

TextBox

Control in the edit form

Content

1	Basic information	5
1.1	Description of the control	5
1.2	Create a new control.....	5
1.3	Edit or delete a control.....	6
2	List of tabs in the control settings dialog	7
2.1	“General” tab	8
2.1.1	Name.....	8
2.1.2	Abbreviated name	8
2.1.3	Dictionary... ..	9
2.1.4	Data type – validation.....	9
2.1.5	Unit.....	21
2.1.6	Width	22
2.1.7	Maximum length.....	22
2.1.8	Rows.....	22
2.1.9	Alignment	22
2.1.10	Options.....	23
2.2	“Default value” tab.....	28
2.2.1	Default value	28
2.2.2	Options.....	31
2.3	“Events” tab	32
2.3.1	OnChange	32
2.3.2	OnClick	33
2.3.3	OnFocus	33
2.3.4	OnBlur	33
2.4	“Button” tab.....	34
2.4.1	Title	34
2.4.2	Width	34
2.4.3	Client.....	34
2.4.4	Server.....	34
2.5	“Help” tab	35
2.5.1	Help.....	35

2.5.2	Comment	36
2.6	"Filter" tab	37
2.6.1	Options.....	37
2.6.2	The number of records per page	37
2.7	"Rights" tab	38
2.7.1	Rights.....	38
2.7.2	Show value	39
2.7.3	Replace rights with other controls.....	39
2.8	"Administration" tab	40
2.8.1	Identifier	40
2.8.2	Full identifier	41
2.8.3	Index	41
2.8.4	Search in the View table	41
2.8.5	Other.....	42
2.8.6	Statistics.....	42
3	Picker.....	43
3.1	Basic information	43
3.2	"General" tab	44
3.2.1	Tooltip	44
3.2.2	Data source	44
3.2.3	Records per page.....	45
3.2.4	Options.....	45
3.2.5	Assignment table.....	46
3.3	"Help" tab	47
3.4	"Activate search" tab.....	48
3.5	"Other" tab	49
4	Examples	50
4.1	Javascript functions	50
4.1.1	General functions	50
4.1.2	Text strings	50
4.1.3	Whole number	50
4.1.4	Decimal numbers	50
4.1.5	Currencies.....	50
4.1.6	Dates.....	51

4.1.7	Times.....	51
4.1.8	E-mails	51
4.2	Server functions	51
4.2.1	General functions	51
4.2.2	Whole numbers	52
4.2.3	Decimal numbers	52
4.2.4	Currencies.....	52
4.2.5	Dates.....	52
4.2.6	Times.....	52
4.2.7	E-mails	52

1 Basic information

1.1 Description of the control

TextBox is a text field of any width and height intended for entering plain text with a maximum length of up to 250 characters for the Firebird database and 4000 characters for the MSSQL database without the possibility of inserting a forced transition to a new line – enter. When the width is set to 0 pixels, the text box does not display the typical border, and the text editing option is also missing – the value of the text box is displayed as a regular paragraph with plain text.

1.2 Create a new control

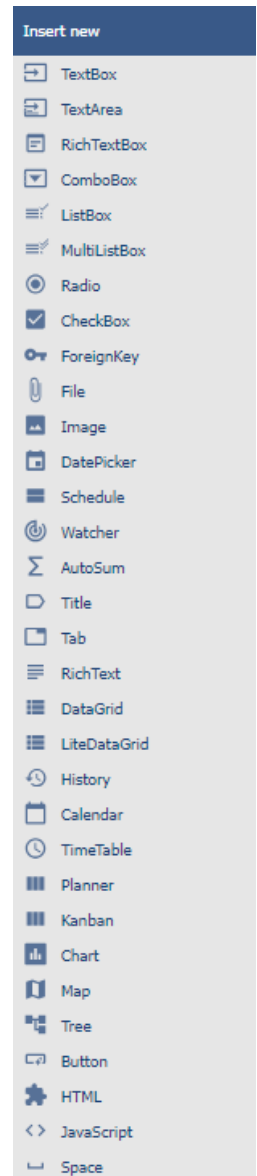
There are two basic ways to create a new textbox:

From the edit form

- The most common way is to create a new textbox directly from the edit form using the toolbar with controls.
- First, the position in which the new textbox is to be placed is determined by clicking the mouse in the edit form. This will highlight the selected position with a gray horizontal line. Then just click on the “TextBox” item in the toolbar of the edit form, fill in the attributes of the new control in the newly opened dialog, and then save. This will insert a new textbox at a pre-selected location on the edit form.
- If the desired position is not determined before inserting a new textbox, the new textbox will be inserted at the end of the edit form.

From the view table

- The view table has an icon on the right edge to add a new column. The following list of controls allows you to select a new textbox, which will be placed in the edit form immediately after the control that appears last in the view table. After filling in the attributes of the new control and saving it, a new textbox will be inserted into the edit form, and at the same time a new column will be added to the view table, which is associated with the textbox just created.



1.3 Edit or delete a control

- For each existing control, a pencil icon appears on the left side of the edit form to change or delete the control's settings.
- Individual controls can be moved vertically in the edit form using the Drag&Drop method. Moving consists of grabbing the control, preferably after its name located in the left column of the edit form, and then dragging it to the desired place in the edit form. For safety reasons, it is important to have the function of moving controls enabled using the "Drag&Drop" check box located above the control toolbar. This field is unchecked each time you log in to the application.
- Deleting a textbox results in the irreversible deletion of the database column associated with this textbox in the database, and thus in the deletion of its values in all records of the database table. Some text boxes in system edit forms cannot be deleted because they are necessary for the basic functionality of the entire application.

2 List of tabs in the control settings dialog

General	Setting general properties
Default value	Setting the default value for newly created records
Events	Setting javascript events
Button	Button settings that can be displayed to the right of the control
Help	Settings for help text that can be displayed on the right side of the edit form
Filter	Setting the filter that can be displayed in the view table in the column header
Rights	Rights settings
Administration	Setting other properties

2.1 “General” tab

General	Default value	Events	Button	Help	Filter	Rights
* Name	<input type="text"/>					Dictionary... ?
Data type	Text string					?
* Width	<input type="text" value="150"/> px					0 - 10 000px
* Maximum length	<input type="text" value="50"/> characters					1 - 250
Rows	<input type="text"/>					1 - 250
Options	<input type="checkbox"/> Required field <input type="checkbox"/> Read-only <input type="checkbox"/> Hidden field <input type="checkbox"/> Show Color Picker in the Edit form <input type="checkbox"/> Show as link in the View table to open the record on the new tab <input type="checkbox"/> Allow multiple assignments <input type="checkbox"/> Snap to previous control					
<div> <div>Save</div> <div>Save and continue editing</div> <div>Delete</div> </div>						

2.1.1 Name

- The name of the control displayed in the edit form in a separate column with a colored background to the left of the text box itself.

2.1.2 Abbreviated name

- Optional control name displayed in the view table header if the control is selected as one of the view table columns.
- If an abbreviated name is specified, the full name of the control is displayed in the tooltip above the abbreviated name.
- If an abbreviated name is not specified, the full name of the control is displayed in the header of the view table.
- The abbreviated name "" (fixed space) can completely hide the name of the control displayed in the view table.

2.1.3 Dictionary...

- The dictionary displays a list of names of other controls, forms, view pages, categories, applications, and application groups used throughout the application.
- The dictionary search is used to design the correct control name and to maintain the consistency of the nomenclature throughout the application so that controls with the same meaning located in different places in the application are named in the same way.

2.1.4 Data type – validation

- The data type defines the type or meaning of the values that the control can take.
- The data type specifies the database type of the column in the database table that is associated with the current form.
- The drop-down list contains the following data types:

Text string

General	Default value	Events	Button	Help	Filter	Rights
* Name	<input type="text"/>					Dictionary... ?
Data type	Text string					?
* Width	<input type="text" value="150"/> <input type="text"/>	px				0 - 10 000px
* Maximum length	<input type="text" value="50"/> characters					1 - 250
Rows	<input type="text"/>					1 - 250
Options	<input type="checkbox"/> Required field <input type="checkbox"/> Read-only <input type="checkbox"/> Hidden field <input type="checkbox"/> Show Color Picker in the Edit form <input type="checkbox"/> Show as link in the View table to open the record on the new tab <input type="checkbox"/> Allow multiple assignments <input type="checkbox"/> Snap to previous control					
<div> <input type="button" value="Save"/> <input type="button" value="Save and continue editing"/> <input type="button" value="Delete"/> </div>						

- “String” with a maximum length of 250 characters for the Firebird database and 4000 characters for the MSSQL database, represented as the database type “varchar” for the Firebird database and “nvarchar” for the MSSQL database.
- The value of this field in the edit form is validated only to the specified maximum length.

Whole number

General	Default value	Events	Button	Help	Filter	Rights
* Name	<input type="text"/>					Dictionary... ?
Data type	Whole number					?
* Width	<input type="text"/> 150 <input type="text"/> px					0 - 10 000px
* Maximum length	50 characters					1 - 250
Alignment	Left					?
Options	<input type="checkbox"/> Required field <input checked="" type="checkbox"/> Read-only <input type="checkbox"/> Disable value assignment using JavaScript <input type="checkbox"/> Hidden field <input type="checkbox"/> Show the "Sort order..." icon in the View table <input type="checkbox"/> Show as link in the View table to open the record on the new tab <input checked="" type="checkbox"/> Allow multiple assignments <input type="checkbox"/> Run OnBeforeSave/OnAfterSave before the record is saved to the database <input checked="" type="checkbox"/> Hide the sum of the column in the View table <input checked="" type="checkbox"/> Snap to previous control					
<div> <input type="button" value="Save"/> <input type="button" value="Save and continue editing"/> <input type="button" value="Delete"/> </div>						

- Whole number "with a length of 4 bytes represented as a database type" int ".
- The value of this field in the edit form is validated to the specified maximum length, to characters from the set 0 to 9, and to the range of the entered value in the range from -2147483648 to 2147483647.
- In the view table, an "whole number" column can be automatically added to the footer of the table.

Big whole number

General	Default value	Events	Button	Help	Filter	Rights
* Name	<input type="text"/>					Dictionary... ?
Data type	Big whole number					?
* Width	<input type="text"/> 150 <input type="text"/> px					0 - 10 000px
* Maximum length	50 characters					1 - 250
Alignment	Left					?
Options	<input type="checkbox"/> Required field <input checked="" type="checkbox"/> Read-only <input type="checkbox"/> Disable value assignment using JavaScript <input type="checkbox"/> Hidden field <input type="checkbox"/> Show as link in the View table to open the record on the new tab <input checked="" type="checkbox"/> Allow multiple assignments <input type="checkbox"/> Run OnBeforeSave/OnAfterSave before the record is saved to the database <input checked="" type="checkbox"/> Hide the sum of the column in the View table <input checked="" type="checkbox"/> Snap to previous control					
<div> <input type="button" value="Save"/> <input type="button" value="Save and continue editing"/> <input type="button" value="Delete"/> </div>						

- “Long” with a length of 8 bytes represented as a database type “bigint”.
- The value of this field in the edit form is validated to the specified maximum length, to characters from the set 0 to 9, and to the range of the entered value in the range from -9223372036854775808 to 9223372036854775807.
- In the view table, a “big whole number” column can be automatically added to the footer of the table.

Decimal number

General	Default value	Events	Button	Help	Filter	Rights
* Name	<input type="text"/>					Dictionary... ?
Data type	Decimal number					?
Unit	<input type="text"/>	<input type="text"/>	px			%, {0.00}, {0.00}% , ...
* Width	<input type="text"/>	150	<input type="text"/>	px		0 - 10 000px
* Maximum length	50	characters				1 - 250
Alignment	Right					?
Options	<input type="checkbox"/> Required field <input checked="" type="checkbox"/> Read-only <input type="checkbox"/> Disable value assignment using JavaScript <input type="checkbox"/> Hidden field <input checked="" type="checkbox"/> Allow multiple assignments <input type="checkbox"/> Run OnBeforeSave/OnAfterSave before the record is saved to the database <input checked="" type="checkbox"/> Hide the sum of the column in the View table <input checked="" type="checkbox"/> Snap to previous control					
<div> <div>Save</div> <div>Save and continue editing</div> <div>Delete</div> </div>						

- "Double" with a length of 8 bytes represented as a database type "float".
- The value of this field in the edit form is validated to the specified maximum length, to characters from the set 0 to 9 and a comma as a decimal separator, and to a range of the entered value from - 1000000000000000 to 1000000000000000.
- Everywhere in the application, except for the text field itself in the edit form, the decimal number is displayed, including the optional symbol, and the displayed value is not rounded.
- In the view table, a "decimal number" column can be automatically added to the footer of the table.

Currency

General	Default value	Events	Button	Help	Filter	Rights
* Name	<input type="text"/>					Dictionary... ?
Data type	Currency					?
Unit	<input type="text"/>	<input type="text"/>	px			%, {0.00}, {0.00}%, ...
* Width	<input type="text"/>	150	<input type="text"/>	px		0 - 10 000px
* Maximum length	50	characters				1 - 250
Alignment	Right					?
Options	<input type="checkbox"/> Required field <input checked="" type="checkbox"/> Read-only <input type="checkbox"/> Disable value assignment using JavaScript <input type="checkbox"/> Hidden field <input checked="" type="checkbox"/> Allow multiple assignments <input type="checkbox"/> Run OnBeforeSave/OnAfterSave before the record is saved to the database <input checked="" type="checkbox"/> Hide the sum of the column in the View table <input checked="" type="checkbox"/> Snap to previous control					
<div> <div>Save</div> <div>Save and continue editing</div> <div>Delete</div> </div>						

- "Double" with a length of 8 bytes represented as a database type "float".
- The value of this field in the edit form is validated to the specified maximum length, to characters from the set 0 to 9 and a comma as a decimal separator, and to a range of the entered value from - 1000000000000000 to 1000000000000000.
- Everywhere in the application except the text box itself in the edit form, the currency is displayed, including an optional symbol, rounded to two decimal places.
- In the view table, a "currency" column can be automatically added to the footer of the table.

Date

General	Default value	Events	Button	Help	Filter	Rights
<p>* Name <input type="text"/> <input type="text"/> Dictionary... ?</p> <p>Data type Date ?</p> <p>* Width <input type="text"/> <input type="text" value="150"/> <input type="text"/> px 0 - 10 000px</p> <p>* Maximum length <input type="text" value="50"/> characters 1 - 250</p> <p>Options</p> <p><input type="checkbox"/> Required field</p> <p><input checked="" type="checkbox"/> Read-only</p> <p><input type="checkbox"/> Disable value assignment using JavaScript</p> <p><input type="checkbox"/> Hidden field</p> <p><input checked="" type="checkbox"/> Show time</p> <p><input type="checkbox"/> Including seconds</p> <p><input type="checkbox"/> Show abbreviated name of the day ?</p> <p><input type="checkbox"/> Show week number ?</p> <p><input checked="" type="checkbox"/> Allow multiple assignments</p> <p><input type="checkbox"/> Run OnBeforeSave/OnAfterSave before the record is saved to the database</p> <p><input checked="" type="checkbox"/> Snap to previous control</p>						

- "Datetime" with a length of 8 bytes represented as a database type "datetime".
- The value of this field in the edit form is validated to the specified maximum length, to characters from the set 0 to 9 and a period or slash as a separator of individual date components, and to the range of the entered value in the range from 1.1.1901 to 31.12.9999.
- Everywhere in the application, the date is displayed according to the basic settings of the portal (dd/mm/yyyy or mm/dd/yyyy) and according to the language habits of the currently logged in user (dot or slash as a separator), optionally with day name or week number.
- The date may or may not be entered, including the time. If the time is used, it is displayed in the edit form and only the hour and minute components are stored in the database – seconds, milliseconds and ticks are removed from the date.

Time (hh: mm)

General	Default value	Events	Button	Help	Filter	Rights
<div> <div>* Name</div> <div> <input type="text"/> <input type="text"/> </div> <div>Dictionary... ?</div> </div>						
<div> <div>Data type</div> <div>Time (hh:mm) ▼</div> <div>?</div> </div>						
<div> <div>* Width</div> <div> <input type="text"/> <input type="text" value="150"/> <input type="text"/> </div> <div>px</div> <div>0 - 10 000px</div> </div>						
<div> <div>* Maximum length</div> <div> <input type="text" value="50"/> </div> <div>characters</div> <div>1 - 250</div> </div>						
<div>Options</div> <div> <input type="checkbox"/> Required field <input checked="" type="checkbox"/> Read-only <input type="checkbox"/> Disable value assignment using JavaScript <input type="checkbox"/> Hidden field <input checked="" type="checkbox"/> Allow multiple assignments <input type="checkbox"/> Run OnBeforeSave/OnAfterSave before the record is saved to the database <input checked="" type="checkbox"/> Hide the sum of the column in the View table <input checked="" type="checkbox"/> Snap to previous control </div>						
Save		Save and continue editing		Delete		

- A 6-byte "string" represented as the "varchar (6)" database type.
- Time is a text value expressing the number of hours and the number of minutes. These two data are separated by a colon, with the possibility of writing a negative value using the "minus" sign at the beginning.
- The value of this field in the edit form is validated to the specified maximum length, to characters from the set 0 to 9, to a hyphen as a "minus" sign, and to a colon as a separator of hours and minutes.
- In the view table, the "time" column can be automatically added to the footer of the table.

Time (hh: mm: ss)

General	Default value	Events	Button	Help	Filter	Rights
* Name	<input type="text"/>					Dictionary... ?
Data type	Time (hh:mm:ss) ▼					?
* Width	<input type="text"/> 150 <input type="text"/> px					0 - 10 000px
* Maximum length	<input type="text"/> 50 characters					1 - 250
Options	<input type="checkbox"/> Required field <input checked="" type="checkbox"/> Read-only <input type="checkbox"/> Disable value assignment using JavaScript <input type="checkbox"/> Hidden field <input checked="" type="checkbox"/> Allow multiple assignments <input type="checkbox"/> Run OnBeforeSave/OnAfterSave before the record is saved to the database <input checked="" type="checkbox"/> Hide the sum of the column in the View table <input checked="" type="checkbox"/> Snap to previous control					
<div> <div>Save</div> <div>Save and continue editing</div> <div>Delete</div> </div>						

- A 13-byte "string" represented as the "varchar (13)" database type.
- Time is a text value expressing the number of hours, minutes, seconds, and optionally milliseconds. These three data are separated by a colon, with the possibility of writing a negative value using the "minus" sign at the beginning. Optional milliseconds are expressed by a one- to three-digit number written at the end after the dot.
- The value of this field in the edit form is validated to the specified maximum length, to characters from the set 0 to 9, to a hyphen as a "minus" sign, to a colon as a separator of hours, minutes, and seconds, and to a dot as a millisecond separator.
- In the view table, the "time" column can be automatically added to the footer of the table.

E-mail

General	Default value	Events	Button	Help	Filter	Rights
* Name	<input type="text"/>					Dictionary... ?
Data type	Email					?
* Width	<input type="text"/> 150 <input type="text"/> px					0 - 10 000px
* Maximum length	50 characters					1 - 250
Options	<input type="checkbox"/> Required field <input checked="" type="checkbox"/> Read-only <input type="checkbox"/> Disable value assignment using JavaScript <input type="checkbox"/> Hidden field <input checked="" type="checkbox"/> Allow multiple assignments <input type="checkbox"/> Run OnBeforeSave/OnAfterSave before the record is saved to the database <input checked="" type="checkbox"/> Snap to previous control					
<div> <input type="button" value="Save"/> <input type="button" value="Save and continue editing"/> <input type="button" value="Delete"/> </div>						

- "String" with a maximum length of 250 characters for the Firebird database and 4000 characters for the MSSQL database, represented as the database type "varchar" for the Firebird database and "nvarchar" for the MSSQL database.
- The value of this field in the edit form is validated to the specified maximum length, and using the regular expression "\w+([-+.]w+)*[-]?@w+([-.]w+)*\w+([-.]w+)*" so that it is not possible to enter a value that is not a valid e-mail address.
- Everywhere in the application except the text box itself in the edit form, the value of the control is displayed as a link to the e-mail address.

URL

General	Default value	Events	Button	Help	Filter	Rights
<div> <div>* Name</div> <div> <input type="text"/> <input type="text"/> </div> <div>Dictionary... ?</div> </div>						
<div> <div>Data type</div> <div> <div>URL</div> <div>▼</div> </div> <div>?</div> </div>						
<div> <div>* Width</div> <div> <input type="text"/> <input type="text" value="150"/> <input type="text"/> </div> <div>px</div> <div>0 - 10 000px</div> </div>						
<div> <div>* Maximum length</div> <div> <input type="text" value="50"/> </div> <div>characters</div> <div>1 - 250</div> </div>						
<div>Options</div> <div> <input type="checkbox"/> Required field <input checked="" type="checkbox"/> Read-only <div> <input type="checkbox"/> Disable value assignment using JavaScript </div> <input type="checkbox"/> Hidden field <input type="checkbox"/> Show shortened version of the URL <input checked="" type="checkbox"/> Allow multiple assignments <div> <input type="checkbox"/> Run OnBeforeSave/OnAfterSave before the record is saved to the database </div> <input checked="" type="checkbox"/> Snap to previous control </div>						
<div> <div>Save</div> <div>Save and continue editing</div> <div>Delete</div> </div>						

- “String” with a maximum length of 250 characters for the Firebird database and 4000 characters for the MSSQL database, represented as the database type “varchar” for the Firebird database and “nvarchar” for the MSSQL database.
- The value of this field in the edit form is validated only to the specified maximum length.
- Everywhere in the application except the text box itself in the edit form, the value of the control is displayed as a link to a web address.

Network link

General	Default value	Events	Button	Help	Filter	Rights
* Name	<input type="text"/>					Dictionary... ?
Data type	Network link					?
* Width	<input type="text"/> 150 <input type="text"/> px					0 - 10 000px
* Maximum length	<input type="text"/> 50 characters					1 - 250
Options	<input type="checkbox"/> Required field <input checked="" type="checkbox"/> Read-only <input type="checkbox"/> Disable value assignment using JavaScript <input type="checkbox"/> Hidden field <input checked="" type="checkbox"/> Allow multiple assignments <input type="checkbox"/> Run OnBeforeSave/OnAfterSave before the record is saved to the database <input checked="" type="checkbox"/> Snap to previous control					
<div> <input type="button" value="Save"/> <input type="button" value="Save and continue editing"/> <input type="button" value="Delete"/> </div>						

- "String" with a maximum length of 250 characters for the Firebird database and 4000 characters for the MSSQL database, represented as the database type "varchar" for the Firebird database and "nvarchar" for the MSSQL database.
- The value of this field in the edit form is validated only to the specified maximum length.
- Everywhere in the application, except for the text box itself in the edit form, the value of the control is displayed as a reference to the network location.

Telephone number

General	Default value	Events	Button	Help	Filter	Rights
<div> <div>* Name</div> <div> <input type="text"/> <input type="text"/> </div> <div>Dictionary... ?</div> </div>						
<div> <div>Data type</div> <div> <div>Telephone number</div> <div>▼</div> </div> <div>?</div> </div>						
<div> <div>* Width</div> <div> <input type="text"/> <input type="text" value="150"/> <input type="text"/> </div> <div>px</div> <div>0 - 10 000px</div> </div>						
<div> <div>* Maximum length</div> <div> <input type="text" value="50"/> </div> <div>characters</div> <div>1 - 250</div> </div>						
<div>Options</div> <div> <input type="checkbox"/> Required field <input checked="" type="checkbox"/> Read-only <input type="checkbox"/> Disable value assignment using JavaScript <input type="checkbox"/> Hidden field <input checked="" type="checkbox"/> Allow multiple assignments <input type="checkbox"/> Run OnBeforeSave/OnAfterSave before the record is saved to the database <input checked="" type="checkbox"/> Snap to previous control </div>						
<div> <div>Save</div> <div>Save and continue editing</div> <div>Delete</div> </div>						

- "String" with a maximum length of 250 characters for the Firebird database and 4000 characters for the MSSQL database, represented as the database type "varchar" for the Firebird database and "nvarchar" for the MSSQL database.
- The value of this field in the edit form is validated only to the specified maximum length.
- Everywhere in the application, except for the text box itself in the edit form, the value of the control is displayed as a reference to a telephone number, divided by spaces into groups of three digits.

IP address

General	Default value	Events	Button	Help	Filter	Rights
* Name	<input type="text"/>					Dictionary... ?
Data type	IP address					?
* Width	<input type="text"/> 150 <input type="text"/> px					0 - 10 000px
* Maximum length	50 characters					1 - 250
Options	<input type="checkbox"/> Required field <input checked="" type="checkbox"/> Read-only <input type="checkbox"/> Disable value assignment using JavaScript <input type="checkbox"/> Hidden field <input checked="" type="checkbox"/> Allow multiple assignments <input type="checkbox"/> Run OnBeforeSave/OnAfterSave before the record is saved to the database <input checked="" type="checkbox"/> Snap to previous control					
<div> <div>Save</div> <div>Save and continue editing</div> <div>Delete</div> </div>						

- "String" with a maximum length of 250 characters for the Firebird database and 4000 characters for the MSSQL database, represented as the database type "varchar" for the Firebird database and "nvarchar" for the MSSQL database.
- The value of this field in the edit form is automatically validated so that it is not possible to enter a value that is not a valid IP address.
- Everywhere in the application except the text box itself in the edit form, the value of the control is displayed as a reference to the RIPE database.

2.1.5 Unit

↶ For data types: Decimal number, Currency

- The unit specifies a symbol (such as "%") that appears everywhere in the application, along with the value of the control. The symbol is always placed to the right of the value itself, separated by a space.
- The unit may also include a decimal format that can be used to force automatic rounding of the value (for example, "{0.00}" or "{0.00}%").
- Along with the unit, it is possible to define the width of the space designated for the unit in the edit form so as to maintain a space between the unit located to the right of the text box and the help of the control located to the right of the unit.

2.1.6 Width

↩ For all data types

- The width of the control, expressed in pixels.
- The width consists of two parts, the width of the control itself in the edit form, and optionally the width of the column in the view table if the control is selected as one of the columns in the view table.
- When the width is set to 0 pixels, the text box does not display the typical border, and the text editing option is also missing – the value of the text box is displayed as a regular paragraph with plain text.

2.1.7 Maximum length

↩ For all data types

- The maximum length determines the maximum number of characters that can be entered in a text box in an edit form and then in the database.
- For the Firebird database, the maximum length is limited to 250 characters, for the MSSQL database to 4000 characters. The maximum length for the “Text string” data type affects the database type of the column that is associated with the current control in the database – “varchar (maxlength)” for the Firebird database and “nvarchar (maxlength)” for the MSSQL database. For other data types, the maximum length only affects the number of characters that can be entered in the text box in the edit form.

2.1.8 Rows

↩ For data type: Text string

- The number of lines determines the optional height of the text box in the edit form.
- The default number of rows is 1. For a text string, you can enter a value from 1 to 250, for other data types, the option to enter the number of lines is not displayed.

2.1.9 Alignment

↩ For data type: Whole number, Big whole number, Decimal, Currency

- Alignment determines the position of the text inside the text box itself in the edit form, and the position of the control values in the view table if the control is selected as one of the columns in the view table.
- The drop-down list contains the following types of alignment:
 - Left (default)
 - Centre
 - Right

2.1.10 Options

• Required field

↩ For all data types

- Checking the required field determines whether the edit form should be required to fill in a valid value before saving it to the database.
- The required field setting does not affect the database type of the column that is associated with the current control in the database. All database columns are set to accept the database value "null".
- A "string" can be stored in the database as a non-zero-length text string, a zero-length text string (empty string), or as a "null" database value. When compiling the conditions of a database query, it is therefore important to take into account the alternative that in addition to a zero-length text string, the database value "null" is also stored in a column in the database table. Such a condition may look like this in NG speech: `tb equals ""` or `tb is not defined`. And now in SQL: `ng_tb = "" OR ng_tb IS NULL`.
- Other data types can be stored either as a valid value of the given data type or as a database value "null".

• Read-only

↩ For all data types

- The read-only control setting determines whether the text box should be disabled in the edit form.
- It is not possible to enter values manually into the disabled field, the text field itself has a gray background, and the text of the value in this field is displayed in a different font than the rest of the controls.
- It is possible to fill in the values in the disabled field using javascript:
 - Text string
 - `control_SetValue(#ng_#, 'abc');`
 - Whole number or big whole number
 - `control_SetValue(#ng_#, 123);`
 - Decimal number or currency:
 - `control_SetValue(#ng_#, jsFN(1.23));`
 - Date:
 - `control_SetValue(#ng_#, '1.1.2000');`
 - `control_SetValue(#ng_#, '1.1.2000', '12:34');`
 - `control_SetValue(#ng_#, new Date());`
- Controls can generally be disabled using javascript: `control_Disable(#ng_#);`
- It is also possible to enable controls using javascript: `control_Enable(#ng_#);`
- By default, it is not possible to enforce a control that is set to read-only; the javascript function `control_Enable` must be passed the second optional parameter "ignorereadonly": `control_Enable(#ng_#, true);`

- **Disable value assignment using JavaScript**

↩ *For all data types in conjunction with the "Read-only" option*

- Checking the javascript ban on changing the value will ensure that read-only controls are not spoofed in the web browser by potential attackers on the edit form.
- Before saving the record to the database, it is always checked whether the value has been changed only on the server side, and if necessary, sets the value to the original valid one.

- **Hidden field**

↩ *For all data types*

- The hidden field setting determines whether the control should be part of an edit form, but should not be visible in user mode.
- The hidden control, including the filled value, can be found in the source code of the web page. It cannot therefore be used as part of security measures, the rights to the control are used for this.

- **Show Color Picker in the Edit form**

↩ *For data type: Text string*

- The color designer serves as a picker to compile a color value in the "R, G, B" format. In the edit form, the background color of the text field is set according to the currently entered value, and it is possible to select a color from the list of colors defined in the portal settings using the picker. If no color is defined in the portal settings, a list of frequently used colors common in web applications is displayed in the picker.
- The color is used in look-up tables, where it is possible to display a small colored rectangle for each row of the table, which optically facilitates orientation in the data. The display of the color rectangle display is done in the graphical database query designer (QueryBuilder), which defines the data source of the view table. In this designer, it is necessary to select one of the controls from the "Color by" drop-down list, which has "Show Color Picker in the Edit form" checked.
- The look-up table also allows you to color the text of each line according to the color value. The settings are made in the graphical database query designer (QueryBuilder), which defines the data source of the view table. In this designer, you need to select one of the controls in the "Color by" drop-down list, and then check "Color Text". Text coloring supports the following color formats:
 - "255,0,0" – red text
 - "B" – bold text without color
 - "B255,0,0" – bold red text
 - "N" – normal text without color
 - "N255,0,0" – normal red text
 - "255,0,0 @ 01/03/2019 @ B255,0,0" – red text, from 1.3.2019 bold red text
 - "N @ 01/03/2019 @ B" – normal text without color, from 1.3.2019 bold text without color

- **Show the “Sort order...” icon in the View table**

↩ *For data type: Whole number*

- Checking this box determines whether the meaning of the control is to be understood as the order in which the records in the view tables or tree structures are sorted. The value of this field is then changed in bulk directly via a look-up table or tree structure, from which it is possible to change the order of displayed records using the order designer, and “renumber” the value of this control in bulk.
- In order for the view table header to display an icon with a reference to the order designer (“A..Z”), the view table must have multiple data changes enabled. For a tree structure, an icon with a reference to the order designer is displayed if this column is set to “Sort by” in the graphical designer of the database query that defines the data source of the tree structure.

- **Show as a link in the View table to open the record on the new tab**

↩ *For data type: Text string, Whole number, Big whole number*

- Checking this box determines whether the control values should be displayed everywhere in the application as a link to open the record on a new tab.

- **Show time**

↩ *For data type: Date*

- Checking the Show time field determines whether the time should also be entered in the edit form along with the date. The time is displayed in a separate text box to the right of the date text box. The hourly and minute component is stored in the database, possibly even seconds, if the Including seconds field is checked.
- If the date, including the time, is stored in the database, it is displayed in the format “dd/mm/yyyy HH:mm” everywhere in the application, eventual “dd/mm/yyyy HH: mm:ss”. This is despite the fact that the date was stored in the database in a different way than via the edit form, ie. including seconds, milliseconds, and ticks (for example, using an external application).

- **Show the abbreviated name of the day**

↩ *For data type: Date*

- Checking this box determines whether the abbreviated name of the day should be displayed along with the value of the control everywhere in the application, in addition to the text box itself in the edit form.
- The abbreviated name of the day is placed at the beginning, and is separated from the date by a space, for example
“Fri March 1, 2019”.

- **Show week number**

↩ *For data type: Date*

- Checking this box determines whether the number of the week should be displayed along with the value of the control everywhere in the application, in addition to the text box itself in the edit form.
- The week number is placed in parentheses at the end, and is separated from the date by a space, for example
“01.03.2019 (9. week)”.

- **Show shortened version of the URL**

↩ *For data type: URL*

- Checking this box determines whether the value of the control should be displayed only in the form "domain/...", with three dots placed after the slash, everywhere in the application except the text field in the edit form. The URL value is always displayed as a link to a web address, so the entire form of the web address can be found by clicking on the link, or read directly in the edit form.

- **Allow multiple assignments**

↩ *For all data types*

- Enabling multiple assignments determines whether a text box with that value should be displayed in the view table or multi-source planning calendar instead of the control value itself, with the ability to change the value without having to open and edit the record in the edit form. This text box only appears in view tables or calendars that have multiple data changes enabled.
- Multiple assignments can only be made by a logged-in user; an anonymous user will see a read-only text box.

- **Run OnBeforeSave/OnAfterSave before the record is saved to the database**

↩ *For all data types in conjunction with the "Allow multiple assignments" option*

- Checking this box determines whether the script "OnBeforeSave" and then "OnAfterSave" should be run when saving the value of the control to the database directly from the view table, which
 - can prevent the value from being stored by triggering an interrupt,
 - or it can run additional database operations, send emails, etc.
- If the "OnBeforeSave" script is interrupted, the user receives the text of the error message, the value of the text field in the view table returns to its original state, and the value is not stored in the database.
- If the "OnAfterSave" script is interrupted, the user will receive an appropriate error message,

however, this no longer affects the successful storage of the value in the database.

- **Hide the sum of the column in the View table**

↩ *For data types: Whole number, Big whole number, Decimal, Currency, Time*

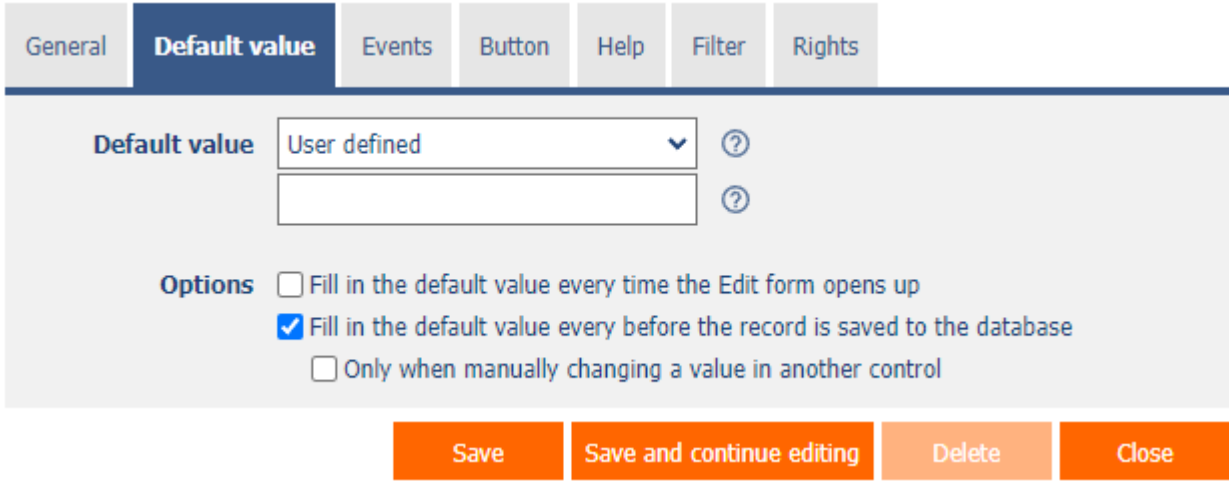
- Checking this box determines whether the sum of the control values from all filtered view table entries should be displayed in the footer of the view table.
- In order for column totals to be displayed in the view table, the view table must have column totals displayed. The default settings for displaying columns are taken directly from the properties of the view table, however, each user can change these settings themselves in their personalized settings of the view table.

- **Snap to previous control**

↩ For all data types

- Checking this box determines whether the control itself should be displayed to the right of the previous one in the edit form, or whether it will be displayed below the previous control in the usual way.
- If the control snaps to the previous one, you can define the offset width from the previous control
 - The offset is defined in pixels in a separate text box, which is located to the left of the “Width” text box.
 - Within the entire edit form, the controls are usually indented so that they are optically placed ideally in two columns or three. It is generally recommended to have as few vertical lines as possible in the form – to fit the individual controls – and thus to use a uniform width of controls that are optically placed one below the other in one column.
 - The same is true for indents, which should be used with the same value for all controls that are snapped, and optically form a separate column of the edit form.
 - When designing columns for an edit form, it is recommended practice to first consider the width of the controls in the first column, and then the overall width of the second column. The first column intentionally uses the width of the controls instead of the width of the entire column, because the controls in the first column start on the same vertical line throughout the edit form, and the control names to the left of the control itself automatically allocate the width needed for that control. the longest of them. The second column of the edit form, which consists of snapped controls, must have a fixed indent that displays the names of the snapped controls. It can happen that the name of the attached control is longer than the indentation itself, which violates the design of the edit form, and the control itself jumps more to the right. To the right of the offset, therefore, follows the attached control element.
 - If we remember the total width of the second column, which consists of the width of the indent and the width of the attached control, and we have already set the width of the control, it is possible to write a value in the format indent in the format “500-”, which ensures that the value the offset is calculated automatically according to the formula “Offset = Total column width – Control width”.

2.2 “Default value” tab



2.2.1 Default value

- The default value is used for newly created records in the edit form. This is a value that is automatically pre-populated into the control when an edit form is opened to create a new record that does not already exist in the database.
- The default value can be scheduled using the script function “SCHEDULEDEFAULTVALUE(ng_, abc)” from the script that is part of the button for opening a new edit form.
- The drop-down list contains the following types of default values:
 - **Used defined** – The default value is defined by a text constant, variable, or a combination of both, written in the text box below the drop-down list. Common variables include “#lastvalue#”, “#lastvalue2#”, etc.
 - **Current user** – login name of the currently logged in user
 - **Current date**
 - **Current time**
 - **Current date and time**
 - **Current value in the database** – the current value in the database is only relevant in connection with the check box “Fill in the default value before each saving of the record in the database”
 - The columns of some database records are also changed in the background of the operating system, for example by external applications. Typically, these are statuses, database record processing dates, etc. While a user has a database record open in the edit form, the value of such a column may be changed in the background by an external application. In this case, the value currently present in the edit form must not be used to save the record (this value was valid at the time the record was opened), but the value currently written in the database must be used.
 - **From query** – the default value defines the result of the database query, which is evaluated when opening the edit form
 - **JavaScript** – the default value is defined by a javascript function executed when opening the edit form, written in the text field below the drop-down list without a semicolon at the end. The Javascript function must return a text value.

- **Max** – the default value defines for text strings the value of the last created record by the currently logged in user, and for whole numbers the highest value present in the database in this column
- **Max + 1** – the default value defines the highest value present in the database in this column plus 1, generated when saving the record to avoid the creation of duplicate values
 - For new records, the text box remains hidden until max plus one is generated.
 - A typical application of this type of default value is to compile your own record ID, or the order of the record for sorting, often in conjunction with the “Show the Sort order... icon in the View table” function.
 - This type defaults
 - distinguishes nested records, which by default have their own number range for each parent record.
 - addresses the uniqueness of the value in all synchronized NET Genium installations. Until the record is synchronized to other synchronized NET Genium installations, the value is generated in a separate number series starting with 1000000.
 - allows you to create custom number series for database records that have identical values in one or two other columns. These columns can be selected by checking the “Categories by column” box.
 - allows you to set the initial value of a number series, eg 240100000. In this case, the first record will have the value 240100001, the second 240100002, etc.
- **Max + 1 (local)** – the default value defines the highest value present in the database in this column plus 1, generated when saving the record to avoid the creation of duplicate values
 - For new records, the text box remains hidden until max plus one is generated.
 - A typical application of this type of default value is to compile your own record ID, or the order of the record for sorting, often in conjunction with the “Show the Sort order... icon in the View table” function.
 - This type defaults
 - distinguishes nested records, which by default have their own number range for each parent record.
 - does not address the uniqueness of the value in all synchronized NET Genium installations.
 - allows you to create custom number series for database records that have identical values in one or two other columns. These columns can be selected by checking the “Categories by column” box.
 - allows you to set the initial value of a number series, eg 240100000. In this case, the first record will have the value 240100001, the second 240100002, etc.
- **Max + 1: uniform numerical series** – the default value defines the highest value present in the database in this column plus 1, generated when saving the record to avoid the creation of duplicate values
 - For new records, the text box remains hidden until max plus one is generated.
 - A typical application of this type of default value is to compile your own record ID, or the order of the record for sorting, often in conjunction with the “Show the Sort order... icon in the View table” function.
 - This type defaults
 - does not distinguish nested records, which by default have their own number range for each parent record.

- addresses the uniqueness of the value in all synchronized NET Genium installations. Until the record is synchronized to other synchronized NET Genium installations, the value is generated in a separate number series starting with 1000000.
 - allows you to create custom number series for database records that have identical values in one or two other columns. These columns can be selected by checking the “Categories by column” box.
 - allows you to set the initial value of a number series, eg 240100000. In this case, the first record will have the value 240100001, the second 240100002, etc.
- **Max + 1: uniform numerical series (local)** – the default value defines the highest value present in the database in this column plus 1, generated when saving the record to avoid the creation of duplicate values
 - For new records, the text box remains hidden until max plus one is generated.
 - A typical application of this type of default value is to compile your own record ID, or the order of the record for sorting, often in conjunction with the “Show the Sort order... icon in the View table” function.
 - This type defaults
 - does not distinguish nested records, which by default have their own number range for each parent record.
 - does not address the uniqueness of the value in all synchronized NET Genium installations.
 - allows you to create custom number series for database records that have identical values in one or two other columns. These columns can be selected by checking the “Categories by column” box.
 - allows you to set the initial value of a number series, eg 240100000. In this case, the first record will have the value 240100001, the second 240100002, etc.
 - **Max + 1: fixed** – the default value defines the highest value present in the database in this column plus 1, generated when opening the edit form
 - A typical application of this type of default is to sort the record order for sorting, often in conjunction with the “Show the Sort order... icon in the View table” function.
 - This type defaults
 - distinguishes nested records, which by default have their own number range for each parent record.
 - does not address the uniqueness of value. When multiple users work together, it allows records to be created with a duplicate value.

2.2.2 Options

- **Fill in the default value every time the Edit form opens up**

- Checking this box determines whether the current value in the edit form is overwritten to the default value when the existing record is opened.
- Controls that have an external function call set as the default, such as "ngef2(test)", and have the default check box selected each time the edit form is opened, may have an empty or blank value in the database, and still appear this value is evaluated everywhere in the application except for the edit form by an external function set as the default value.

- **Fill in the default value every before the record is saved to the database**

- Checking this box determines whether the current value in the edit form is overwritten to the default value before each save of the record.
- Controls that have an external function call set as the default value, such as "ngef2(test)", and have the default value checked before each record is saved to the database, may have an empty or blank value in the database, and still try to displaying this value everywhere in the application except the edit form evaluates the external function set as the default value.

- **Only when manually changing a value in another control**

↩ *In conjunction with the "Fill in the default value every before the record is saved to the database" option*

- Checking this box determines whether a value change in one of the other edit form controls will be evaluated before each record is saved. Only in this case will the current value in the edit form be overwritten to the default value.
- This option is the default for the Watcher controls, i.e. "Modified by" and "Modified on"

2.3 “Events” tab

General	Default value	Events	Button	Help	Filter	Rights
<div> <div>OnChange</div> <div> <input type="text"/> ? </div> <div> <input type="checkbox"/> Remove unnecessary spaces <input type="checkbox"/> Remove all spaces <input type="checkbox"/> First letter to upper case </div> </div> <div> <div>OnClick</div> <div> <input type="text"/> ? </div> </div> <div> <div>OnFocus</div> <div> <input type="text"/> ? </div> </div> <div> <div>OnBlur</div> <div> <input type="text"/> ? </div> </div>						
			Save	Save and continue editing	Delete	

2.3.1 OnChange

- The “onchange” event specifies a semicolon-separated list of javascript commands that are executed after a value change and then exiting the control.
 - **Remove unnecessary spaces**
 - Checking this box determines whether the “onchange” event should include the removal of unnecessary spaces, including
 - replacement of whitespace with spaces,
 - removing spaces at the beginning and end of a text string,
 - replacing double spaces with simple ones.
 - Unnecessary spaces will be removed before running the list of javascript commands defined in the “onchange” field.
 - **Remove all spaces**
 - Checking this box determines whether the “onchange” event should remove all spaces and whitespace from the text string.
 - All spaces will be removed before running the list of javascript commands defined in the “onchange” field.
 - **First letter to upper case**
 - Checking this box determines whether the “onchange” event should replace the first letter of the text string with an uppercase one.
 - The first letter is replaced with a capital letter before the list of javascript commands defined in the “onchange” field is run.

2.3.2 OnClick

- The “onclick” event specifies a semicolon-separated list of javascript commands that are executed when the mouse is clicked on the control.

2.3.3 OnFocus

- The “onfocus” event specifies a semicolon-separated list of javascript commands that are executed after entering the control (when the control receives focus).

2.3.4 OnBlur

- The “onblur” event specifies a semicolon-separated list of javascript commands that are executed when you exit the control with a tab or click outside the control.

2.4 “Button” tab

General	Default value	Events	Button	Help	Filter	Rights
<div> <div>Title</div> <input type="text"/> </div> <div> <div>Width</div> <input type="text"/> px </div> <div> <div>Events</div> <div> <div>Client</div> <div> <input type="text"/> ? </div> </div> <div> <div>Server</div> <div> Modify script ? </div> </div> </div> <div> <div>Save</div> <div>Save and continue editing</div> <div>Delete</div> </div>						

2.4.1 Title

- The title specifies the name of the button, which optionally appears to the right of the control itself in the edit form.
- An empty name means that the button will not be displayed.

2.4.2 Width

- Button width in pixels.

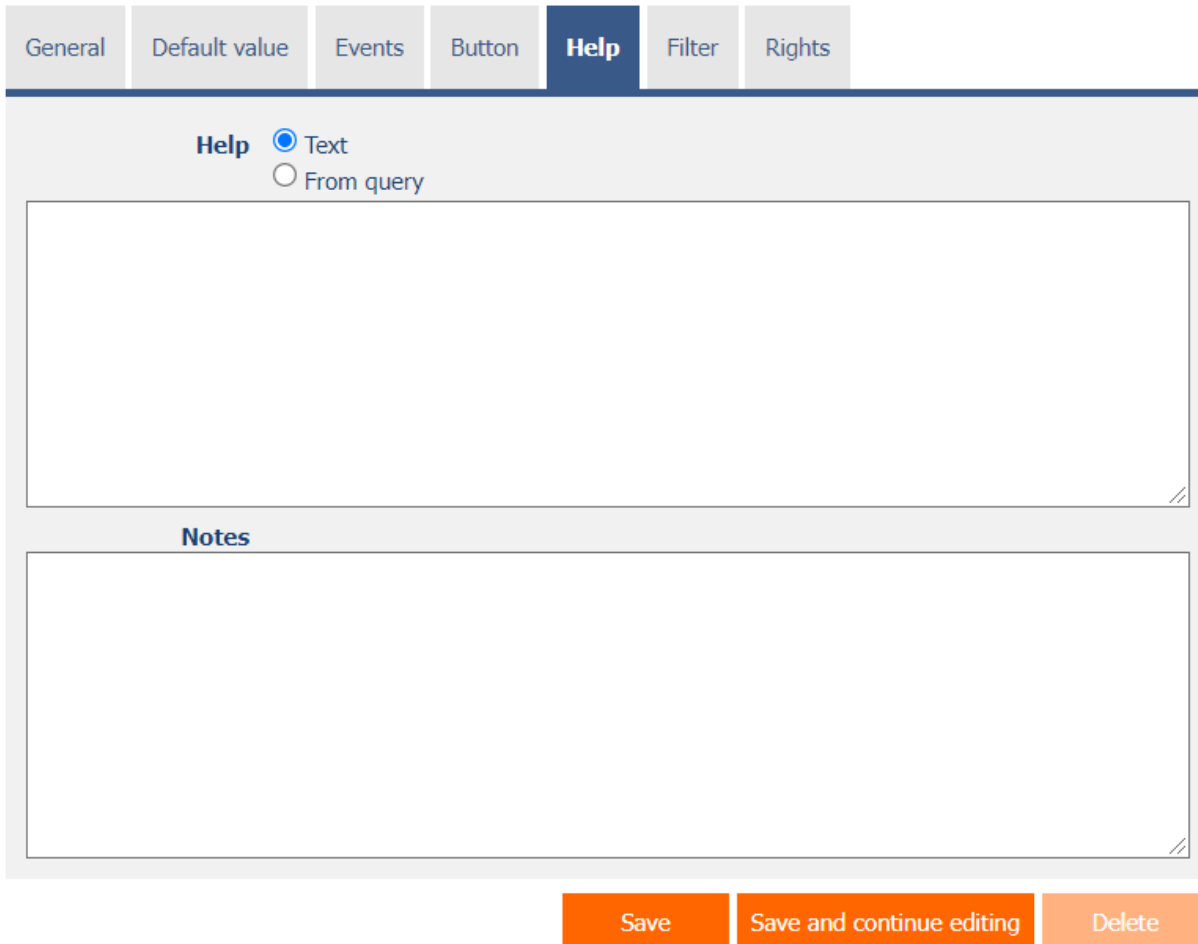
2.4.3 Client

- The “Client” field specifies a list of javascript commands that will be executed when the button is pressed. The drop-down list offers the following frequently used javascript functions:
 - **Are you sure you want to ...?** – return confirm ('Are you sure you want to ...?');
 - **AJAX...** – return bt_Eval (this, 'ngef.aspx?test', 'evalFunction');
 - **control_SetValue** – control_SetValue(#ng_#, "");

2.4.4 Server

- If the javascript commands are successfully evaluated, or if no javascript commands are defined, the server-side script will then be run. The “Edit script” link is used to open the integrated graphic script designer, in which the script is defined, resp. server part of the button.
- Execution of javascript commands can be interrupted by the command “return false;”, which prevents the subsequent execution of the script on the server side.

2.5 “Help” tab



2.5.1 Help

- Help is the text displayed to the right of the control itself in the edit form. The help of all controls starts on the same vertical line in the entire edit form – they are optically placed in a separate help column.
- The display of help text differs when snapping controls.
 - If “Automatically group help for snapped controls” is checked in the edit form settings, a bulleted list of all help controls located in the same row will be automatically generated in the help column, with the option to click the bulleted list item and display the help text.
 - If “Automatically group snap-in help help” is not checked in the edit form settings, the help for controls to the left of snapped controls is not displayed.
- Help text supports simple text formatting using wiki syntax:
 - `'''Bold text'''`
 - `''Italics''`
 - `''''Bold italics''''`
 - `''' Text enclosed in double quotes '''`

- Lists:
 - * Heading 1
 - ** Subheading 1.1
 - ** Subheading 1.2
 - * Heading 2
 - * Heading 3
 - # Numbered heading 1
 - ## Numbered subheading 1.1
 - ## Numbered subheading 1.2
 - # Numbered heading 2
 - # Numbered heading 3
- The text of the help can be displayed only after clicking on the icon with a question mark, which is located in a common place as help, only visually it takes up less space. Alternatively, abbreviated help text may appear in front of this icon. The question mark icon is defined by placing three dots at the end of the first line of help and then entering a new line (enter). The three dots may be preceded by an abbreviated help text. This is followed by help text of any length, including transitions to new lines.
 - ...
Help text displayed after clicking on the question mark icon
 - Short help text...
Help text displayed after clicking on the question mark icon
- In conjunction with the question mark icon, a variant can be used in which, after clicking on the icon, the help text followed by three dots is not displayed, but on the contrary, the content of, for example, the wonder located anywhere in the edit form is displayed/hidden. On the new line after the three dots, it is necessary to specify the ID of this wonder using the javascript function "el".
 - ...
el ('D1')
 - Short help text...
el ('D1')
- The help radio button contains the following two types of help definitions:
 - **Text** – help text defines a manually entered text string located in a multi-line text field below the radio button
 - **From query** – the help text defines the result of a database query, which is evaluated when opening the edit form

2.5.2 Comment

- Notes are used to enter any text intended for the application administrator.

2.6 “Filter” tab

General	Default value	Events	Button	Help	Filter	Rights
<div> <div>Options</div> <div> <input checked="" type="checkbox"/> Show filter in the View table <input type="checkbox"/> Allow multiple assignments </div> <div> <div>Records per page</div> <div>Default ▾</div> </div> </div>						
<div> <div>Save</div> <div>Save and continue editing</div> <div>Delete</div> </div>						

2.6.1 Options

- **Show the filter in the View table**
 - Checking this box determines whether the view table should be allowed to filter records by the values present in the database in the column associated with this control. The filter drop-down list is located in the header of the view table under the control name.
- **Allow multiple assignments**
 - Checking this box determines whether the “Assign value...” item should be available in the filter drop-down list, which is used to assign a value to all filtered records in the view table.
 - Selecting this item opens a new window displaying the control associated with the filtered column. Using the “Save” button, you can then change the column value for all filtered records in the view table to the selected value. In order for “Assign value...” to appear in the filter drop-down list, the view table must have multiple data changes enabled.
 - Multiple assignments cannot be applied to controls that are read-only or hidden.

2.6.2 The number of records per page

- The number of records per page determines the maximum number of unique values that appear in the filter based on the analysis of the read data in the view table in the column associated with the filter. If there are more occurrences of unique values in the data than the defined number of records on the page, the analysis of unique values in the given column is interrupted, and the item “Load values...” is displayed in the filter. all unique values in the given column and read data in the view table.
- The view table itself can read a large amount of data, or the filtered column can contain a large number of unique values, which affects the speed of compiling a list of these values. In cases where it is not important to analyze the values at all when generating the look-up table, and it is sufficient to use the function “Load values...”, it is advisable to set the number of records on the page to the value “0”.

2.7 “Rights” tab

General
Default value
Events
Button
Help
Filter
Rights

Rights

Available

External Users
Anonymous

Selected

Users

>>
>
<
<<

<<
<
>
>>

Show value
☒ to all users
☐ only to the owner or co-owner of the record

Replace rights to other controls
Show...

Save
Save and continue editing
Delete

2.7.1 Rights

- Rights define a list of user groups and users who have permission to display a control on an edit form and to display the values of that control elsewhere in the application.
- Users who do not have control rights
 - they do not see the control or its value in the edit form, nor can they find it in the source code of the page in a web browser,
 - they cannot display the column associated with this control in a view table or other visual control for viewing data.
 - they can create an entry in the edit form in which there is a control to which they do not have rights, the value of these controls is then set to the database value “null”,
 - they cannot delete any record in an edit form in which there is a control to which they do not have rights.

- An application administrator can cause a logical error when using a javascript function to reference a control that may not be accessible from a rights perspective for some users. The variable “ng_” is then replaced by the value stored in the database in this control instead of a reference to the object of the control itself. In most cases, this error stops loading the edit form, it disables itself, and the user is shown a message that he should contact the application administrator. This type of error can be partially eliminated using the report “Controls (JavaScript + insufficient rights)”.

2.7.2 Show value

- The display of the control value can be limited only to users who created the database record, or are set as co-owners of all records in the edit form settings.
- The radio button contains the following types of value display:
 - **To all users** – the control, including its value, will be displayed to all users who have the right to the control
 - **Only to the owner or co-owner of the record** – the control, including its value, will be displayed only to users who have the right to the control and at the same time are the owners of the currently open database record in the edit form (created the record) or are set as co-owners of all records in the edit form settings.

2.7.3 Replace rights with other controls

- The “Replace rights to other controls” function is used to unify the rights within the entire edit form for controls that are related in terms of rights.
- Thanks to the “Show...” link, it is possible to display a list of all controls in the edit form, including setting the rights of these controls. For each control, a check box is displayed that allows you to select the control. When saving a control, the same rights are then set for these selected controls.
- An alternative for unifying rights within the entire edit form is to display the report available from the “Subordinate object rights” tab in the edit form settings.

2.8 “Administration” tab

General	Default value	Events	Button	Help	Filter	Rights	Administration
<div> Identifier <input type="text" value="#ng_tb#"/> <input type="button" value="Correct"/> </div> <div> <input type="text" value="#ng_form#.#ng_tb#"/> </div>							
<div> Index <input checked="" type="checkbox"/> Index column "tb" along with the following columns... <input type="text"/> <input type="button" value="?"/> </div>							
<div> Search in the View table <input checked="" type="radio"/> LIKE (contains) <input type="radio"/> CONTAINS (MSSQL Full-Text) <input type="radio"/> = (equal to) <input type="radio"/> Deny <input type="checkbox"/> Search without diacritics </div>							
<div> Color by <input type="text" value=""/> </div>							
Other							
<div> Current value in the database <input type="text"/> <input type="button" value="Save"/> </div>							
<div> Change type to <input type="text" value="TextArea"/> <input type="button" value="Change"/> </div>							
<div> Change data type to <input type="text" value="Whole number"/> <input type="button" value="Change"/> </div>							

- The “Administration” tab is only displayed for existing controls.

2.8.1 Identifier

- The identifier specifies the name of the database column in which the values entered through the control in the edit form are stored in the database.
- The identifier is derived from the name of the control. It starts with the prefix “ng_”, and also contains the alphanumeric characters used in the control name (system controls do not include the prefix “ng_”). When changing the name of the control, the “Correct” button is available, which is used to correct the identifier, and thus to rename the database column in the database. For clarity and to eliminate possible misunderstandings, it is recommended to keep the identifier in accordance with the name of the control element. If external applications also access the database and, for example, read or write data to this column, it is necessary to schedule the change of the identifier until a suitable time.

2.8.2 Full identifier

- The full identifier of the control also contains the identifier of the edit form, and is used in printing templates.

2.8.3 Index

- Checking this box determines whether the column associated with this control should be indexed in the database. In the text field located below the check box it is possible to define the names (identifiers) of other columns that are to be in the associated index with this database column – they should be part of the index key.
- For the MSSQL database, it is also possible to define “INCLUDE” columns to be attached to the index. However, they are not part of the index key, but the values of that key. The Firebird database does not allow this functionality.

2.8.4 Search in the View table

- The search determines the way records in the view table will be searched in the database after entering a search term or phrase. The search method differs for text field data types.
- Text strings support the following types of searches:
 - **LIKE (contains)** – normal “case-insensitive” search using the “LIKE” clause
 - **CONTAINS (MSSQL Full-Text)** – full-text “case-insensitive” search using the “CONTAINS” clause, which is supported by MSSQL databases. Checking this option will automatically create a full-text index on the database.
 - **= (equal to)** – “case-insensitive” search using the “=” operator, which is very fast, especially in conjunction with the index on the database column. In this way, it is not possible to search for substrings in a database column, always only the whole value.
 - **Deny** – Searching in this column is disabled, and there is no check box in the view table to include the table column in the list of columns to be searched.
 - **Search without diacritics** – Checking this box will ensure that diacritics are first removed from the search term or phrase entered in the view table. For such a search to work, it is important to ensure that the values in the database in this column are also stored without diacritics, typically in the “OnBeforeSave” script using the “REMOVEDIACRITICS()” function.
- Numbers and dates support the following types of searches:
 - **Deny** – Searching in this column is disabled, and there is no check box in the view table to include the table column in the list of columns to be searched. This type is the default setting.
 - **Allow** – Searching in this column is enabled, and a check box appears in the view table, which allows you to include the table column in the list of searched columns.
- **Color by**
 - Everywhere in the application except the text box in the edit form itself, the value of the control can be displayed, including a colored rectangle or a picture icon.
 - The drop-down list selects from a list of “TextBox” controls with the color designer checked, or from “Image” controls located in the same edit form that will be used as a source of color or image information to visually highlight the control's value.

2.8.5 Other

- **Current value in the database**

- The text box displays the current value of the currently open record in the edit form in the given control, with the possibility to change this value.

- **Change type to**

- The drop-down list offers the types of controls to which conversion of the control is supported, including automatic conversion of the data present in the database table.

- **Change the data type to**


- The drop-down list offers data types to which control conversion is supported, including automatic conversion of data present in the database table.

2.8.6 Statistics

- Statistics display aggregated information about control usage across the entire application.
- Using the “Statistics” button, a detailed report of all places where the control is used or referenced is displayed, including the possibility to open the settings of such a place or control.
- Before each change of control type or data type, or before each deletion of a control from the edit form, it is recommended to thoroughly go through the control statistics, and eliminate any misunderstandings.
- Before each deletion of the control from the edit form, the evaluation of the statistics starts automatically. If the control is used or referenced somewhere in the application, it is necessary to confirm the deletion of the control again.

3 Picker

3.1 Basic information

General	Default value	Events	Button	Help	Filter	Rights	Administration
* Name	tb						Dictionary... ?
Data type	Text string						?
* Width	150		px				0 - 10 000px
* Maximum length	50		characters				1 - 250
Rows							1 - 250
Options	<input type="checkbox"/> Required field <input type="checkbox"/> Read-only <input type="checkbox"/> Hidden field <input type="checkbox"/> Show Color Picker in the Edit form <input type="checkbox"/> Show as link in the View table to open the record on the new tab <input type="checkbox"/> Allow multiple assignments <input type="checkbox"/> Snap to previous control						
						<div>  <div>SaveDelete</div> </div>	

- Picker is a dialog box that displays a look-up table with database records, the values of which can be taken over or copied to the currently open edit form. The picker is typically used to copy entire database records or to retrieve a ForeignKey value.
- The link to the pickup dialog box with the view table and database records is in the form of a magnifying glass icon, and appears in the edit form to the right of the control to which the picker is associated.
- The picker also takes the form of an ajax whisperer, which starts offering database records as soon as you enter a value in a text field in the edit form. For ajax whispering to work, you must enable searching for values entered in the edit form.
- Picker settings are made in a separate dialog. The link to this dialog is in the form of a magnifying glass icon, and is located in the lower left corner of the control settings dialog to which the picker is associated.

3.2.3 Records per page

- For clarity, the look-up table uses the paging of the displayed records. This is useful in situations where the view table is retrieving large amounts of data.
- Records per page determines the maximum number of records that appear on one page. You can easily switch between pages using the tools that each view table contains in its footer.

3.2.4 Options

- **Search for the value entered in the Edit form**
 - Checking this box determines whether the value entered in the Edit form should be found in the pickup view table and whether the edit form should offer an ajax picker.
 - The Ajax picker does not work if one of the fields selected is a "File" or "Image" control.
- **Optimized search**
 - Optimized search means that each column of the picker's view table uses its own text box to enter a search term or phrase.
 - A normal (non-optimized) search means that one common text field is used to enter a search term, in which it is possible to list all search terms, and these are then searched in all columns of the view table at once.
- **Show search results only**
 - Checking this box determines whether the picker's view table should display records only when a search term or phrase is entered.
- **Replace the value entered in the edit form**
 - Checking this field determines whether the transfer of the field value from the picker view table to the edit form should replace the value or add it to the end.
- **Allow creating new records**
 - Checking this box determines whether a button for entering a new record of the same agenda, which is used as a data source in the view table, will be displayed below the pickup view table.
- **Allow opening records**
 - Checking this box determines whether it should be possible to open individual records from the picker's view table and display the detail of a specific database record in the picker's dialog box directly in its edit form.
- **Allow multiple records to be selected at once**
 - Checking this box allows you to select multiple records in the pickup at once. Individual records are selected in the picker by checking the checkbox located on each row of the viewer's view table in the first column. In this way, however, it is possible to select only the column "(ID)", ie. primary key. The selected records are transferred to the edit form as individual primary key values separated by a semicolon.

3.2.5 Assignment table

- **Columns**

- The column settings determine which column of the view table in the picker will be copied, which control in the edit form.
- Picked columns – those that have a source and destination set – are automatically displayed in the picker's view table in the order in which they appear in the assignment table.
- The columns in the viewer's view table can be hidden by selecting the “Hide” command from the drop-down list located to the right of the “Destination” drop-down list.
- For columns in the picker's view table, it is possible to activate the search by selecting the “Activate search” command from the drop-down list located to the right of the “Destination” drop-down list.
- If no search column is selected, the search is automatically activated for all columns of the pickup view table, where this is allowed in terms of the general view table lookup settings defined in the control's properties.

3.3 “Help” tab

General

Help

Activate search

Other

Columns

Available

(ID)
tb
Latitude
Longitude
Created by
Created on
Modified by
Modified on

Selected

»

>

<

«

»

>

<

«

Save

Reset assignment table

Hide - all at once

- On the “Help” tab, it is possible to set a list of all other columns to be displayed in the picker's view table, even though none of these columns are peaked.

3.4 “Activate search” tab

General

Help

Activate search

Other

Columns

Available

(ID)
tb
Latitude
Longitude
Created by
Created on
Modified by
Modified on

Selected

»

>

<

«

»

>

<

«

Save

Reset assignment table

Hide - all at once

- On the “Activate search” tab it is possible to set a list of all other columns in which the search is to be activated.

3.5 “Other” tab

General	Help	Activate search	Other
---------	------	-----------------	--------------

Template name

Save

Reset assignment table

Hide - all at once

- **Template name**

- Picker settings can be saved to a template so that another control field that retrieves data from the same data source can be easily copied when setting up the picker.

4 Examples

4.1 Javascript functions

4.1.1 General functions

- `var s = control_GetValue(#ng_#);`
- `control_Disable(#ng_#);` // General javascript function for all database controls
- `tb_Disable(#ng_#);`
- `control_Enable(#ng_#);` // General javascript function for all database controls
- `tb_Enable(#ng_#);`
- `picker_Open(#ng_#);`

4.1.2 Text strings

- `var s = control_GetValue(#ng_#);`
- `var s = #ng_#.value;`
- `control_SetValue(#ng_#, s);`
- `#ng_#.value = s;`
- `control_SetValue(#ng_#, 'abc');`
- `#ng_#.value = 'abc';`

4.1.3 Whole number

- `var n = jsN(control_GetValue(#ng_#));`
- `var n = jsN(#ng_#.value);`
- `control_SetValue(#ng_#, n);`
- `#ng_#.value = n;`
- `control_SetValue(#ng_#, 123);`
- `#ng_#.value = 123;`
- `var ok = jsIsNumber(#ng_#.value);`
- `var s = jsNumberInWords(12345678, 'cs');`

4.1.4 Decimal numbers

- `var n = jsN(control_GetValue(#ng_#));`
- `var n = jsN(#ng_#.value);`
- `control_SetValue(#ng_#, jsFN(n));`
- `#ng_#.value = jsFN(n);`
- `control_SetValue(#ng_#, jsFN(1.23));`
- `#ng_#.value = jsFN(1.23);`
- `var ok = jsIsNumber(#ng_#.value);`
- `var s = jsNumberInWords(12345.678, 'cs');`
- `var s = jsFormatDouble(1100.5412, 'kg');`

4.1.5 Currencies

- `var n = jsN(control_GetValue(#ng_#));`

- `var n = jsN(#ng_#.value);`
- `control_SetValue(#ng_#, jsFN(n));`
- `#ng_#.value = jsFN(n);`
- `control_SetValue(#ng_#, jsFN(1.23));`
- `#ng_#.value = jsFN(1.23);`
- `var ok = jsIsNumber(#ng_#.value);`
- `var s = jsNumberInWords(12345.678, 'cs');`
- `var s = jsFormatCurrency(1100.5412, 'CZK');`

4.1.6 Dates

- `var date = Str2Date(control_GetValue(#ng_#));`
- `var date = Str2Date(#ng_#.value);` // Attention, it does not work for the date including the time – the time is located in a different text field than the date itself
- `control_SetValue(#ng_#, date);`
- `#ng_#.value = Date2Str(date);` // Attention, it does not work for the date including the time – the time is located in a different text field than the date itself
- `control_SetValue(#ng_#, '1.1.2000');`
- `control_SetValue(#ng_#, '1.1.2000', '12:34');`
- `control_SetValue(#ng_#, new Date());`

4.1.7 Times

- `var n = Time2Int(control_GetValue(#ng_#));`
- `var n = Time2Int(#ng_#.value);`
- `control_SetValue(#ng_#, Int2Time(n));`
- `#ng_#.value = Int2Time(n);`
- `control_SetValue(#ng_#, '12:23');`
- `#ng_#.value = '12:23';`
- `var s = jsFormatTime(new Date());`
- `InsertTime(#ng_#);`

4.1.8 E-mails

- `var s = control_GetValue(#ng_#);`
- `var s = #ng_#.value;`
- `control_SetValue(#ng_#, s);`
- `#ng_#.value = s;`
- `control_SetValue(#ng_#, 'info@netgenium.com');`
- `#ng_#.value = 'info@netgenium.com';`
- `var ok = jsValidateEmails('i@netgenium.com, n@netgenium.com; f@netgenium.com');`
- `var ok = jsValidateEmails('info@netgenium.com');`

4.2 Server functions

4.2.1 General functions

- `SCHEDULEDEFAULTVALUE(ng_, #ng_#)`
- `FORMATSTRING({0} – {1}, A, B)`

- REQUESTQUERYSTRING(abc)
- SUBSTRING(abc, 0, 1)

4.2.2 Whole numbers

- SUBTRACTINTEGERS(10, 1)

4.2.3 Decimal numbers

- COMPAREDOUBLE(1, <, 2, Yes, No)
- FORMATDOUBLE(1000)
- FORMATDOUBLE(1000, EUR)

4.2.4 Currencies

- FORMATCURRENCY(1000)
- FORMATCURRENCY(1000, EUR)

4.2.5 Dates

- ADDTIMESPANINWORDS(#today#, 1 month)
- ADDTIMESPANINWORDS(#today#, 1 month, 2 days)
- COMPAREDATE(#today#, <, #now#, Yes, No)
- FORMATDATE(#today#)
- FORMATDATE(#today#, dd.MM.yyyy)
- FORMATDATES(abc #today# def, true, true)
- SUBTRACTDATES(#today# + 14, #today#)

4.2.6 Times

- FORMATTIME(1000)
- INT2TIME(480)
- TIME2INT(08:00)

4.2.7 E-mails

- EMAIL(#loginname#)
- EMAIL(#userid#)
- EMAILS(1, 2, 3)
- EMAILS(loginname1, loginname2, loginname3)
- VALIDEMAIL(info@netgenium.com)