

I had started this project a couple of years ago, then I abandoned it, so I decided to go back to it, if I had to start it from scratch today I would change many things since they are not the most optimal, but this does not prevent the calendar from working well.

Now, the control is a calendar to which we can add events, this can be used for many things such as showing staff vacations, collection events, shifts, time activities or graphically showing date intervals.

The control has support for drag & drop of events, such as being able to modify its date range with the mouse, if we press the Ctrl key and do drag & drop it duplicates this event. Then the data entry part is in charge of the programmer, in the examples I added some sample forms, what I did not do was create a series of events, but it is possible if you have a little patience, any doubt about the latter I can give a guide how to do it.

Inside the examples in the "Advance" folder you will find an example linking the calendar to a SQLite database, you should add the reference to the dll J3cnn.dll

The calendar events are not handled by index but by keys, because internally they are ordered by date and in this way the index is lost (I did not like how I approached it at first).

The control does not have any dependencies, even the icons are painted with lines.

The scrollbar wheel works correctly when it is compiled, I did this as I did not use Safe subclass, to avoid headaches when programming I leave it like this, compiled it works correctly the same for the MouseLeave event. If someone needs it to work in the ide they can remove the "If App.LogMode Then".

To close, I know that there are many things that were pending, I would like you to comment if something should be implemented or corrected, also if you see any errors; that way it's easier for me. In the first days of uploading, I will surely be updating it frequently as they inform me or find details.

---

Here is a quick detail of its functions and properties:

**AddEvents** : Main function to add events to the calendar, parameters required; subject (descriptive title), start date and time, end date and time and color of the event, the rest of the parameters are optional, return value an event key.

**CenterCalendarInNow** : Moves the scroll to the current time.

**Clear** : Remove all events.

**DateValue** : Assigns or returns the current date of the calendar.

**DayHaveEvents** : Returns True/False if there are events on a specific day.

**DropDownColor** : When there are many events in a day and in the Month view mode there are more than can be shown, a dropdown bar is shown which we can change the color with this property.

**EventsCount** : Number of events added.

**EventsRoundCorner** : Boolean property to show or not show rounded corners in events and buttons.

**FirstDayOfWeek** : Here we can assign which day we want to be displayed as the first day of the week, by default it uses the system day.

**GetAllEvents** : Gets a collection of the keys of the added events.

**GetEventData** : Obtains the data of an event, its first parameter is the key of the event which we can obtain with GetAllEvents or by some event, the rest of the parameters are return values.

**GetEventsFromDay** : Returns a collection of keys of the events of a specific day,

**GetSelectionRangeDate**: Function to obtain the selected date range.

**HeaderColor** : Color of the header and part of the theme.

**HideEvent** : Hides the event, useful for filtering.

**LinesColor** : Color of the lines.

**Redraw** : Enables or disables calendar repainting, this serves to speed up the loading of events.

**RemoveEvent** : Removes an event.

**Refresh** : Refreshes the calendar redraw.

**SelectedEvent** : Returns the selected event.

**SelectionColor** : Color of the selection.

**SetStrLanguage** : Here we can pass the translation of the words used.

**ShowToolTipEvents**: If you want to show or not the tooltip window, it can be replaced by a custom one with more detailed information. See the **EventMouseEnter** and **EventMouseLeave** events.

**Update** : It is more complete than refresh, it reorders the events by date and alphabetically, recalculates the position and finally repaints everything.

**UpdateEventData** : Function to update the data of an event. The key of the event that we want to modify must be passed.

**UserCanChangeDate** : Enable or disable the user to be able to change the current calendar page.

**UserCanChangeEvents** : Enables or disables whether the user can change events (by stretching or dragging).

**UserCanChangeViewMode**: Hide all the buttons at the top of the right side. In this way the user cannot change the view mode or the programmer takes control of which view he wants to show.

**UserCanScrollMonth** : In month view mode, you can scroll infinitely if you need to change pages, only if this option is enabled.

**ViewMode** : Change by code in the view mode (Day, Week, Month, Year)

### Events:

**DateBackColor** : When this event is triggered, we can choose the background color of the cell based on the date or time, as well as its **HatchStyle**.

**DateChange** : When the user changes the calendar page this event is fired.

**DragNewEvent** : When the user has the Control key pressed and drags & drops, the event is duplicated, that is, a new event is added by the user, with this event we are informed and thus we can save it in the database or as it is being driving.

**DropDownViewMore**: Event when the view more button is pressed, this button is in the "Months" view, it appears when there are many events in a day and they do not enter the cell, pressing this will change to view mode = "Day", We can cancel this and show a custom window with the events of the day, we can retrieve the events of the day with **GetEventsFromDay**.

**EventChangeDate** : This event fires when the user drags & drops or resizes the event with the mouse. it is useful for storing the new dates in the database.

**EventMouseEnter** : When the mouse enters an event.

**EventMouseLeave** : When the mouse leaves over the event.

**EventClick** : Click on an event.

**PreDateChange**: Before the user pages to another date, here we can cancel it.

**PreEventChangeDate** : Before the user changes the data of an event with the drag & drop or the mouse, this event is very useful to prevent dragging events on certain dates or times in which we do not want an event to be added.