netgenium

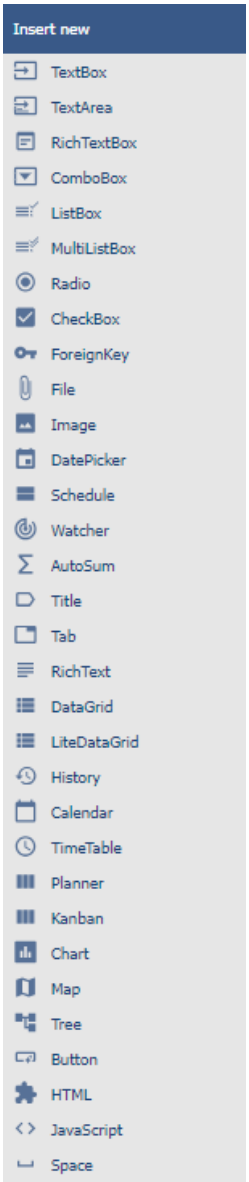
TextArea

Control in the edit form

Content

1 Basic information

1.1 Description of the control



TextArea is a text field of any width and height intended for entering plain text with unlimited length with 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 textarea:

From the Edit form

- The most common way is to create a new textarea directly from the edit form using the toolbar with controls.
- First, the position in which the new textarea 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 "TextArea" 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 textarea at the pre-selected location of the edit form.
- If the desired position is not determined before inserting a new textarea, the new textarea 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 textarea, 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 textarea 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 textarea just created.

1.3 Edit or delete a control

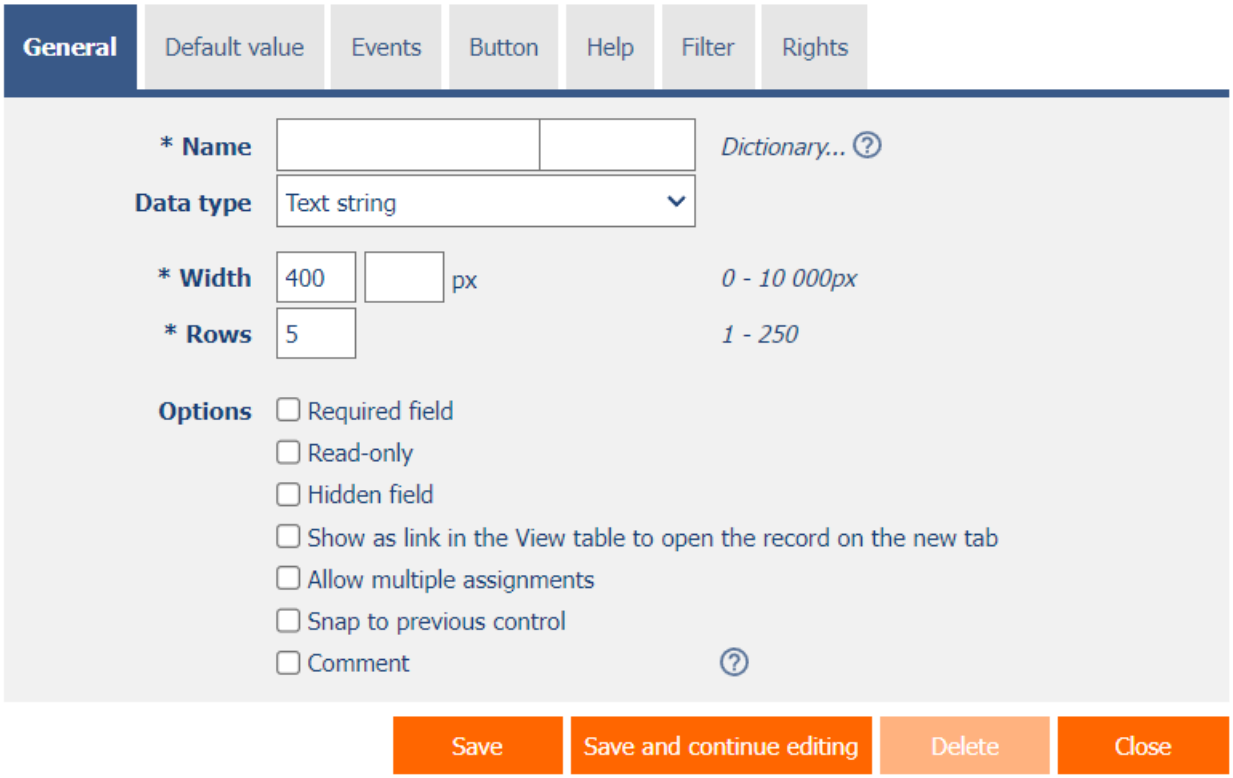
- For each existing control, a pencil icon appears on the left side of the edit form to change or delete the control's settings.
- Individual controls can be moved vertically in the edit form using the Drag&Drop method. Moving consists of grabbing the control, preferably behind its name located in the 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 textarea results in the irreversible deletion of the database column associated with that textarea in the database, and thus in the deletion of its values in all records in the database table. Some textarea 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	Set the default value for newly created records
Events	Javascript event settings
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



2.1.1 Name

For all data types

- 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

For all data types

- 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…

For all data types

- 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.

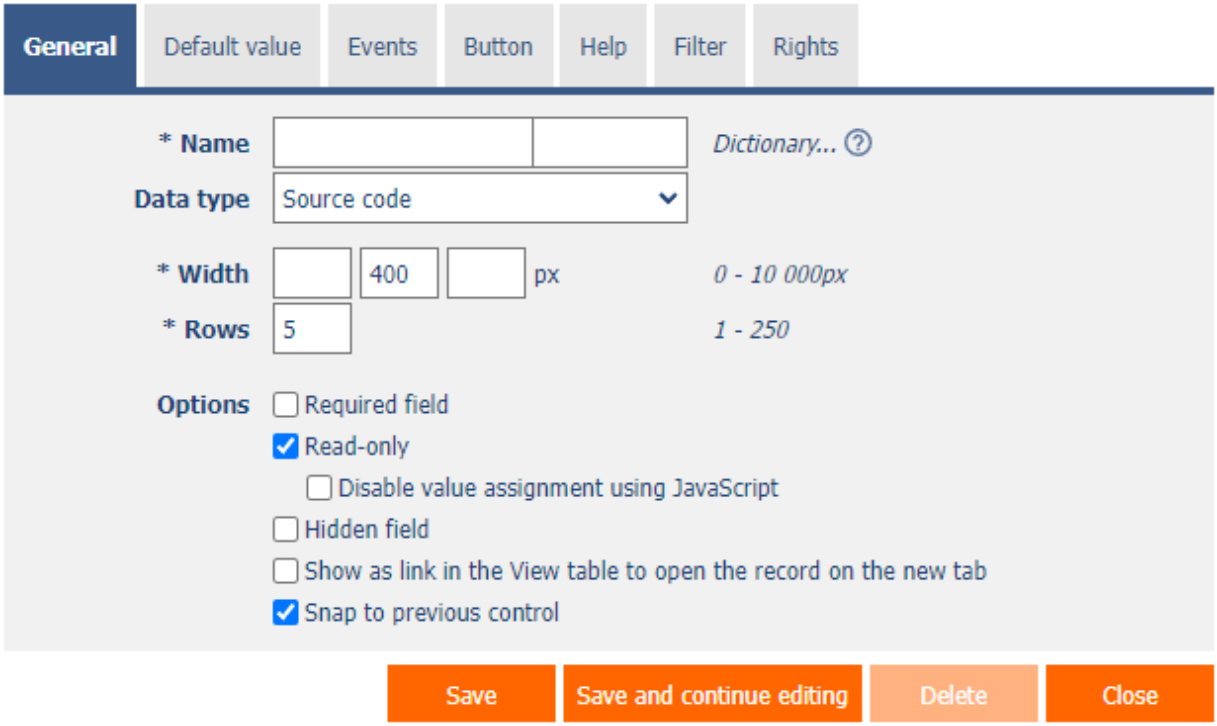
2.1.4 Data type

- The data type defines the type or meaning of the values that the control can take.
- Textarea 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.
- The drop-down list contains the following data types

Text string

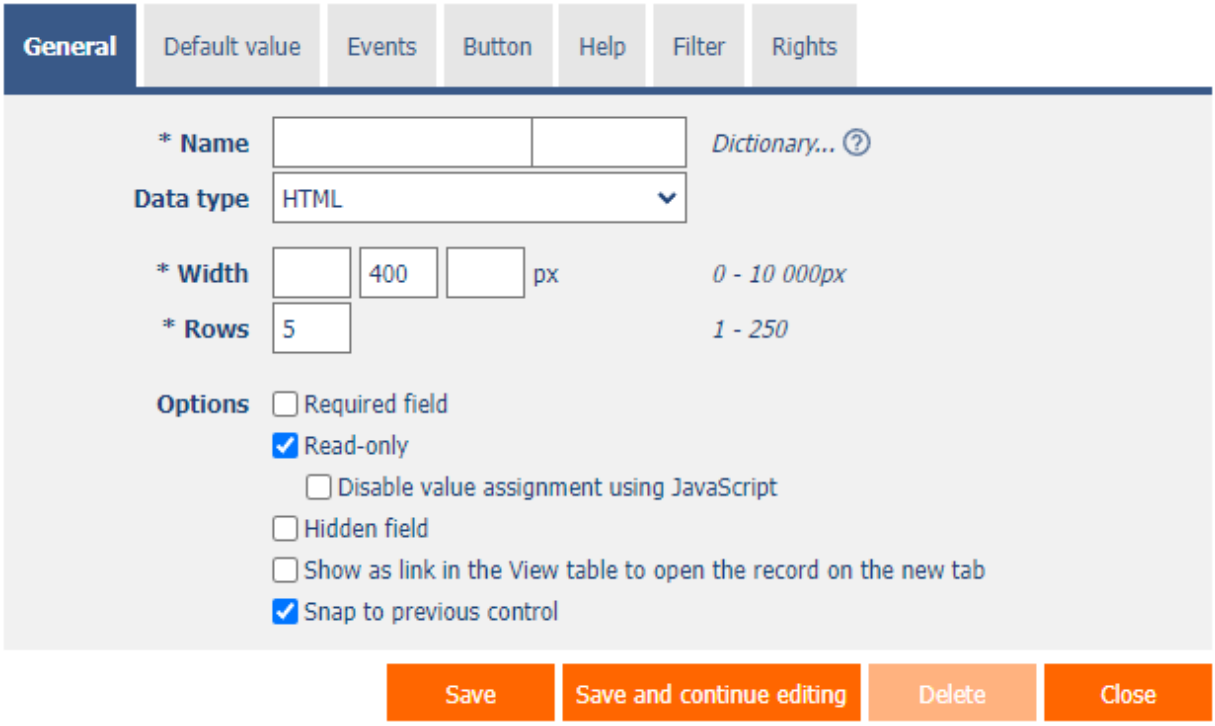
- Classic multiline text box with framing.
- “String” of unlimited length, represented as the database type “text” for the Firebird database and “nvarchar(max)” for the MSSQL database.

Source code



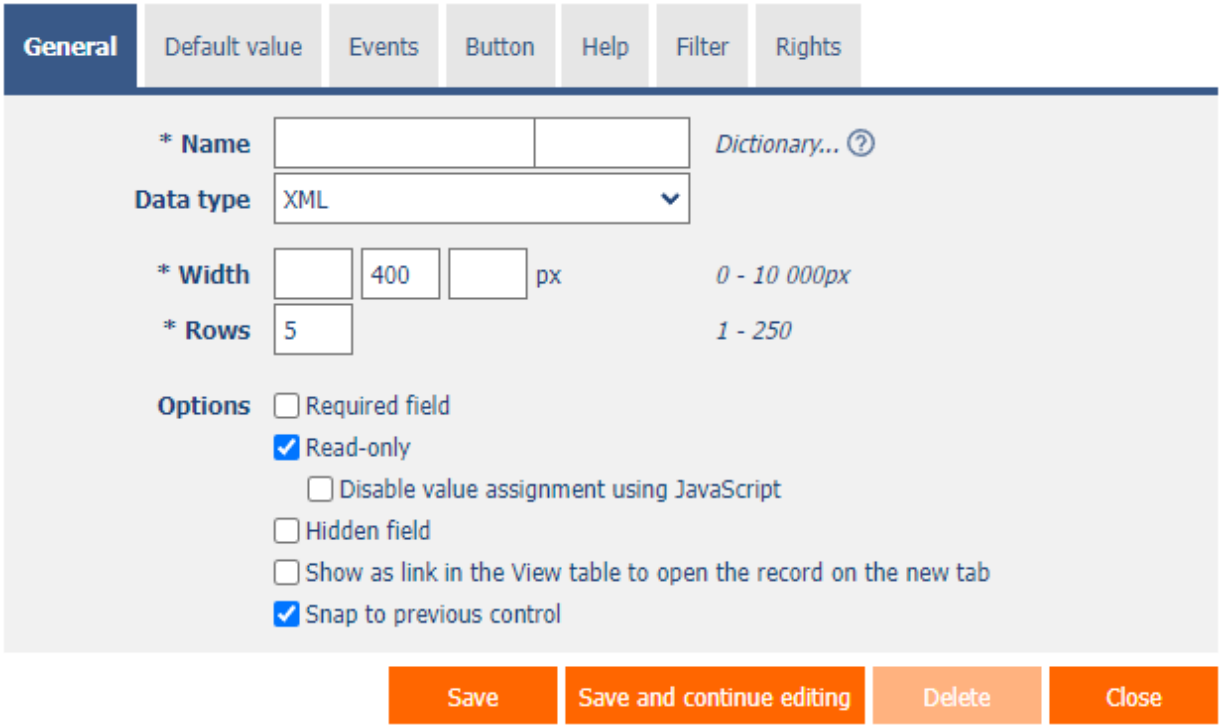
- CodeMirror text box with numbered lines for writing general source code with syntax highlighting.
- “String” of unlimited length, represented as the database type “text” for the Firebird database and “nvarchar(max)” for the MSSQL database.

HTML



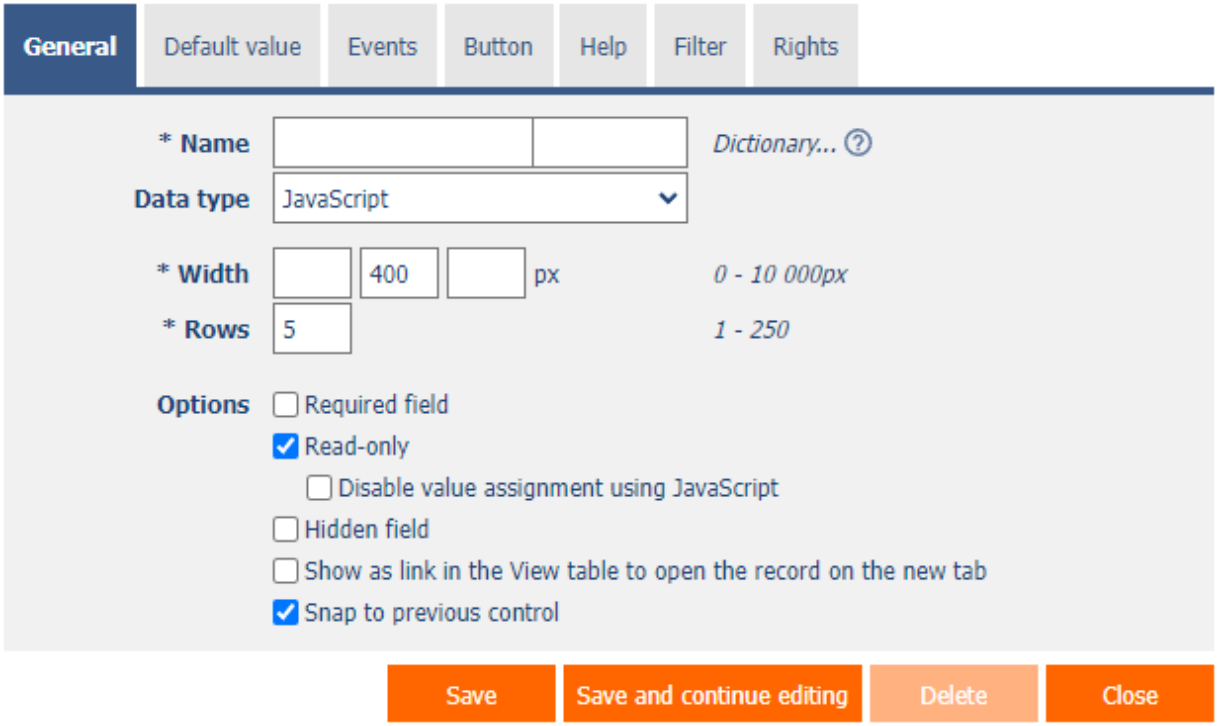
- CodeMirror text box with numbered lines for writing general HTML code with color syntax highlighting.
- “String” of unlimited length, represented as the database type “text” for the Firebird database and “nvarchar(max)” for the MSSQL database.

XML



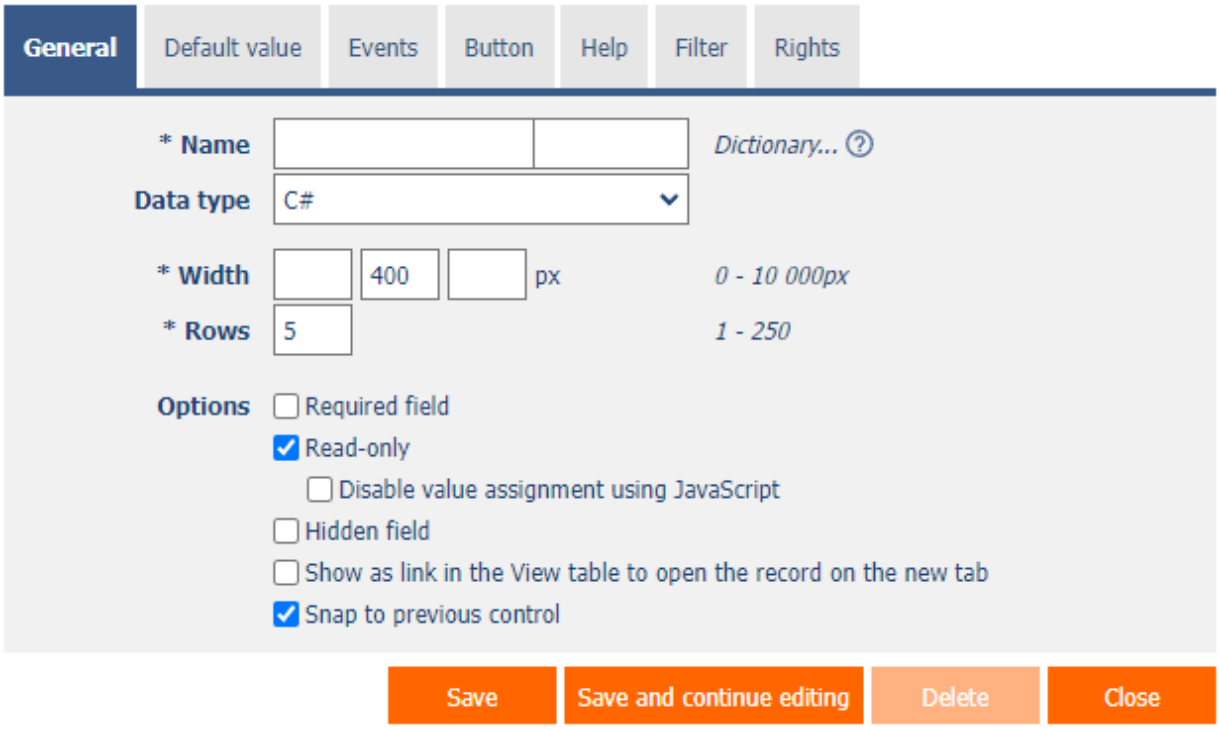
- CodeMirror text box with numbered lines for writing XML code with color syntax highlighting.
- "String" of unlimited length, represented as the database type "text" for the Firebird database and "nvarchar(max)" for the MSSQL database.

JavaScript



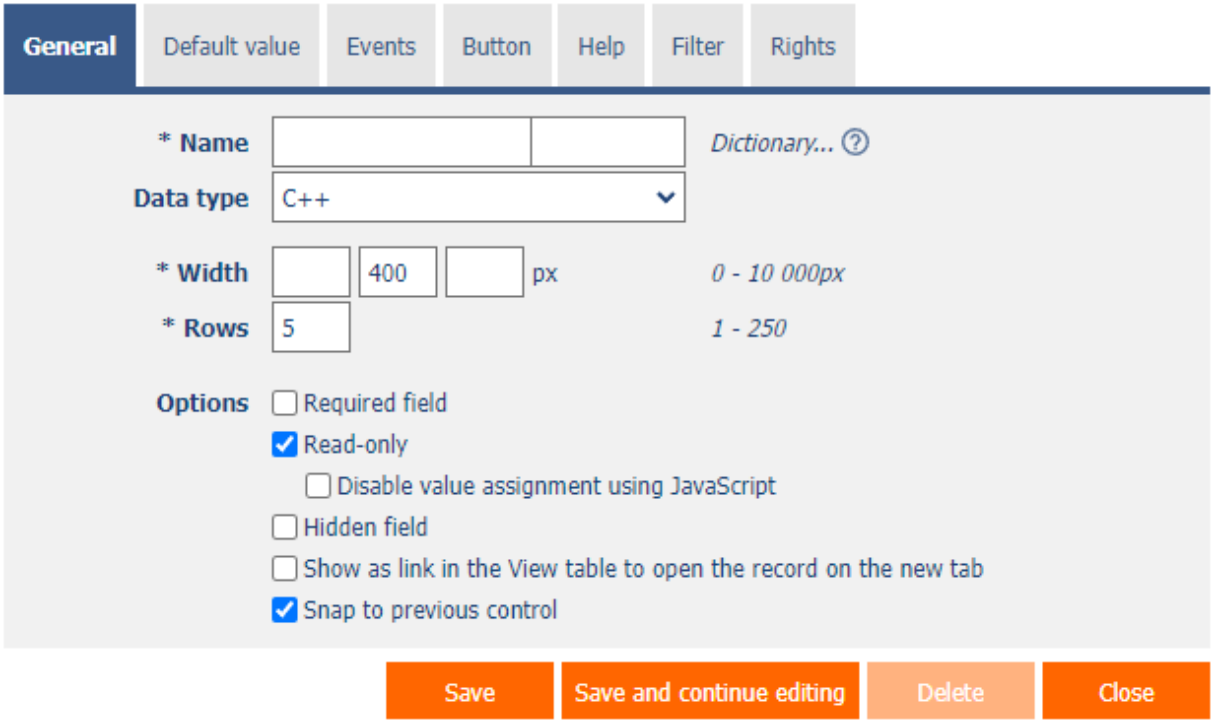
- CodeMirror text box with numbered lines for writing javascript code with color syntax highlighting.
- “String” of unlimited length, represented as the database type “text” for the Firebird database and “nvarchar(max)” for the MSSQL database.

C#



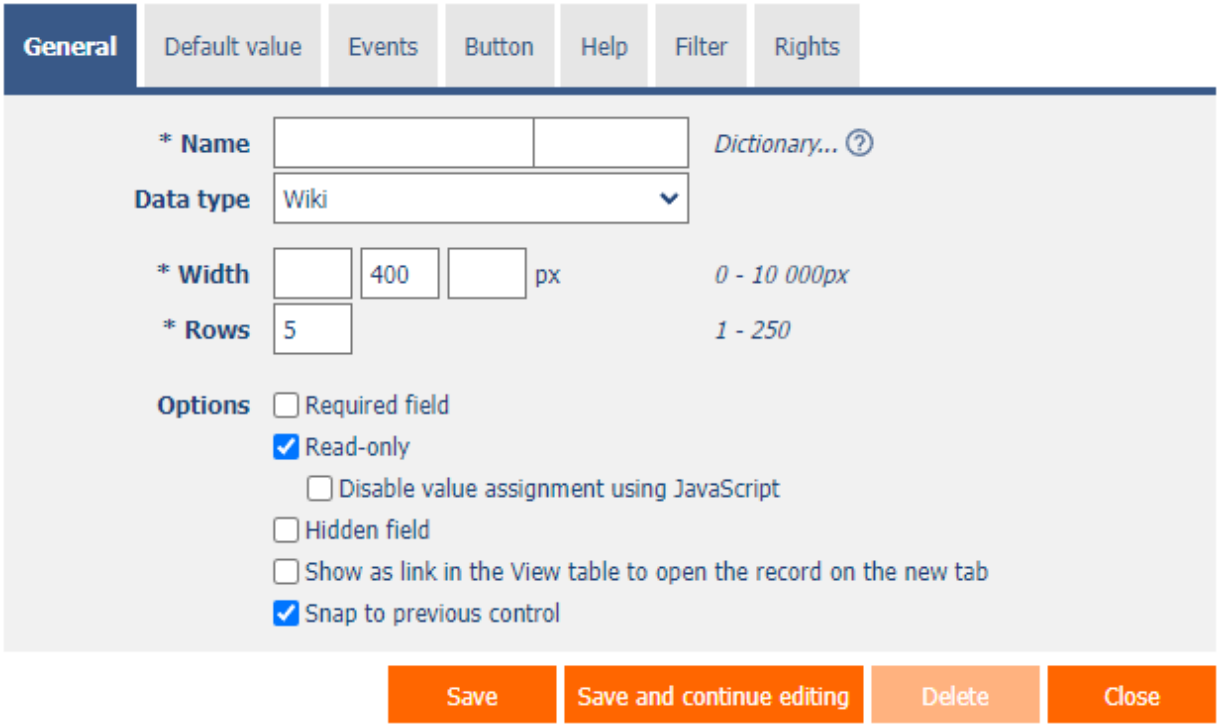
- CodeMirror text box with numbered lines for writing C# source code with color syntax highlighting.
- “String” of unlimited length, represented as the database type “text” for the Firebird database and “nvarchar(max)” for the MSSQL database.

C++



- CodeMirror text box with numbered lines for writing C++ source code with color syntax highlighting.
- “String” of unlimited length, represented as the database type “text” for the Firebird database and “nvarchar(max)” for the MSSQL database.

Wiki



- CodeMirror text box with numbered lines for writing general Wiki code.
- "String" of unlimited length, represented as the database type "text" for the Firebird database and "nvarchar(max)" for the MSSQL database.

2.1.5 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.

2.1.6 Rows

For all data types

- The number of lines determines the optional height of the text box in the edit form.
- The default number of rows is 5. The minimum number of rows is 1, the maximum is 250.

2.1.7 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”.
  - 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: it equals “” or is not defined. And now in SQL: ng_ta = '' OR ng_ta IS NULL.

- **Read only**

For all data types

  - The read-only control setting determines whether the text box should be disabled in the edit form.
  - You cannot manually enter values in a disabled field, the text box itself has a gray background, and the text of the value in that 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:
    - control_SetValue(#ng_#, 'abc');
    - ta_Insert(#ng_#, 'abc');
  - Controls can generally be disabled using javascript:
    - control_Disable(#ng_#); // General javascript function for all database controls
    - ta_Disable(#ng_#);
  - It is also possible to enable controls using javascript:
    - control_Enable(#ng_#); // General javascript function for all database controls

    - ta_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 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**

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

  - 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.

- **Allow multiple assignments**

For data type: Text string

  - Allowing multiple assignments determines whether a link with the ability to change the value should be displayed below the value of the control in the view table or multi-source planning calendar without having to open and edit the record in the edit form. The editable link only appears in view tables or calendars that have multiple data assignments allowed.
  - Multiple assignments can only be made by a logged in user, the link is not displayed to an anonymous user.

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

For a text string data type 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 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**

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 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-" to ensure that the value the offset is calculated automatically according to the formula "Offset = Total column width – Control width".

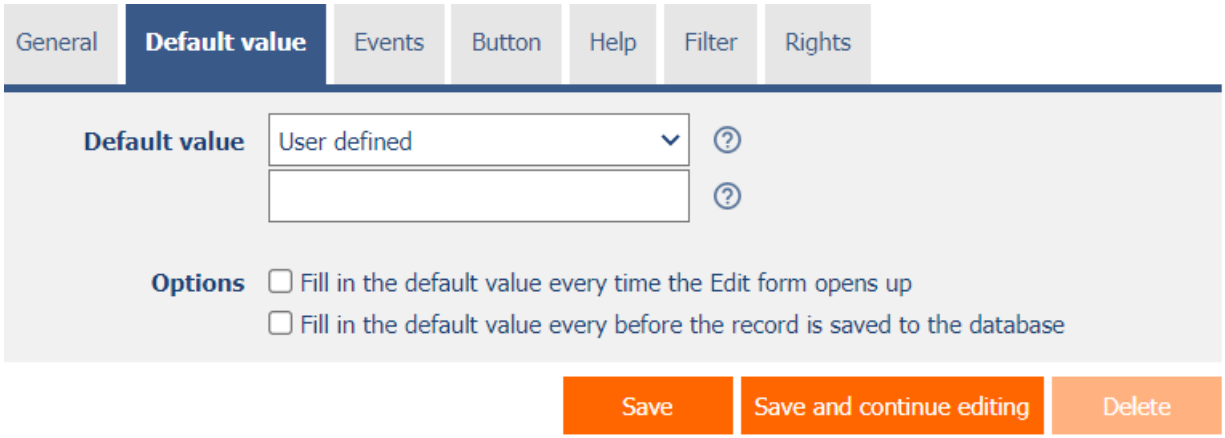
- **Comment**

For data type: Text string

- The comment determines whether the text box should be displayed read-only with the option to enter comments using the "New Comment" button located below the text box in the edit form.
- Clicking on the "New comment" button will display a new dialog with the option to enter a comment in plain text.
- Subsequent confirmation of the entered comment using the "Insert comment" button or using the "CTRL + ENTER" key combination inserts it into the control field in the edit form
  - blank line
  - see the name of the author of the comment below,
  - comma and space,
  - current date and time,
  - go to a new line and
  - the text of the written comment itself.

- **Author**
  - Login name – the author's name will be the login name of the currently logged in user
  - Name – the author's name will be the full name of the currently logged in user
- **Sorting**
  - Ascending – the entered comment will be inserted at the end after a blank line
  - Descending – the written comment will be inserted at the beginning before the blank line
- **Button name**
  - New comment – the text of the button will be set to the default text “New comment”
  - User definer – the button text will be set to a custom value

2.2 "Default value" tab



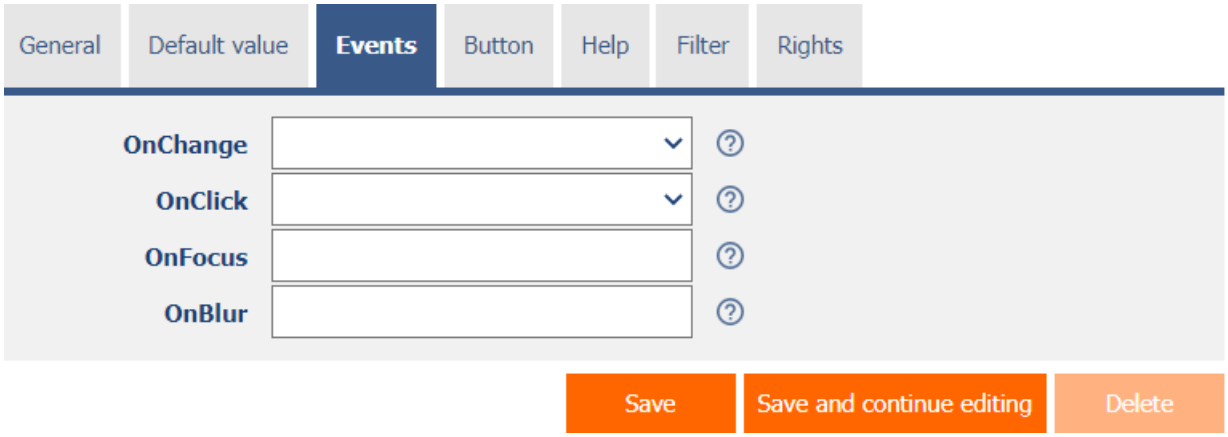
2.2.1 Default value

- 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 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:
    - **User 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 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
    - **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.

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 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



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.

2.3.2 OnClick

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

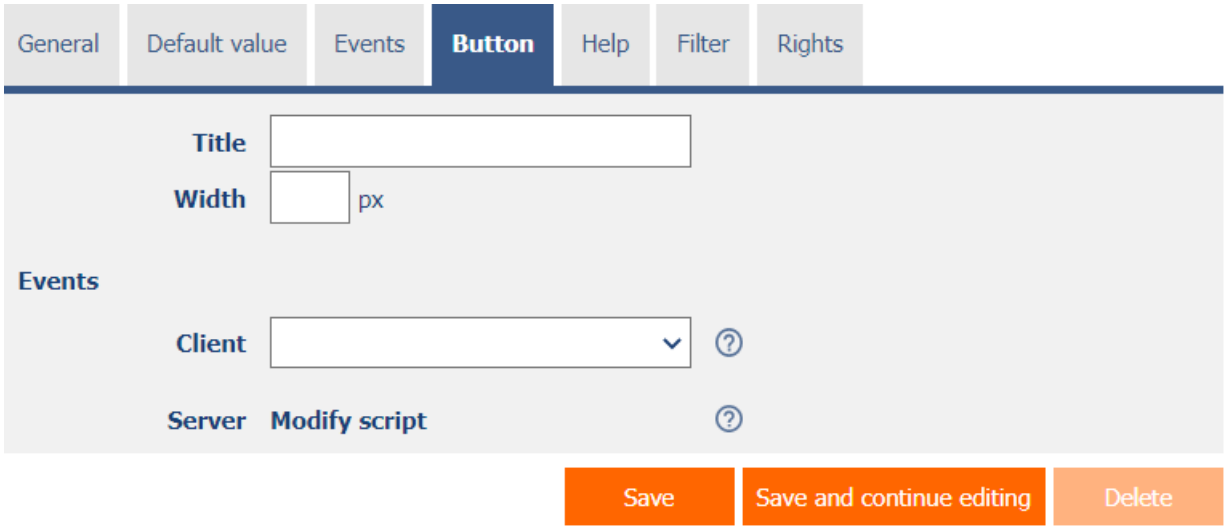
2.3.3 OnFocus

- 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



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.

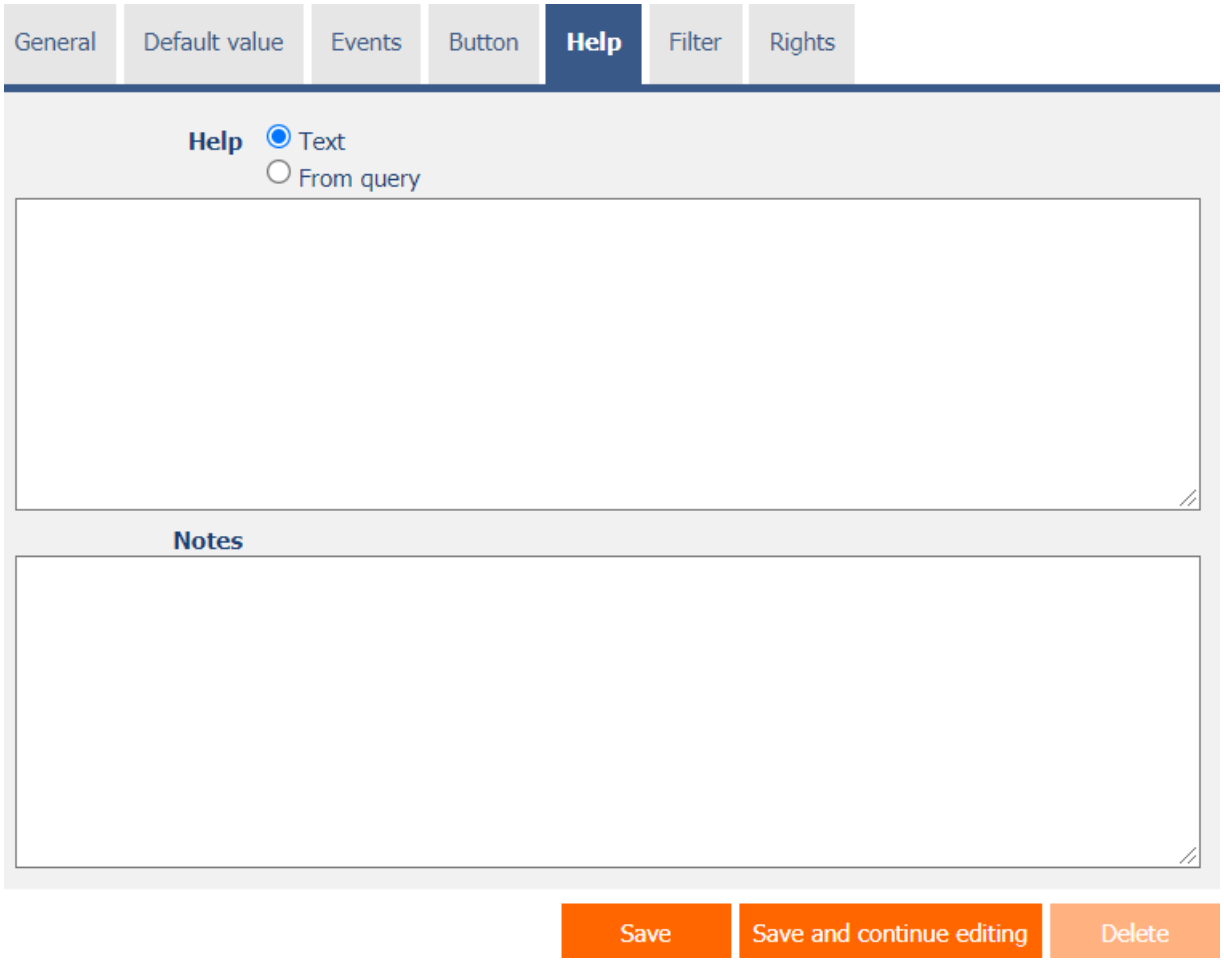
2.4.3 Client

- 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:
  - **Do you really want to?** – return confirm('Do you really want to...?');
  - **AJAX...** – return bt_Eval(this, 'ngef.aspx? Test', 'evalFunction');

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 “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

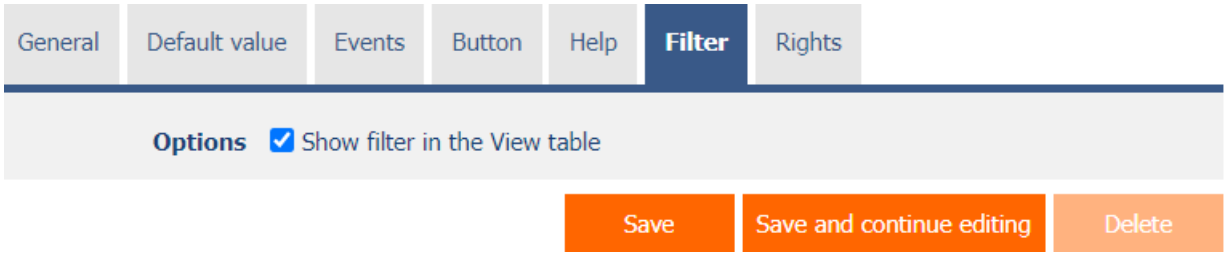
- In conjunction with the question mark icon, a variant can be used in which, after clicking on the icon, the help text followed by three dots is not displayed, but on the contrary, the content of, for example, the wonder located anywhere in the edit form is displayed/hidden. On the new line after the 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

2.5.2 Notes

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

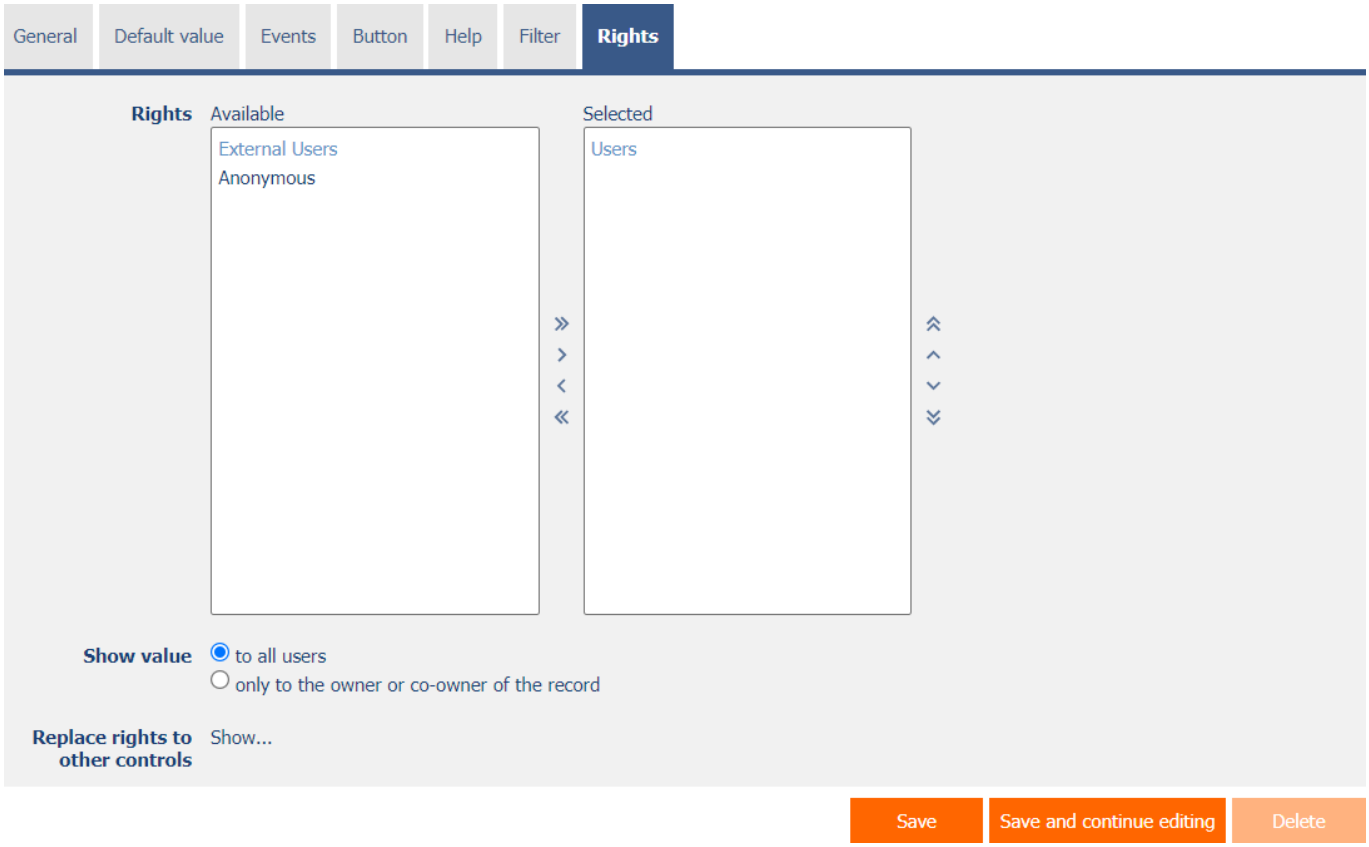
2.6 "Filter" tab



2.6.1 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

2.7 "Rights" tab



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 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)”.

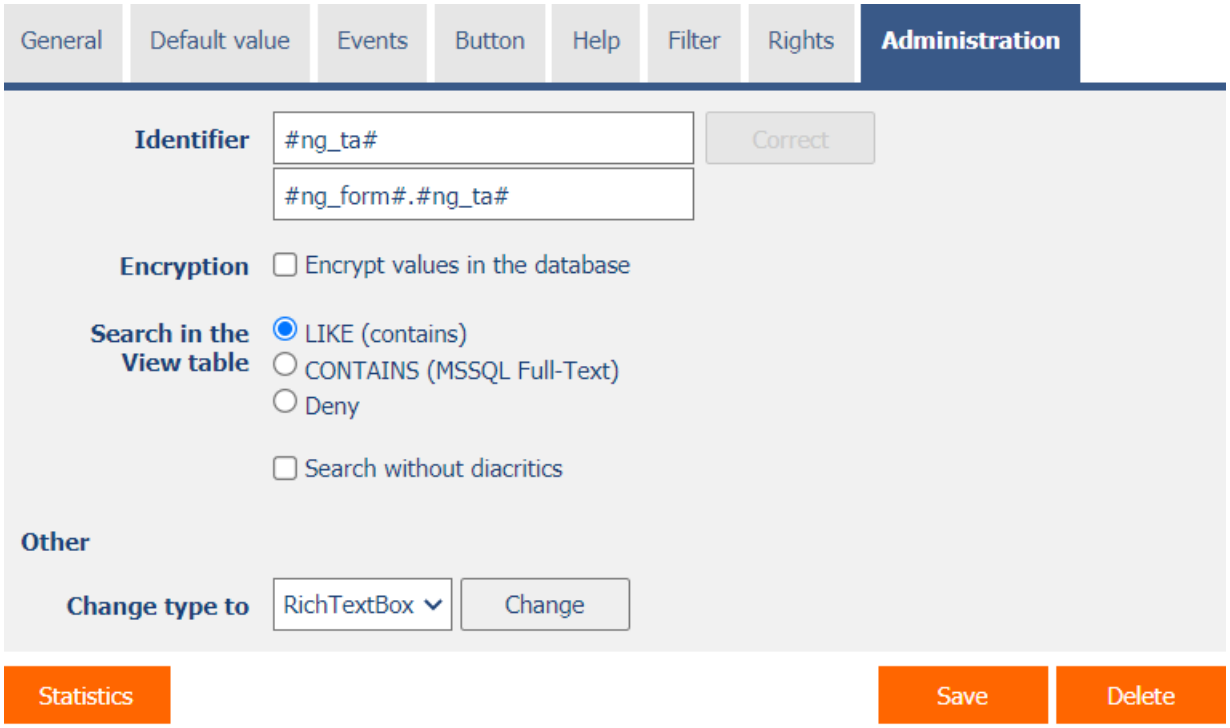
2.7.2 Show value

- 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.

2.7.3 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.

2.8 "Administration" tab



- 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 Encryption

- Encryption determines whether the values stored in this column in the database will be encrypted using either the default encryption key or the encryption key specified in the optional configuration file "Config\EncryptionKey.txt ".
- Decrypted values can be displayed only in the edit form, everywhere else in the application the placeholder string "*****" is displayed instead of specific values stored in the database.
- When encryption is turned on again, all values stored in this column in the database are encrypted.
- When you turn off encryption again, all values stored in this column in the database are decrypted.

2.8.4 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.
- Textarea supports the following search types:
    - **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.

2.8.5 Other

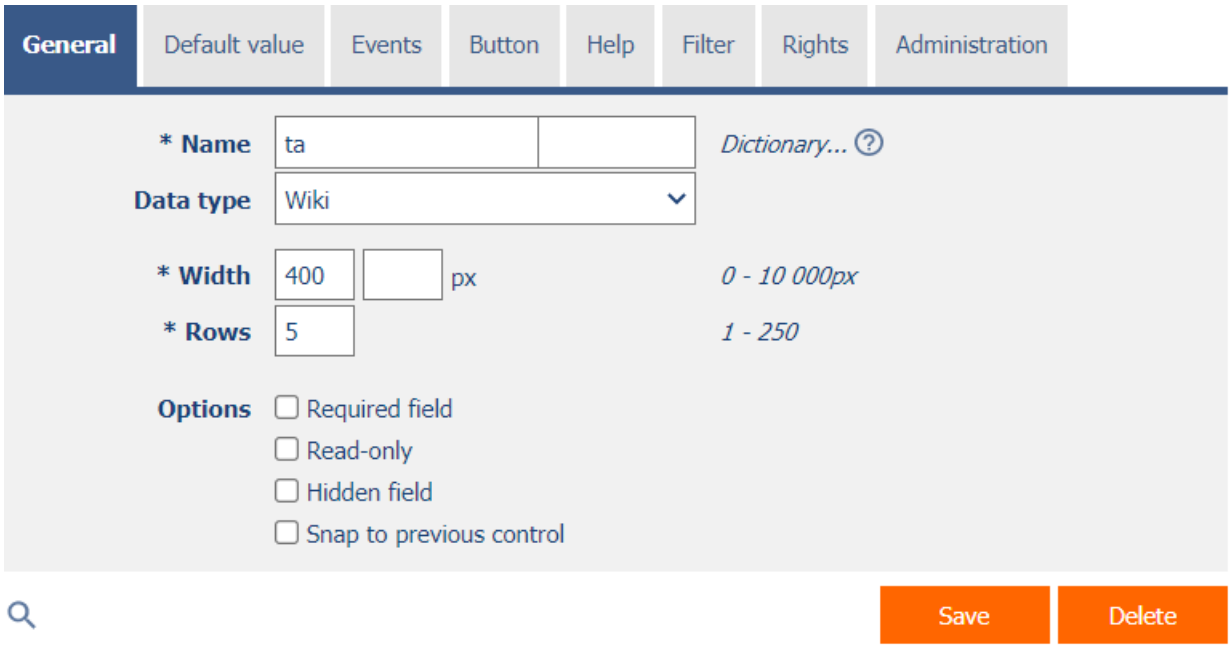
- **Change type to**
    - The drop-down list offers the types of controls to which conversion of the control is supported, including automatic conversion of the data present in the database table.

2.8.6 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.

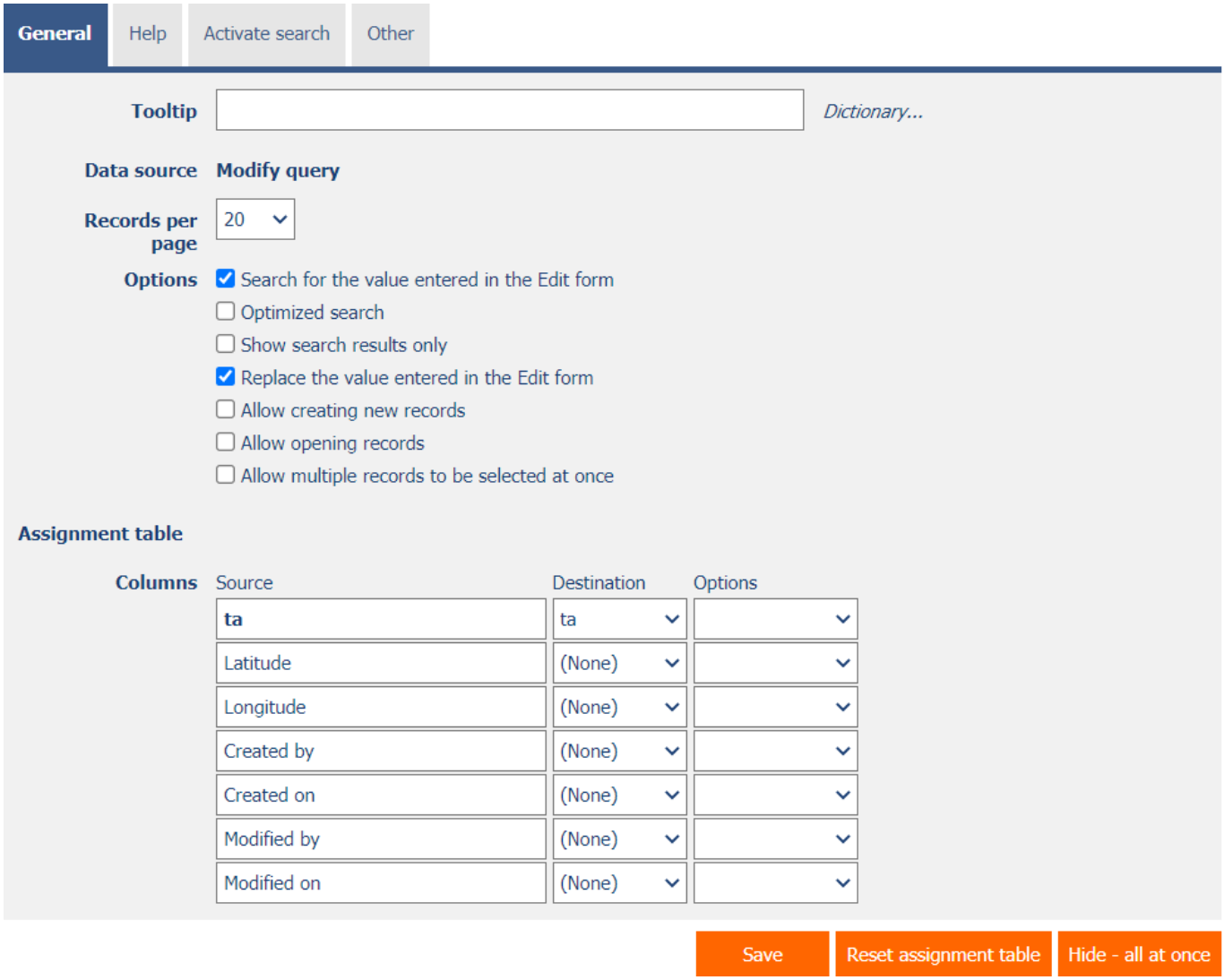
3 Picker

3.1 Basic information



- 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



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.

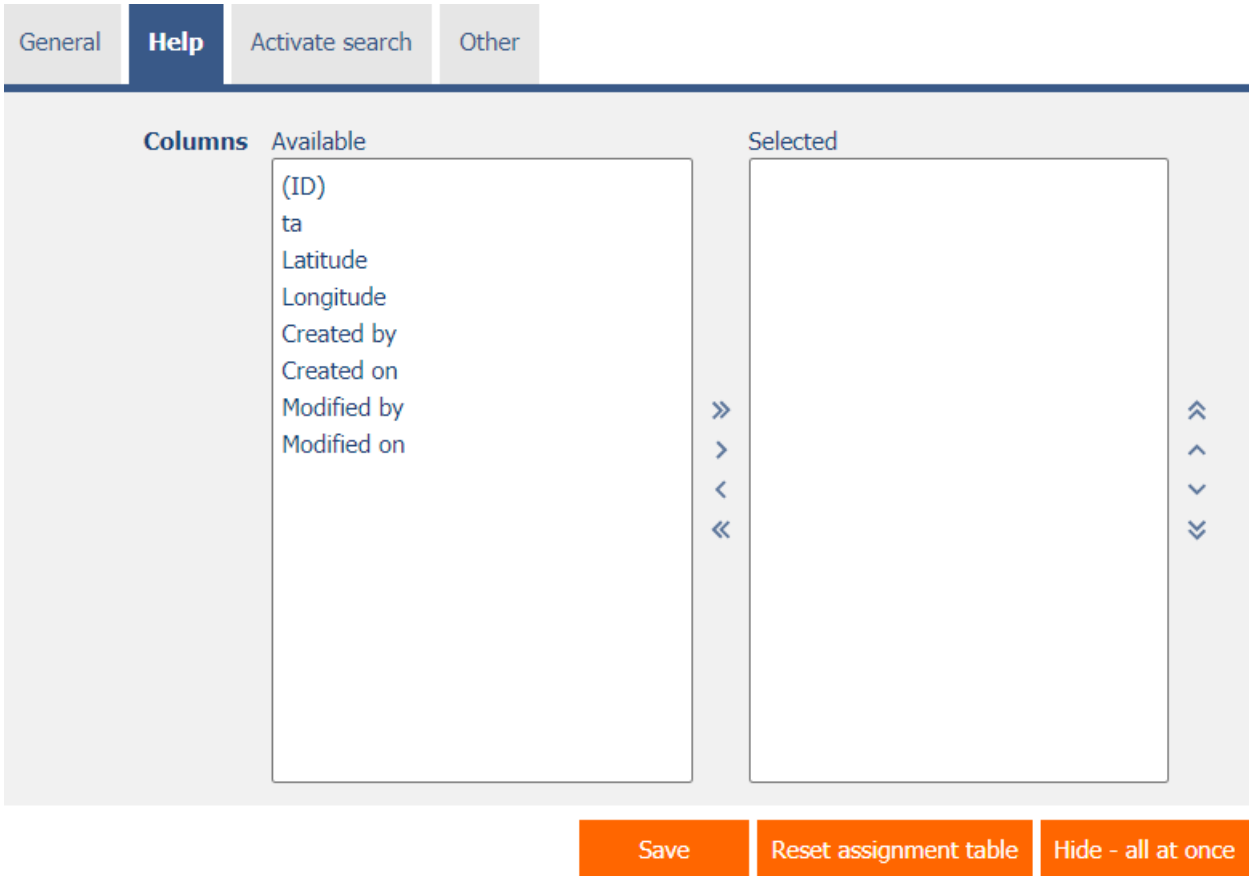
3.2.4 Options

- **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 picker's view table and whether the edit form should offer to display an ajax picker.
- **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 overwrite 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 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

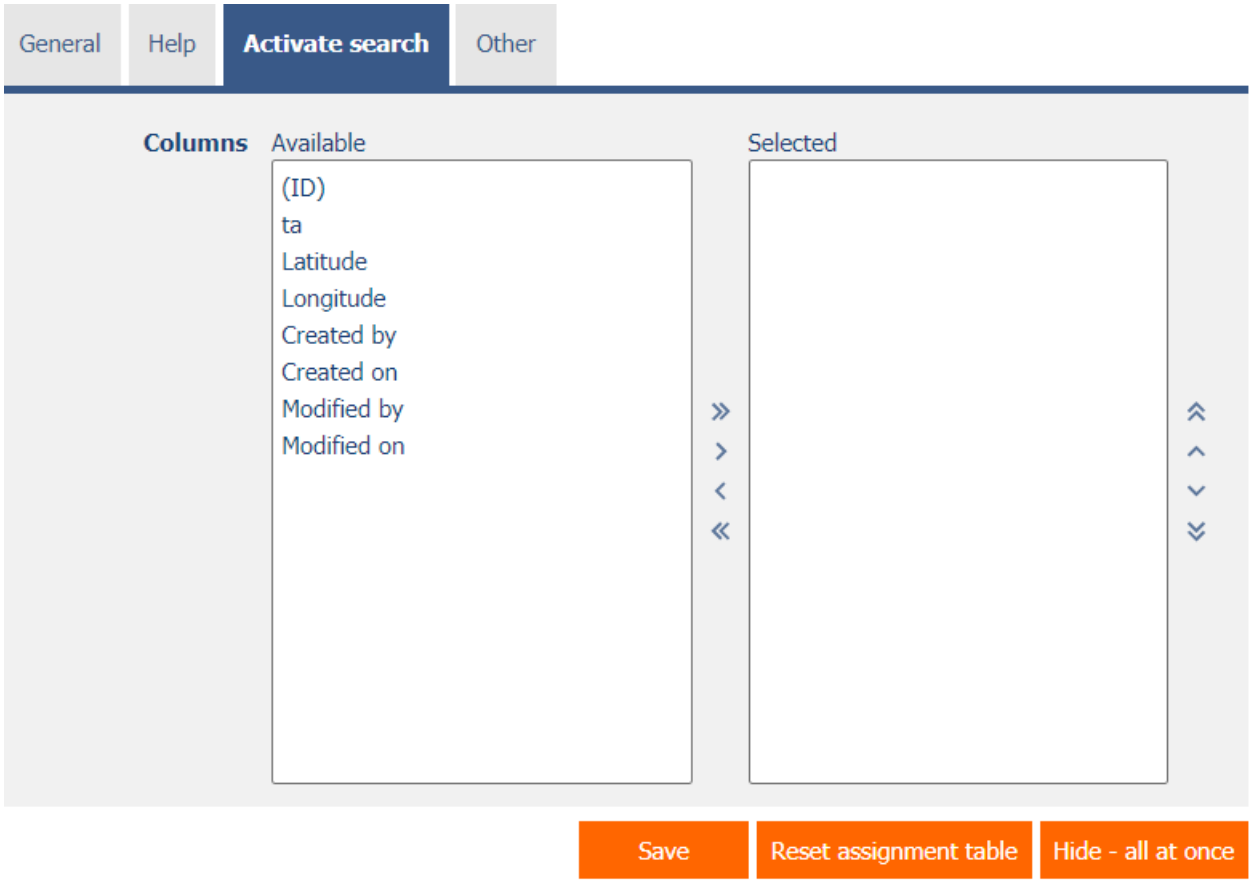
- **Columns**
  - The column settings determine which column of the view table in the picker will be copied to which control in the edit form.
  - Picked columns – those that have a source and destination set – are automatically displayed in the picker's view table in the order in which they appear in 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 view settings defined in the control's properties.

3.3 “Help” tab



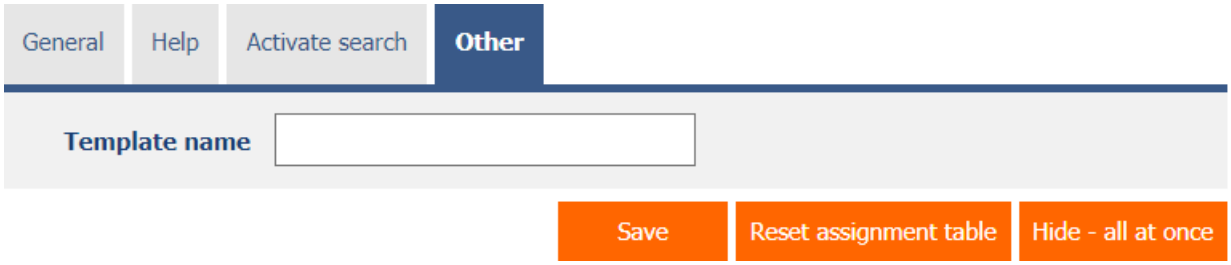
- On the “Help” tab, it is possible to set a list of all other columns to be displayed in the picker's view table, even though none of these columns are peaked.

3.4 “Activate search” tab



- 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



- **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_#);
- var s = #ng_#.value;
- control_SetValue(#ng_#, s);
- control_SetValue(#ng_#, 'abc');
- ta_Insert(#ng_#, 'abc');
- picker_Open(#ng_#);

4.1.2 Text strings

- #ng_#.value = 'abc';

4.1.3 Source codes, HTML, XML, JavaScript, C#, C++

- ta_InitColorWords(['a', 'b']);
- ta_FormatAll(#ng_#);

4.2 Server functions

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