XML code.

No parents.

This is an abstract element that serves as the root tag for the form XML code.

65.3 Fields

Name	Type	Description
Database	optional Database	This is the database to which the form is linked.
RootContainer	optional AbstractContainer	This is the root container of the form, all the other containers and widgets are placed inside it. In the XML everything is placed inside the root container tag, except fot Toolbar , MenuBar and ScreenRecord tags.
TopMenu	optional MenuBar	This is the main menu of the form.
MainToolbar	optional Toolbar	This is the toobar of the form.
Actions	list of Action	No information
ClassNames	list of ClassName	The name of a class that is applied to the UI element. There can be a customly created class or one of the default classes. The default classes depend on the 4GL attributes applied to the element by means of the 4GL code or form file and usually specify the colour or intensity attribute.

66 FunctionEventHandler Not-referenced

66.1 Diagram
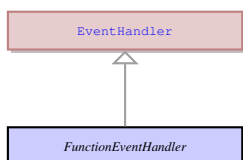
66.2 Description

Name: FunctionEventHandler

This event handler triggers the function specified in it, when the even to which it is assigned is invoked.

Parent: EventHandler - This is common class for all the specific event handler types.

This event handler triggers the function specified in it, when the even to which it is assigned is invoked.

66.3 Fields

Name	Type	Description
FunctionName	optional String	This is the form of a 4GL function without parentheses.

67 FunctionField Not-referenced

67.1 Diagram
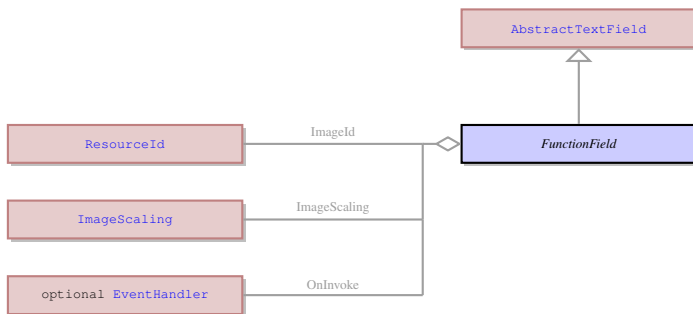
67.2 Description

Name: FunctionField

This is a form widget in a form of a text field with a button attached to its right side.

Parent: AbstractTextField - It is an abstract UI element, which unites a subset of AbstractStringField elements with the exception of TextArea , ComboBox , and Button . Typically it includes the UI elements which allow entering values, like normal text fields, and usually are only one line wide.

This is a form widget in a form of a text field with a button attached to its right side.

67.3 Fields

Name	Type	Description
ImageId	ResourceId	A reference to an image file.
ImageScaling	ImageScaling	It specifies whether the image should be scaled to fit the UI element it is applied to.
OnInvoke	optional EventHandler	The event which is triggered when the UI element is invoked. It can be invoked by mouse click, by pressing Enter, or in some cases Space, when the cursor is in the element.
InvisibleValue	optional Bool	If enabled, the value displayed to the field will be invisible. During input the value will be masked with *.
LabelText	optional String	No information
HelperText	optional String	No information

68 GlobalContentFilter Not-referenced

68.1 Diagram

68.2 Description

Name: GlobalContentFilter

No information

No parents.

No information

68.3 Fields

Name	Type	Description
ApplicationName	optional String	No information
WindowName	optional String	No information
FormName	optional String	No information

69 GreaterThen Not-referenced

69.1 Diagram

69.2 Description

Name: GreaterThen

It is a part of the conditional 4GL display attributes - ¿. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator ¿.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - ¿. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator ¿.

70 GreaterThenOrEqualTo Not-referenced

70.1 Diagram

70.2 Description

Name: GreaterThenOrEqualTo

It is a part of the conditional 4GL display attributes - ¿=. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator ¿=.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - ¿=. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator ¿=.

71 GridColumnDefinition Not-referenced

71.1 Diagram
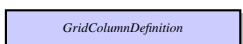
71.2 Description

Name: GridColumnDefinition

This UI element defines the properties of a columns in a GridPanel container and their properties.

No parents.

This UI element defines the properties of a columns in a GridPanel container and their properties.

71.3 Fields

Name	Type	Description
GridLengthValue	String	The width of the grid column or row in the units specified by the length type.
GridMinLength	optional String	This the minimum size of a grid column or row to which it can be resized.
GridMaxLength	optional String	This the maximum size of a grid column or row to which it can be resized.

72 GridItemLocation

72.1 Diagram
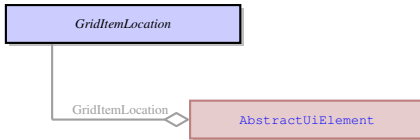
72.2 Description

Name: GridItemLocation

This property defines the position of an element located within a GridPanel in relation to this grid panel. The grid panel is divided into cells which are created by means of grid rows and columns. Each element placed inside the grid panel must occupy at least one cell. It can occupy more than one cell, but two elements cannot occupy one and the same cell. Each element inside a grid panel is located inside the cells, it cannot occupy half of a cell.

No parents.

This property defines the position of an element located within a GridPanel in relation to this grid panel. The grid panel is divided into cells which are created by means of grid rows and columns. Each element placed inside the grid panel must occupy at least one cell. It can occupy more than one cell, but two elements cannot occupy one and the same cell. Each element inside a grid panel is located inside the cells, it cannot occupy half of a cell.

72.3 Fields

Name	Type	Description
GridX	optional Int	It is the number of column in which the grid cell with the UI element is located. It is treated as the X coordinate of an element within the grid panel.
GridY	optional Int	It is the number of row in which the grid cell with the UI element is located. It is treated as the Y coordinate of an element within the grid panel.
GridWidth	optional Int	It specifies the number of horizontal cells that the element occupies. It cannot be less than 1.
GridHeight	optional Int	It specifies the number of vertical cells that the element occupies. It cannot be less than 1.

72.4 Referenced in

- GridItemLocation field in optional AbstractUiElement - This property defines the position of an element located within a GridPanel in relation to this grid panel. The grid panel is divided into cells which are created by means of grid rows and columns. Each element placed inside the grid panel must occupy at least one cell. It can occupy more than one cell, but two elements cannot occupy one and the same cell. Each element inside a grid panel is located inside the cells, it cannot occupy half of a cell.

73 GridLength Not-referenced

73.1 Diagram

73.2 Description

Name: GridLength

This UI element defines the length of the grid columns and width of the rows. Thus it can define the size of the GridPanel cells. The size can be absolute or relative. It can also define the length of the table columns.

No parents.

This UI element defines the length of the grid columns and width of the rows. Thus it can define the size of the GridPanel cells. The size can be absolute or relative. It can also define the length of the table columns.

73.3 Fields

Name	Type	Description
GridLengthValue	String	The width of the grid column or row in the units specified by the length type.
GridMinLength	optional String	This the minimum size of a grid column or row to which it can be resized.
GridMaxLength	optional String	This the maximum size of a grid column or row to which it can be resized.

74 GridPanel Not-referenced

74.1 Diagram
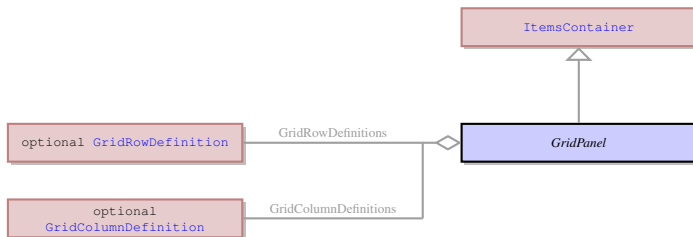
74.2 Description

Name: GridPanel

It is a container that is used to arrange the layout of other UI elements placed inside. The elements inside the grid panel are placed inside the grid cells that are formed by the grid rows and columns. Each element must occupy at least 1 grid cell, two elements cannot occupy one and the same grid cell. The number of the grid cells can be defined by the user.

Parent: ItemsContainer - The containers that can contain any number of UI elements inherit their properties from the ItemsContainer UI element. These are the containers that can contain any number of form fields and other containers, as opposed to the containers belonging to ElementContainer class.

It is a container that is used to arrange the layout of other UI elements placed inside. The elements inside the grid panel are placed inside the grid cells that are formed by the grid rows and columns. Each element must occupy at least 1 grid cell, two elements cannot occupy one and the same grid cell. The number of the grid cells can be defined by the user.

74.3 Fields

Name	Type	Description
GridRowDefinitions	list of GridRowDefinition	This UI element defines the number of rows in a grid panel container and their properties.
GridColumnDefinitions	list of GridColumnDefinition	This UI element defines the number of rows in a grid panel container and their properties.

75 GridRowDefinition Not-referenced

75.1 Diagram
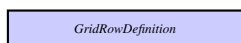
75.2 Description

Name: GridRowDefinition

This UI element defines the properties of a row in a GridPanel container.

No parents.

This UI element defines the properties of a row in a GridPanel container.

75.3 Fields

Name	Type	Description
GridLengthValue	String	The width of the grid column or row in the units specified by the length type.
GridMinLength	optional String	This the minimum size of a grid column or row to which it can be resized.
GridMaxLength	optional String	This the maximum size of a grid column or row to which it can be resized.

76 GroupBox Not-referenced

76.1 Diagram

76.2 Description

Name: GroupBox

It is a container that groups the UI elements inside a visible border with an optional title at the top. It can contain only one other UI element. It can be another container or a form widget. Thus though it can encompass UI elements of the AbstractField group, having only one element of this group in a container makes little sense. So it should include one of the other containers first.

Parent: ElementContainer - This UI element unites all the containers which can contain exactly one element. The containers that derive from ElementContainer UI element can be logically opposed to containers derived from ItemsContainer UI element that can contain any number of elements of any type. The elements that inherit their properties from ElementContainer can encompass such elements as ring menu area or any other container. They can also contain an element belonging to ui.AbstractFiled class, but only one such element.

It is a container that groups the UI elements inside a visible border with an optional title at the top. It can contain only one other UI element. It can be another container or a form widget. Thus though it can encompass UI elements of the AbstractField group, having only one element of this group in a container makes little sense. So it should include one of the other containers first.

76.3 Fields

Name	Type	Description
Title	optional String	This is the inscription attached to the UI element. Usually this is the text of all sorts of labels.
TitleJustification	TitleJustification	It specifies the horizontal alignment of the text of the title.

77 HorizontalAlignment

77.1 Diagram
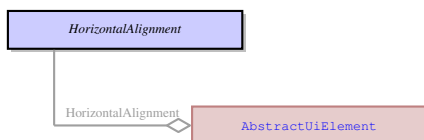
77.2 Description

Name: HorizontalAlignment

This enum specifies the horizontal alignment of a UI element inside a container. It is applicable to UI elements inside any container except coord panel. It defines to which border of the container (or container cell) - left or right - the element must adjoin.

No parents.

This enum specifies the horizontal alignment of a UI element inside a container. It is applicable to UI elements inside any container except coord panel. It defines to which border of the container (or container cell) - left or right - the element must adjoin.

77.3 Options

Name	Description
Default	The window size is the size with which it was opened or which was set after opening by 4GL or graphical theme means.
Stretch	The UI element will be stretched to fit the container (or container cell) without preserving the aspect ratio.
Left	The UI element will be aligned to the left side of the container (or container cell).
Center	The UI element will be equidistant from both sides.
Right	The UI element will be aligned to the right side of the container (or container cell).

77.4   Referenced in

- HorizontalAlignment field in optional AbstractUiElement - This enum specifies the horizontal alignment of a UI element inside a container. It is applicable to UI elements inside any container except coord panel. It defines to which border of the container (or container cell) - left or right - the element must adjoin.

78   HorizontalTextAlignment

78.1   Diagram
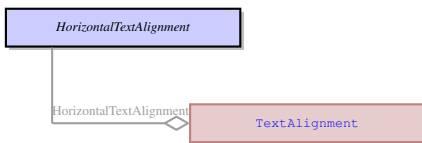
78.2   Description

Name: HorizontalTextAlignment

   No parents.

78.3   Options

Name	Description
Default	The window size is the size with which it was opened or which was set after opening by 4GL or graphical theme means.
Left	The UI element will be aligned to the left side of the container (or container cell).
Center	The UI element will be equidistant from both sides.
Right	The UI element will be aligned to the right side of the container (or container cell).

78.4   Referenced in

- HorizontalTextAlignment field in optional TextAlignment -

79 Image

79.1 Diagram
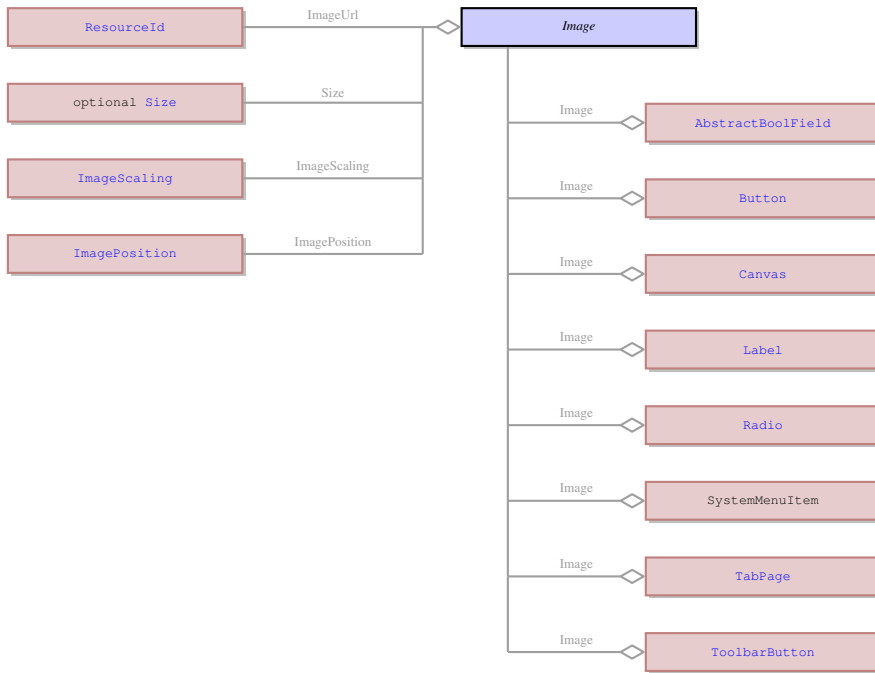
79.2 Description

Name: Image
>It is an image that can be applied to other UI elements, e.g. to a button.
>No parents.
>It is an image that can be applied to other UI elements, e.g. to a button.

79.3 Fields

Name	Type	Description
ImageUrl	ResourceId	It specifies the URI of an image file. The image should be located on the application server and inside the folder into which the application is deployed. The URL should begin with: qx://application/... .
Size	optional Size	The size of the UI element in pixels that.
ImageScaling	ImageScaling	It specifies whether the image should be scaled to fit the UI element it is applied to.
ImagePosition	ImagePosition	No information

79.4 Referenced in

- Image field in optional AbstractBoolField - It is an image that can be applied to other UI elements, e.g. to a button.

- Image field in optional Button - It is an image that can be applied to other UI elements, e.g. to a button.

- Image field in optional Canvas - It is an image that can be applied to other UI elements, e.g. to a button.

- Image field in optional Label - It is an image that can be applied to other UI elements, e.g. to a button.

- Image field in optional Radio - It is an image that can be applied to other UI elements, e.g. to a button.

- Image field in optional SystemMenuItem - It is an image that can be applied to other UI elements, e.g. to a button.

- Image field in optional TabPage - It is an image that can be applied to other UI elements, e.g. to a button.

- Image field in optional ToolbarButton - It is an image that can be applied to other UI elements, e.g. to a button.

80 ImagePosition

80.1 Diagram
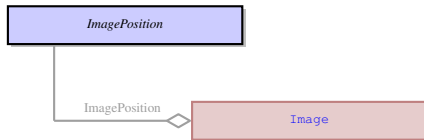
80.2 Description

Name: ImagePosition
    No information
    No parents.
    No information

80.3 Options

Name	Description
Left	The UI element will be aligned to the left side of the container (or container cell).
Right	The UI element will be aligned to the right side of the container (or container cell).
Top	The UI element will be aligned to the top of the container (or container cell).
Bottom	The UI element will be aligned to the bottom of the container (or container cell).

80.4 Referenced in

- ImagePosition field in optional Image - No information

81 ImageScaling

81.1 Diagram
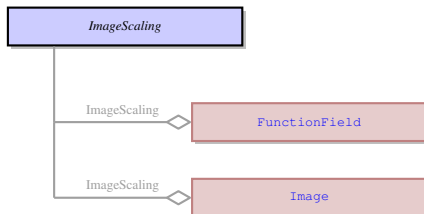
81.2 Description

Name: ImageScaling
    It specifies whether the image should be scaled (resized) to fit the UI element it is applied to. The scaling preserves the aspect ratio of an image, so in case the image is scaled by the larger side of the UI element, a part of it might be cut off.
    No parents.
    It specifies whether the image should be scaled (resized) to fit the UI element it is applied to. The scaling preserves the aspect ratio of an image, so in case the image is scaled by the larger side of the UI element, a part of it might be cut off.

81.3 Options

Name	Description
None	The property is not applied and the default behaviour is used.
Horizontal	The image will be scaled to fit the width of the UI element.
Vertical	The image will be scaled to fit the height of the UI element.
Both	The image will be scaled to fit the smallest dimension (either height or width) of the UI element.

81.4 Referenced in

- ImageScaling field in optional FunctionField - It specifies whether the image should be scaled (resized) to fit the UI element it is applied to. The scaling preserves the aspect ratio of an image, so in case the image is scaled by the larger side of the UI element, a part of it might be cut off.

- ImageScaling field in optional Image - It specifies whether the image should be scaled (resized) to fit the UI element it is applied to. The scaling preserves the aspect ratio of an image, so in case the image is scaled by the larger side of the UI element, a part of it might be cut off.

82 In Not-referenced

82.1 Diagram
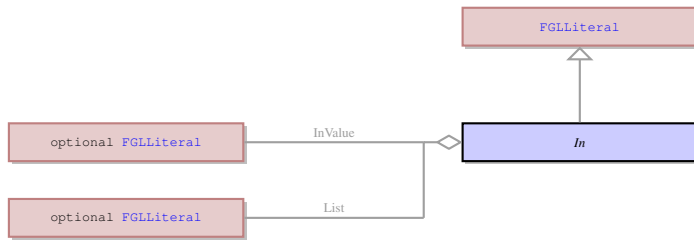
82.2 Description

Name: In

It is a part of the conditional 4GL display attributes - IN(). The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. Here is an example of the operator of membership IN(): filed_tag IN("mother", "father", "aunt").

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

It is a part of the conditional 4GL display attributes - IN(). The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. Here is an example of the operator of membership IN(): filed_tag IN("mother", "father", "aunt").

82.3 Fields

Name	Type	Description
InValue	optional FGLLiteral	No information
List	list of FGLLiteral	This a list of 4GL literals.

83 Include Not-referenced

83.1 Diagram
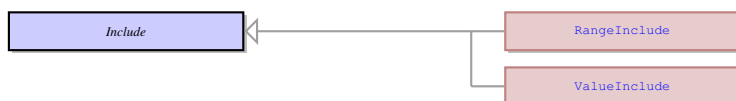
83.2 Description

Name: Include

This class defines what literal values are allowed for input into a form field. It can be either individual literals or a range of values. No parents.

This class defines what literal values are allowed for input into a form field. It can be either individual literals or a range of values.

83.3 Children

- RangeInclude - This class defines the range of values that are allowed for being entered into a field. These are normally numeric values or letters, e.g. from a to z, or any other values that can form a range.

- ValueInclude - This class defines individual literals included into a list of allowed values for a field.

84 IntegerLiteral Not-referenced

84.1 Diagram
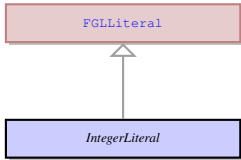
84.2 Description

Name: IntegerLiteral

These are integer values taking up to 4 bytes maximum. It corresponds in the format to the requirements of the INTEGER, SMALL-INT and TINYINT 4GL data types. E.g. 1234.

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

These are integer values taking up to 4 bytes maximum. It corresponds in the format to the requirements of the INTEGER, SMALL-INT and TINYINT 4GL data types. E.g. 1234.

84.3 Fields

Name	Type	Description
IntegerValue	Int	This is an integer number.

85 IntervalLiteral Not-referenced

85.1 Diagram
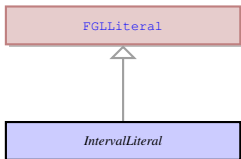
85.2 Description

Name: IntervalLiteral

These are interval values in different time units, e.g. INTERVAL (10-11) YEAR TO MONTH - means 10 years and 11 months. It corresponds in the format to the requirements of the INTERVAL 4GL data type.

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

These are interval values in different time units, e.g. INTERVAL (10-11) YEAR TO MONTH - means 10 years and 11 months. It corresponds in the format to the requirements of the INTERVAL 4GL data type.

85.3 Fields

Name	Type	Description
YearValOpt	optional Int	This value represents the year in the DATETIME or INTERVAL VALUE. This value is optional and may be present or absent depending on the Start and End qualifiers.
MonthValOpt	optional Int	This value represents the number of months in the DATETIME or INTERVAL value. This value is optional and may be present or absent depending on the Start and End qualifiers. It includes integers from 1 to 12.
DayValOpt	optional Int	This value represents the number of days in the datetime or interval value. This value is optional and may be present or absent depending on the Start and End qualifiers. It includes integers from 1 to 31.
HourValOpt	optional Int	This value represents the number of hours in the DATETIME or INTERVAL value. This value is optional and may be present or absent depending on the Start and End qualifiers. It includes integers from 0 to 23.

MinuteValOpt	optional Int	This value represents the number of minutes in the DATETIME or INTERVAL value. This value is optional and may be present or absent depending on the Start and End qualifiers. It includes integers from 0 to 59.
SecondValOpt	optional Int	This value represents the number of seconds in the DATETIME or INTERVAL value. This value is optional and may be present or absent depending on the Start and End qualifiers. It includes integers from 0 to 59.
FractionValOpt	optional Int	This value represents the number of fractions of second in the DATETIME or INTERVAL value. This value is optional and may be present or absent depending on the Start and End qualifiers.
FromValOpt	optional String	It represents the first part of the qualifier.
ToValOpt	optional String	It represents the second part of the qualifier.
PrecisionValOpt	optional Int	No information
ScaleValOpt	optional Int	It represents scale of the End qualifier.

86   IsNull Not-referenced

86.1   Diagram
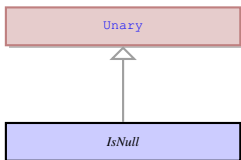
86.2   Description

Name: IsNull

It defines whether the field is allowed to accept NULL values.

Parent: Unary - These are binary operators. This class includes plus and minus operators that specify whether a number is positive or negative.

It defines whether the field is allowed to accept NULL values.

87   ItemsContainer Not-referenced

87.1   Diagram
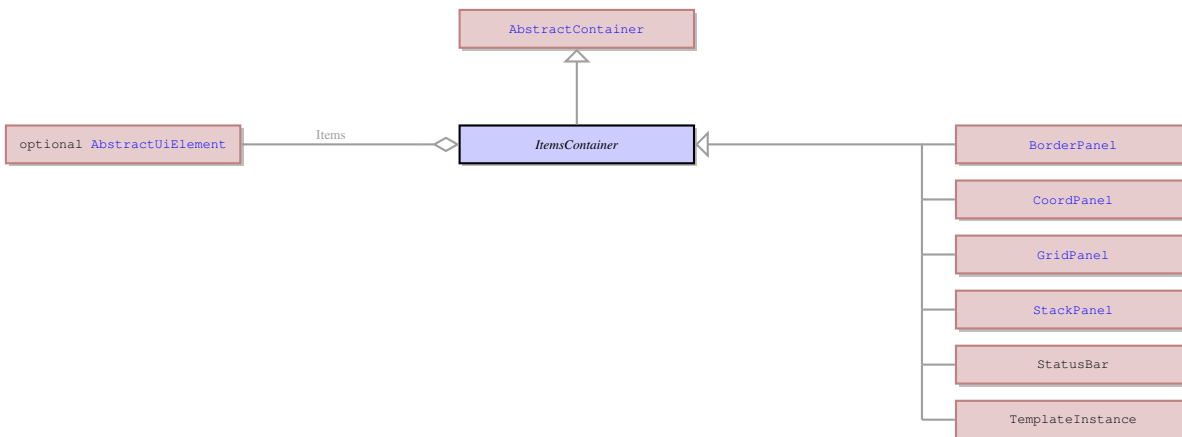
87.2   Description

Name: ItemsContainer

The containers that can contain any number of UI elements inherit their properties from the ItemsContainer UI element. These are the containers that can contain any number of form fields and other containers, as opposed to the containers belonging to ElementContainer class.

Parent: AbstractContainer - This UI element represents an abstract container from which all the form containers their properties. This abstract UI element unites all form containers - elements that determine the form layout.

The containers that can contain any number of UI elements inherit their properties from the ItemsContainer UI element. These are the containers that can contain any number of form fields and other containers, as opposed to the containers belonging to ElementContainer class.

87.3 Children

- **BorderPanel** - It is a concrete UI element - a container for arranging the layout of other UI elements. Other UI elements can be located either alongside the panel borders or in its center, thus this panel can incorporate up to 5 elements - 1 for each side and 1 in the center. The elements are stretched by default, one element can take up more than one position cell. The position of an element inside the Border panel (that is which of the) is defined by the **BorderPanelItemLocation** property of this element.

- **CoordPanel** - This is a container the location of the elements inside which is determined by the coordinates of the component. The coordinates are stored in pixels and specify the **Location** on the coord panel where the top left corner of the child element is placed.

- **GridPanel** - It is a container that is used to arrange the layout of other UI elements placed inside. The elements inside the grid panel are placed inside the grid cells that are formed by the grid rows and columns. Each element must occupy at least 1 grid cell, two elements cannot occupy one and the same grid cell. The number of the grid cells can be defined by the user.

- **StackPanel** - This is a container which arranges the elements in horizontal or vertical stacks. Any number of elements can be placed inside this container one next to the other. At runtime the contents of the stack panel can be resized only in the direction opposite to the orientation of the container.

87.4 Fields

Name	Type	Description
Items	list of AbstractUiElement	A set of UI elements that are placed inside the container.

88 KeyEvent Not-referenced

88.1 Diagram
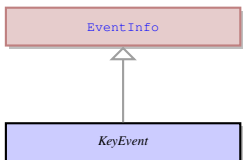
88.2 Description

Name: KeyEvent

It is an event that is triggered when the specified key on the keyboard is pressed. This event is sent to the Application server on the keypress.

Parent: EventInfo - It is an abstract UI entity which is the root class for the KeyEvent . It is used to send the information to the server about the event triggered on the client side.

It is an event that is triggered when the specified key on the keyboard is pressed. This event is sent to the Application server on the keypress.

88.3 Fields

Name	Type	Description
KeyValue	optional String	The name of the key pressed. The key name is the name written on the key, e.g. F12 or A.
ControlModifier	Bool	It indicates whether the Ctrl key should be held down when the key is pressed.
AltModifier	Bool	It indicates whether the Alt key should be held down when the key is pressed.
ShiftModifier	Bool	It indicates whether the Shift key should be held down when the key is pressed.

89 KeyEventHandler Not-referenced

89.1 Diagram

89.2 Description

Name: KeyEventHandler

This is an event handler that is invoked when the KeyEvent is triggered.

Parent: EventHandler - This is common class for all the specific event handler types.

This is an event handler that is invoked when the KeyEvent is triggered.

89.3 Fields

Name	Type	Description
KeyName	optional KeyName	This is a keyboard key or key combination that triggers the handler.

90 KeyName

90.1 Diagram
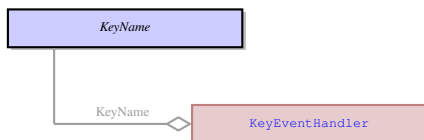
90.2 Description

Name: KeyName

No information

No parents.

No information

90.3 Fields

Name	Type	Description
KeyValue	optional String	The name of the key pressed. The key name is the name written on the key, e.g. F12 or A.
ControlModifier	Bool	It indicates whether the Ctrl key should be held down when the key is pressed.
AltModifier	Bool	It indicates whether the Alt key should be held down when the key is pressed.
ShiftModifier	Bool	It indicates whether the Shift key should be held down when the key is pressed.

90.4 Referenced in

- KeyName field in optional KeyEventHandler - No information

91 Label Not-referenced

91.1 Diagram
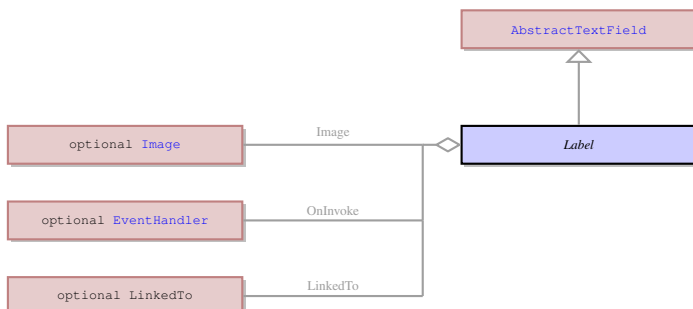
91.2 Description

Name: Label

It is a concrete UI element that has the form of a label with some text, image or both. The label is not an interactive widget and cannot be used for input, but the information displayed by it can be changed dynamically.

Parent: AbstractTextField - It is an abstract UI element, which unites a subset of AbstractStringField elements with the exception of TextArea , ComboBox , and Button . Typically it includes the UI elements which allow entering values, like normal text fields, and usually are only one line wide.

It is a concrete UI element that has the form of a label with some text, image or both. The label is not an interactive widget and cannot be used for input, but the information displayed by it can be changed dynamically.

91.3 Fields

Name	Type	Description
Image	optional Image	The image that is displayed to a label.
IsDynamic	Bool	It specifies whether the information displayed by the label can be changed dynamically by means of the DISPLAY TO statement.
OnInvoke	optional EventHandler	The event which is triggered when the UI element is invoked. It can be invoked by mouse click, by pressing Enter, or in some cases Space, when the cursor is in the element.
AllowNewlines	Bool	This property specifies whether the Enter key will be used to move to another form element at runtime (if the value is FALSE), or it will create a newline symbol inside the current field (if the value is TRUE). It is typically applied for the TextArea element.

92 Length Not-referenced

92.1 Diagram
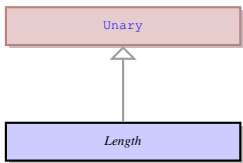
92.2 Description

Name: Length

No information

Parent: Unary - These are binary operators. This class includes plus and minus operators that specify whether a number is positive or negative.

No information

93 LessThen Not-referenced

93.1 Diagram
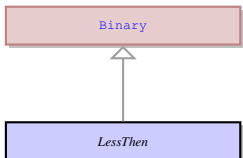
93.2 Description

Name: LessThen

It is a part of the conditional 4GL display attributes - ¡. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator ¡.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - ¡. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator ¡.

94 LessThenOrEqualTo Not-referenced

94.1 Diagram

```
Binary
  △
  │
LessThenOrEqualTo
```

94.2 Description

Name: LessThenOrEqualTo

It is a part of the conditional 4GL display attributes - ¡=. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator ¡=.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - ¡=. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator ¡=.

95 Like Not-referenced

95.1 Diagram

```
Binary
  △
  │
 Like
```
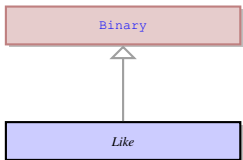
95.2 Description

Name: Like

It is a part of the conditional 4GL display attributes - LIKE. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of string comparison LIKE.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - LIKE. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of string comparison LIKE.

96 LineBorder Not-referenced

96.1 Diagram

```
Border
  △
  │
LineBorder
```
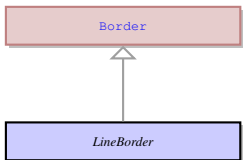
96.2 Description

Name: LineBorder

This UI element is used to apply a custom line border to any concrete UI element. A line border is just a line of the defined thickness and colour that surrounds the element. The line border allows the CornerRadius to be set to round the corners.

Parent: Border - It defines the properties of a custom border around a concrete UI element. The properties border can be applied to one of the three border types: BevelBorder , EtchedBorder , and LineBorder .

This UI element is used to apply a custom line border to any concrete UI element. A line border is just a line of the defined thickness and colour that surrounds the element. The line border allows the CornerRadius to be set to round the corners.

97 ListBox Not-referenced

97.1 Diagram
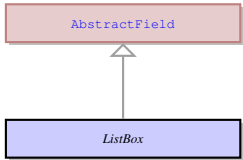
97.2 Description

Name: ListBox

It is a concrete UI element that has the form of a form field with a list of values inside available for selection. It does not accept values entered from the keyboard, but can participate in the input and records into the underlying variable the value that was selected from the list.

Parent: AbstractField - This UI element represents an abstract field from which all the form widgets inherit their properties. This abstract UI element unites all form fields - the form elements that can accept and display data - as opposed tp form containers - elements that determine the form layout.

It is a concrete UI element that has the form of a form field with a list of values inside available for selection. It does not accept values entered from the keyboard, but can participate in the input and records into the underlying variable the value that was selected from the list.

97.3 Fields

Name	Type	Description
EnableMultiSelection	Bool	It specifies how many items can be simultaneously selected inside a list box widget. If set to FALSE, only one item cal be selected at a time.
HelperText	optional String	No information

98 ListBoxItem Not-referenced

98.1 Diagram

98.2 Description

Name: ListBoxItem

It is an individual item that is a part of the list box list of values.

No parents.

It is an individual item that is a part of the list box list of values.

98.3 Fields

Name	Type	Description
Text	optional String	This is the value of the list box list item, which is recorded to the underlying variable after the input.
Value	FGLLiteral	This is any 4GL literal usually specifying the default value of a widget.

99 Locale

99.1 Diagram
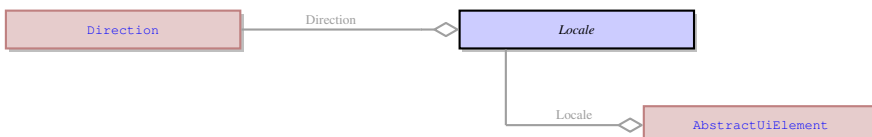
99.2 Description

Name: Locale

It specifies a custom locate of a UI element that can be different from the default application locale. It can mainly be used for to make a form fir the requirements of several locales at once.

No parents.

It specifies a custom locate of a UI element that can be different from the default application locale. It can mainly be used for to make a form fir the requirements of several locales at once.

99.3 Fields

Name	Type	Description
Language	optional String	The language of the locale, e.g. FR for French.
Country	optional String	The territory where the specified locale language is used. E.g. CA - for French language in Canada.
Variant	optional String	The code set of the selected locale. E.g. ISO-8859-1 or UTF-8.
Direction	Direction	The direction of the text: from left to right or from right to left.

99.4 Referenced in

- Locale field in optional AbstractUiElement - It specifies a custom locate of a UI element that can be different from the default application locale. It can mainly be used for to make a form fir the requirements of several locales at once.

100 Location

100.1 Diagram
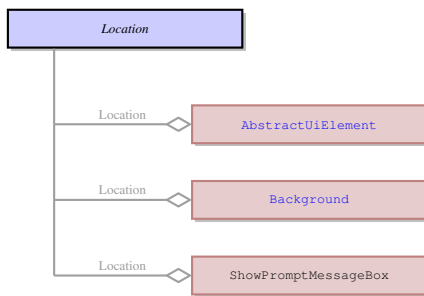
100.2 Description

Name: Location

This is the coordinates of the position of a UI element inside a coordinate panel in pixels.

No parents.

This is the coordinates of the position of a UI element inside a coordinate panel in pixels.

100.3 Fields

Name	Type	Description
XCoord	String	The coordinate of the top left corner of the element on X axis of the coord panel.
YCoord	String	The coordinate of the top left corner of the element on Y axis of the coord panel.

100.4 Referenced in

- Location field in optional AbstractUiElement - This is the coordinates of the position of a UI element inside a coordinate panel in pixels.

- Location field in optional Background - This is the coordinates of the position of a UI element inside a coordinate panel in pixels.

- Location field in optional ShowPromptMessageBox - This is the coordinates of the position of a UI element inside a coordinate panel in pixels.

101 LongIntegerLiteral Not-referenced

101.1 Diagram

101.2 Description

Name: LongIntegerLiteral

These are integer values taking up to 8 bytes maximum. It corresponds in the format to the requirements of the INTEGER 4GL data type. E.g. 123456.

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

These are integer values taking up to 8 bytes maximum. It corresponds in the format to the requirements of the INTEGER 4GL data type. E.g. 123456.

101.3 Fields

Name	Type	Description
LongIntegerValue	Integer64	This is an integer number.

102 Matches Not-referenced

102.1 Diagram
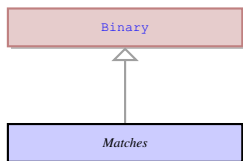
102.2 Description

Name: Matches

It is a part of the conditional 4GL display attributes - MATCHES. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of string comparison MATCHES.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - MATCHES. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of string comparison MATCHES.

103 MenuBar Not-referenced

103.1 Diagram
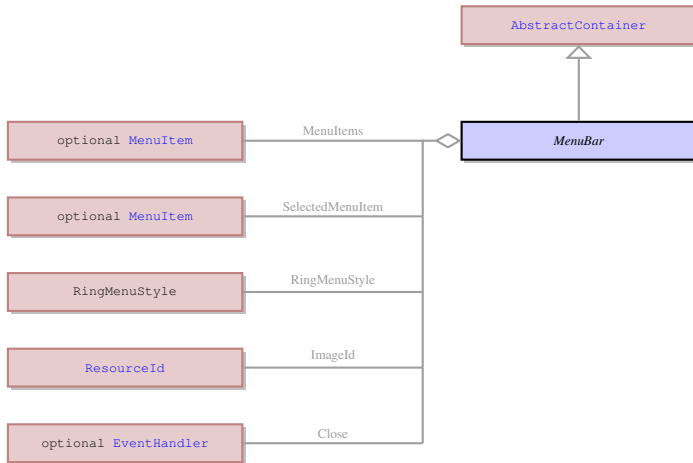
103.2 Description

Name: MenuBar

This is the area for the top menu (is not applied to ring menus). It includes menu options and menu option groups.

Parent: AbstractContainer - This UI element represents an abstract container from which all the form containers their properties. This abstract UI element unites all form containers - elements that determine the form layout.

This is the area for the top menu (is not applied to ring menus). It includes menu options and menu option groups.

103.3 Fields

Name	Type	Description
MenuItems	list of MenuItem	A set of menu options belonging to the same menu.

104 MenuCommand Not-referenced

104.1 Diagram
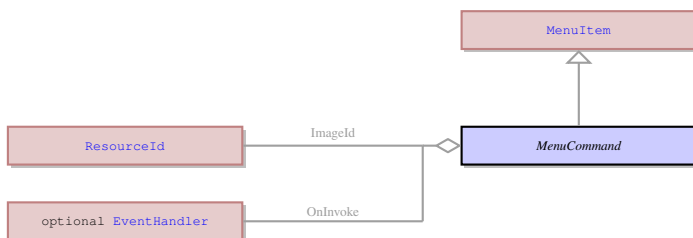
104.2 Description

Name: MenuCommand

This is the menu option that can be invoked by the user. It has a label and/or icon and an even attached.

Parent: MenuItem - This UI element serves as the base class for all menu items: menu commands, menu groups, and menu separators.

This is the menu option that can be invoked by the user. It has a label and/or icon and an even attached.

104.3 Fields

Name	Type	Description
Text	optional String	This is the label of the menu option.
ImageId	ResourceId	The image that will be used as the icon on the menu option button.
OnInvoke	optional EventHandler	The event which is triggered when the UI element is invoked. It can be invoked by mouse click, by pressing Enter, or in some cases Space, when the cursor is in the element.

105 MenuGroup Not-referenced

105.1 Diagram
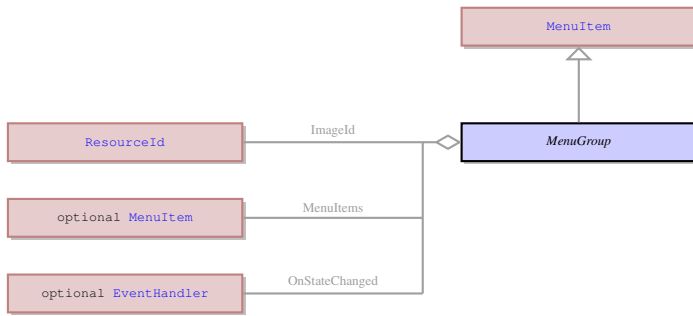
105.2 Description

Name: MenuGroup

It is a group that unites several menu options and possibly menu separators. It offers a drop-down menu containing these options and separators, when the mouse cursor hovers over its label.

Parent: MenuItem - This UI element serves as the base class for all menu items: menu commands, menu groups, and menu separators.

It is a group that unites several menu options and possibly menu separators. It offers a drop-down menu containing these options and separators, when the mouse cursor hovers over its label.

105.3 Fields

Name	Type	Description
Text	optional String	This is the of the menu group.
ImageId	ResourceId	A reference to an image file.
MenuItems	list of MenuItem	A set of menu options belonging to the same menu.
IsExpanded	Bool	No information
OnStateChanged	optional EventHandler	No information

106 MenuItem Not-referenced

106.1 Diagram
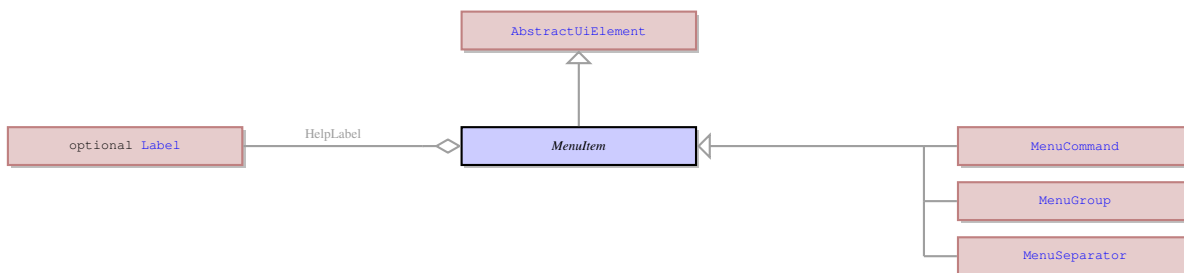
106.2 Description

Name: MenuItem

This UI element serves as the base class for all menu items: menu commands, menu groups, and menu separators.

Parent: AbstractUiElement - AbstractUiElement is the base class for UI widgets. It is a generic UI element that can accept user actions. Most of concrete UI elements must inherit the properties and action types from the AbstractUIElement.

This UI element serves as the base class for all menu items: menu commands, menu groups, and menu separators.

106.3 Children

- MenuCommand - This is the menu option that can be invoked by the user. It has a label and/or icon and an even attached.

- MenuGroup - It is a group that unites several menu options and possibly menu separators. It offers a drop-down menu containing these options and separators, when the mouse cursor hovers over its label.

- MenuSeparator - It is a horizontal line that visually separates menu options in the drop-down list of the menu group.

107 MenuSeparator Not-referenced

107.1 Diagram

```
MenuItem
   ^
   |
MenuSeparator
```
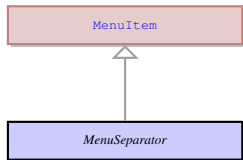
107.2 Description

Name: MenuSeparator

It is a horizontal line that visually separates menu options in the drop-down list of the menu group.

Parent: MenuItem - This UI element serves as the base class for all menu items: menu commands, menu groups, and menu separators.

It is a horizontal line that visually separates menu options in the drop-down list of the menu group.

108 Mod Not-referenced

108.1 Diagram

```
Binary
   ^
   |
  Mod
```
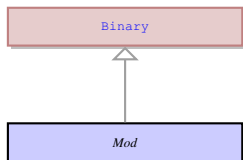
108.2 Description

Name: Mod

It is a part of the conditional 4GL display attributes - MOD. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of module MOD.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - MOD. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of module MOD.

109 ModelItem Not-referenced

109.1 Diagram

```
ModelItem
```

109.2 Description

Name: ModelItem

EMPTY.

No parents.

EMPTY.

110 Multiply Not-referenced

110.1 Diagram

```
Binary
   ^
   |
Multiply
```
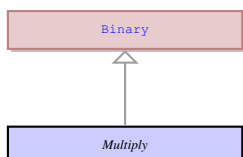
110.2 Description

Name: Multiply

It is a part of the conditional 4GL display attributes - *. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator *.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - *. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator *.

111 Negative Not-referenced

111.1 Diagram

111.2 Description

Name: Negative

It is a part of the conditional 4GL display attributes - -. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic unary operator - used to differentiate negative numbers from positive.

Parent: Unary - These are binary operators. This class includes plus and minus operators that specify whether a number is positive or negative.

It is a part of the conditional 4GL display attributes - -. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic unary operator - used to differentiate negative numbers from positive.

112 Not Not-referenced

112.1 Diagram

112.2 Description

Name: Not

It is a part of the conditional 4GL display attributes - NOT. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator NOT.

Parent: Unary - These are binary operators. This class includes plus and minus operators that specify whether a number is positive or negative.

It is a part of the conditional 4GL display attributes - NOT. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator NOT.

113 NotEqualTo Not-referenced

113.1 Diagram

113.2 Description

Name: NotEqualTo

It is a part of the conditional 4GL display attributes - !=. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator !=.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - !=. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is one of these operator !=.

114 NotLike Not-referenced

114.1 Diagram
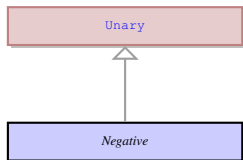

114.2 Description

Name: NotLike

It is a part of the conditional 4GL display attributes - NOT LIKE. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of string comparison NOT LIKE.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.
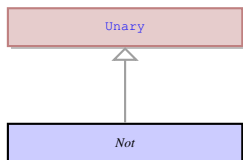
It is a part of the conditional 4GL display attributes - NOT LIKE. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of string comparison NOT LIKE.

115 NotMatches Not-referenced

115.1 Diagram

115.2 Description

Name: NotMatches

It is a part of the conditional 4GL display attributes - NOT MATCHES. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of string comparison NOT MATCHES.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - NOT MATCHES. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of string comparison NOT MATCHES.

116 NullLiteral Not-referenced

116.1 Diagram
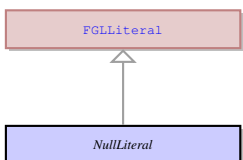
116.2 Description

Name: NullLiteral

This is a 4GL literal that contains no value. This means the value of this literal is displayed as NULL or an empty space.

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

This is a 4GL literal that contains no value. This means the value of this literal is displayed as NULL or an empty space.

117 OpenUrlEventHandler Not-referenced

117.1 Diagram
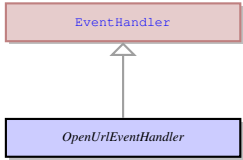
117.2 Description

Name: OpenUrlEventHandler

This is an event handler that can be assigned to any event. This handler opens the URL specified in the default system web browser.

Parent: EventHandler - This is common class for all the specific event handler types.

This is an event handler that can be assigned to any event. This handler opens the URL specified in the default system web browser.

117.3 Fields

Name	Type	Description
Url	optional String	An URL, generally it requires the explicit specification of the protocol: http, ftp, etc..

118 Or Not-referenced
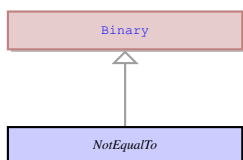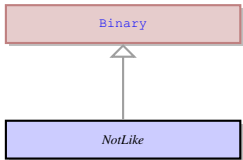
118.1 Diagram
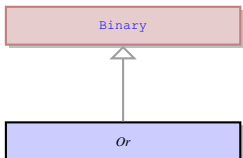
118.2 Description

Name: Or

It is a part of the conditional 4GL display attributes - OR. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of logical intersection OR.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.
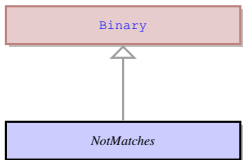
It is a part of the conditional 4GL display attributes - OR. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of logical intersection OR.

119 Orientation

119.1 Diagram

119.2 Description

Name: Orientation

This enum specifies whether the UI element should have vertical or horizontal layout. The horizontal layout is the default one. It is applied to some containers which defines the layout of the elements inside the container. It is also applied to Slider , ProgressBar and ScrollBar UI elements.

No parents.

This enum specifies whether the UI element should have vertical or horizontal layout. The horizontal layout is the default one. It is applied to some containers which defines the layout of the elements inside the container. It is also applied to Slider , ProgressBar and ScrollBar UI elements.

119.3 Options

Name	Description
Horizontal	The UI element will be placed horizontally and directed from left to right.
Vertical	The UI element will be placed vertically and directed from top to bottom.

119.4 Referenced in

- Orientation field in optional RadioGroup - This enum specifies whether the UI element should have vertical or horizontal layout. The horizontal layout is the default one. It is applied to some containers which defines the layout of the elements inside the container. It is also applied to Slider , ProgressBar and ScrollBar UI elements.

- Orientation field in optional ScrollBar - This enum specifies whether the UI element should have vertical or horizontal layout. The horizontal layout is the default one. It is applied to some containers which defines the layout of the elements inside the container. It is also applied to Slider , ProgressBar and ScrollBar UI elements.

- Orientation field in optional Slider - This enum specifies whether the UI element should have vertical or horizontal layout. The horizontal layout is the default one. It is applied to some containers which defines the layout of the elements inside the container. It is also applied to Slider , ProgressBar and ScrollBar UI elements.

- Orientation field in optional StackPanel - This enum specifies whether the UI element should have vertical or horizontal layout. The horizontal layout is the default one. It is applied to some containers which defines the layout of the elements inside the container. It is also applied to Slider , ProgressBar and ScrollBar UI elements.

120 PlaceHolder Not-referenced

120.1 Diagram

120.2   Description

Name: PlaceHolder

  No information

  Parent: AbstractUiElement - AbstractUiElement is the base class for UI widgets. It is a generic UI element that can accept user actions. Most of concrete UI elements must inherit the properties and action types from the AbstractUIElement.

  No information

121   Power Not-referenced

121.1   Diagram

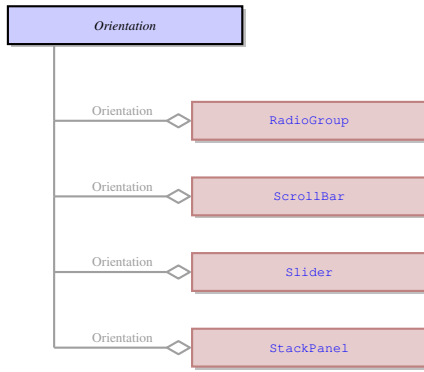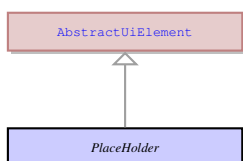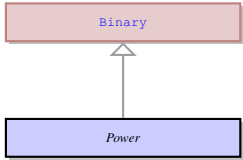## 121.2   Description

Name: Power

  It is a part of the conditional 4GL display attributes - **. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of exponentiation **.

  Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

  It is a part of the conditional 4GL display attributes - **. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator of exponentiation **.

122   ProgressBar Not-referenced

122.1   Diagram
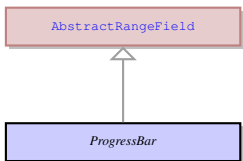
122.2   Description

Name: ProgressBar

  This is a concrete UI element that has a form of a rectangular bar that can show the progress of the application execution by means of being filled with colour background gradually. For it to reflect the progress, the DISPLAY TO statement should be used to indicate the degree to which it must be filled after each stage. The progress bar should have the maximum value (when it is displayed to the progress bar it becomes 100 percent filled) and minimum value (when displayed makes the progress bar 0 percent filled).

  Parent: AbstractRangeField - It is an abstract UI element, which unites the concrete UI elements which accept only the values included into the specified range. It is typically a range or numeric values, for example from 1 to 100. The concrete UI elements that inherit their properties from the AbstractRangeField are Slider , ProgressBar , Spinner , and ScrollBar .

  This is a concrete UI element that has a form of a rectangular bar that can show the progress of the application execution by means of being filled with colour background gradually. For it to reflect the progress, the DISPLAY TO statement should be used to indicate the degree to which it must be filled after each stage. The progress bar should have the maximum value (when it is displayed to the progress bar it becomes 100 percent filled) and minimum value (when displayed makes the progress bar 0 percent filled).

122.3   Fields

Name	Type	Description
Step	Int	This is a number by which the value of the UI element can be increased or decreased at a time. It must be within the maximum and minimum value range. It prevents floating value changing.

123 Radio Not-referenced

123.1 Diagram
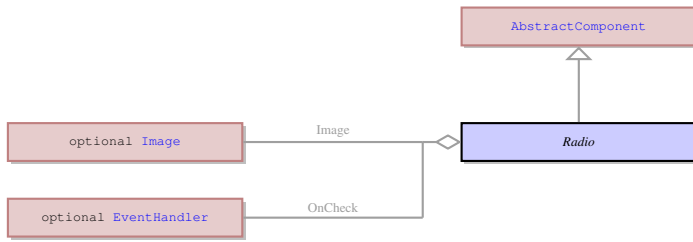
123.2 Description

Name: Radio

A Radio is a UI element that can only occur inside a RadioGroup . It can be in either of the two states at a time - checked or unchecked. The state of one Radio in a list influences and depends on the state of other items in the same list.

Parent: AbstractComponent - This is the common parent of all UI elements.

A Radio is a UI element that can only occur inside a RadioGroup . It can be in either of the two states at a time - checked or unchecked. The state of one Radio in a list influences and depends on the state of other items in the same list.

123.3 Fields

Name	Type	Description
Title	optional String	This is the inscription attached to the UI element. Usually this is the text of all sorts of labels.
Image	optional Image	It is an image that can be applied to other UI elements, e.g. to a button.
OnCheck	optional EventHandler	The OnCheck field defines the event which will be triggered if the IsChecked field of the UI element is changed to TRUE.
AllowNewlines	Bool	This property specifies whether the Enter key will be used to move to another form element at runtime (if the value is FALSE), or it will create a newline symbol inside the current field (if the value is TRUE). It is typically applied for the TextArea element.

124 RadioGroup Not-referenced

124.1 Diagram
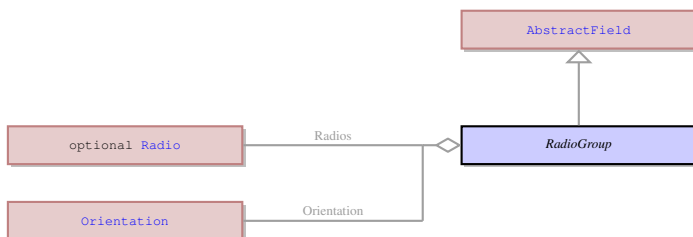
124.2 Description

Name: RadioGroup

The Radio is a UI element - a form widget - that contains a set of Radio which are either in selected or deselected state. The user can select only one Radio belonging to the same RadioGroup at a time, selecting a new item from the set deselects the previously selected element.

Parent: AbstractField - This UI element represents an abstract field from which all the form widgets inherit their properties. This abstract UI element unites all form fields - the form elements that can accept and display data - as opposed tp form containers - elements that determine the form layout.

The Radio is a UI element - a form widget - that contains a set of Radio which are either in selected or deselected state. The user can select only one Radio belonging to the same RadioGroup at a time, selecting a new item from the set deselects the previously selected element.

124.3 Fields

Name	Type	Description
Radios	list of Radio	This is the list of Radios that belong to the specified RadioGroup element.
Orientation	Orientation	This enum specifies whether the UI element should have vertical or horizontal layout.
Required	Bool	No information

125 RangeInclude Not-referenced

125.1 Diagram
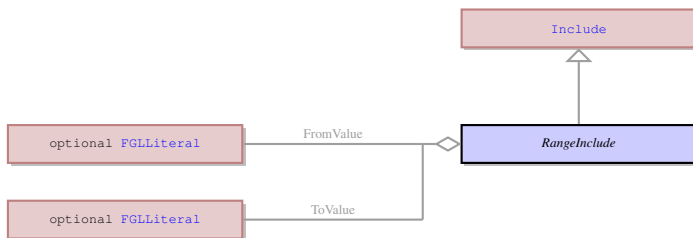
125.2 Description

Name: RangeInclude

This class defines the range of values that are allowed for being entered into a field. These are normally numeric values or letters, e.g. from a to z, or any other values that can form a range.

Parent: Include - This class defines what literal values are allowed for input into a form field. It can be either individual literals or a range of values.

This class defines the range of values that are allowed for being entered into a field. These are normally numeric values or letters, e.g. from a to z, or any other values that can form a range.

125.3 Fields

Name	Type	Description
FromValue	FGLLiteral	This is the minimum value of the range of values.
ToValue	FGLLiteral	This is the maximum value of the range of values.

126 ResourceId Not-referenced

126.1 Diagram

126.2 Description

Name: ResourceId

This is the specification of a media resource that is to be applied to the UI element, normally of an image or an icon. It specifies the media file, the path to it and other information about this media file.

No parents.

This is the specification of a media resource that is to be applied to the UI element, normally of an image or an icon. It specifies the media file, the path to it and other information about this media file.

126.3 Fields

Name	Type	Description
Uri	String	It is the URI of a media resource. The resource should be located on the application server and the URI should begin with qx://application/... .

127 ScaleType

127.1 Diagram
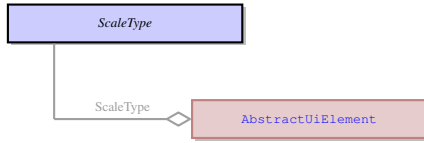
127.2 Description

Name: ScaleType

It indicates whether the UI element contents will be scaled, when the element is resized. The element resizing depends on the layout of the form and is predefined by the container. The scaling does not influence whether or not the physical size of the element will be changed by the attempt to resize it, it only influences the element contents. during the resizing.

No parents.

It indicates whether the UI element contents will be scaled, when the element is resized. The element resizing depends on the layout of the form and is predefined by the container. The scaling does not influence whether or not the physical size of the element will be changed by the attempt to resize it, it only influences the element contents. during the resizing.

127.3 Options

Name	Description
NoScale	The scaling is not applied when the element is resized. It will be resized only according to its layout position; e.g. the button will be enlarged, but the text on it will remain unchanged.
Both	When an element is resized, its contents is also resized: if a button gets bigger, the text in it also gets the bigger font.

127.4 Referenced in

- ScaleType field in optional AbstractUiElement - It indicates whether the UI element contents will be scaled, when the element is resized. The element resizing depends on the layout of the form and is predefined by the container. The scaling does not influence whether or not the physical size of the element will be changed by the attempt to resize it, it only influences the element contents. during the resizing.

128 ScreenRecord Not-referenced

128.1 Diagram

128.2 Description

Name: ScreenRecord

This class specifies a list of widgets present on the form. They are united into a record that can be referenced from within the 4GL code.

No parents.

This class specifies a list of widgets present on the form. They are united into a record that can be referenced from within the 4GL code.

128.3 Fields

Name	Type	Description
Identifier	String	It is a unique name of a UI element by which it can be referenced.
Fields	non-empty list of Field	No information
ScrollId	optional String	No information

129 ScrollBar Not-referenced

129.1 Diagram

129.2 Description

Name: ScrollBar

It is a concrete UI element that is represented by a scrollbar. It as the maximum and minimum values and the slider can be moved by the user at runtime or by displaying values to the element.

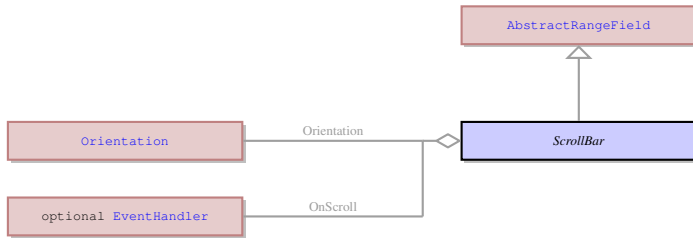**Parent:** AbstractRangeField - It is an abstract UI element, which unites the concrete UI elements which accept only the values included into the specified range. It is typically a range or numeric values, for example from 1 to 100. The concrete UI elements that inherit their properties from the AbstractRangeField are Slider , ProgressBar , Spinner , and ScrollBar .

It is a concrete UI element that is represented by a scrollbar. It as the maximum and minimum values and the slider can be moved by the user at runtime or by displaying values to the element.

129.3 Fields

Name	Type	Description
Orientation	Orientation	This enum specifies whether the UI element should have vertical or horizontal layout.
LargeStep	Int	It indicates the value by which the slider will be moved at a time, if the user moves it by holding down the arrow key.
SmallStep	Int	It indicates the smallest value by which the slider can be moved at a time. The slider cannot move smoothly and stop at values that won't make a complete step. E.g.: if the step is 2, the slider cannot stop at values 1, 3, 5, etc., it can stop at values 0,2,4,6 and so on. The small step is used when the user moves the slider by a single press of the arrow key on the keyboard.
OnScroll	optional EventHandler	This is the event invoked when the slider of the UI element moves.

130 ScrollViewer Not-referenced

130.1 Diagram
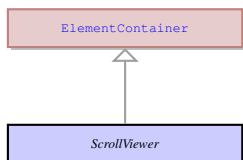
130.2 Description

Name: ScrollViewer

It is a container the content of which can be bigger than the container. The scrollbars are used to view the content that does not fit. It can contain exactly one element. E.g. it can contain a stack panel container, the number of elements inside which can be bigger than fit the size of the Scroll Viewer.

Parent: ElementContainer - This UI element unites all the containers which can contain exactly one element. The containers that derive from ElementContainer UI element can be logically opposed to containers derived from ItemsContainer UI element that can contain any number of elements of any type. The elements that inherit their properties from ElementContainer can encompass such elements as ring menu area or any other container. They can also contain an element belonging to ui.AbstractFiled class, but only one such element.

It is a container the content of which can be bigger than the container. The scrollbars are used to view the content that does not fit. It can contain exactly one element. E.g. it can contain a stack panel container, the number of elements inside which can be bigger than fit the size of the Scroll Viewer.

131 Separator Not-referenced

131.1 Diagram
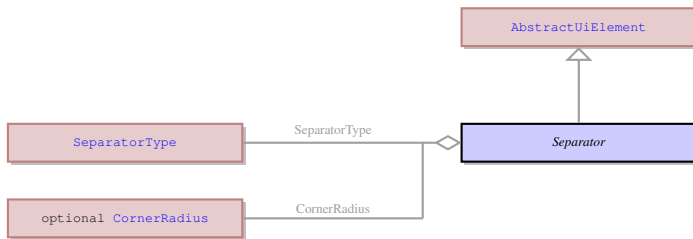
131.2 Description

Name: Separator

Any kind of separator, e.g. the status bar separator.

Parent: AbstractUiElement - AbstractUiElement is the base class for UI widgets. It is a generic UI element that can accept user actions. Most of concrete UI elements must inherit the properties and action types from the AbstractUIElement.

Any kind of separator, e.g. the status bar separator.

131.3 Fields

| Name	Type	Description
SeparatorType	SeparatorType	This is the type of the separator to be displayed
CornerRadius	optional CornerRadius	The radius of a corner of a custom border around the UI element. It is used to make the border corners rounded.

132 SeparatorType

132.1 Diagram
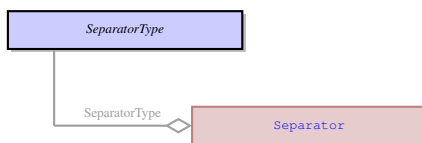
132.2 Description

Name: SeparatorType

This is the type of the separator to be displayed

No parents.

This is the type of the separator to be displayed

132.3 Options

Name	Description
Horizontal	Separator in the form of a single horizontal line
Vertical	Separator in the form of a single vertical line.
LeftTop	Separator in the form of two short lines adjoining orthogonally and forming a left top corner of a rectangle.
RightTop	Separator in the form of two short lines adjoining orthogonally and forming a right top corner of a rectangle.
LeftBottom	Separator in the form of two short lines adjoining orthogonally and forming a left bottom corner of a rectangle.
RightBottom	Separator in the form of two short lines adjoining orthogonally and forming a right bottom corner of a rectangle.
Cross	Separator in the form of two short lines intersecting orthogonally and forming an equilateral cross. Serves for connecting vertical and horizontal separators that overlap separators.
LeftJunction	Separator in the form of one longer vertical and one shorter horizontal line with the shorter line adjoining the longer one orthogonally at the middle from its left side. Serves for connecting a horizontal separator to the middle of vertical one.

RightJunction	LeftJunction - Separator in the form of one longer vertical and one shorter horizontal line with the shorter line adjoining the longer one orthogonally at the middle from its right side. Serves for connecting a horizontal separator to the middle of vertical one.
TopJunction	Separator in the form of one longer horizontal and one shorter vertical line with the shorter line adjoining the longer one orthogonally at the middle from the top. Serves for connecting a vertical separator to the middle of horizontal one.
BottomJunction	Separator in the form of one longer horizontal and one shorter vertical line with the shorter line adjoining the longer one orthogonally at the middle from the bottom. Serves for connecting a vertical separator to the middle of horizontal one.

132.4   Referenced in

- SeparatorType field in optional Separator - This is the type of the separator to be displayed

133   Size

133.1   Diagram
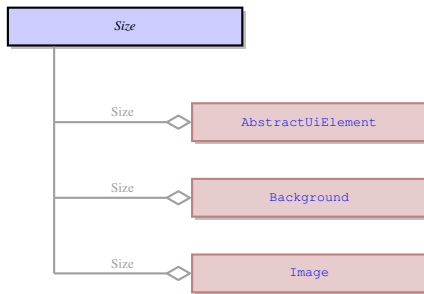
133.2   Description

Name: Size
    The size of a UI element in pixels.
    No parents.
    The size of a UI element in pixels.

133.3   Fields

Name	Type	Description
Width	optional String	The width of the UI element in pixels.
Height	optional String	The height of the UI element in pixels.

133.4   Referenced in

- Size field in optional AbstractUiElement - The size of a UI element in pixels.

- Size field in optional Background - The size of a UI element in pixels.

- Size field in optional Image - The size of a UI element in pixels.

134   Slider Not-referenced

134.1   Diagram

134.2 Description

Name: Slider

This is a concrete UI element that consists of a scale and a slider that can move across this scale. The slider widget has the minimum and maximum value which present the start and the end of the scale. It can be moved directly by the user during the input, or it can be moved if a value within its values range is displayed to it by the 4GL means.

Parent: AbstractRangeField - It is an abstract UI element, which unites the concrete UI elements which accept only the values included into the specified range. It is typically a range or numeric values, for example from 1 to 100. The concrete UI elements that inherit their properties from the AbstractRangeField are Slider , ProgressBar , Spinner , and ScrollBar .

This is a concrete UI element that consists of a scale and a slider that can move across this scale. The slider widget has the minimum and maximum value which present the start and the end of the scale. It can be moved directly by the user during the input, or it can be moved if a value within its values range is displayed to it by the 4GL means.

134.3 Fields

Name	Type	Description
Step	Int	This is a number by which the value of the UI element can be increased or decreased at a time. It must be within the maximum and minimum value range. It prevents floating value changing.
Orientation	Orientation	This enum specifies whether the UI element should have vertical or horizontal layout.

135 Sorted

135.1 Diagram
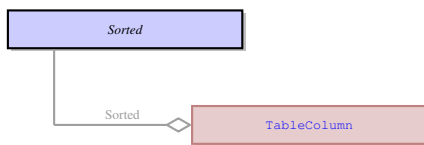
135.2 Description

Name: Sorted
No information
No parents.
No information

135.3 Options

Name	Description
None	The property is not applied and the default behaviour is used.
Asc	Not described yet
Desc	Not described yet

135.4 Referenced in

- Sorted field in optional TableColumn - No information

136 SpecificKeyEventHandler Not-referenced

136.1 Diagram
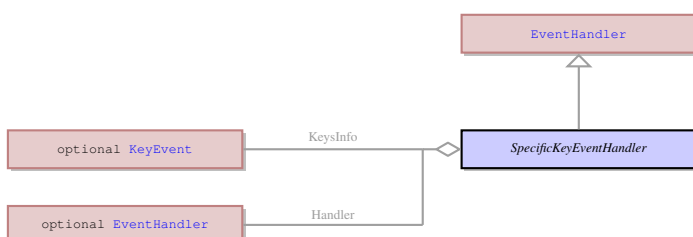
136.2 Description

Name: SpecificKeyEventHandler

This event handler specifies what event handler should be triggered when a specific key is pressed. It links the keypress with a 4GL event.

Parent: EventHandler - This is common class for all the specific event handler types.

This event handler specifies what event handler should be triggered when a specific key is pressed. It links the keypress with a 4GL event.

136.3 Fields

Name	Type	Description
KeysInfo	list of KeyEvent	It specifies the concrete keys that must be pressed to trigger the event.
Handler	optional EventHandler	It specifies the event handler that should be invoked on the keypress.

137 Spinner Not-referenced

137.1 Diagram
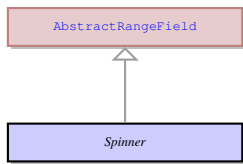
137.2 Description

Name: Spinner

This is a concrete UI element that has a form of a field available for inputting and displaying data that accepts only values inside the allowed range of values. It has the up and down arrows on the right that allow the user to scroll through the acceptable values and prevents the user from entering values from keyboard.

Parent: AbstractRangeField - It is an abstract UI element, which unites the concrete UI elements which accept only the values included into the specified range. It is typically a range or numeric values, for example from 1 to 100. The concrete UI elements that inherit their properties from the AbstractRangeField are Slider , ProgressBar , Spinner , and ScrollBar .

This is a concrete UI element that has a form of a field available for inputting and displaying data that accepts only values inside the allowed range of values. It has the up and down arrows on the right that allow the user to scroll through the acceptable values and prevents the user from entering values from keyboard.

137.3 Fields

Name	Type	Description
Step	Int	This is a number by which the value of the UI element can be increased or decreased at a time. It must be within the maximum and minimum value range. It prevents floating value changing.

138 StackPanel Not-referenced

138.1 Diagram

138.2 Description

Name: StackPanel

This is a container which arranges the elements in horizontal or vertical stacks. Any number of elements can be placed inside this container one next to the other. At runtime the contents of the stack panel can be resized only in the direction opposite to the orientation of the container.

Parent: ItemsContainer - The containers that can contain any number of UI elements inherit their properties from the ItemsContainer UI element. These are the containers that can contain any number of form fields and other containers, as opposed to the containers belonging to ElementContainer class.

This is a container which arranges the elements in horizontal or vertical stacks. Any number of elements can be placed inside this container one next to the other. At runtime the contents of the stack panel can be resized only in the direction opposite to the orientation of the container.

138.3 Fields

Name	Type	Description
Orientation	Orientation	This enum specifies whether the UI element should have vertical or horizontal layout.
Reverse	Bool	No information

139 StartProgramEventHandler Not-referenced

139.1 Diagram

139.2 Description

Name: StartProgramEventHandler

This event handler specifies the child 4GL program that should be launched and the parameters of this program. It is normally used for the MDI mode, but can be used in other cases.

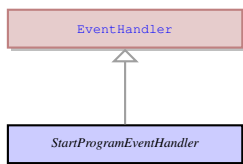**Parent:** EventHandler - This is common class for all the specific event handler types.

This event handler specifies the child 4GL program that should be launched and the parameters of this program. It is normally used for the MDI mode, but can be used in other cases.

139.3 Fields

Name	Type	Description
ProgramName	optional String	The name of the child program.
ProgramParameters	optional String	The parameters of the child program.
ProgramServer	optional String	The name of the host - the application server on which the program is deployed and should run.
ProgramPort	optional String	The port on the application server.
UserId	optional String	The name of the user who runs the application.
Waiting	Bool	It indicates whether the parent program should be suspended until the child program is closed.

140 StringLiteral Not-referenced

140.1 Diagram
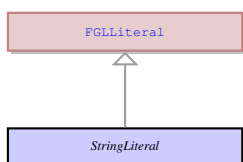
140.2 Description

Name: StringLiteral

These are character strings enclosed in quotation marks. It corresponds in the format to the requirements of the STRING, CHAR, VARCHAR 4GL data types. E.g. "abcde".

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

These are character strings enclosed in quotation marks. It corresponds in the format to the requirements of the STRING, CHAR, VARCHAR 4GL data types. E.g. "abcde".

140.3 Fields

Name	Type	Description
StringValue	String	This is one or more printable characters or white space characters enclosed in quotation marks.

141 Subtract Not-referenced

141.1 Diagram
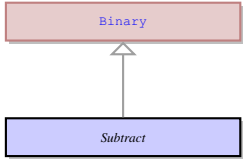
141.2 Description

Name: Subtract

It is a part of the conditional 4GL display attributes - -. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator -.

Parent: Binary - These are binary operators which have two operands - left operand and right operand. This class includes all the arithmetic and logical operators for the conditional properties.

It is a part of the conditional 4GL display attributes - -. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic operator -.

142 SystemColor Not-referenced

142.1 Diagram

142.2 Description

Name: SystemColor

The system color defines a list of preset colours that can be applied to widgets, as opposed to the custom colour where the user needs to specify RGB of the color.

Parent: Color - It is the root element to all color properties that can be applied to any UI element.

The system color defines a list of preset colours that can be applied to widgets, as opposed to the custom colour where the user needs to specify RGB of the color.

142.3 Fields

Name	Type	Description
SystemColorName	SystemColorName	It is the name of one of the predefined system colors.

143 SystemColorName

143.1 Diagram

143.2 Description

Name: SystemColorName

It is a name of a preset system color the color code for which is hard-coded and associated with this name.

No parents.

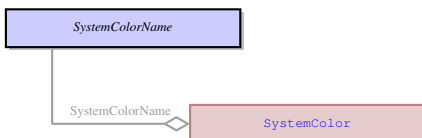It is a name of a preset system color the color code for which is hard-coded and associated with this name.

143.3 Options

Name	Description
None	The property is not applied and the default behaviour is used.
Black	RGB 0 0 0.
Gray	RGB 230 230 230.
DarkGray	RGB 75 75 75.
LightGray	RGB 217 217 217.
White	RGB 255 255 255.
Red	RGB 156 0 6.
LightRed	RGB 255 183 186.
Magenta	RGB 197 28 90.
LightMagenta	RGB 250 207 221.
Green	RGB 0 97 0.
LightGreen	RGB 190 240 200.
Blue	RGB 31 73 125.
LightBlue	RGB 190 210 240.
Cyan	RGB 49 134 155.
LightCyan	RGB 205 235 235.
Yellow	RGB 156 101 0.
LightYellow	RGB 255 235 156.
Purple	RGB 172 5 76.
LightPurple	RGB 228 186 232.
Orange	RGB 226 107 10.
LightOrange	RGB 253 233 217.

143.4 Referenced in

- SystemColorName field in optional SystemColor - It is a name of a preset system color the color code for which is hard-coded and associated with this name.

144 Tab Not-referenced

144.1 Diagram
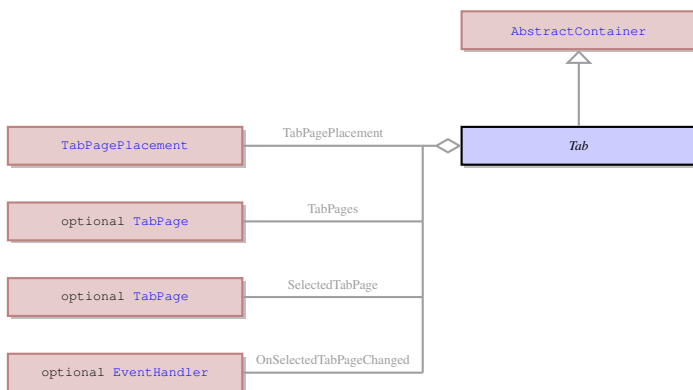
144.2 Description

Name: Tab

This is a special type of container which can contain any number of elements, but these elements can only be of TabPage . The Tab serves as the container for a stack of tab pages with only one page visible at a time. Other pages can be brought forward by clicking on their tabs.

Parent: AbstractContainer - This UI element represents an abstract container from which all the form containers their properties. This abstract UI element unites all form containers - elements that determine the form layout.

This is a special type of container which can contain any number of elements, but these elements can only be of TabPage . The Tab serves as the container for a stack of tab pages with only one page visible at a time. Other pages can be brought forward by clicking on their tabs.

144.3   Fields

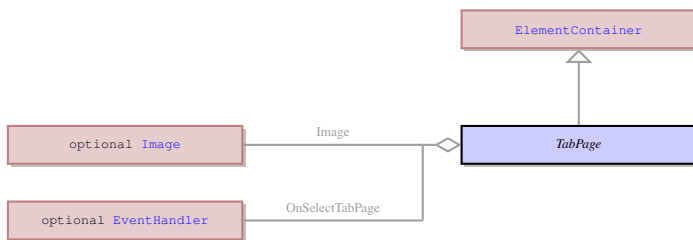| Name	Type	Description
TabPagePlacement	TabPagePlacement	It defines where the tabs should be located - to which side of the tab panel should they adjoin.
TabPages	list of TabPage	This is the set of tab pages that belong to the same tab container.
OnSelectedTabPageChanged	optional EventHandler	This is an event that is triggered every time the current tab page is changed.

145   TabPage Not-referenced

145.1   Diagram

145.2   Description

Name: TabPage

This is a container that can only be placed inside the Tab container. A tab page can contain a single element of any type. Each tab page has a tab with the page title which is used to bring the page forward from the stack of other tab pages at runtime or during form modification.

Parent: ElementContainer - This UI element unites all the containers which can contain exactly one element. The containers that derive from ElementContainer UI element can be logically opposed to containers derived from ItemsContainer UI element that can contain any number of elements of any type. The elements that inherit their properties from ElementContainer can encompass such elements as ring menu area or any other container. They can also contain an element belonging to ui.AbstractFiled class, but only one such element.

This is a container that can only be placed inside the Tab container. A tab page can contain a single element of any type. Each tab page has a tab with the page title which is used to bring the page forward from the stack of other tab pages at runtime or during form modification.

145.3   Fields

Name	Type	Description
Title	optional String	This is the inscription attached to the UI element. Usually this is the text of all sorts of labels.
Image	optional Image	This is an icon that can be displayed to the tab of the page with or instead of the page title.
OnSelectTabPage	optional EventHandler	This is an event that is triggered every time the tab page becomes the current tab page of the tab container and its contents is brought forward.

146   TabPagePlacement

146.1   Diagram
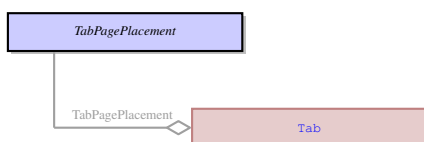
146.2   Description

Name: TabPagePlacement

This enum defined where th list of tabs should be located. By default it is located horizontally below the top border of the tab container. They can also be located horizontally at the bottom of the container or vertically at its either side.

No parents.

This enum defined where th list of tabs should be located. By default it is located horizontally below the top border of the tab container. They can also be located horizontally at the bottom of the container or vertically at its either side.

146.3   Options

Name	Description
Top	The UI element will be aligned to the top of the container (or container cell).
Left	The UI element will be aligned to the left side of the container (or container cell).
Right	The UI element will be aligned to the right side of the container (or container cell).
Bottom	The UI element will be aligned to the bottom of the container (or container cell).

146.4   Referenced in

- TabPagePlacement field in optional Tab - This enum defined where th list of tabs should be located. By default it is located horizontally below the top border of the tab container. They can also be located horizontally at the bottom of the container or vertically at its either side.

147   Table Not-referenced

147.1   Diagram

147.2   Description

Name: Table

This is a container that can only contain a specific type of element - TableColumn . It serves as the root container of a table with rows and columns of widgets used to display and input data.

Parent: AbstractDataTable - This UI element is used to display and edit data in a customized two-dimensional table of cells. The data in the cell therefore can be retrieved by specifying the row and column identifier of that cell in the table. AbstractDataTable UI element manages the overall appearance and behavior of the table, but does not have direct influence on the columns and rows.

This is a container that can only contain a specific type of element - TableColumn . It serves as the root container of a table with rows and columns of widgets used to display and input data.

148   TableColumn Not-referenced

148.1   Diagram

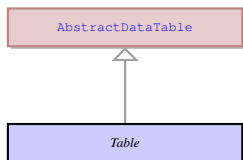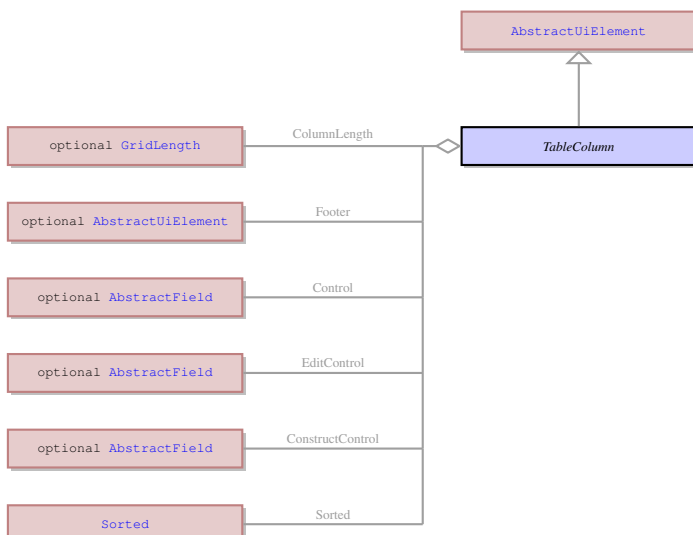## 148.2 Description

Name: TableColumn

This is a container that can only be placed inside the Table container or TreeTable container. It can contain only one element belonging to the AbstractField class. Though only one element can be placed into a column, this element will be repeated till the bottom of the column, creating table row together with the elements in other columns, if any. All the duplicates of the element will have the same identifier and will be treated as a single element by th e form designer. The 4GL can differentiate between the instances of the element belonging to different rows by means of using the element identifier together with the number of the table row. The table row numbers start at number 1 at the top of the table.

Parent: AbstractUiElement - AbstractUiElement is the base class for UI widgets. It is a generic UI element that can accept user actions. Most of concrete UI elements must inherit the properties and action types from the AbstractUIElement.
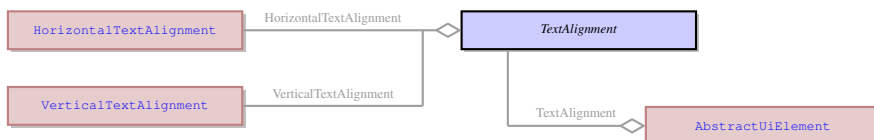
This is a container that can only be placed inside the Table container or TreeTable container. It can contain only one element belonging to the AbstractField class. Though only one element can be placed into a column, this element will be repeated till the bottom of the column, creating table row together with the elements in other columns, if any. All the duplicates of the element will have the same identifier and will be treated as a single element by th e form designer. The 4GL can differentiate between the instances of the element belonging to different rows by means of using the element identifier together with the number of the table row. The table row numbers start at number 1 at the top of the table.

148.3 Fields

Name	Type	Description
Text	optional String	This is the text used as the header of the column.
ColumnLength	optional GridLength	It specifies the length of a column. The column length determines how many rows of widgets the table will have.
Resizable	Bool	It indicates whether the user is allowed to resize the column at runtime using the mouse cursor.
AllowNewlines	Bool	This property specifies whether the Enter key will be used to move to another form element at runtime (if the value is FALSE), or it will create a newline symbol inside the current field (if the value is TRUE). It is typically applied for the TextArea element.
Control	optional AbstractField	No information
Unsortable	Bool	No information
Sorted	Sorted	No information

149 TextAlignment

149.1 Diagram

149.2 Description

Name: TextAlignment

It defines the alignment of the text inside the UI element to which it belongs. For example, it can define the alignment of the text inside a table cell or inside a text area.

No parents.

It defines the alignment of the text inside the UI element to which it belongs. For example, it can define the alignment of the text inside a table cell or inside a text area.

149.3 Fields

Name	Type	Description
HorizontalTextAlignment	HorizontalTextAlignment	
VerticalTextAlignment	VerticalTextAlignment	

149.4 Referenced in

- TextAlignment field in optional AbstractUiElement - It defines the alignment of the text inside the UI element to which it belongs. For example, it can define the alignment of the text inside a table cell or inside a text area.
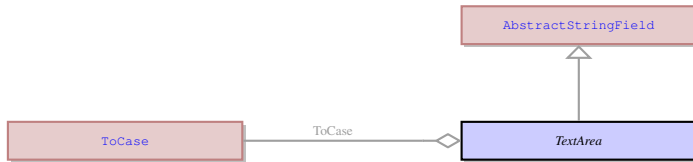
150   TextArea Not-referenced

150.1   Diagram

150.2   Description

Name: TextArea

This is a concrete UI element that has the form of a text field and shares many features with TextField , but is designed for working with multiline text instead of single lines of text. It does not have some features of the text field that deal with the navigation between fields, but instead it had improved facilities for navigating inside the field.

Parent: AbstractStringField - It is an abstract UI element, which unites the concrete UI elements that accept a character string as their value. Most of the concrete UI elements that are not containers inherit their properties from this element.
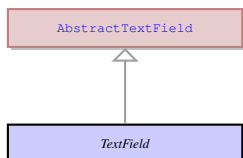
This is a concrete UI element that has the form of a text field and shares many features with TextField , but is designed for working with multiline text instead of single lines of text. It does not have some features of the text field that deal with the navigation between fields, but instead it had improved facilities for navigating inside the field.

150.3   Fields

Name	Type	Description
ToCase	ToCase	This property specifies the case of a UI element. It can be applied to any UI element that allows entering text from keyboard. By default its value is None, meaning that the case of the letters does not change and remains as they were inputted.
MaxLength	optional Int	It specifies the maximum length in bytes allowed for entering into the filed. Its value is normally taken from the data type and size of the variable linked to the field.
AllowTabulation	Bool	It indicates whether the Tab key will move the cursor to the next field (FALSE - default value) or create a TAB symbol inside the field.
Editor	optional String	Specifies the program to be used for opening and editing the BYTE or TEXT value.
Autonext	Bool	If enabled, moves the cursor to the next field during input automatically, when the MaxLength of the current field is met.
Required	Bool	No information
PlaceholderText	optional String	No information
LabelText	optional String	No information
HelperText	optional String	No information

151   TextField Not-referenced

151.1   Diagram

151.2   Description

Name: TextField

This is a concrete UI element that is commonly used for input and displaying information. Normally it is used to process a single line of data.

Parent: AbstractTextField - It is an abstract UI element, which unites a subset of AbstractStringField elements with the exception of TextArea , ComboBox , and Button . Typically it includes the UI elements which allow entering values, like normal text fields, and usually are only one line wide.

This is a concrete UI element that is commonly used for input and displaying information. Normally it is used to process a single line of data.

151.3   Fields

Name	Type	Description
AllowNewlines	Bool	This property specifies whether the Enter key will be used to move to another form element at runtime (if the value is FALSE), or it will create a newline symbol inside the current field (if the value is TRUE). It is typically applied for the TextArea element.
InvisibleValue	optional Bool	If enabled, the value displayed to the field will be invisible. During input the value will be masked with *.
PlaceholderText	optional String	No information
LabelText	optional String	No information
HelperText	optional String	No information

152   TextInjectionEventHandler Not-referenced

152.1   Diagram
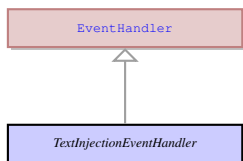
152.2   Description

Name: TextInjectionEventHandler

This event handler injects the text specified as its parameter into the current input widget. It can be assigned to any event.

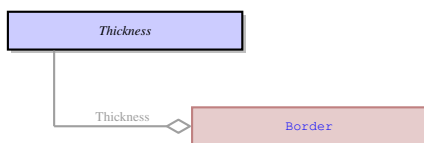**Parent:** EventHandler - This is common class for all the specific event handler types.

This event handler injects the text specified as its parameter into the current input widget. It can be assigned to any event.

152.3   Fields

Name	Type	Description
Text	optional String	A character string.

153   Thickness

153.1   Diagram

153.2   Description

Name: Thickness

This is a property which defines the thickness of elements or their parts. It is use to define the thickness of the border, the width or padding and margin offsets. The parts of the same object (e.g. border) can have different thickness in its different parts - for example a border can be 1 pixel wide at the top and 2 pixels wide at the bottom. If the thickness of any side is set to 0 - this side of the element absent.

No parents.

This is a property which defines the thickness of elements or their parts. It is use to define the thickness of the border, the width or padding and margin offsets. The parts of the same object (e.g. border) can have different thickness in its different parts - for example a border can be 1 pixel wide at the top and 2 pixels wide at the bottom. If the thickness of any side is set to 0 - this side of the element absent.

153.3   Fields

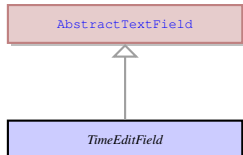| Name	Type	Description
Left	Int	The size of the left standoff in pixels.
Top	Int	The size of the top standoff in pixels.

| Right | Int | The size of the right standoff in pixels. |
| Bottom | Int | The size of the bottom standoff in pixels. |

153.4   Referenced in

- Thickness field in optional Border - This is a property which defines the thickness of elements or their parts. It is use to define the thickness of the border, the width or padding and margin offsets. The parts of the same object (e.g. border) can have different thickness in its different parts - for example a border can be 1 pixel wide at the top and 2 pixels wide at the bottom. If the thickness of any side is set to 0 - this side of the element absent.

154   TimeEditField Not-referenced

154.1   Diagram

154.2   Description

Name: TimeEditField

This is a concrete UI element that accepts a limited range of time values. The value inside the field is formatted into hh:mm:ss format. It also has up and down arrows that can scroll the data in the field - whether hours, minutes or seconds are scrolled depends on there inside the field the cursor is located.

Parent: AbstractTextField - It is an abstract UI element, which unites a subset of AbstractStringField elements with the exception of TextArea , ComboBox , and Button . Typically it includes the UI elements which allow entering values, like normal text fields, and usually are only one line wide.

This is a concrete UI element that accepts a limited range of time values. The value inside the field is formatted into hh:mm:ss format. It also has up and down arrows that can scroll the data in the field - whether hours, minutes or seconds are scrolled depends on there inside the field the cursor is located.

154.3   Fields

Name	Type	Description
LabelText	optional String	No information
HelperText	optional String	No information
PlaceholderText	optional String	No information

155   TitleJustification

155.1   Diagram
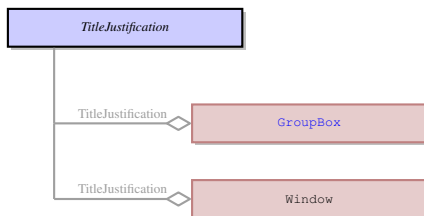
155.2   Description

Name: TitleJustification

This enum defines the horizontal justification of the title text. It is typically is applied to window titles, column header titles, tab page titles, etc..

No parents.

This enum defines the horizontal justification of the title text. It is typically is applied to window titles, column header titles, tab page titles, etc..

155.3 Options

Name	Description
Left	The UI element will be aligned to the left side of the container (or container cell).
Center	The UI element will be equidistant from both sides.
Right	The UI element will be aligned to the right side of the container (or container cell).

155.4 Referenced in

- TitleJustification field in optional GroupBox - This enum defines the horizontal justification of the title text. It is typically is applied to window titles, column header titles, tab page titles, etc..

- TitleJustification field in optional Window - This enum defines the horizontal justification of the title text. It is typically is applied to window titles, column header titles, tab page titles, etc..
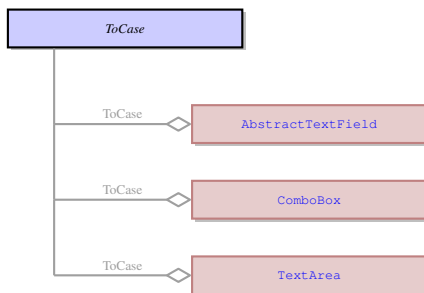
156 ToCase

156.1 Diagram

156.2 Description

Name: ToCase

This is the case (lower case or upper case) to be applied to the text in the UI element.
No parents.
This is the case (lower case or upper case) to be applied to the text in the UI element.

156.3 Options

Name	Description
None	The property is not applied and the default behaviour is used.
Up	All the letters entered into the UI element will be uppercase letters regardless of their original case.
Down	All the letters entered into the UI element will be lowercase letters regardless of their original case.

156.4 Referenced in

- ToCase field in optional AbstractTextField - This is the case (lower case or upper case) to be applied to the text in the UI element.

- ToCase field in optional ComboBox - This is the case (lower case or upper case) to be applied to the text in the UI element.

- ToCase field in optional TextArea - This is the case (lower case or upper case) to be applied to the text in the UI element.

157 Today Not-referenced

157.1 Diagram
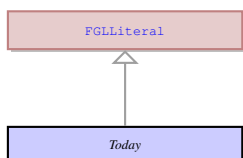
157.2 Description

Name: Today

It is a part of the conditional 4GL display attributes - TODAY(). The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals.

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.
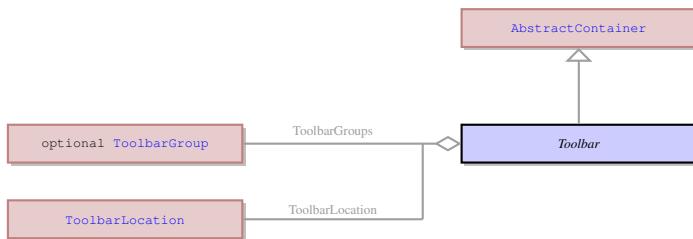
It is a part of the conditional 4GL display attributes - TODAY(). The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals.

158 Toolbar Not-referenced

158.1 Diagram

158.2 Description

Name: Toolbar

This is the container that incorporates toolbar buttons.

Parent: AbstractContainer - This UI element represents an abstract container from which all the form containers their properties. This abstract UI element unites all form containers - elements that determine the form layout.

This is the container that incorporates toolbar buttons.

158.3 Fields

Name	Type	Description
ToolbarGroups	list of ToolbarGroup	A set of all toolbar groups that belong to the toolbar.
HideLabels	Bool	It specifies whether the text on the toolbar buttons should be visible or not. If set to true - only the icons will be visible.
ToolbarLocation	ToolbarLocation	No information

159 ToolbarButton Not-referenced

159.1 Diagram
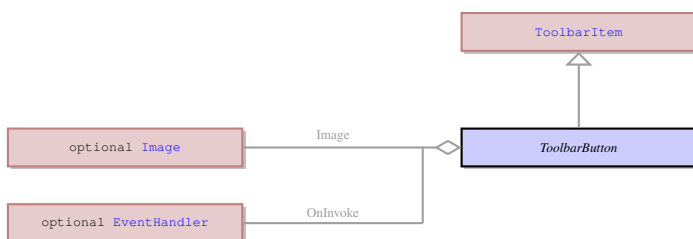
159.2 Description

Name: ToolbarButton

This is an individual toolbar button that belongs to the toolbar.

Parent: ToolbarItem - This is an abstract element that unites the toolbar buttons and toolbar separators.

This is an individual toolbar button that belongs to the toolbar.

159.3 Fields

Name	Type	Description
Text	optional String	This is the label of the toolbar button.

AllowNewlines	Bool	This property specifies whether the Enter key will be used to move to another form element at runtime (if the value is FALSE), or it will create a newline symbol inside the current field (if the value is TRUE). It is typically applied for the TextArea element.
Image	optional Image	It specifies the icon that should be displayed to the toolbar button. The button is resized to the size of the icon applied.
OnInvoke	optional EventHandler	The event which is triggered when the UI element is invoked. It can be invoked by mouse click, by pressing Enter, or in some cases Space, when the cursor is in the element.

160 ToolbarGroup Not-referenced

160.1 Diagram

160.2 Description

Name: ToolbarGroup

This is a set of toolbar buttons that are united into a single group. The group unites the toolbar buttons that have the same conditions for being displayed. It was designed to make the toolbar more dynamic - to display or hide the toolbar groups depending on what widgets are active and to combine different groups freely.

Parent: AbstractContainer - This UI element represents an abstract container from which all the form containers their properties. This abstract UI element unites all form containers - elements that determine the form layout.

This is a set of toolbar buttons that are united into a single group. The group unites the toolbar buttons that have the same conditions for being displayed. It was designed to make the toolbar more dynamic - to display or hide the toolbar groups depending on what widgets are active and to combine different groups freely.

160.3 Fields

Name	Type	Description
ToolbarItems	list of ToolbarItem	This is the list of Toolbar elements - toolbar buttons, toolbar separators - present in the toolbar UI element.

161 ToolbarItem Not-referenced

161.1 Diagram

161.2 Description

Name: ToolbarItem

This is an abstract element that unites the toolbar buttons and toolbar separators.

Parent: AbstractUiElement - AbstractUiElement is the base class for UI widgets. It is a generic UI element that can accept user actions. Most of concrete UI elements must inherit the properties and action types from the AbstractUIElement.

This is an abstract element that unites the toolbar buttons and toolbar separators.
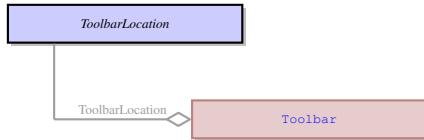
161.3 Children

- ToolbarButton - This is an individual toolbar button that belongs to the toolbar.

- ToolbarSeparator - This is a visual separator that can visually divide the toolbar into logical sets of buttons.

161.4 Fields

Name	Type	Description
Place	optional String	No information

162 ToolbarLocation

162.1 Diagram

162.2 Description

Name: ToolbarLocation
　　No information
　　No parents.
　　No information
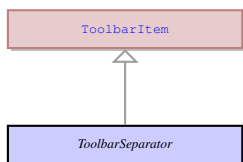
162.3 Options

Name	Description
Top	The UI element will be aligned to the top of the container (or container cell).
Right	The UI element will be aligned to the right side of the container (or container cell).

162.4 Referenced in

- ToolbarLocation field in optional Toolbar - No information

163 ToolbarSeparator Not-referenced

163.1 Diagram
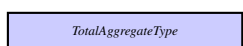

163.2 Description

Name: ToolbarSeparator
　　This is a visual separator that can visually divide the toolbar into logical sets of buttons.
　　Parent: ToolbarItem - This is an abstract element that unites the toolbar buttons and toolbar separators.
　　This is a visual separator that can visually divide the toolbar into logical sets of buttons.

164 TotalAggregateType Not-referenced

164.1 Diagram

164.2 Description

Name: TotalAggregateType
　　It defines the type of the calculations that should be performed on the values of the whole column.
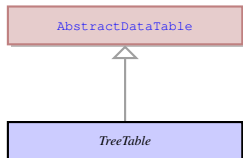　　No parents.
　　It defines the type of the calculations that should be performed on the values of the whole column.

164.3 Options

Name	Description
Sum	An aggregate type that calculates the total sum of all the values in a column.
Program	This aggregate does not calculate anything automatically and indicates that the aggregate values will be calculated by 4GL and then displayed to the aggregate field by the programmer.
Avg	An aggregate type that calculates the average value among all the values in a column.
Min	An aggregate type that calculates the minimum value among all the values in a column.
Max	An aggregate type that calculates the maximum value among all the values in a column.
Count	An aggregate type that calculates how many values there are in a column.

165 TreeTable Not-referenced

165.1 Diagram

165.2 Description

Name: TreeTable

This is a special container that can contain only TableColumn elements. It is similar to a table, but arranges the items in a hierarchical order and allows to fold and unfold rows.

Parent: AbstractDataTable - This UI element is used to display and edit data in a customized two-dimensional table of cells. The data in the cell therefore can be retrieved by specifying the row and column identifier of that cell in the table. AbstractDataTable UI element manages the overall appearance and behavior of the table, but does not have direct influence on the columns and rows.
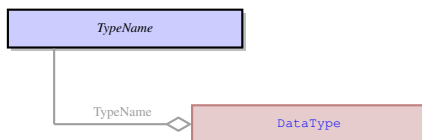
This is a special container that can contain only TableColumn elements. It is similar to a table, but arranges the items in a hierarchical order and allows to fold and unfold rows.

166 TypeName

166.1 Diagram

166.2 Description

Name: TypeName

This is the data type of the form widget. It restricts the type of data that can be entered into this widget, e.g. Integer field type does not allow other characters besides digits.

No parents.

This is the data type of the form widget. It restricts the type of data that can be entered into this widget, e.g. Integer field type does not allow other characters besides digits.

166.3 Options

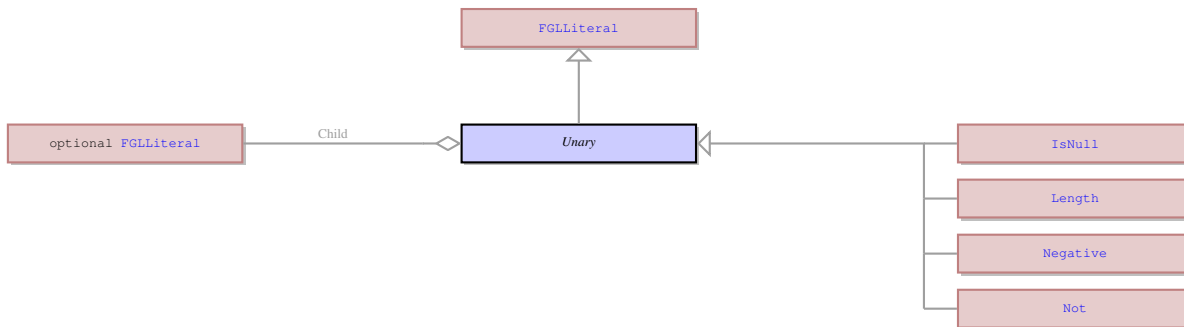| Name	Description
BigInt	Integer numbers taking up to 8 bytes.
Boolean	Boolean values represented by TRUE, FALSE, or NULL.
Byte	Large binary data (e.g. images).
Char	A fixed size character string.
Date	Calendar dates whose formatting depends on the locale.
DateTime	Data that contain calendar date and/or time.
Decimal	Decimal numbers with fixed decimal point.
Float	Floating point decimal numbers taking up to 8 bytes.
Integer	Integer numbers taking up to 4 bytes.

Interval	Time range with time units from years to fractions of second.
Money	Decimal values with the automatically added currency symbol.
SmallFloat	Floating point decimal numbers taking up to 4 bytes.
SmallInt	Integer numbers taking up to 2 bytes.
Text	Large binary data (e.g. plain text).
VarChar	A variable size character string.
String	A dynamic character string.

166.4   Referenced in

- TypeName field in optional DataType - This is the data type of the form widget. It restricts the type of data that can be entered into this widget, e.g. Integer field type does not allow other characters besides digits.

167   Unary Not-referenced

167.1   Diagram

167.2   Description

Name: Unary

These are binary operators. This class includes plus and minus operators that specify whether a number is positive or negative.

Parent: FGLLiteral - This is common class for all the specific literals that can be accepted by the fields. The 4GL literals are typically applied to the default and included values of the form fields.

These are binary operators. This class includes plus and minus operators that specify whether a number is positive or negative.

167.3   Children

- IsNull - It defines whether the field is allowed to accept NULL values.

- Length - No information

- Negative - It is a part of the conditional 4GL display attributes - -. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This an arithmetic unary operator - used to differentiate negative numbers from positive.

- Not - It is a part of the conditional 4GL display attributes - NOT. The condition of the attribute is specified as a string which is later parsed and turned into a set of relational operators, boolean operators and 4GL literals. This is the operator NOT.

167.4   Fields

Name	Type	Description
Child	optional FGLLiteral	This the right operand of an unary operator.

168   ValueInclude Not-referenced

168.1   Diagram

168.2 Description

Name: ValueInclude

This class defines individual literals included into a list of allowed values for a field.

Parent: Include - This class defines what literal values are allowed for input into a form field. It can be either individual literals or a range of values.
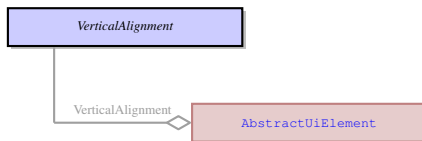
This class defines individual literals included into a list of allowed values for a field.

168.3 Fields

Name	Type	Description
Value	FGLLiteral	This is any 4GL literal usually specifying the default value of a widget.

169 VerticalAlignment

169.1 Diagram

169.2 Description

Name: VerticalAlignment

This enum specifies the vertical alignment of a UI element inside a container. It is applicable to UI elements inside any container except coord panel. It defines to which border of the container (or container cell) - top or bottom - the element must adjoin.

No parents.

This enum specifies the vertical alignment of a UI element inside a container. It is applicable to UI elements inside any container except coord panel. It defines to which border of the container (or container cell) - top or bottom - the element must adjoin.
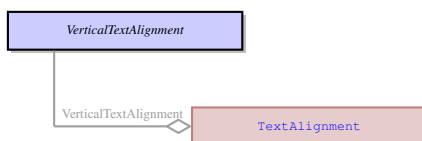
169.3 Options

Name	Description
Default	The window size is the size with which it was opened or which was set after opening by 4GL or graphical theme means.
Stretch	The UI element will be stretched to fit the container (or container cell) without preserving the aspect ratio.
Top	The UI element will be aligned to the top of the container (or container cell).
Center	The UI element will be equidistant from both sides.
Bottom	The UI element will be aligned to the bottom of the container (or container cell).

169.4 Referenced in

- VerticalAlignment field in optional AbstractUiElement - This enum specifies the vertical alignment of a UI element inside a container. It is applicable to UI elements inside any container except coord panel. It defines to which border of the container (or container cell) - top or bottom - the element must adjoin.

170 VerticalTextAlignment

170.1 Diagram

170.2 Description

Name: VerticalTextAlignment

No parents.

170.3 Options

Name	Description
Default	The window size is the size with which it was opened or which was set after opening by 4GL or graphical theme means.
Top	The UI element will be aligned to the top of the container (or container cell).
Center	The UI element will be equidistant from both sides.
Bottom	The UI element will be aligned to the bottom of the container (or container cell).

170.4 Referenced in

- VerticalTextAlignment field in optional TextAlignment -

171 WebComponent Not-referenced

171.1 Diagram
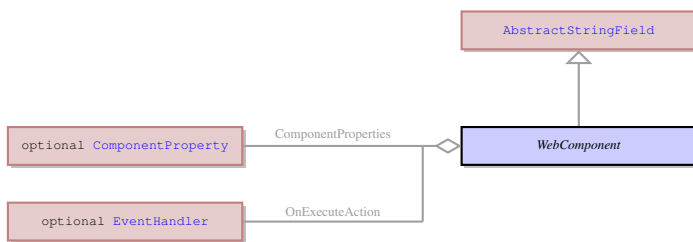
171.2 Description

Name: WebComponent

It is a concrete UI element that serves as a container for third party web components. It is basically just the space which is filled by the web component at runtime.

Parent: AbstractStringField - It is an abstract UI element, which unites the concrete UI elements that accept a character string as their value. Most of the concrete UI elements that are not containers inherit their properties from this element.

It is a concrete UI element that serves as a container for third party web components. It is basically just the space which is filled by the web component at runtime.

171.3 Fields

Name	Type	Description
ComponentType	optional String	This is the name of a web component. The web component folder should be located in the components directory on the application server. The HTML file describing the component should be located in the same folder as the component sources and have the same name as the component folder. For example: C:/ProgramDat/Querix/Lycia 6/components/Charts/charts.html - in this case the component type will be 'charts'.
ComponentProperties	list of ComponentProperty	These are specific properties. Their types and number are defines by the HTML file describing the web component.

172 YesNoAuto Not-referenced

172.1 Diagram
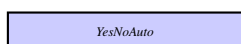
172.2 Description

Name: YesNoAuto

This enum is used by the fields that accept values Yes, No and Auto. It was introduced for compatibility reasons.

No parents.

This enum is used by the fields that accept values Yes, No and Auto. It was introduced for compatibility reasons.

172.3   Options

Name	Description
Auto	This is the Auto value.
Yes	This is the Yes value..
No	This is the No value.