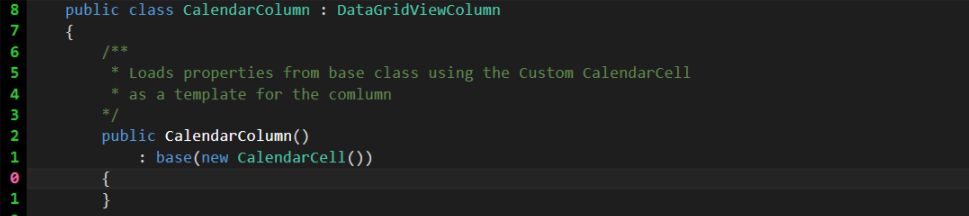
## Calendar Column

The calendar column is an extension of the DatagridViewColumn that utilyzes three classes inheritting from the DatagridViewColumn, DataGridviewTextboxCell, and the DateTimePicker as well as incorporating the IDataGridViewEditingControl interface as a means of embedding the DateTimePicker class in the form of the CalendarEditingControl.

#### The Calendar Column

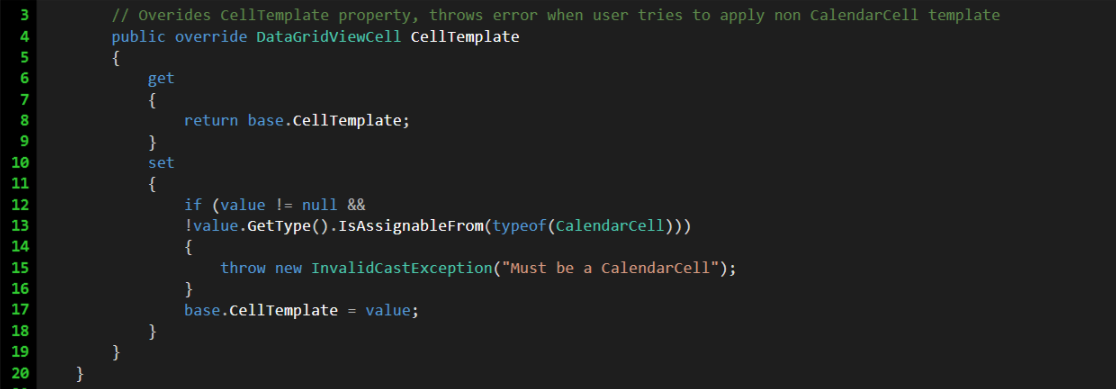
##### The Constructor



Calendar Column

The CalendarColumn class inherits from the DatagridViewColumn. This control will ultimately be available under the DatagridView control using the add a column feature. The class can also be hard coded in later. The public CalendarColumn is the entry point for the class, the colon base(new CalendarCcell()) runs the base class constructor (DatagridViewColumn) and then adds the CalendarCell() class as the cell template (we will build the CalendarCell later).

##### CellTemplate Override

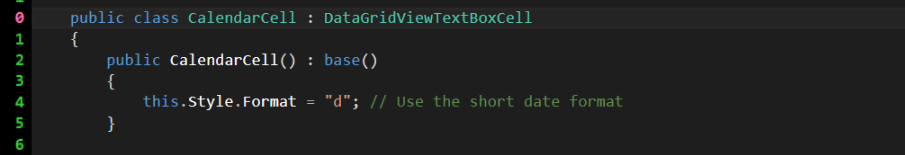


Calendar Column CellTemplate Overide

Overides the CellTemplate property on the DatagridViewColumn. Causes any other CellTemplate setting other than the CalendarCell to throw an error.

#### The Calendar Cell

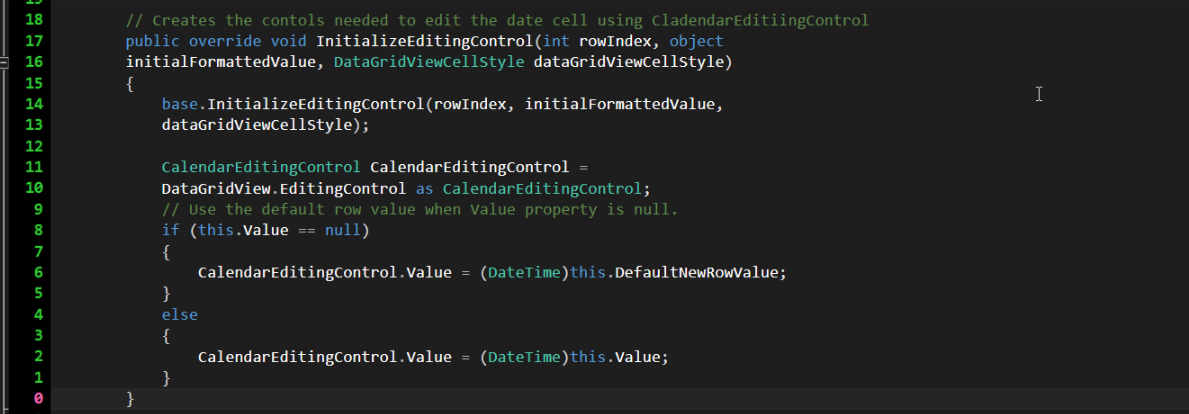
##### The Constructor



Calendar Cell

The CalendarCell class inherits from the DatagridViewTextrBoxCell. Like the CalendarColumn, the CalendarCell constructor calls the base constructor as well. Then the Format for the custom cell is set to short date.

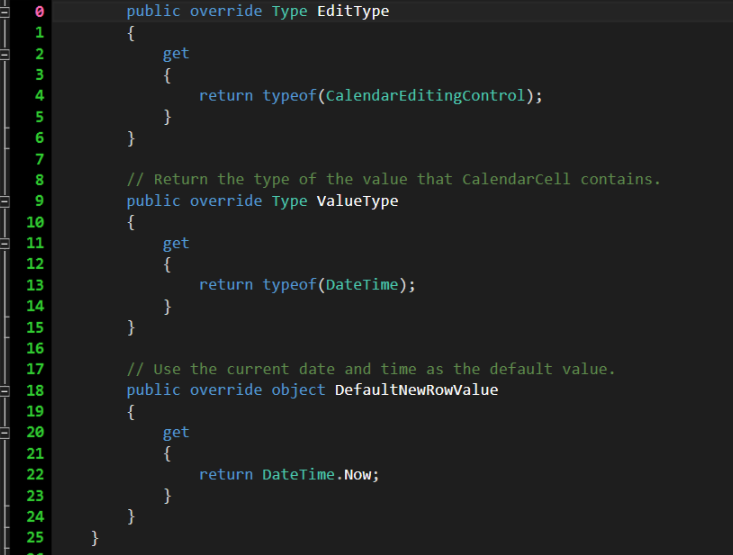
#### InitializeEditingControl Override



Initialize Editing Control

When editing is initiated cell information is sent to the editing control. If the cell value is null, the CalendarEditingControl default value is set to today.

#### Additional Properties



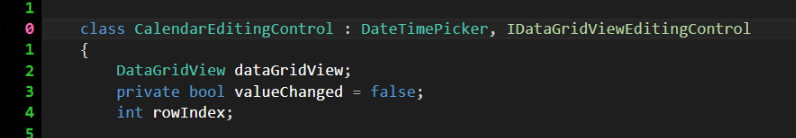
Additional Properties

These properites override basic information about the class.

#### The CalendarEditingControl

Will be called when the user starts to edit a cell in a calendar column. Inherits from the DateTimePicker and uses the IDatagridViewEditingControl inorder to apply DateTimePicker edits to the cell.

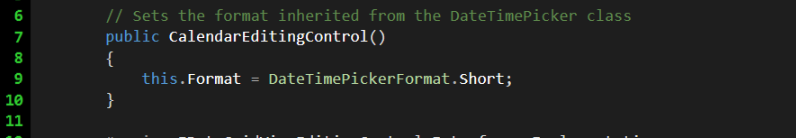
##### Properties



Calendar Editing Control Properties

Properties needed for class

#### The Constructor



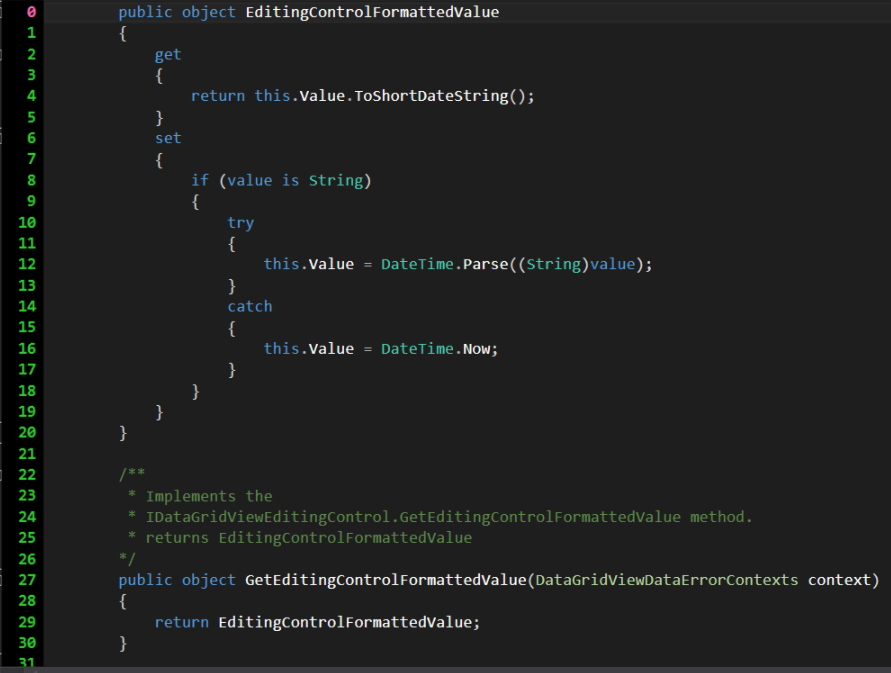
Calendar Editing Control Constructor

Sets the editing format to Short Date

#### Interface Implemenatons

The following sections of code are the required implementations for the IDatagridViewEditingControl. These are the parts of the code that need to be defined so that when the DatagridView that parents our CalendarColumn is able to apply the user's edits.

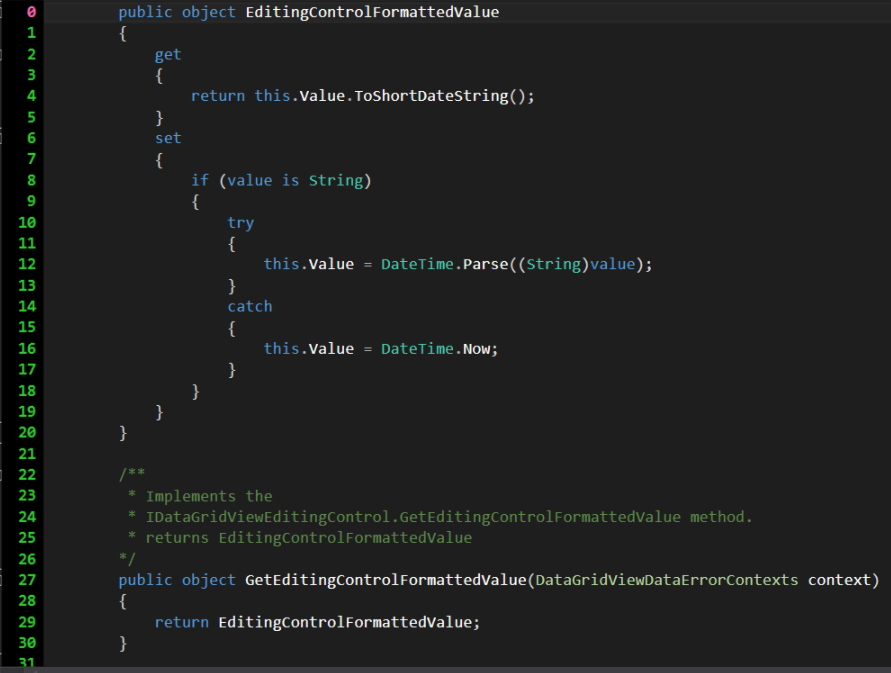
##### Editing Control Formatted Value



Editing Control Formatted Value

Handles getting and setting the value of the Editing Control, returns the value as a short date, as well as attempts to store it as a short date. If the value cannot be stored as a short date, the value is set to todays date. GetEditingControlFormattedValue handles retrieving the formatted value, and also has it's own set of error handling.

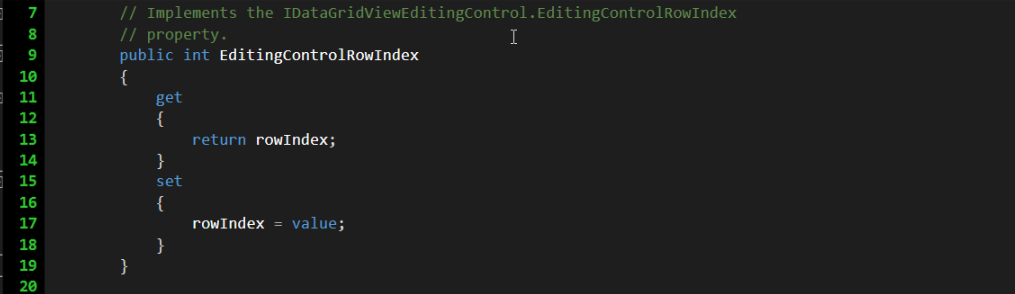
##### Apply Cell Styling to Editing Control



Apply Cell Styling to Editing Control

Sets the look of the CalendarEditingControl to match the parent DatagridView

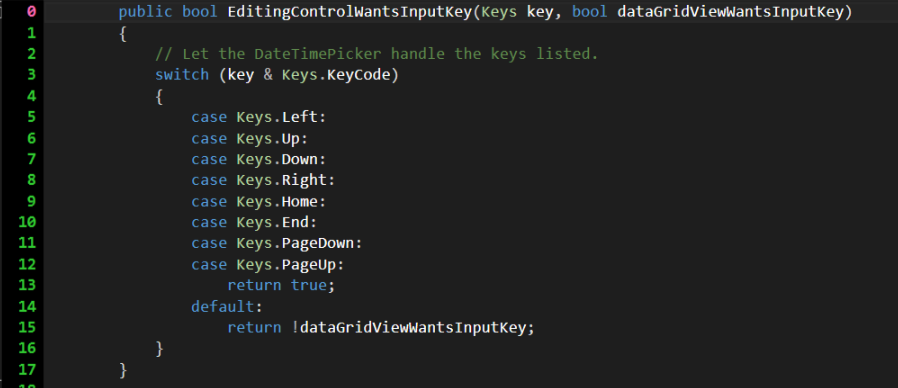
##### Editing Control Row Index



Editing Control Row Index

Property to store the cell index that is being editted.

##### Editing Control Wants Input Key



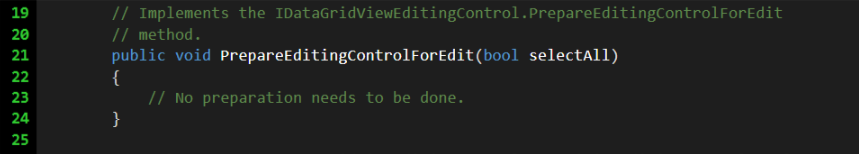
Editing Control Wants Input Key

Defines how to handle the editing control needing a key input.

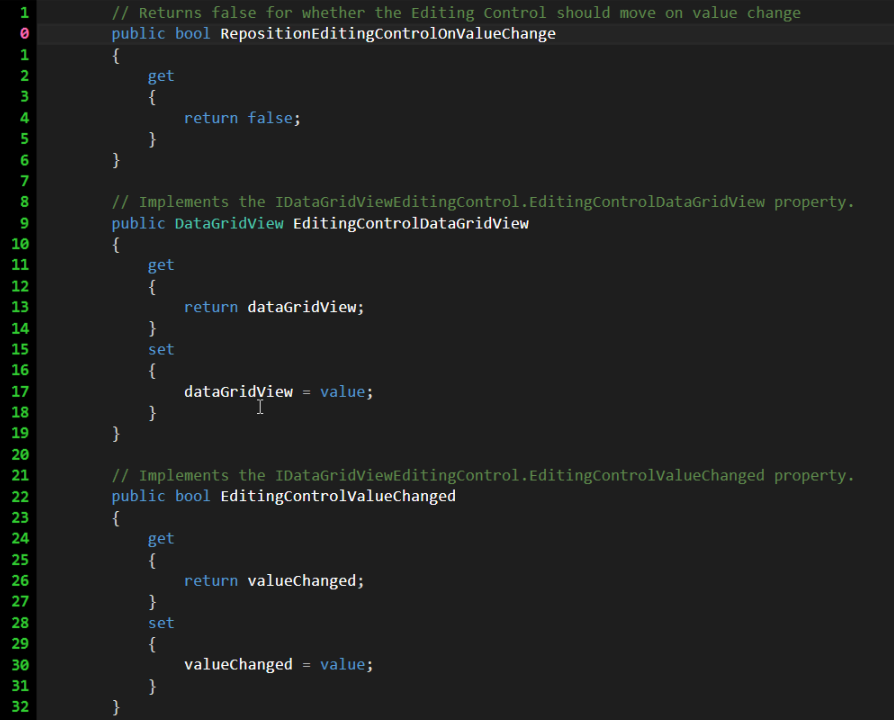
##### Prepare Editing Control for Edit

No preparation is needed prior to edit

##### Misc. Implemenations



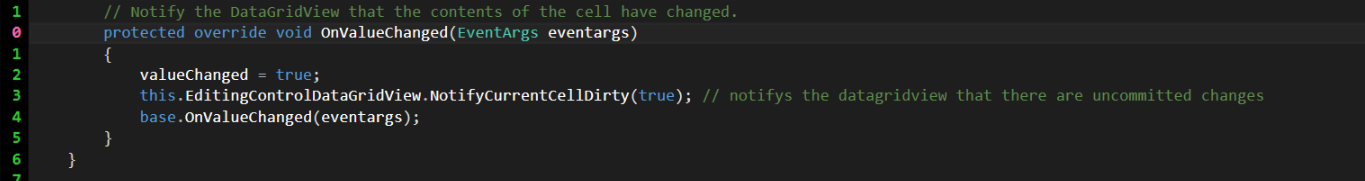
Prepare Editing Control for Edit



Repositon Editing Control on Value Change

PrepareEditingControlforEdit does not need set as there is no preparation needed before editing can begin. RepositionEditingControlOnValueChange is set to return false so the control won't reposition when the value changes. We have the EditingControlDatagridView set to return our class' dataGridView property. EditingControlValueChanged boolean is set to return to return the our class' valueChanged boolean. Finally the EditingPanelCursor is set to return the default DateTimePicker cursor.

##### On Value Changed

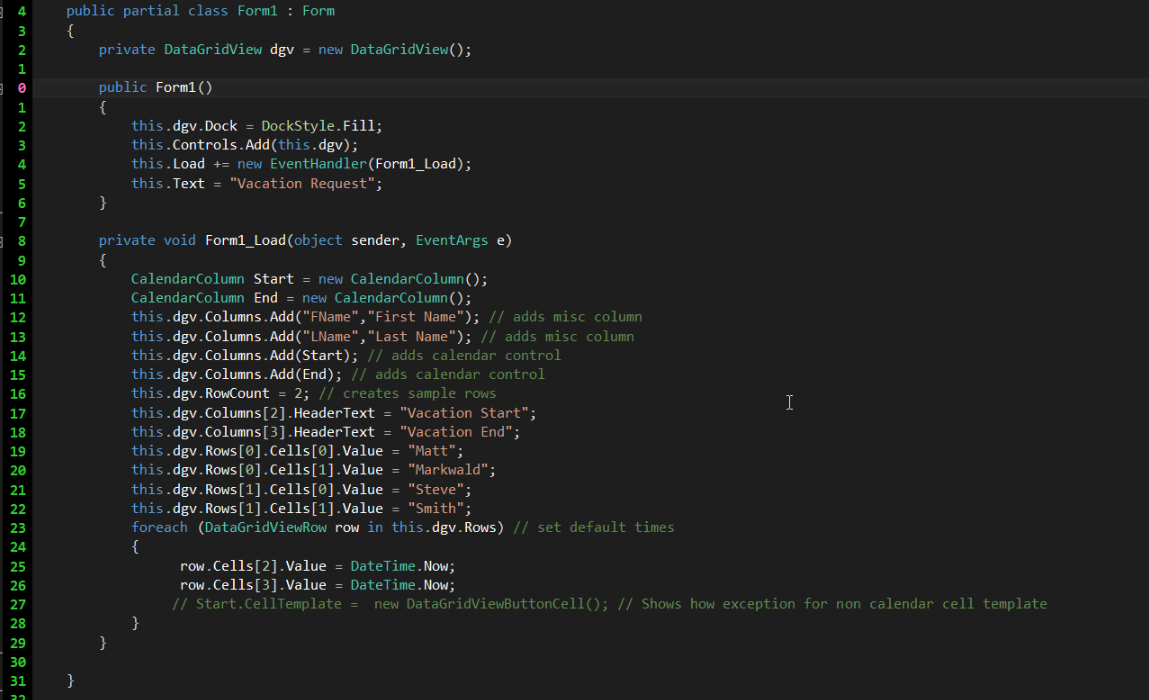


On Value Changed

The last bit of code we need is an override for our base class void, OnValueChanged. This handles when the value changes by setting our valueChanged boolean to true, then notify the parent DatagridView, before calling the base OnValueChanged sending the eventargs.

## Demo

Go to form1 constructor and add the following code.



Demo

Run the program. Notice what happens when you click on the Vacation Start and Vacation End cells.