

MultiListBox

Control in the edit form

Content

1	Basic information	4
1.1	Description of the control	4
1.2	Create a new control	4
1.3	Edit or delete a control	4
2	List of tabs in the control settings dialog	6
2.1	“General” tab	7
2.1.1	Name	7
2.1.2	Abbreviated name	7
2.1.3	Dictionary...	8
2.1.4	Separator	8
2.1.5	Width	8
2.1.6	Rows	8
2.1.7	Options	9
2.2	“Values” tab	12
2.2.1	Values	12
2.2.2	Default value	13
2.2.3	Options	13
2.3	“Events” tab	14
2.3.1	OnChange	14
2.4	“Help” tab	15
2.4.1	Help	15
2.4.2	Notes	16
2.5	“Filter” tab	17
2.5.1	Options	17
2.6	“Rights” tab	18
2.6.1	Rights	18
2.6.2	Show value	19
2.6.3	Replace rights to other controls	19
2.7	“Administration” tab	20
2.7.1	Identifier	20
2.7.2	Full identifier	20

2.7.3	Search in the View table	21
2.7.4	Other.....	21
2.7.5	Statistics.....	21
3	Picker.....	22
3.1	Basic information	22
3.2	“General” tab	23
3.2.1	Tooltip	23
3.2.2	Data source	23
3.2.3	Records per page.....	24
3.2.4	Options.....	24
3.2.5	Assignment table.....	25
3.3	“Help” tab	26
3.4	“Activate search” tab.....	27
3.5	“Other” tab	28
4	Examples	29
4.1	Javascript functions.....	29
4.2	Server functions	29

1 Basic information

1.1 Description of the control

MultiListBox is a multi-line list of any width and height with the option to select multiple values from a list of available values. The selected values are stored in the database separated by a tab in the form of a text string of unlimited length. Multilistbox uses a single database column type that is associated with the current control in the database – “text” for the Firebird database and “nvarchar(max)” for the MSSQL database.

1.2 Create a new control

There are two basic ways to create a new multilistbox:

From the Edit form

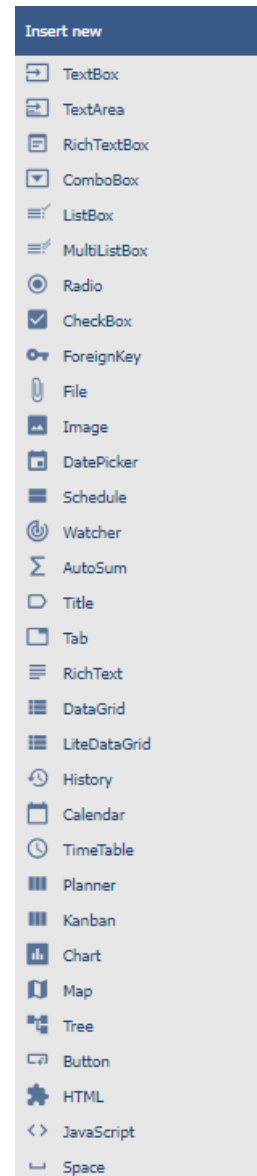
- The most common way is to create a new multilistbox directly from the edit form using the toolbar with controls.
- First, the position in which the new multilistbox is to be placed is determined by clicking the mouse in the edit form. This will highlight the selected position with a gray horizontal line. Then just click on the “MultiListBox” 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 multilistbox at a pre-selected location on the edit form.
- If the desired position is not determined before inserting a new multilistbox, the new multilistbox 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 subsequently displayed list of controls allows you to select a new multilistbox, which will be placed in the edit form immediately after the control, which is displayed last in the view table. After filling in the attributes of the new control and saving it, a new multilistbox 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 multilistbox 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 behind its name located in the left column of the edit form, and then dragging it to the desired position in the edit form.
- For security reasons, it is important to have the function of moving controls enabled using the “Drag&Drop” check box located above the toolbar with controls. This field is unchecked each time you log in to the application.
- Deleting a multilistbox results in the irreversible deletion of the database column associated with this multilistbox in the database, and thus in the deletion of its values in all records of the database table.



2 List of tabs in the control settings dialog

General	Setting general properties
Values	Setting values for newly created records
Events	Javascript event settings
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	Values	Events	Help	Filter	Rights
* Name	<input type="text"/>	<input type="text"/>	<i>Dictionary... ?</i>		
Separator	<input type="text"/>	<input type="text" value="
"/>			
* Width	<input type="text"/>	<input type="text" value="350"/>	<input type="text"/>	px <i>150 - 10 000px</i>	
* Rows	<input type="text" value="8"/>	<i>1 - 250</i>			
Options	<input checked="" 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 checked="" type="checkbox"/> Run OnBeforeSave/OnAfterSave before the record is saved to the database <input checked="" type="checkbox"/> Snap to previous control				
Save		Save and continue editing		Delete	
Close					

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 multiline list 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 lists the names of other controls, forms, view pages, categories, applications, and application groups used across the entire 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 Separator

- Everywhere in the application except the multiline list itself in the edit form, the value of the control is displayed as a comma-separated list of selected values. This delimiter can optionally be changed to any text string other than a comma.
- The value separator displayed in the details of the view table is set to "
", Resp. jump to a new line, and it is not possible to change it.
- The delimiter of values stored in the database is set to tab, and cannot be changed.
 - Multilistbox is the only database control that uses this method of separating values, and stores in the database a tab-separated list of selected values in the form of a single text string of unlimited length. It is important to consider this fact when compiling database query conditions. NET Genium allows the database query designer to use the syntax of "multilistbox equals value" conditions, automatically applying the user-defined database function "JoinText2(ng_mlb, 'value') 0" to the Firebird database, and "dbo.JoinNtext(ng_mlb, 'value') 0" for an MSSQL database, instead of the regular "ng_mlb = 'value' "operator. Function "JoinText2" resp. "Dbo.JoinNtext" splits the value of the first parameter into tab-separated values, and cross-compares them with the value of the second parameter, which also splits it into tab-separated values. The result of this function is the number of elements that match in the first and second parameters, and thus the function can be used as a search for a specific value in the multilistbox.u.
 - Thanks to the automatic cross-search for multilistboxes, it is not possible to use a condition in the form "ng_mlb = 'value'" in a database query, however, due to the nature of the "MultiListBox" control, this is not even necessary.
 - Cross-searching of values can also be used in other cases in the application, not only in multilistboxes, where the use of the functions "JoinText2" resp. "Dbo.JoinNtext" works completely automatically – using a syntax in the form of "column with a list of values separated by a tab equal to (mlb) value", where the key position of the substring "(mlb)" is in the right part of the condition before the searched value.
 - Cross-searching for values is not suitable for traversing a large number of database records.

2.1.5 Width

- 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.

2.1.6 Rows

- The number of rows determines the height of the multiline list in the edit form.
- The default number of rows is 8. The minimum number of rows is 1, the maximum is 250.

2.1.7 Options

- **Required field**

- 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".
- The value 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: `mlb equals ""` or `mlb is not defined`. And now in SQL: `ng_mlb = " OR ng_mlb IS NULL`.

- **Read-only**

- The read-only control setting determines whether the multilistbox should be disabled in the edit form.
- The disabled multilistbox cannot be checked manually.
- The disabled multilistbox can be checked using javascript:
 - `control_SetValue(#ng_#, 'x');`
- Controls can generally be disabled using javascript:
 - `control_Disable(#ng_#);` // General javascript function for all database controls
 - `mlb_Disable(#ng_#);`
- It is also possible to enable controls using javascript:
 - `control_Enable(#ng_#);` // General javascript function for all database controls
 - `mlb_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 assignments using JavaScript**

↩ *Only in conjunction with the "Read Only" option*

- Checking the value change ban using javascript will ensure that read-only controls are not spoofed by the value 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**

- 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 selected 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.

- **Allow multiple assignments**

- Allowing multiple assignments determines whether a list of checkboxes should be displayed in the view table or in the multi-source planning calendar with the option to change the value without having to open and edit the record in the edit form.
- Multiple assignments can only be made by a logged in user, an anonymous user is shown a read-only list of checkboxes.

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

↩ *Only in conjunction with the "Allow multiple assignments" option*

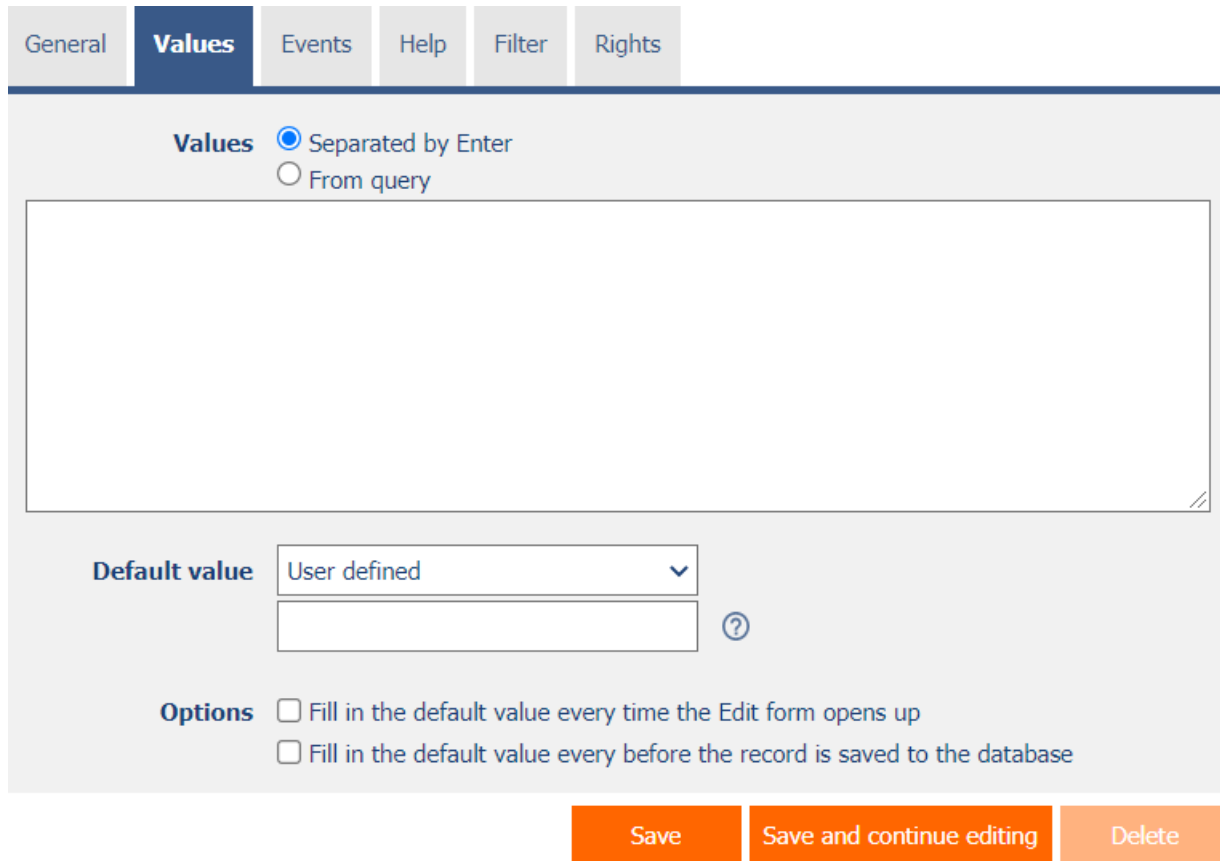
- Checking this box determines whether the script "OnBeforeSave" and then "OnAfterSave" should be run when saving the value of the control to the database directly from the view table, which
 - can prevent the value from being stored by triggering an interrupt,
 - or it can run additional database operations, send emails, etc.
- If the "OnBeforeSave" script is interrupted, the user will see the text of the error message, the checkmark of the button in the view table will return to its original state, and the value will not be saved to the database.
If the "OnAfterSave" script interrupts, the user will receive an error message, but this will no longer affect the successful storage of the value in the database.

- **Snap to previous control**

- 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 indent 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 automatically calculates according to the formula "Offset = Total column width - Control width".

2.2 “Values” tab



The screenshot shows the 'Values' tab of a configuration interface. At the top, there are tabs for 'General', 'Values', 'Events', 'Help', 'Filter', and 'Rights'. The 'Values' tab is active. Below the tabs, there are two radio button options: 'Separated by Enter' (selected) and 'From query'. A large empty text area is provided for entering values. Below this, there is a 'Default value' section with a dropdown menu set to 'User defined' and an empty text input field with a help icon. At the bottom, there are two checkboxes under 'Options': 'Fill in the default value every time the Edit form opens up' and 'Fill in the default value every before the record is saved to the database'. At the very bottom, there are three buttons: 'Save', 'Save and continue editing', and 'Delete'.

2.2.1 Values

- **Separated by Enter**
 - The list of values of the multiline list is set manually – in the text field using values separated by enter.
- **From query**
 - The list of multiline list values defines a database query that is evaluated when opening or updating an edit form.

2.2.2 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 a 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:
 - **Own** – 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, with the option to use a semicolon as a value separator. Common variables include "#lastvalue#", "#lastvalue2#", etc.
 - **Current user** – login name of the currently logged in user
 - **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 an open database record 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

2.2.3 Options

- **Fill in the default value every 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 value, 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 fail when attempted to display 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 check box selected each time a 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.

2.3 “Events” tab

General	Values	Events	Help	Filter	Rights
---------	--------	---------------	------	--------	--------

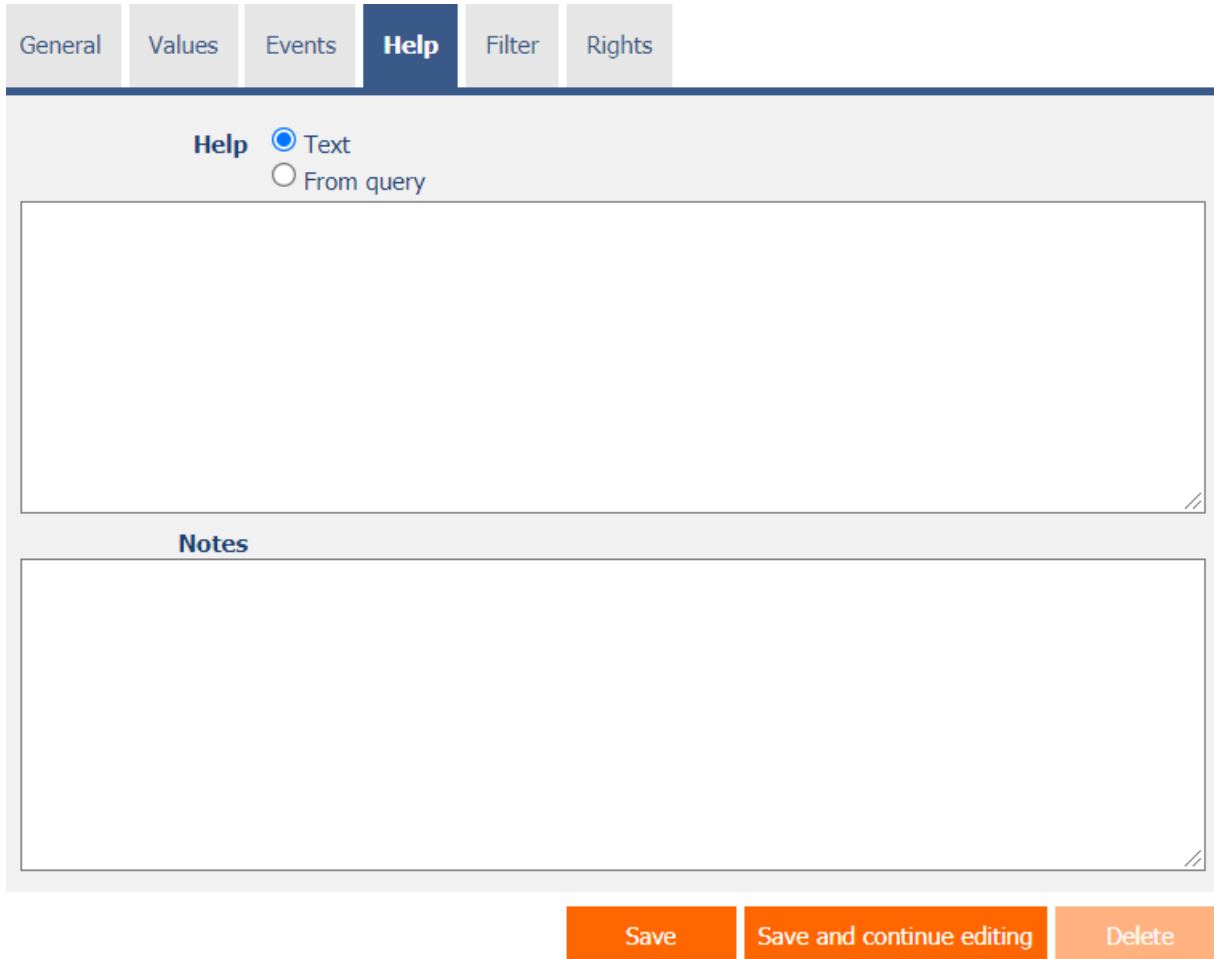
OnChange ?

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.

2.4 “Help” tab



General Values Events **Help** Filter Rights

Help Text
 From query

Notes

Save Save and continue editing Delete

2.4.1 Help

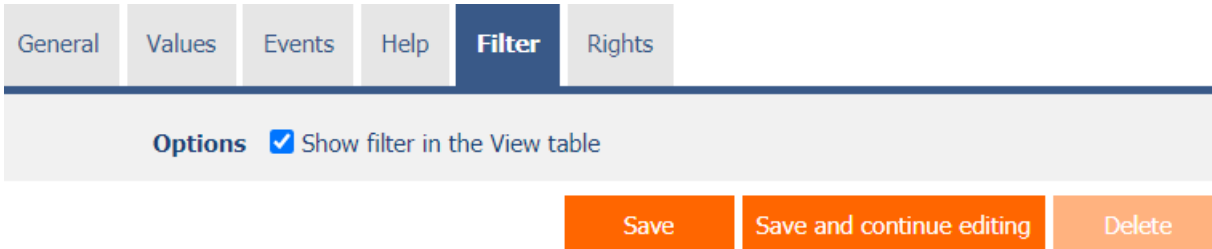
- Help is the text displayed to the right of the control itself in the edit form. The help of all controls starts on the same vertical line in the entire edit form – they are optically placed in a separate help column.
- The display of help text differs when snapping controls.
 - If “Automatic grouping” 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 “Automatic grouping” 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

2.4.2 Notes

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

2.5 “Filter” tab



The screenshot shows a configuration interface for the 'Filter' tab. At the top, there are five tabs: 'General', 'Values', 'Events', 'Help', and 'Filter' (which is selected and highlighted in dark blue), and 'Rights'. Below the tabs is a light gray panel with the text 'Options' followed by a checked checkbox and the text 'Show filter in the View table'. At the bottom right of the panel are three orange buttons: 'Save', 'Save and continue editing', and 'Delete'.

2.5.1 Options

- **Show 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 name of the control.
 - The filter drop-down list does not display the specific values present in the database in the column associated with this control, only the options:
 - Search for the entered value
 - Find empty values in the database
 - Finding non-empty values in the database
 - Search by compound condition

2.6 “Rights” tab

General
Values
Events
Help
Filter
Rights

Rights Available

External Users

Anonymous

Selected

Users

»
>
<
«

»
^
v
«

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

2.6.1 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 intended 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 a javascript function refers to 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)”.

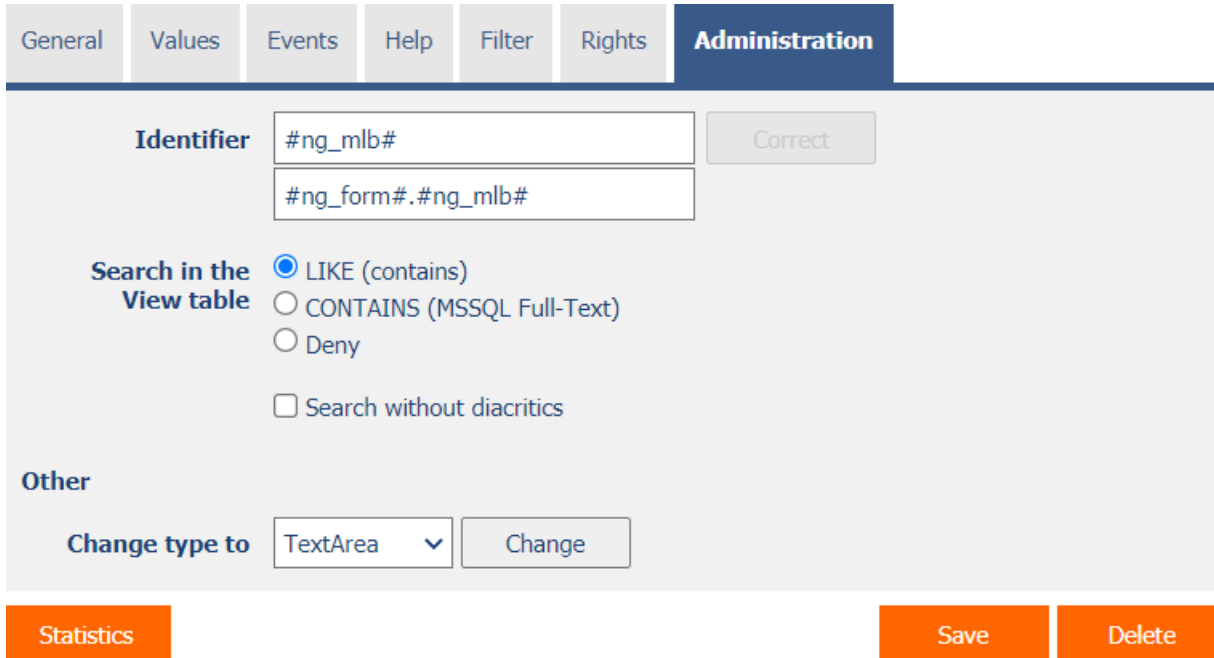
2.6.2 Show value

- The display of the control value can be limited only to the 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.

2.6.3 Replace rights to 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 you save a control, the same rights are then set for those 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.

2.7 “Administration” tab



General Values Events Help Filter Rights **Administration**

Identifier

Search in the View table LIKE (contains)
 CONTAINS (MSSQL Full-Text)
 Deny

Search without diacritics

Other

Change type to

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

2.7.1 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.

2.7.2 Full identifier

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

2.7.3 Search in the View table

- The search determines the way the records in the view table will be searched in the database after entering the search term or phrase.
- Multilistbox supports 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.
 - **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.

2.7.4 Other

- **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.

2.7.5 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	Values	Events	Help	Filter	Rights	Administration
* Name	<input type="text" value="mlb"/>	<input type="text"/>	<i>Dictionary... ?</i>			
Separator	<input type="text"/>	<input type="text" value="
"/>				
* Width	<input type="text" value="350"/>	<input type="text"/>	px	<i>150 - 10 000px</i>		
* Rows	<input type="text" value="8"/>	<i>1 - 250</i>				
Options	<input checked="" type="checkbox"/> Required field <input type="checkbox"/> Hidden field <input type="checkbox"/> Snap to previous control					
						<input type="button" value="Save"/> <input type="button" value="Delete"/>

- Picker is a dialog box that displays a view table with database records, the values of which can be taken or copied to the currently open edit form. The picker is typically used to copy entire database records or to take 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 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 "General" tab

General
Help
Activate search
Other

Tooltip Dictionary...

Data source **Modify query**

Records per page ▼

Options

- Optimized search
- Show search results only
- Allow creating new records
- Allow opening records
- Allow multiple records to be selected at once

Assignment table

	Source	Destination	Options
Columns	mlb	mlb ▼	▼
	Latitude	(None) ▼	▼
	Longitude	(None) ▼	▼
	Created by	(None) ▼	▼
	Created on	(None) ▼	▼
	Modified by	(None) ▼	▼
	Modified on	(None) ▼	▼

Save
Reset assignment table
Hide - all at once

3.2.1 Tooltip

- The tooltip determines the help text that appears when you hover over the picker icon in the edit form.

3.2.2 Data source

- The data source defines the data displayed in the viewer's view table. This data is the result of a database query that is evaluated when the picker window is opened.
- Data source settings are made in the graphical designer of database queries.

3.2.3 Records per page

- For clarity, the view table uses the paging of the displayed records. This is useful in situations where the view table is retrieving large amounts of data.
- The number of 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

- **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.
- **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 picker's view table in the first column. In this way, however, it is possible to select only the column "(ID)", i.e. 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 to the 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 view settings defined in the control's properties.

3.3 "Help" tab

General

Help

Activate search

Other

Columns

Available	Selected
(ID)	
mlb	
Latitude	
Longitude	
Created by	
Created on	
Modified by	
Modified on	

»»
»
<
««

««
«
^
∨
∨∨

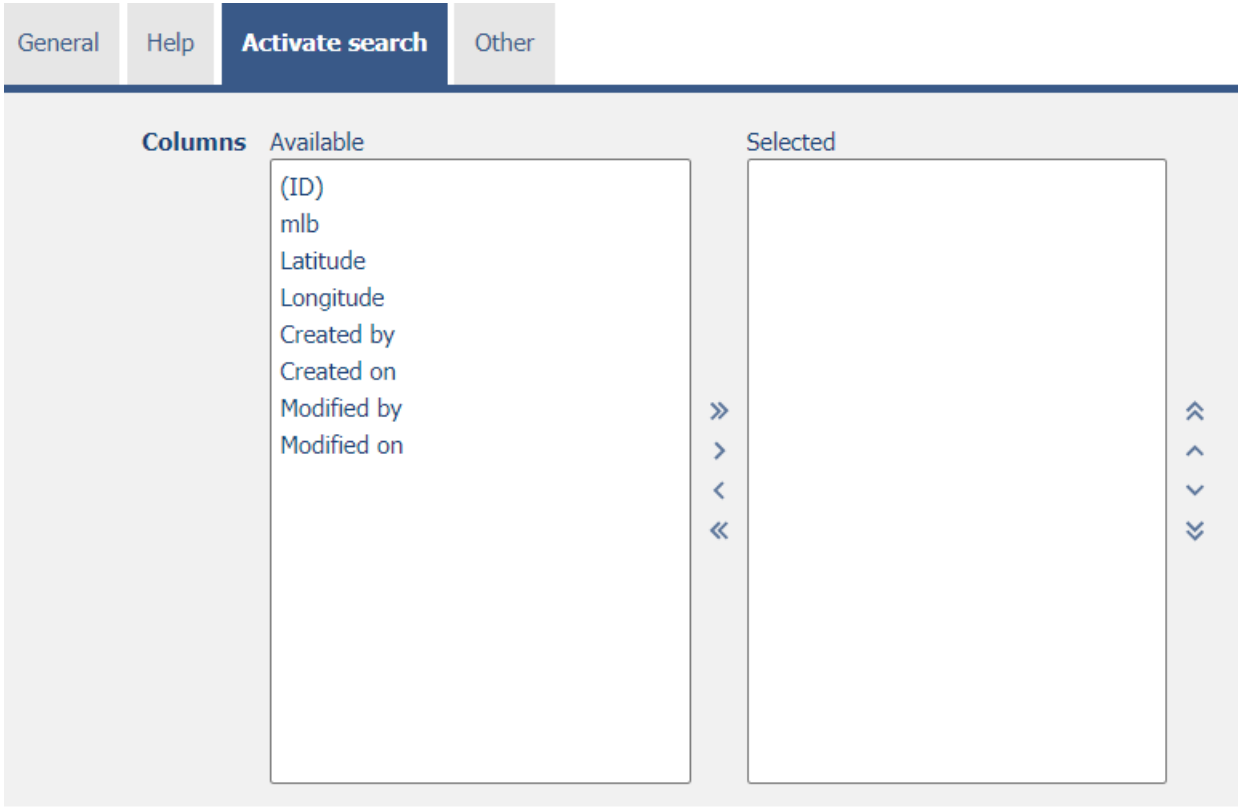
Save

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



Columns Available Selected

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

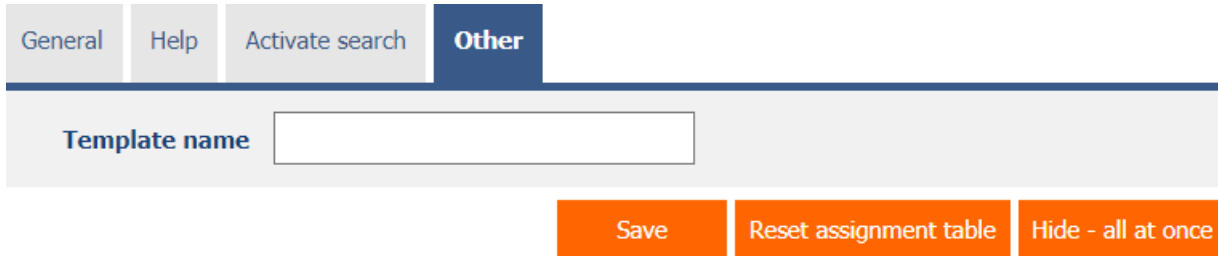
»
>
<
«

⏪
⏩
⏴
⏵

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



The screenshot shows a configuration interface with four tabs: 'General', 'Help', 'Activate search', and 'Other'. The 'Other' tab is selected and highlighted in dark blue. Below the tabs is a light gray panel containing a text input field labeled 'Template name'. To the right of this panel are three orange buttons: 'Save', 'Reset assignment table', and '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

- `var s = control_GetValue(#ng_#);`
- `var s = #ng_#.value;`
- `control_SetValue(#ng_#, s);`
- `control_SetValue(#ng_#, 'a\tb\tc');`
- `control_Disable(#ng_#);` // General javascript function for all database controls
- `mlb_Disable(#ng_#);`
- `control_Enable(#ng_#);` // General javascript function for all database controls
- `mlb_Enable(#ng_#);`
- `picker_Open(#ng_#);`

4.2 Server functions

- `SCHEDULEDEFAULTVALUE(ng_#,ng_#)`
- `FORMATSTRING({0} - {1}, A, B)`
- `REQUESTQUERYSTRING(abc)`
- `SUBSTRING(abc, 0, 1)`