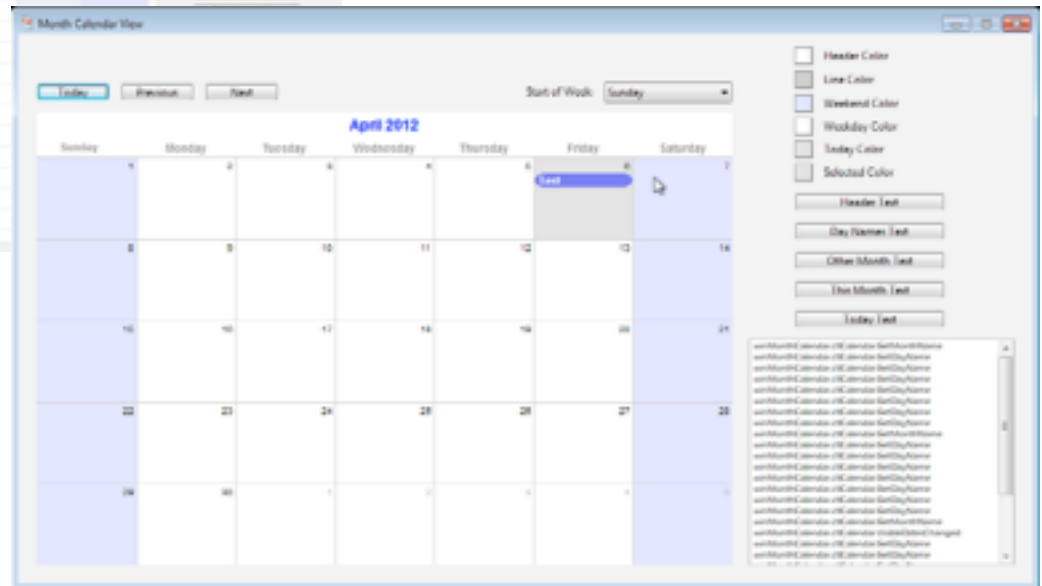
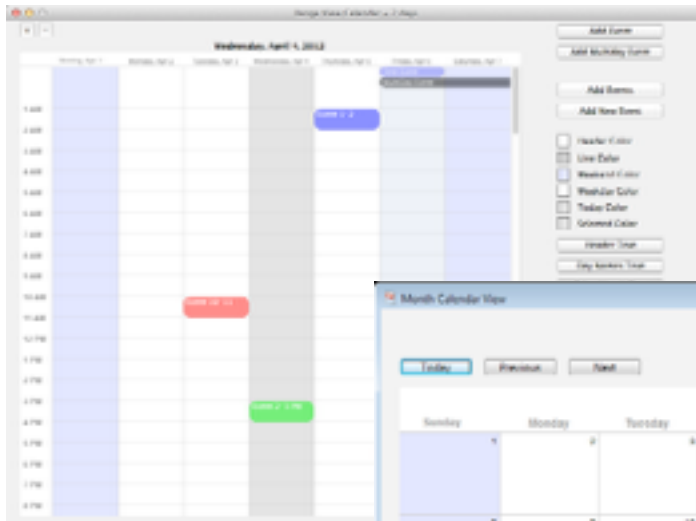


# BKS\_Calendar

Version 1.1.2



<b>About the Classes</b>	<b>4</b>
<b>Requirements</b>	<b>4</b>
<b>Adding the classes to your project</b>	<b>5</b>
<b>mConfig</b>	<b>6</b>
<i>mConfig Constants</i>	<i>6</i>
<i>mConfig Properties</i>	<i>6</i>
<i>BKS_CalendarBase</i>	<i>7</i>
<i>BKS_CalendarBase Events</i>	<i>7</i>
<i>Open</i>	<i>7</i>
<i>BKS_CalendarBase Methods</i>	<i>14</i>
<i>BKS_CalendarBase Properties</i>	<i>17</i>
<b>BKS_MonthCalendar</b>	<b>21</b>
<i>BKS_MonthCalendar Events</i>	<i>21</i>
<i>BKS_MonthCalendar Methods</i>	<i>22</i>
<i>BKS_MonthCalendar Properties</i>	<i>23</i>
<i>BKS_CalendarEvent</i>	<i>24</i>
<i>BKS_CalendarEvent Methods</i>	<i>25</i>
<i>BKS_CalendarEvent Properties</i>	<i>26</i>
<i>bAllDay</i>	<i>26</i>
<b>BKS_clsEventBlock</b>	<b>28</b>
<b>BKS_clsCalendarCell</b>	<b>28</b>
<b>BKS_TextProperties Class</b>	<b>29</b>
<i>BKS_TextProperties Methods</i>	<i>29</i>
<i>BKS_TextProperties Properties</i>	<i>30</i>

<b>BKS_DateExtends</b>	<b>32</b>
<b>BKS_PopupmenuExtends</b>	<b>34</b>
<b>BKSColorPicker</b>	<b>35</b>
<i>BKSColorPicker Events</i>	<i>35</i>
<i>BKSColorPicker Methods</i>	<i>35</i>
<i>BKSColorPicker Properties</i>	<i>35</i>
<b>Release Notes</b>	<b>36</b>

## About the Classes

BKS\_Calendar is a set of drop-in Real Studio and Xojo classes that provide a flexible set of calendar classes that provide variable view calendars. Daily, Weekly, and Monthly view calendars are available as well as being able to change the number of weeks you can view at any given time.

BKS\_Calendar will compile to all three desktop platforms supported by Real Studio and Xojo (Macintosh OS X, Windows, and Linux). It has been thoroughly tested on the Mac (both Carbon and Cocoa) and Windows. These classes do NOT work with Web Edition and we have no plans to do so (however if you purchase the full source code you could adopt some of this for use in Web Edition).

The BKS\_Calendar suite is comprised of three main classes. BKS\_CalendarBase is a canvas subclass, BKS\_MonthCalendar is a subclass of the BKS\_CalendarBase that functions only in Month View and, finally, the BKS\_CalendarEvent is how events are placed into the calendar.

There are also three supporting classes that are required to in your project as well. BKS\_clsEventBlock, BKS\_clsCalendarCell, and BKS\_TextProperties. Of the three, only BKS\_TextProperties is ever used by an end developer.

## Requirements

BKS\_Calendar will run on Macintosh OS X, Windows, and Linux versions of Real Studio versions 2010 R4 and better or Xojo 2013 Release 1 and better.

## Adding the classes to your project

To use BKS\_Calendar in a Real Studio project, you must add all of its required components to the 'Project Tag' in the Real Studio IDE. The easiest way to do this is to open the demo project in Real Studio and copy the "BKS\_Library" folder to your own project. This adds all required classes to your project.

To add a BKS\_Calendar to your project simply drag an instance of either BKS\_CalendarBase or BKS\_MonthCalendar to your window or dialog.

You may have to switch the Controls list PopupMenu to "Project Controls" to see these items. Alternatively, you can drag a generic Canvas control to your window or dialog and change its Super to that of BKS\_CalendarBase or BKS\_MonthCalendar.

The default colors and text properties will most likely not be what you want for your application so you'll want to become familiar with changing those settings.

If you have any problems regarding the use of the BKS\_Calendar controls or would like to make suggestions or file a bug report then please email us at [support@bkeeney.com](mailto:support@bkeeney.com).

# mConfig

## mConfig Constants

### kUseEinhugurLocale

When set to true the internal class will use the Einhugur Utils Library plugin to gather Month and Day names.

```
Const kUseEinhugurLocale = false
```

Note: If the user returns true from the GetMonthName or GetDayName event the class will use the Month and Day names the user supplies.

### kUseMBSLocale

When set to true the internal class will use the Monkeybread plugins to gather Month and Day names. In Windows it uses the WinLocalizationClass. In Mac OS X it uses the NSLocal and NSLocalData classes.

```
Const kUseMBSLocale = false
```

Note: If the user returns true from the GetMonthName or GetDayName event the class will use the Month and Day names the user supplies.

If these two constants are set to false the control will still use the GetMonthName and GetDayName events but if no value is returned by the user, the class will automatically use North American English for both days and months.

## mConfig Properties

### Interval

When set to something other than zero, this will cause drag events (either the entire event or start/stop drag changes) to correct to the nearest time interval. For example, if the drag results in a minutes of 33 and the interval is set to 15 minutes it will reset it to 30. If the resulting minutes increments to the next hour, the hour variable is automatically changed too.

This value may be changed at runtime.

# BKS\_CalendarBase

## BKS\_CalendarBase Events

### Open

The open event occurs after some basic initialization of the control.

**Event** Open()

### Close

The close event occurs during the destruction of the control. Normally you won't have to use this event but you might want to save some data though it would probably be better to do it sooner than the close event.

**Event** Close()

### Paint

The paint event is called just after the calendar is drawn allowing you to do your own drawing in the graphics object of the calendar.

**Event** Paint(g as Graphics)

### CellBackgroundPaint

Returning true from the CellBackGroundPaint prevents the control from doing its default drawing routine. A Cell is a DAY on the control.

**Event** CellBackgroundPaint(g as Graphics, dtm as Date) As boolean

Parameters:

g as graphics. The graphics object for the control.

dt as Date. The date of the cell.

Return value, Boolean. When true is returned the control will not draw that event.

Example Usage:

```
Function CellBackgroundPaint(g as Graphics, dtm as Date) As boolean
  if dt.DayOfWeek = 2 then
    g.ForeColor = &cFF0000
    g.FillRect 0, 0, g.Width, g.Height
```

```
    return true
end
End Function
```

### **CellDayPaint**

Returning true from the CellDayPaint event prevents the control from doing its default drawing routine. A Cell is a Day of the Week on the control.

This event is only called when more than one week is displayed.

**Event** CellDayPaint(g as Graphics, dtm as Date, x as integer, y as integer) As boolean

Parameters:

g as graphics. The graphics object of the control.

dt as date. The date of the cell.

x as integer. The x location the the cell text will be drawn.

y as integer. The y location of the cell text will be drawn.

Return value, Boolean. When true is returned, the control will not draw the cell.

Example Usage:

```
Function CellDayPaint(g as Graphics, dt as Date, x as integer, y as integer)
As boolean
```

```
    if dt.Day = 30 then
        g.ForeColor = &cFF0000
```

```
        g.drawstring "BOB", x, y
```

```
    return true
```

```
mConfig.Interval = 15
```

```
end
End Function
```



### **EventBackgroundPaint**

The EventBackgroundPaint event is called just before an event is drawn on the calendar. If the user returns true the control will not draw the background of the event and it is up to the user to do the drawing in the supplied graphics object.

**Event** EventBackgroundPaint(g **as** Graphics, oEvent **as** BKS\_CalendarEvent) **As boolean**

Parameters:

g as graphics. The graphics object of the control.

oEvent as BKS\_CalendarEvent. The event about to be drawn.

Return value, Boolean. Returning true will prevent the control from drawing the background of the event.

### **EventTextPaint**

The EventTextPaint event is called just before an event is drawn on the calendar. If the user returns true the control will not draw the background of the event and it is up to the user to do the drawing in the supplied graphics object.

**Event** EventTextPaint(g **as** Graphics, oEvent **as** BKS\_CalendarEvent) **As boolean**

Parameters:

g as graphics. The graphics object of the control.

oEvent as BKS\_CalendarEvent. The event about to be drawn.

Return value, Boolean. Returning true will prevent the control from drawing the text of the event.

### **EventToolTip**

The EventTooTip event is called when the cursors hovers over an event. The developer has the option of returning a string which is then displayed on the control next to the mouse.

`Event EventTextPaint(oEvent as BKS_CalendarEvent) As String`

Parameters:

oEvent as BKS\_CalendarEvent. The event the cursor is on.

Return value, String. Returning anything other than a blank string will put that text into the Tooltip.

### **EventClicked**

The EventClicked event occurs whenever the user clicks on the event.

`Event EventClicked(oEvent as BKS_CalendarEvent)`

Parameters:

oEvent as BKS\_CalendarEvent. The event that was clicked.

### **EventDoubleClicked**

The EventDoubleClicked event occurs whenever the user double clicks on the event.

`Event EventDoubleClicked(oEvent as BKS_CalendarEvent)`

Parameters:

oEvent as BKS\_CalendarEvent. The event that was clicked.

### **EventLayoutHeight**

The EventLayoutHeight event is called whenever the calendar control changes dimensions allows the user to override the default dimensions on the event it passes in. If the user returns true, the control will use the iEventHeight value.

`Event EventLayoutHeight(iCellWidth as integer, oEvent as BKS_CalendarEvent, byref iEventHeight as integer) As boolean`

Parameters:

iCellWidth as integer. The CellWidth the control has calculated.

oEvent as BKS\_CalendarEvent. The event about to be drawn.

iEventHeight as integer. The user supplied height for the event.

Return value, Boolean. Returning true will prevent the control from drawing the text of the event.

### **EventChanged**

The EventChanged event is called whenever an event is changed (move or start/stop changes) on the control.

```
Event EventChanged(oEvent as BKS_CalendarEvent)
```

Parameters:

oEvent as BKS\_CalendarEvent.

Note that there is no guarantee that there is actually a change in the event start/stop times. This event simply means it was moved.

### **EventMoved**

The EventMoved event is called whenever an event is dragged on the control. If the user returns true it is up to them to handle the movement.

```
Event EventMoved(oEvent as BKS_CalendarEvent,  
dtmDateEventMovedTo as Date) As boolean
```

Parameters:

oEvent as BKS\_CalendarEvent.

dtmDateEventMovedTo is the date the event was moved to. Note that this is NOT the event time.

Return value, Boolean. Returning true will prevent the control from moving the event to the new date.

### **DateChanged**

The DateChanged event occurs whenever the selected date changes.

```
Event DateChanged()
```

### **DateClicked**

The DateClicked event occurs after the MouseDown event is recognized.

```
Event DateClicked(dateClicked as date)
```

Parameters:

dateClicked as Date. The new date.

### **DateDoubleClicked**

The DateDoubleClicked event occurs after a double click mousedown event.

**Event** DateDoubleClicked(dateClicked **as** date, blnHeader **as** boolean)

Parameters:

dateClicked as Date. The new date.

blnHeader as boolean. The doubleclick occurred in the header.

### **GetDayName**

The GetDayName event occurs whenever the controls wants to draw the Day Of the Week name. If the user returns true the control uses the value specified by DayName.

**Event** GetDayName(DayOfWeek **as** integer, byref DayName **as** String)  
**As Boolean**

Parameters:

DayOfWeek as integer. The day of the week. 0 is Sunday, 6 is Saturday.

byRef DayName as string. Set the value of this string to be used by the control.

Return value, Boolean. Returning true tells the control to use the user supplied value.

### **GetHourName**

The GetHourName event occurs whenever the control wants to draw the Hour string. If the user returns true the control uses the value specified by DayName.

**Event** GetHourName(Hour **as** integer, byref Name **as** String) **As Boolean**

Parameters:

Hour as integer. The hour of the day. 0 is 00:00 and 23 is 23:00.

byRef Name as string. Set the value of this string to be used by the control.

Return value, Boolean. Returning true tells the control to use the user supplied value.

### **VisibleDatesChanged**

The VisibleDatesChanged event occurs whenever the dates in the calendar are changed.

<b>Event</b> VisibleDatesChanged()
------------------------------------

## BKS\_CalendarBase Methods

### **WeekCount**

To determine how many full weeks are displayed, call the WeekCount Function.

```
Function WeekCount() As integer
```

Returns integer.

### **Today**

To set the Selected Date to today, call this method.

```
Sub Today()
```

### **SelectedEvent**

To get the currently selected event, call this function. If no event selected a NIL object may be returned.

```
Function SelectedEvent() As BKS_CalendarEvent
```

### **SelectedDate**

To set the currently selected date, call this method and pass in the date you want.

```
Sub SelectedDate(assigns dtm as Date)
```

Parameters:

dtm as date. The date to set the selected date to.

To get the currently selected date, call this function. If no date is currently selected it may pass back a NIL object.

```
Function SelectedDate() As Date
```

### **SelectDateRange**

To select a range of dates, call this method and pass in the start and end dates. If the end date is less than or equal to the start date a runtime exception is thrown with the error message "Invalid Date Range".

```
Sub SelectDateRange(dtmStart as Date, dtmEnd as Date)
```

Parameters:

dtmStart as date. The start date of the new selection.

dtmEnd as date. The end date of the new selection.

### **Events**

Call the events function to get all of the events currently in the calendar.

```
Function Events() As BKS_CalendarEvent()
```

Return Value: Array of BKS\_CalendarEvent objects.

### **AddEvents**

To add an event to the calendar, call the AddEvent method and pass in a BKS\_CalendarEvent object. It will then refresh the calendar control.

```
Sub AddEvent(oEvent as BKS_CalendarEvent)
```

Parameters:

oEvent as BKS\_CalendarEvent. An event with all of the data set for the CalendarEvent.

### **ClearEvents**

Clears all events from the calendar and forces a redraw.

```
Sub ClearEvents()
```

### **ReplaceEvents**

If you wish to replace all of the events in the calendar, call this method and pass in an array of BKS\_CalendarEvent objects. This will replace all existing events. It will then refresh the calendar control.

```
Sub ReplaceEvents(aoEvent() as BKS_CalendarEvent)
```

Parameters:

aoEvent() as BKS\_CalendarEvent. An array of BKS\_CalendarEvent objects.

### **RemoveEvent**

To remove an event from the calendar, pass the BKS\_CalendarEvent into this method. This will remove the event and refresh the calendar control.

```
Sub RemoveEvent(oEvent as BKS_CalendarEvent)
```

**GetDayCount**

To determine how many days are currently being displayed on the calendar, call this function.

`Function GetDayCount() As Integer`

Returns integer.



## **BKS\_CalendarBase Properties**

### **cHeaderColor**

To change the header and side bar colors(in week and day mode) change this value.

cHeaderColor [As Color](#)

Read/Write. When this value is set it will refresh the calendar.

### **cLineColor**

To change the line colors in the calendar change this value.

cLineColor [As Color](#)

Read/Write. When this value is set it will refresh the calendar.

### **cSelectedFillColor**

To change the fill color of the selected day, change this value.

cSelectedFillColor [As Color](#)

Read/Write. When this value is set it will refresh the calendar.

### **cTodayFillColor**

To change the fill color of 'Today' on the calendar, change this value.

cTodayFillColor [As Color](#)

Read/Write. When this value is set it will refresh the calendar.

### **cWeekdayFillColor**

To change the fill color of weekday days in the calendar change this value.

cWeekDayFillColor [As Color](#)

Read/Write. When this value is set it will refresh the calendar.

### **cWeekEndFillColor**

To change the fill color of weekend days in the calendar change this value.

cWeekendFillColor [As Color](#)

Read/Write. When this value is set it will refresh the calendar.

**Format24Hour**

When true the calendar will display times in 24 hour format.

Format24Hour As Boolean

Read/Write. When this value is set it will refresh the calendar.

**oDayText**

To change the text properties of the Day text set this object. Changing this object will cause the calendar to refresh.

oDayText As BKS\_TextProperties

Read/Write. When this value is set it will refresh the calendar.

**oHeaderText**

To change the text properties of the Header text set this object. Changing this object will cause the calendar to refresh.

oHeaderText As BKS\_TextProperties

Read/Write. When this value is set it will refresh the calendar.

**oOtherMonthText**

To change the text properties of the cell for Other Month text set this object. Changing this object will cause the calendar to refresh.

oOtherMonthText As BKS\_TextProperties

Read/Write. When this value is set it will refresh the calendar.

**oThisMonthText**

To change the text properties of the cell text for This Month set this object. Changing this object will cause the calendar to refresh.

oThisMonthText As BKS\_TextProperties

Read/Write. When this value is set it will refresh the calendar.

**oTodayText**

To change the text properties of the cell for Today text set this object. Changing this object will cause the calendar to refresh.

oTodayText As BKS\_TextProperties

Read/Write. When this value is set it will refresh the calendar.

### **SelectedEndDate**

Return the selected end date.

SelectedEndDate **As** Date

Read Only.

### **SelectedStartDate**

Return the selected start date.

SelectedStartDate **As** Date

Read Only.

### **SingleDay**

Returns true if the calendar is showing a single day.

SingleDay **As** boolean

Read only.

### **StartOfWeek**

If you want to change the first day of the week to something other than Sunday change the value of the property.

StartOfWeek **As** Integer

Read/Write. When this value is set it will refresh the calendar.

The following values are accepted:

- 1 Sunday
- 2 Monday
- 3 Tuesday
- 4 Wednesday
- 5 Thursday
- 6 Friday
- 7 Saturday

If a value outside of 1 and 7 is passed in a Runtime Exception is generated with an error message of "Invalid Start of Week".

### **VerticalScrollBar**

If the calendar control should use a scrollbar assign the scrollbar it should use.

VerticalScrollbar [As](#) Scrollbar

Read/Write.

**VisibleEndDate**

Returns the visible end date being shown in the calendar.

VisibleEndDate [As](#) Date

Read only.

**VisibleStartDate**

Returns the visible start date being shown in the calendar.

VisibleStartDate [As](#) Date

Read only.

## BKS\_MonthCalendar

The BKS\_MonthCalendar is a subclass of the BKS\_CalendarBase and it is designed to only show a single month at a time. Since it is a subclass it inherits all of the methods, events, and properties of the BKS\_CalendarBase class. It does, however, have a few additional events, methods and properties.

### BKS\_MonthCalendar Events

#### GetMonthName

The GetMonthName event occurs whenever the controls wants to draw the Month name. If the user returns true the control uses the value specified by DayName.

**Event** GetMonthName(Month **as integer**, byref MonthName **as String**)  
**As Boolean**

Parameters:

Month as integer. The month of the year as..

byRef MonthName as string. Set the value of this string to be used by the control.

Return value, Boolean. Returning true tells the control to use the user supplied value.

#### MonthChanged

The MonthChanged event is fired whenever the month in the calendar is changed.

**Event** MonthChanged(dtm **as Date**)

Parameters:

dtm as date. The date of the month being displayed.

## **BKS\_MonthCalendar Methods**

### **NextMonth**

To go to the next month in the calendar, call this method.

```
Sub NextMonth()
```

This will cause the calendar to refresh.

### **PreviousMonth**

To go to the previous month in the calendar, call this method.

```
Sub PreviousMonth()
```

This will cause the calendar to refresh.

### **Today**

Calling the Today method will cause the calendar to display the current month.

```
Sub Today()
```

This will cause the calendar to refresh.

## **BKS\_MonthCalendar Properties**

### **SelectedMonth**

Returns the displayed month.

SelectedMonth <a href="#">As Integer</a>
--

Read Only. Gets the currently selected month number. The range of values is 1 to 12.

### **SelectedYear**

Returns the display year.

SelectedYear <a href="#">As Integer</a>
---

Read Only. Gets the currently selected year.

**BKS\_CalendarEvent**

The BKS\_CalendarEvent is a data class that contains the relevant data for the events that are added to the calendar.

You can subclass this object if you want more methods and properties than are supplied.

It has no events.



## BKS\_CalendarEvent Methods

### Constructor

To create a new instance of the CalendarEvent call its constructor.

```
Sub Constructor()
```

Using the default constructor will create an event with the current date/time.

Passing in a date will create an event with the passed in date.

```
Sub Constructor(dtm as Date)
```

Example Usage:

```
dim oEvent as BKS_CalendarEvent

oEvent = new BKS_CalendarEvent
oEvent.bDraggable = true
oEvent.dtmStart = new Date(dtToday.Year, dtToday.Month, dtToday.Day, 10, 15)
oEvent.sEventName = "Test"
oEvent.cEventColor = &c66FFFF00
oEvent.Tag = kTest
ctlCalendar.AddEvent(oEvent)
```

### Clone

To create a clone of an event, use the clone function to create an exact duplicate.

```
Function Clone() As BKS_CalendarEvent
```

Example Usage:

```
Dim oNewEvent as BKS_CalendarEvent
oNewEvent = oEvent.Clone
```

## **BKS\_CalendarEvent Properties**

### **bAllDay**

Set the bAllDay boolean to true to tell the calendar that the event is all day. If it is all day then any changes to hour, minutes, and seconds will have no effect on where the event is drawn on the calendar.

bAllDay [As boolean](#)

Read/Write. Boolean

### **bDraggable**

The bDraggable boolean tells the calendar whether to allow the user to drag the event around on the calendar.

bDraggable [As boolean](#)

Read/Write. Boolean

### **bSelected**

The bSelected boolean tells the user whether or not the event is selected.

bSelected [As Boolean](#)

Read Only.

### **cEventColor**

The cEventColor is the color that the event is filled with.

cEventColor [As Color](#)

Read/Write.

### **dtmEnd**

The end date/time of the event. If bAllDay is true, the hours, minutes, and seconds are ignored.

dtmEnd [As Date](#)

Read/Write.

### **dtmStart**

The start date/time of the event. If bAllDay is true, the hours, minutes, and seconds are ignored.

dtmStart As Date

Read/Write.

**iEventID**

The iEventID is an integer value the developer can use in whatever manner they want. It is not used internally.

iEventID As Integer

Read/Write.

**sEventName**

The sEventName is the name that is drawn on the calendar for the event.

sEventName As String

Read/Write.

**Tag**

The tag property is for the developer to use in whatever manner they want. It is not used internally.

Tag As Variant

Read/Write.

## **BKS\_clsEventBlock**

The BKS\_clsEventBlock is used internally by the other classes and should never be called directly or indirectly.

## **BKS\_clsCalendarCell**

The BKS\_clsCalendarCell is used internally by the other classes and should never be called directly or indirectly.

## BKS\_TextProperties Class

The BKS\_TextProperties is a data class that contains the properties to control and manipulate the various text displays in the calendar views.

### BKS\_TextProperties Methods

#### Constructor

Creates a new instance of BKS\_TextProperties class using the default properties.

Font = "Arial"

FontSize = 12

FontColor = &c00000000

Bold = false

Italic = false

Underline = false

Example usage:

```
Dim oText as New BKS_TextProperties
```

```
oText.sFont = "Arial"
```

```
oText.iFontSize = 14
```

**Constructor(Font as string, TextSize as integer, FontColor as color = &c000000, Bold as boolean = false, Italic as boolean = false, underline as boolean = false)**

Creates a new instance of the BKS\_TextProperties class using the passed in parameters.

#### Parameters

##### *Font*

The Font for the text object.

##### *FontSize*

The size of the font for the text object.

##### *FontColor*

The color of the font for the text object. If no parameter is passed in it defaults to black.

##### *Bold*

Set to true if the font is to be displayed in Bold form. This may not work with all font choices.

### *Italic*

Set the true if the font is to be display in Italic form. This may not work with all font choices.

### *Underline*

Set to true if the font is to be displayed in Underline form.

Example Usage:

```
Dim oText as New BKS_TextProperties("Arial", 18, &c000000, TRUE)
```

### **BKS\_TextProperties Properties**

#### **bBold**

**bBold as Boolean**

Read-write. Set this property to set the bold property of the text.

#### **bItalic**

**bItalic as Boolean**

Read-write. Set this property to set the italic property of the text.

#### **bUnderline**

**bUnderline as Boolean**

Read-write. Set this property to set the underline property of the text.

#### **cFontColor**

**cFontColor as color**

Read-write. Set this property to set the font color property of the text.

#### **iTextSize**

**iTextSize as integer**

Read-write. Set this property to set the text size property of the text.

## **sFont**

<b>sFont as string</b>
------------------------

Read-write. Set this property to set the Font property of the text.

## BKS\_DateExtends

The BKS\_DateExtends is a required module that contains publicly available methods to extend the built-in REALbasic Date class. You may use any of these functions in your projects.

### AddDays

**Function** AddDays(Extends dt **As** Date, DayCount **As** Integer) **As** Date

Adds the number of day to the extended date and returns the new date.

Parameters:

DayCount as Integer. The number of days to add to the date.

Returns:

Date object

### GetNextDay

**Function** GetNextDay(Extends dt **As** Date) **As** Date

Returns the next day of the extended date.

Parameters:

None

Returns:

Date object

### GetNextMonth

**Function** GetNextMonth(Extends dt **As** Date) **As** Date

Returns the next month of the extended date.

Parameters:

None

Returns:

Date object

### GetPreviousDay



<b>Function</b> GetPreviousDay(Extends dt As Date) As Date
--

Returns the previous day of the extended date.

Parameters:  
None

Returns:  
Date object

### **GetPreviousMonth**

<b>Function</b> GetPreviousMonth(Extends dt As Date) As Date
--

Returns the previous month of the extended date.

Parameters:  
None

Returns:  
Date object

## BKS\_PopupmenuExtends

The BKS\_PopupMenuExtends is an optional module that contains publicly accessible methods to the extend the built-in REALbasic PopupMenu control. You may use any of these functions in your projects.

### SetText

`Sub SetText(extends pm as PopupMenu, assigns s as String)`

Sets the text of the extended popup menu by iterating through its List array and when it matches the assigned string (s) it sets the list index. If no match is found the listindex is NOT set and no error is generated.

Parameters:

s as string. The string to set the popup menu to.

Returns:

nothing

## BKSColorPicker

BKSColorPicker is a subclassed Canvas control that is a way to display and invoke the Color Picker dialog. You may use this control in your projects.

### BKSColorPicker Events

#### ColorChange

`Event ColorChange()`

This event occurs when the user has selected a new color.

Parameters:

None

### BKSColorPicker Methods

#### SetColor

`Sub SetColor(assigns inColor as color)`

Sets the color of the control.

Parameters:

InColor as color. The color to draw in the control.

Returns:

Nothing

### BKSColorPicker Properties

#### c

`c As color = &c000000`

The color setting of the control. The default color is black (&c000000).

# Release Notes

## **Version 1.1.2 (May, 2014)**

- Added ToolTips to Events
- Added EventTooltip event to base calendar
- Changed the Event (GUI) handle size so it's easier to grab

## **Version 1.1.1 (April, 2014) - Not made public**

- Minor bug fixes.

## **Version 1.1.0 (March, 2014)**

- Made Retina Ready (Mac OS X only)
- Events no longer overlap
- Added ClearEvents method
- Added Constant, kVersion to BKS\_CalendarBase
- Can you drag events to different times
- Added a time interval to mConfig to constrain time drags and start/stop drag changes
- Changed EventMoved event parameter to make it more clear
- Added ScrollToTime method to get the calendar to scroll to that time
- Added a new EventChanged event to tell the user that the event may have changed due to a time drag (move or change start/stop time)

## **Version 1.0.3 (July 8, 2013)**

- Events that spanned more than one week weren't displaying correctly.
- Added GetHourName event so users can use their own localization.

## **Version 1.0.2 (February 27, 2013)**

- Fixed a compile error when mConfig.kUseMBSLocale was set to true.
- Added Format24Hour to BKS\_CalendarBase. When true times are shown in 24 hour format.

## **Version 1.0.1 (November 8, 2012)**

- Moved mConfig module to the top of the BKS\_Calendar folder.
- Now set mConfig.kUseEinhugurLocale = false by default.
- Fixed a code error in BKS\_CalendarBase when neither MonkeyBread or Einhugur plugins are installed.
- Changed a deprecated RS procedure call.
- Updated the installation procedure in this document.

## **Version 1.0.0 (July 11, 2012)**

- Events in Week and Day views that overlap are now offset so each is selectable.
- User can now select and drag start and end times of events in Week and Day views.
- User contributed fix. In Single Day or Single Week view, events are no longer clipped.

## **Version 0.9.2 (June 7, 2012)**

- Fixed the Today Color
- Fixed the EventClicked event that didn't fire when the event was draggable.

**Version 0.9.1 (April 15, 2012)**

- Removed a MonkeyBread Software dependency that kept it from compiling if MBS plugins not installed.

**Version 0.9 (April 1, 2011)**

- First public release