Common Properties

- accessKey :- Specifies the shortcut key that sets focus on the UI component.
- activeStateEnabled - Specifies whether the UI component changes its state as a result of user interaction.
- disabled - Specifies whether the UI component responds to user interaction.
- elementAttr - Specifies the global attributes to be attached to the UI component's container element.
- focusStateEnabled - Specifies whether the UI component can be focused using keyboard navigation.
- height - Specifies the UI component's height.
- hint - Specifies text for a hint that appears when a user pauses on the UI component.
- hoverStateEnabled - Specifies whether the UI component changes its state when a user pauses on it.
- inputAttr - Specifies the attributes to be passed on to the underlying HTML element.
- isValid - Specifies or indicates whether the editor's value is valid.
- name The value to be assigned to the name attribute of the underlying HTML element.
- max - The last date that can be selected within the UI component.
- maxLength - Specifies the maximum number of characters you can enter into the textbox.
- min - The minimum date that can be selected within the UI component.
- placeholder - Specifies a placeholder for the input field.
- readOnly - Specifies whether the editor is read-only.
- rtlEnabled - Switches the UI component to a right-to-left representation.
- tabIndex - Specifies the number of the element when the Tab key is used for navigating.
- text - Specifies the text displayed by the check box.
- validationError - Information on the broken validation rule. Contains the first item from the validationErrors array.
- spellcheck - Specifies whether or not the UI component checks the inner text for spelling mistakes.
- stylingMode - Specifies how the UI component's text field is styled.
- type - A format used to display date/time information.
- validationErrors - An array of the validation rules that failed.
- validationMessageMode - Specifies how the message about the validation rules that are not satisfied by this editor's value is displayed.
- validationStatus - Indicates or specifies the current validation status.
- value - Specifies the UI component state.
- valueChangeEvent - Specifies the DOM events after which the UI component's value should be updated.
- visible - Specifies whether the UI component is visible.
- width - Specifies the UI component's width.

Common Methods

- beginUpdate() - prevents the UI from refreshening until the endUpdate() calls.
- defaultOptions() - applies default options for all the specified checkbox, button, etc.
- dispose() - dispose the UI component's styles, scripts and other properties which are initialized.
- element() - gets the root element
- endUpdate() - refreshens the UI component after a call of beginUpdate() method
- focus() - focus that UI component when page refresh.
- getInstance(element) - gets the UI component's instance.
- instance() - gets the UI component's instance for use it to access other methods of the UI component.
- off(eventName) - Detaches all event handlers from a single event.
- off(eventName, eventHandler) - Detaches a particular event handler from a single event.
- on(eventName, eventHandler) - Subscribes to an event.
- on(events) - Subscribes to events.
- option() - Gets all UI component properties.
- option(optionName) - Gets the value of a single property.
- option(optionName, optionValue) - Updates the value of a single property.
- option(options) - Updates the values of several properties.
- registerKeyHandler(key, handler) - Registers a handler to be executed when a user presses a specific key.
- repaint() - Repaints the UI component without reloading data. Call it to update the UI component's markup.
- reset() - Resets the value property to the default value.
- resetOption(optionName) - Resets a property to its default value.

Common Events

- change - Raised when the UI component loses focus after the text field's content was changed using the keyboard.
- closed - Raised once the drop-down editor is closed.
- contentReady - Raised when the UI component's content is ready.
- copy - Raised when the UI component's input has been copied.
- cut - Raised when the UI component's input has been cut.
- disposing - Raised before the UI component is disposed of
- enterKey - Raised when the Enter key has been pressed while the UI component is focused.
- focusIn - Raised when the UI component gets focus.
- focusOut - Raised when the UI component loses focus..
- initialized - Raised only once, after the UI component is initialized.
- input - Raised each time the UI component's input is changed while the UI component is focused.
- keyDown - Raised when a user is pressing a key on the keyboard.
- keyUp - Raised when a user releases a key on the keyboard.
- opened - Raised once the drop-down editor is opened.

- optionChanged - Raised after a UI component property is changed.
- paste - Raised when the UI component's input has been pasted.
- valueChanged - Raised after the UI component's value is changed.

Checkbox

The CheckBox is a small box, which when selected by the end user, shows that a particular feature has been enabled or a specific property has been chosen.

Datebox

The DateBox is a UI component that displays date and time in a specified format, and enables a user to pick or type in the required date/time value.

Properties

- acceptCustomValue - Specifies whether or not the UI component allows an end-user to enter a custom value.
- adaptivityEnabled - Specifies whether or not adaptive UI component rendering is enabled on a small screen.
- applyButtonText - The text displayed on the Apply button.
- applyValueMode - Specifies the way an end-user applies the selected value.
- buttons[] - Allows you to add custom buttons to the input text field.
- calendarOptions - Configures the calendar's value picker. Applies only if the pickerType is "calendar".
- cancelButtonText - The text displayed on the Cancel button.
- dateOutOfRangeMessage - Specifies the message displayed if the specified date is later than the max value or earlier than the min value.
- dateSerializationFormat - Specifies the date-time value serialization format. Use it only if you do not specify the value at design time.
- deferRendering - Specifies whether to render the drop-down field's content when it is displayed. If false, the content is rendered immediately.
- disabledDates - Specifies dates that users cannot select. Applies only if pickerType is "calendar".
- displayFormat - Specifies the date display format. Ignored if the pickerType property is "native"
- dropDownButtonTemplate - Specifies a custom template for the drop-down button.
- dropDownOptions - Configures the drop-down field which holds the content.
- interval - Specifies the interval between neighboring values in the popup list in minutes.
- invalidDateMessage - Specifies the message displayed if the typed value is not a valid date or time.
- opened - Specifies whether or not the drop-down editor is displayed.
- openOnFieldClick - Specifies whether a user can open the drop-down list by clicking a text field.
- pickerType - Specifies the type of the date/time picker.

- showAnalogClock - Specifies whether to show the analog clock in the value picker. Applies only if type is "datetime" and pickerType is "calendar".
- showClearButton - Specifies whether to display the Clear button in the UI component.
- showDropDownButton - Specifies whether the drop-down button is visible.
- useMaskBehavior - Specifies whether to control user input using a mask created based on the displayFormat.

DropDown Box

The DropDownBox UI component consists of a text field, which displays the current value, and a drop-down field, which can contain any UI element.

Properties

- acceptCustomValue- Specifies whether the UI component allows a user to enter a custom value.
- buttons[] - Allows you to add custom buttons to the input text field.
- contentTemplate - Specifies a custom template for the drop-down content.
- dataSource - Binds the UI component to data.
- deferRendering - Specifies whether to render the drop-down field's content when it is displayed. If false, the content is rendered immediately.
- displayExpr - Specifies the data field whose values should be displayed.
- displayValueFormatter - Customizes text before it is displayed in the input field.
- dropDownButtonTemplate - Specifies a custom template for the drop-down button.
- dropDownOptions - Configures the drop-down field which holds the content.
- fieldTemplate - Specifies a custom template for the text field. Must contain the TextBox UI component.
- items - An array of items used to synchronize the DropDownBox with an embedded UI component.
- opened - Specifies whether or not the drop-down editor is displayed.
- openOnFieldClick - Specifies whether a user can open the drop-down list by clicking a text field.
- showClearButton - Specifies whether to display the Clear button in the UI component.
- showDropDownButton - Specifies whether the drop-down button is visible.
- valueExpr - Specifies which data field provides unique values to the UI component's value.

Number Box

The NumberBox is a UI component that displays a numeric value and allows a user to modify it by typing in a value, and incrementing or decrementing it using the keyboard or mouse.

Properties

- buttons[] - Allows you to add custom buttons to the input text field.

- format - Specifies the value's display format and controls user input accordingly.
- invalidValueMessage - Specifies the text of the message displayed if the specified value is not a number.
- mode - Specifies the value to be passed to the type attribute of the underlying <input> element.
- showClearButton - Specifies whether to display the Clear button in the UI component.
- showSpinButtons - Specifies whether to show the buttons that change the value by a step.
- step - Specifies how much the UI component's value changes when using the spin buttons, Up/Down arrow keys, or mouse wheel.
- useLargeSpinButtons - Specifies whether to use touch friendly spin buttons. Applies only if showSpinButtons is true.

Select Box

The SelectBox UI component is an editor that allows an end user to select an item from a drop-down list.

Properties

- acceptCustomValue - Specifies whether the UI component allows a user to enter a custom value. Requires the onCustomItemCreating handler implementation.
- buttons[] - Allows you to add custom buttons to the input text field.
- dataSource - Binds the UI component to data.
- deferRendering - Specifies whether to render the drop-down field's content when it is displayed. If false, the content is rendered immediately.
- displayExpr - Specifies the data field whose values should be displayed.
- displayValue - Returns the value currently displayed by the UI component.
- dropDownButtonTemplate - Specifies a custom template for the drop-down button.
- dropDownOptions - Configures the drop-down field which holds the content.
- fieldTemplate - Specifies a custom template for the text field. Must contain the TextBox UI component.
- grouped - Specifies whether data items should be grouped.
- items - An array of items displayed by the UI component.
- itemTemplate - Specifies a custom template for items.
- minSearchLength - The minimum number of characters that must be entered into the text box to begin a search. Applies only if searchEnabled is true.
- noDataText - Specifies the text or HTML markup displayed by the UI component if the item collection is empty.
- opened - Specifies whether or not the drop-down editor is displayed.
- openOnFieldClick - Specifies whether a user can open the drop-down list by clicking a text field.
- searchEnabled - Specifies whether to allow searching.
- searchExpr - Specifies the name of a data source item field or an expression whose value is compared to the search criterion.
- searchMode - Specifies a comparison operation used to search UI component items.

- searchTimeout - Specifies the time delay, in milliseconds, after the last character has been typed in, before a search is executed.
- selectedItem - Gets the currently selected item.
- showClearButton - Specifies whether to display the Clear button in the UI component.
- showDataBeforeSearch - Specifies whether or not the UI component displays unfiltered values until a user types a number of characters exceeding the minSearchLength property value.
- showDropDownButton - Specifies whether the drop-down button is visible.
- showSelectionControls - Specifies whether or not to display selection controls.
- useItemTextAsTitle - Specifies whether the SelectBox uses the item's text as a title attribute.
- valueExpr - Specifies which data field provides unique values to the UI component's value.
- wrapItemText - Specifies whether text that exceeds the drop-down list width should be wrapped.

Events

- onCustomItemCreating - A function that is executed when a user adds a custom item. Requires acceptCustomValue to be set to true.
- onItemClick - A function that is executed when a list item is clicked or tapped.
- onSelectionChanged - A function that is executed when a list item is selected or selection is canceled.

TextArea

The TextArea is a UI component that enables a user to enter and edit a multi-line text.

Properties

- autoResizeEnabled - A Boolean value specifying whether or not the auto resizing mode is enabled.
- maxHeight - Specifies the maximum height of the UI component.
- minHeight - Specifies the minimum height of the UI component.

Text Box

The TextBox is a UI component that enables a user to enter and edit a single line of text.

Properties

- buttons[] - Allows you to add custom buttons to the input text field.
- mask - The editor mask that specifies the custom format of the entered string.
- maskChar - Specifies a mask placeholder. A single character is recommended.
- maskInvalidMessage - A message displayed when the entered text does not match the specified pattern.
- maskRules - Specifies custom mask rules.

- mode - The "mode" attribute value of the actual HTML input element representing the text box.
- showClearButton - Specifies whether to display the Clear button in the UI component.
- showMaskMode - Specifies when the UI component shows the mask. Applies only if useMaskedValue is true.
- useMaskedValue - Specifies whether the value should contain mask characters or not.

Button

The JavaScript Button is a simple button that performs specified commands when a user clicks it.

Properties

- icon - Specifies the icon to be displayed on the button.
- onClick - A function that is executed when the Button is clicked or tapped.
- template - Specifies a custom template for the Button UI component.
- type - Specifies the button type.
- useSubmitBehavior - Specifies whether the button submits an HTML form.
- validationGroup - Specifies the name of the validation group to be accessed in the click event handler.

File Uploader

The FileUploader UI component enables an end user to upload files to the server. An end user can select files in the file explorer or drag and drop files to the FileUploader area on the page.

Properties

- abortUpload - A function that cancels the file upload.
- accept - Specifies a file type or several types accepted by the UI component.
- allowCanceling - Specifies if an end user can remove a file from the selection and interrupt uploading.
- allowedFileExtensions - Restricts file extensions that can be uploaded to the server.
- chunkSize - Specifies the chunk size in bytes. Applies only if uploadMode is "instantly" or "useButtons". Requires a server that can process file chunks.
- dialogTrigger - Specifies the HTML element which invokes the file upload dialog.
- dropZone - Specifies the HTML element in which users can drag and drop files for upload.
- invalidFileExtensionMessage - The text displayed when the extension of the file being uploaded is not an allowed file extension.
- invalidMaxFileSizeMessage - The text displayed when the size of the file being uploaded is greater than the maxFileSize.
- invalidMinFileSizeMessage - The text displayed when the size of the file being uploaded is less than the minFileSize.

- labelText - Specifies the text displayed on the area to which an end-user can drop a file.
- maxFileSize - Specifies the maximum file size (in bytes) allowed for uploading. Applies only if uploadMode is "instantly" or "useButtons".
- minFileSize - Specifies the minimum file size (in bytes) allowed for uploading. Applies only if uploadMode is "instantly" or "useButtons".
- multiple - Specifies whether the UI component enables an end-user to select a single file or multiple files.
- progress - Gets the current progress in percentages.
- readyToUploadMessage - The message displayed by the UI component when it is ready to upload the specified files.
- selectButtonText - The text displayed on the button that opens the file browser.
- showFileList - Specifies whether or not the UI component displays the list of selected files
- uploadAbortedMessage - The message displayed by the UI component when the file upload is canceled.
- uploadButtonText - The text displayed on the button that starts uploading.
- uploadChunk - A function that uploads a file in chunks.
- uploadCustomData - Specifies custom data for the upload request.
- uploadedMessage - The message displayed by the UI component when uploading is finished.
- uploadFailedMessage - The message displayed by the UI component on uploading failure.
- uploadFile - A function that uploads a file.
- uploadHeaders - Specifies headers for the upload request.
- uploadMethod - Specifies the method for the upload request.
- uploadMode - Specifies how the UI component uploads files.
- uploadUrl - Specifies a target Url for the upload request.

Events

- onBeforeSend - A function that allows you to customize the request before it is sent to the server.
- onDropZoneEnter - A function that is executed when the mouse enters a drop zone while dragging a file.
- onDropZoneLeave - A function that is executed when the mouse leaves a drop zone as it drags a file.
- onFilesUploaded - A function that is executed when the file upload process is complete.
- onProgress - A function that is executed when a file segment is uploaded.
- onUploadAborted - A function that is executed when the file upload is aborted.
- onUploaded - A function that is executed when a file is successfully uploaded.
- onUploadError - A function that is executed when an error occurs during the file upload.
- onUploadStarted - A function that is executed when the file upload is started.

Validation

Validation Group

The ValidationGroup is a UI component that allows you to validate several editors simultaneously.

Validation Result

- brokenRules - An array of the validation rules that failed.
- complete - A promise that is fulfilled when all async rules are validated.
- isValid - Indicates whether all the rules checked for the group are satisfied.
- status - Indicates the validation status.
- validators - Validator UI components included in the validated group.

Validation Summary

A UI component for displaying the result of checking validation rules for editors.

- items - An array of items displayed by the UI component.
- itemTemplate - Specifies a custom template for items.
- onItemClick - A function that is executed when a collection item is clicked or tapped.
- validationGroup - Specifies the validation group for which summary should be generated.

Validator

A UI component that is used to validate the associated DevExtreme editors against the defined validation rules.

- adapter - An object that specifies what and when to validate, and how to apply the validation result.
- validationGroup - Specifies the validation group the editor will be related to.
- validationRules - An array of validation rules to be checked for the editor with which the dxValidator object is associated.

Adapter

An object that specifies what and when to validate, and how to apply the validation result.

- applyValidationResults - A function that the Validator UI component calls after validating a specified value.
- bypass - A function that returns a Boolean value specifying whether or not to bypass validation.
- focus - A function that sets focus to a validated editor when the corresponding ValidationSummary item is focused.
- getValue - A function that returns the value to be validated.
- reset - A function that resets the validated values.
- validationRequestsCallbacks - Callbacks to be executed on the value validation.

Validation Rules

Common Properties

- ignoreEmptyValue - If true, the validationCallback is not executed for null, undefined, false, and empty strings.
- message - Specifies the message that is shown if the rule is broken.
- type - Specifies the rule type.

Async Rule

A custom validation rule that is checked asynchronously. Use async rules for server-side validation.

- reevaluate - Indicates whether the rule should always be checked for the target value or only when the value changes.
- validationCallback - A function that validates the target value.

Compare Rule

A validation rule that demands that a validated editor has a value that is equal to a specified expression.

- comparisonTarget - Specifies the function whose return value is used for comparison with the validated value.
- comparisonType - Specifies the operator to be used for comparing the validated value with the target.

Custom Rule

A rule with custom validation logic.

- reevaluate - Indicates whether the rule should be always checked for the target value or only when the target value changes.
- validationCallback - A function that validates the target value.

Email Rule

A validation rule that demands that the validated field match the Email pattern.

Numeric Rule

A validation rule that demands that the validated field has a numeric value.

Pattern Rule

A validation rule that demands that the validated field match a specified pattern.

- pattern - Specifies the regular expression that the validated value must match.

Range Rule

A validation rule that demands the target value be within the specified value range (including the range's end points).

- max - Specifies the maximum value allowed for the validated value.
- min - Specifies the minimum value allowed for the validated value.

- reevaluate - Indicates whether the rule should be always checked for the target value or only when the target value changes.

Required Rule

A validation rule that demands that a validated field has a value.

- trim - Indicates whether to remove the Space characters from the validated value.

StringLength Rule

A validation rule that demands the target value length be within the specified value range (including the range's end points).

- max - Specifies the maximum length allowed for the validated value.
- min - Specifies the minimum length allowed for the validated value.
- trim - Indicates whether or not to remove the Space characters from the validated value.

RadioGroup

The RadioGroup is a UI component that contains a set of radio buttons and allows an end user to make a single selection from the set.

Properties

- dataSource - Binds the UI component to data.
- displayExpr - Specifies the data field whose values should be displayed.
- items - An array of items displayed by the UI component.
- itemTemplate - Specifies a custom template for items.
- layout - Specifies the radio group layout.
- valueExpr - Specifies which data field provides unique values to the UI component's value.