DatePicker Control Reference

Let users select dates from a calendar.

Reference **Controls** **Editors** DatePicker

DateTimePicker is a graphical component for selecting and editing date values, combining calendar with configurable options like input masks, step intervals, and display settings.

Requirements
Configuration
DateTime Pickers Settings
Comparative
Runtime Execution

Requirements

This component is Portable. It runs both on Windows (WPF) and on Web Pages hosted in any platform.

Configuration

The DatePicker, and TimePicker controls share common and similar configuration options. These options include label text, localization, selected date, binding modes, border styles, label positioning, and control width adjustments. These settings configure the controls for selecting and editing date and time values.

DateTime Pickers Settings

DateTime Pickers configurations	
Field	**Description**
Label Text	Specifies the text label displayed alongside the DateTimePicker control. It guides users on what the control is used for, such as "Select Date".
Localize	When checked, it adjusts the label to align with the user's regional settings, presenting the text in the preferred local language.
SelectedDate	Represents the current value of the selected date and time. This field is used to show the chosen date and time from the DateTime Picker control.
Binding	Determines the binding mode for the selected date. **TwoWay**: Changes in the UI update the data source and vice versa. **OutputOnly**: Only outputs the selected date without accepting user input. **InputOnly**: Only accepts user input without reflecting changes back to the data source.
Preview	Defines the preview for the label. **LinkedTags**: Shows a preview of tags linked to the selected brush. **Placeholder**: Displays a placeholder indicating where the preview will appear without actual content. **Whitespace**: Shows whitespace in the preview area to keep it empty and clean.

InputMask	This option is available at **DateTimePicker**. It determines the format of the date and time input, such as "ShortTime", which configures the control to accept time inputs in a shorter format (e.g., HH). **Custom**: Allows the user to define a custom date and time format. **ShortDate**: Displays the date in a short format (e.g.: "MM/dd/yyyy"). **ShortDateTime**: Displays the date and time in a short format (e.g.: "MM/dd/yyyy HH"). **ShortDateLongTime**: Displays the date in a short format and time in a long format (e.g.: "MM/dd/yyyy HH:mm"). **ShortTime**: Displays the time in a short format (e.g.: "HH"). **LongTime**: Displays the time in a long format (e.g.: "HH:mm"). **LongDateTime**: Displays the date and time in a long format (e.g.: "dddd, MMMM dd, yyyy HH:mm"). **FullDateTime**: Displays the full date and time in a long format, including milliseconds if applicable (e.g.: "dddd, MMMM dd, yyyy HH:mm.fff"). **MonthDay**: Displays the month and day (e.g.: "MMMM dd"). **RFC1123**: Displays the date and time in a format compliant with RFC 1123 (e.g.: "ddd, dd MMM yyyy HH':'mm':'ss 'GMT'"). **SortableDateTime**: Displays the date and time in a sortable format (e.g.: "yyyy'-'MM'-'dd'T'HH':'mm':'ss"). **UniversalSortable**: Displays the date and time in a universal sortable format (e.g.: "yyyy'-'MM'-'dd HH':'mm':'ss'Z'"). **YearMonth**: Displays the year and month (e.g.: "yyyy MMMM").
TimeStep Interval	Sets the step interval for adjusting time values. An interval of 30 means that time can be adjusted in 30-minute increments, making it easier to select specific times. Option include: **15**: Sets the step interval to 15 minutes. The control increments or decrements the time in 15-minute steps. **30**: Sets the step interval to 30 minutes. The control increments or decrements the time in 30-minute steps. **45**: Sets the step interval to 45 minutes. The control increments or decrements the time in 45-minute steps. **60**: Sets the step interval to 60 minutes. The control increments or decrements the time in 60-minute steps.
Border	Sets the border style for the date picker. **OutlineAndFill**: Combines outline and fill effects; **OutlineOnly**: Displays only the outline of the control; and **HorizontalLine**: Shows a horizontal line as the border.
Label Position	Defines the position of the label. Options include: **Auto**: Determines the label's position automatically; **Top**: Places the label above the control; **Left**: Places the label to the left of the control; **Inside**: Places the label inside the control; and **None**: Hides the label.
Width	Sets the width of the control. When "Auto" is selected, the width adjusts automatically based on the content and layout. Width value is 80 if not set to auto.
Preview (Label)	Indicates how the label will preview linked tags or values. **LinkedTags:** Shows the associated tags or values linked to the control for preview purposes; **Placeholder**: Displays a placeholder indicating where the preview will appear without actual content; and **Whitespace**: Shows whitespace in the preview area to keep it empty and clean.

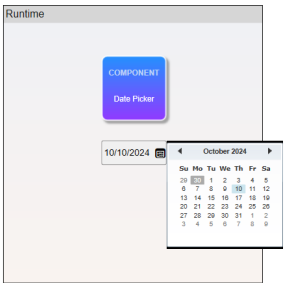
Comparative

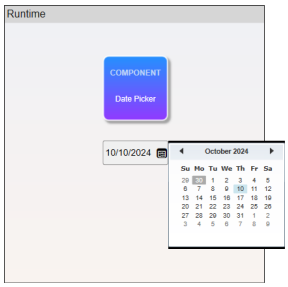
The following comparative table illustrates the configuration items available for the DateTimePicker, DatePicker, and TimePicker controls. It shows that all three controls share common settings such as Label Text, Localize, SelectedDate, Binding, Preview, Border, Label Position, Width, and Preview (Label). However, DateTimePicker uniquely includes the InputMask and TimeStep Interval settings, while TimePicker includes only the TimeStep Interval setting. DatePicker does not include these two settings, focusing solely on date selection.

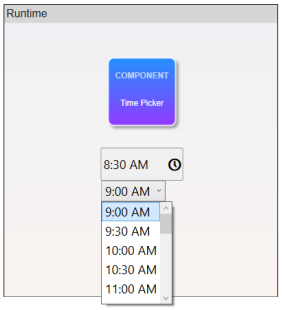
Configuration item	DateTimePicker	DatePicker	TimePicker
Label Text	✓	✓	✓
Localize	✓	✓	✓
SelectedDate	✓	✓	✓
Binding	✓	✓	✓
Preview	✓	✓	✓
InputMask	✓	✗	✗
TimeStep Interval	✓	✗	✓
Border	✓	✓	✓
Label Position	✓	✓	✓
Width	✓	✓	✓
Preview (Label)	✓	✓	✓

Runtime Execution

During runtime, the DateTimePicker, DatePicker, and TimePicker controls become interactive elements within the user interface. These controls are designed with embedded functionalities that permit operators to adjust date and time values in real-time. Configuration settings for these controls are typically bound to specific tags — data references within the application's framework — that are made visible in the display interface. This binding enables operators to manipulate the controls' properties, such as selected dates or times, directly through the user interface. The dynamic nature of these controls ensures that operators can efficiently manage temporal data within the application's operational context.







In this section...
