

TAdrock3dCalendar Contents

Help File last updated: 18th January 1996

[Properties](#)

[DateMethods](#)

[CalendarMethods](#)

[Events](#)

[OwnerDraw](#)

[Registration](#)

The TAdrock3dCalendar contains close on 100 properties, 10 calendar specific methods, 6 events, and what every programmer really wants lots of date routines. There are over 60 methods covering everything from the number of days, weeks, months, quarters, and years two dates are apart to adding or subtracting any number of days, weeks, months, years to a date. You can determine what day of the week a date falls on using any starting day of the week, and more.

There is support for Drag and Drop to and from the calendar, OwnerDrawing, ButtonsBars, Month and Year Combos / Labels, 3D Fonts, 3D Bevels, Keyboard And/Or Mouse Support, and other controls can be placed on it to seamlessly create stunning applications.



The near 100 properties control all aspects of the calendar, including Colors, Fonts, Sizing, 3D Text Fonts, 3D Bevels and every other feature of the calendar.

Day, Month and Year onChange events, and a OnDateChange event are available so you know exactly what is happening on the calendar.

You can show dates in the previous and next months, and there are different mouse selection properties.

The 10 Calendar specific methods allow you to goto a specific date, Goto Today, Previous / Next Day, Previous / Next Week, Previous / Next Month, and Previous / Next Year.

onDateChange

This event gets called when the date on the calendar is changed.

The event gets triggered by setting a date that it not the current date though the property Date, or by selecting a new date on the calendar.

This event would be of most use to people.

See Also : [OnDayChange](#), [OnWeekChange](#), [OnMonthChange](#), [OnYearChange](#).

OnDayChange

This event gets triggered when the day of the month changes.

Example : If the current day is the first of the month and the user goes to the next day by selecting it with a mouse or the keyboard then this event will be triggered.

See Also : [OnWeekChange](#), [OnMonthChange](#), [OnYearChange](#), [OnDateChange](#)

OnMonthChange

This event gets triggered when the month of the year changes

Example : If the month changes from January to February this event will be triggered.

Note : The OnDateChange event also gets called when the month changes on the calendar..

See Also : [OnDayChange](#), [OnWeekChange](#), [OnYearChange](#).

OnYearChange

This event gets triggered when the year changes.

Example : If the year changes from 1995 to 1996 this event will be triggered.

Note : The OnDateChange event also gets called when the year changes.

See Also : OnDayChange, OnWeekChange, OnMonthChange.

OnWeekChange

This event gets triggered when the week of the month changes.

Example : If the current day is the first of the month and the user goes to the next week by selecting it with a mouse or the keyboard then this event will be triggered.

Note : The OnDateChange event also gets called when the Week changes.

See Also : [OnDayChange](#), [OnMonthChange](#), [OnYearChange](#).

Properties

<u>Align</u>	<u>Bevel</u>	<u>BevelWidth</u>
<u>ButtonsFont</u>	<u>ButtonsHeight</u>	<u>Canvas</u>
<u>CellBevel</u>	<u>CellBevelWidth</u>	<u>CellFont</u>
<u>CellSpaceX</u>	<u>CellSpaceY</u>	<u>CellsSurroundBevel</u>
<u>CellsSurroundBevelWidth</u>	<u>CellTextHorizontalAlign</u>	
	<u>CellTextVerticalAlign</u>	
<u>ColorBack</u>	<u>ColorBetweenCellsBackColor</u>	<u>ColorCellsBack</u>
<u>ColorCellsHighlight</u>	<u>ColorCellsShadow</u>	<u>ColorCombosBack</u>
<u>ColorMonFri</u>	<u>ColorPrevNextMonth</u>	
	<u>ColorPrevNextMonthBackColor</u>	<u>ColorSaturday</u>
	<u>ColorSelectedBackColor</u>	<u>ColorSunday</u>
	<u>ColorWeekDayHeaderBackColor</u>	
	<u>ColorWeekDayHeaderMonFri</u>	
	<u>ColorWeekDayHeaderSaturday</u>	
	<u>ColorWeekDayHeaderSunday</u>	
	<u>Cursor</u>	<u>CombosFont</u>
	<u>CursorNonSelectable</u>	<u>CursorSelectable</u>
<u>CursorClick</u>	<u>Day</u>	<u>DragCursor</u>
<u>Date</u>		
<u>DragMode</u>	<u>Height</u>	<u>HelpContext</u>
Enabled	<u>Left</u>	<u>Month</u>
Hint	<u>MonthLabelFont</u>	<u>MouseMode</u>
<u>MonthButtonPercentage</u>	<u>NextMonthCaption</u>	<u>NextYearCaption</u>
Name	<u>PreviousYearCaption</u>	<u>SelectedCellFont</u>
<u>PreviousMonthCaption</u>	<u>SelectPreviousNextMonthDate</u>	<u>ShowButtonBar</u>
<u>SelectMode</u>	<u>ShowOnCalendar</u>	
<u>ShowCustomCursors</u>	<u>ShowPrevNextMonth</u>	<u>ShowSaturdayColor</u>
	<u>ShowSundayColor</u>	
	<u>ShowWeekDayHeader</u>	
<u>SpaceAtBottom</u>	<u>SpaceAtLeft</u>	<u>SpaceAtTop</u>
	<u>SpaceBetweenButtonsAndBottom</u>	
	<u>SpaceBetweenCellsAndButtons</u>	
	<u>SpaceBetweenLabelsAndWeekDayHeader</u>	
	<u>SpaceBetweenTopAndLabels</u>	
	<u>SpaceBetweenWeekDayHeaderAndCells</u>	<u>SpaceFromSides</u>
<u>StartOfWeek</u>	<u>Style</u>	<u>TabStop</u>
Tag	<u>Text3dCells</u>	<u>Text3dHighlight</u>
<u>Text3dWeekDayHeader</u>	Top	Visible
<u>WeekDayHeaderBevel</u>	<u>WeekDayHeaderBevelWidth</u>	
	<u>WeekDayHeaderDisplayLen</u>	
	<u>WeekDayHeaderFont</u>	
	<u>WeekDayHeaderTextAlign</u>	<u>Width</u>
<u>Year</u>	<u>YearLabelFont</u>	

Properties above which are not highlighted are standard delphi properties and HELP is not shown here for them here..

Events

<u>OnDayChange</u>	Any change of the day of the month.
<u>OnWeekChange</u>	Any change of the week.
<u>OnMonthChange</u>	Any change of the Month.
<u>OnYearChange</u>	Any change of the Year
<u>OnDateChange</u>	Any change of the date at all.

Registration

Option #1 **\$20.00** U.S. for unlimited use licence for the component, this does **not** include source code.

Option #2 **\$75.00** U.S. for unlimited use licence and complete source code for the component and date routines.

Upgrade **\$55.00** U.S. upgrade from Option #1 to Option #2. (**You must have registered Option #1 previously**)

Use the Registration Form below for mail-order registrations via Internet, Fax, Phone or standard Mail. We accept VISA, MASTERCARD and International Money Orders. Your Delphi control will be Emailed to you.

Please provide a clearly legible Internet or CompuServe address. We do not post disks.

Adrock Software Shareware Registration Form

Your name: _____

Your address: _____
(email only)

Country: _____

Payment accepted by VISA, MASTERCARD, or international money order only.

If you are ordering from a country other than New Zealand, please DO NOT send checks drawn on non New Zealand banks. Use a cashier's check or international money order valued in "U.S. DOLLARS".

If you do not wish to send your VISA or MASTERCARD number to us via CompuServe or Internet you can fax your order or you can place your order by phoning me personally.

The fax and voice phone number is: 64-3-365-1112. Please check our time zone here in Kiwiland - we're right on the International date line (the first to see light of the day:-), 12hrs < GMT. The 24hr line always accepts fax calls and will record your voice message when unattended.

Bank draft/International money order...

can be sent to:

Adrock Software

Byte Computer and Software Ltd

P.O. Box 13-155

Christchurch

New Zealand

attn: Calendar Component

VISA/MASTERCARD...

Your name on card: _____

Number: _____
(xxxx-xxx-xxx-xxx)

Expiry Date: _____
(mm/yy)

Payment by VISA, MASTERCARD or money order will be processed before we Email you the component - please don't forget your Email address. International postal delivery is not supported at this time due to the cost (I feel this component is *cheap as* at it's current price - and I want to keep it that way! - I'm sure you do to).

Ordering by Compuserves Online Shareware Registration System (SWREG) is the best and quickest method to get your registered component. (go SWREG and search the keyword ADROCK to find our stuff.)

Regards, Duncan McCormack, <CIS: 100353,110 Int: dhmcc@ibm.net>

BYTE Computer & Software Ltd
132 Peterborough Street (don't post to here though)
+64 3 365-1112 (voice/fax)
P.O Box 13-155 (postal address)
Christchurch
New Zealand

P.S. Please enclose details of your Visa/Mastercard, with expiry date, or send International Money Order if ordering directly.

Today

Declaration

Procedure Today;

Description

This method will force the calendar to reset itself to today (The current date).

DateMethods

Calendar Methods

Today - Jump to Today DateString - Return the current calendar date as a string

NextDay - Jump to the Next Day PrevDay - Jump to the Previous Day
NextWeek - Jump to the Next Week PrevWeek - Jump to the Previous Week
NextMonth - Jump to the Next Month PrevMonth - Jump to the Previous month
NextYear - Jump to the next Year PrevYear - Jump to the Previous Year

GotoMonth - Goto a Specific Month GotoYear - Goto a Specific Year

IsIndexInCurrentCalendarMonth(nIndex : Integer)
GetIndexFromPoint(nLeft : Integer ; nTop : Integer)
GetDateFromIndex(Index : Integer)
GetDayFromIndex(Index : Integer)
GetIndexFromDay(Day : Integer)

Year

This property holds the current year that the calendar is displaying. It is a read / write property which means you can set another year by entering the year in this property.

See Also [DateString](#), [Day](#), [Month](#)

Examples

```
{ The example below sets the current year to 1999 }  
procedure TForm1.BitBtn1Click(Sender: TObject);  
begin  
    Adrock3dCalendar1.Year := 1999;  
end;
```

Month

This property holds the current month that the calendar is displaying. It is a read / write property which means you can set another month by entering the month number in this property. 1=January

See Also [Date](#), [Day](#), [Month](#), [Year](#)

Examples

{ The example below sets the current month to 1, January }

```
procedure TForm1.BitBtn1Click(Sender: TObject);  
begin  
    Adrock3dCalendar1.Month := 1;  
end;
```

Day

This property holds the current day that the calendar is displaying. It is a read / write property which means you can set another day in the current month / year by entering the day number in this property.

See Also [Date](#), [Month](#), [Year](#)

Examples

{ The example below sets the current day to 5 }

```
procedure TForm1.BitBtn1Click(Sender: TObject);  
begin  
    Adrock3dCalendar1.Day := 5;  
end;
```

BevelWidth

This property sets the bevel width for the calendar border. This property is ignored if the property bevel is set to bvNone

Example: TAdrock3dCalendar1.BevelWidth := 2;



The calendar above is set to bvRaised and its bevel width is set to 2.

See Also : [Bevel](#)

Bevel

This property defines the calendar bevel, this bevel goes around the edge of the calendar.

bvNone No bevel is drawn.
bvRecessed The bevel is drawn recessed (Sunken)
bvRaised The bevel is drawn raised.

Example: TAdrock3dCalendar1.Bevel := bvRaised;



The calendar above is set to bvRaised and its bevel width is set to 2.

See Also : [BevelWidth](#)

Canvas

This property is a direct link to the canvas for the calendar. It is really only useful when the [Style](#) property is set to Owner Draw.

Once the control is repainted anything you drew on the canvas will be lost, therefore it is really only useful for Owner Draw while updating the cell that requires painting in the owner draw method.

See Also : [Style](#), [OnDrawCell](#)

Colors

There are a number of color settings

SpaceBetweenLabelsAndWeekDayHeader
SpaceBetweenTopAndLabels

SpaceBetweenWeekDayHeaderAndCells
SpaceFromSides

CellBevel

This property defines the bevel that goes around every day cell in the calendar.

bvNone	No bevel is drawn.
bvRecessed	The bevel is drawn recessed (Sunken)
bvRaised	The bevel is drawn raised.

Example: `TAdrock3dCalendar1.CellBevel := bvNone;`

29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

This bevel is set to `bvRecessed` and its bevel width is set to 1.

See Also : [CellBevelWidth](#)

CellBevelWidth

This property sets the bevel width for the cell bevel. This property is ignored if the property [CellBevel](#) is set to [bvNone](#)

Example: `TAdrock3dCalendar1.CellBevelWidth := 2;`

29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

This bevel is set to [bvRecessed](#) and its bevel width is set to 1.

See Also : [CellBevel](#)

WeekDayHeaderBevel

This property defines the Week Day Header Bevel, this bevel goes around the Week Day Names

bvNone	No bevel is drawn.
bvRecessed	The bevel is drawn recessed (Sunken)
bvRaised	The bevel is drawn raised.

Example: TAdrock3dCalendar1.WeekDayBevel := bvNone;



See Also : [WeekDayBevelWidth](#)

WeekDayHeaderBevelWidth




This property sets the bevel width for the WeekDayHeader. This property is ignored if the property WeekDayHeaderBevel is set to byNone

Example: TAdrock3dCalendar1.WeekDayBevelWidth := 2;



CellTextHorizontalAlign




This property defines the horizontal alignment of the days that are drawn into the day cells.

htLeft	The text is aligned to the left	
htCenter	The text is aligned to the center	
htRight	The text is aligned to the right	

See Also : [CellTextVerticalAlign](#)

CellTextVerticalAlign

This property defines the vertical alignment of the day that are drawn into the day cells.

vtTop	The text is aligned to the top	
vtCenter	The text is aligned to the center	
vtBottom	The text is aligned to the bottom	

See Also : [CellTextHorizontalAlign](#)

AllowSpeedKeys

The TabStop property is used to enable or disable the ability of the control to accept the input focus.

TabStop

This property control if the calendar can accept the input focus.

If Enabled then the calendar can accept the input focus to select dates. This means that a user can tab to the calendar as you would any other control. When the user tabs to the control or selects it with the mouse the current day will display a focus rectangle. The user can then use UpArrow, DownArrow, LeftArrow, and Right Arrow to navigate the days in the calendar. The PageUp, PageDown keys are used to navigate the months, and Ctrl-PageUp, CtrlPageDown are used to navigate the years.

If Disabled the user can not tab to the control, and selecting a date on the calendar will not move the input focus to the calendar but will still select the date. The input focus will stay on the control that had it before you clicked the day on the calendar.

Note: If you are displaying Combo Box's on the calendar these will always receive the input focus even if TabStop if set to FALSE.

ColorMonFri

This property defines the foreground text color for days that fall in the current month on Monday through Friday.



The ColorMonFri is set to clBlue

See Also : [ColorBetweenCellsBackColor](#), [ColorCellsBack](#), [ColorCellsShadow](#), [ColorPrevNextMonth](#), [ColorPrevNextMonthBackColor](#), [ColorSaturday](#), [ColorSelectedBackColor](#), [ColorSunday](#)

ButtonsFont

This property defines the Font, Size, and Color of the text for the button bar at the bottom of the calendar.



The buttons above are using a Times New Roman, 10 Point Bold Font, drawn in a Maroon color.

See Also : [ButtonHeight](#)

ButtonHeight

This property defines the height of the buttons in the button bar.



The buttons above are using a Times New Roman, 10 Point Bold Font, drawn in Maroon, and a height of 25.

See Also : [ButtonsFont](#)

AllowDateSelection

DragMode

This property defines whether or not the control will allow dates to be dragged from the calendar to other controls.

Setting this property to `dmEnabled` as well as setting MouseMode to `cmClickAndMoveAroundCells` can cause strange results as the control thinks that it should be dragging a date and not just moving around the calendar selecting cells.

MouseMove

This property controls how the mouse selects can be used to select days in the calendar.

cmClickAndMoveAroundCells - The user can click and while the mouse is still down move and select other days. Depending on the property SelectMode a date change event will be triggered either when the user releases the key or while the user moves around the cells.

cmClickOnCell - The user can click on a cell, moving the mouse around does NOT select any other cell.

See Also : [SelectMode](#)

Date

This property holds the current date that the calendar is displaying. It is a read / write property which means you can set another date by entering the date in this property. If you set a date that is not in the current month then the calendar month is re-drawn to show the new date.

See Also [DateString](#), [Day](#), [Month](#), [Year](#)

Examples

{ The example below display the current date that the calendar is displaying. }

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
    ShowMessage('The current date is : '+DateToStr(Adrock3dCalendar1.Date));
end;
```

{ The example below sets the calendar date. }

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
    Adrock3dCalendar1.Date := StrToDate('01/01/95');
end;
```

Note : You must add SysUtils to the uses clause to use the DateToStr and StrToDate functions.

DateString

This method will return the Date as a string, the string returned is formatted using the LongDateFormat.

Examples

{ The example below display the current date that the calendar is displaying. }

```
procedure TForm1.BitBtn1Click(Sender: TObject);  
begin  
    ShowMessage('The current date is : '+Adrock3dCalendar1.DateString);  
end;
```

See Also [Date](#), [Day](#), [Month](#), [Year](#)

CellFont

This property defines the font to use in the cells for the days of the month.

The color part of the font does not get used because of other properties that define the colors.



The above example CellFont is using a System Bold 10 Point font.

See Also : [SelectedCellFont](#), [ColorMonFri](#), [ColorSaturday](#), [ColorSunday](#), [ShowSaturdayColor](#), [ShowSundayColor](#)

ShowButtonBar

This property defines if the button bar is visible or not.

CellSpaceX

This property defines the horizontal space between each cell in the calendar days.

28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

The CellSpaceX, and the CellSpaceY in the above is set to 6

28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

The CellSpaceX, and the CellSpaceY in the above is set to 0

See Also : [CellSpaceY](#)

CellSpaceY

This property defines the vertical space between each cell in the calendar days.

28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

The CellSpaceX, and the CellSpaceY in the above is set to 6

28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

The CellSpaceX, and the CellSpaceY in the above is set to 0

See Also : [CellSpaceX](#)

CellsSurroundBevel

This property defines the bevel that is used to draw around the calendar cells.



28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

The example above has a raised bevel with a bevel width of 6.

See Also : [CellsSurroundBevelWidth](#)

CellsSurroundBevelWidth

This property defines the width of the bevel around the cells.



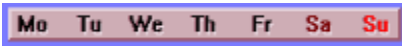
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

The above bevel is set to 6, and the bevel is Raised.

See Also : [CellsSurroundBevel](#)

ColorWeekDayHeaderBackColor

This property defines the background color for the week day header.



CombosFont

This property sets the Font Name, Font Size, and foreground color for the combo box's if they are being displayed.

When you change the font for the combo box's the combos grow or shrink as needed to display the font. The rest of the calendar also adjusts its size for the new combo font.



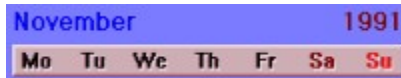
The combos above are using a System Bold 10 point font, and Black color.

See Also : [ShowOnCalendar](#), [ColorCombosBack](#)

ShowOnCalendar

This property defines what is displayed on the Calendar above the week day header.

scNothing	Nothing is displayed above the weekday header.
scCombos	Combo box's are displayed, the user can choose a month, or year from the combo box's.
scLabels	Month and Year labels are displayed.



The above example is displaying Labels: ShowOnCalendar := scLabels;



The above example is displaying Month and Year combo box's: ShowOnCalendar := scCombos;

Note: Even if TabStop is disabled, if you have combos displayed, the combos will accept the input focus even though the actual calendar days will not.

ColorCombosBack

This property sets the background color for the combo box's.



In the above example the ColorCombosBack is set to clWhite;

See Also : [CombosFont](#), [ColorBack](#), [ColorBetweenCellsBackColor](#), [ColorCellsHighlight](#), [ColorCellsShadow](#), [ColorMonFri](#), [ColorPrevNextMonth](#), [ColorPrevNextMonthBackColor](#), [ColorSaturday](#), [ColorSelectedBackColor](#), [ColorSunday](#), [ShowOnCalendar](#)

ShowPrevNextMonth

If this property is set to true the calendar will fill any dates that are not in the current calendar month with dates from the previous and next months.

28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

ShowPrevNextMonth := TRUE;


				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

ShowPrevNextMonth := FALSE;

See Also : [SelectPreviousNextMonthDate](#)

ShowWeekDayHeader

If this property is set to TRUE then the calendar will display a week day header.



Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

ShowWeekDayHeader := TRUE;

See Also : [WeekDayHeaderFont](#), [WeekDayHeaderBevel](#), [WeekDayHeaderBevelWidth](#),
[WeekDayHeaderDisplayLen](#),
[WeekDayHeaderTextAlign](#)

WeekDayHeaderFont

This property defines the font that is used to draw the WeekDayHeader Weekday Names.



Su Mo Tu We Th Fr Sa

In the example above the WeekDayHeaderFont is set to Bold System 10 Point and it is also recessed.

See Also : [Text3dWeekDayHeader](#), [ColorWeekDayHeaderMonFri](#),
[ColorWeekDayHeaderSaturday](#), [ColorWeekDayHeaderSunday](#)

WeekDayHeaderDisplayLen

This property sets the display length for the week day header weekday names.



In the example above the WeekDayHeaderDisplayLen is set to 2. ie : Su, Mo, etc..

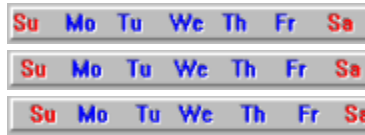
WeekDayHeaderTextAlign

This property sets the alignment of the text in the week day header.

htLeft The text is aligned to the left

htCenter The text is aligned to the center

htRight The text is aligned to the right



See Also : [WeekDayHeaderFont](#)

ColorCellsBack

This property defines the background color for the cells that make up the current month. Not the color of the cells in the previous/next month, and not the back color of the selected day.



ColorCellsBack is set to clSilver in the above example.

See Also : [ColorBack](#), [ColorBetweenCellsBackColor](#), [ColorCellsHighlight](#), [ColorCellsShadow](#), [ColorMonFri](#), [ColorPrevNextMonth](#), [ColorPrevNextMonthBackColor](#), [ColorSaturday](#), [ColorSelectedBackColor](#), [ColorSunday](#)

ColorBack

This property defines the background color for the calendar.



The background color is set to cSilver

See Also : [ColorBetweenCellsBackColor](#), [ColorCellsBack](#), [ColorCellsHighlight](#), [ColorCellsShadow](#), [ColorMonFri](#), [ColorPrevNextMonth](#), [ColorPrevNextMonthBackColor](#), [ColorSaturday](#), [ColorSelectedBackColor](#), [ColorSunday](#)

ColorBetweenCellsBackColor

This property defines the color of the space between each cell in the calendar days.



28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

In the above example the ColorBetweenCellsBackColor is set to clBLUE;

See Also : [CellSpaceX](#), [CellSpaceY](#)

ColorCellsHighlight

This property defines the highlight color that is used to draw the border around the cells for the days in the month.



The ColorCellsHighlight is set to clbtnHighlight

See Also : [ColorBetweenCellsBackColor](#), [ColorCellsBack](#), [ColorCellsShadow](#), [ColorMonFri](#), [ColorPrevNextMonth](#), [ColorPrevNextMonthBackColor](#), [ColorSaturday](#), [ColorSelectedBackColor](#), [ColorSunday](#)

ColorCellsShadow

This property defines the shadow color that is used to draw the border around the cells for the days in the month.



The ColorCellsShadow is set to clBtnShadow

See Also : [ColorBetweenCellsBackColor](#), [ColorCellsBack](#), [ColorCellsHighlight](#), [ColorMonFri](#), [ColorPrevNextMonth](#), [ColorPrevNextMonthBackColor](#), [ColorSaturday](#), [ColorSelectedBackColor](#), [ColorSunday](#)

ColorPrevNextMonth

This property defines the foreground text color that is used to draw days that are not in the current month.



The ColorPrevNextMonth is set to clGray

See Also : [ColorPrevNextMonthBackColor](#), [ColorBetweenCellsBackColor](#), [ColorCellsBack](#), [ColorCellsShadow](#), [ColorMonFri](#), [ColorPrevNextMonth](#), [ColorSaturday](#), [ColorSelectedBackColor](#), [ColorSunday](#)

ColorPrevNextMonthBackColor

This property defines the background color for the days that are not in the current month.



The ColorPrevNextMonthBackColor is set to clSilver

See Also : [ColorBetweenCellsBackColor](#), [ColorCellsBack](#), [ColorCellsShadow](#), [ColorMonFri](#), [ColorPrevNextMonth](#), [ColorSaturday](#), [ColorSelectedBackColor](#), [ColorSunday](#)

ColorSaturday

This property defines the foreground text color for days that fall on a saturday in the current month. This property only gets used if the [ShowSaturdayColor](#) property is set to TRUE.



The ColorSaturday is set to clRed

See Also : [ColorBetweenCellsBackColor](#), [ColorCellsBack](#), [ColorCellsShadow](#), [ColorMonFri](#), [ColorPrevNextMonth](#), [ColorPrevNextMonthBackColor](#), [ColorSelectedBackColor](#), [ColorSunday](#)

ColorSelectedBackColor

ColorSunday

This property defines the foreground text color for days that fall on a Sunday in the current month. This property only gets used if the [ShowSundayColor](#) property is set to TRUE.



The ColorSunday is set to clRed

See Also : [ColorSaturday](#), [ColorBetweenCellsBackColor](#), [ColorCellsBack](#), [ColorCellsShadow](#), [ColorMonFri](#), [ColorPrevNextMonth](#), [ColorPrevNextMonthBackColor](#), [ColorSelectedBackColor](#)

MonthButtonPercentage

This property defines the percentage of the weekday header width to allocate to the month buttons in the button bar.



The month buttons above are set to 65%, which means that the year buttons will be 35% of the weekday header width.



The month buttons above are set to 45%, which means that the year buttons will be 55% of the weekday header width.

NextDay

Declaration

Procedure NextDay;

Description

This method will force the calendar to make the Next Day the current day, redrawing the calendar as needed.

YearLabelFont

This property sets the Font Type, Font Size, and Color for the Year Label.



The image shows a calendar for November 1995. The year '1995' is displayed in the top right corner in a bold, black font. The days of the week are listed at the top: Su, Mo, Tu, We, Th, Fr, Sa. The dates are arranged in a grid, with the 30th of November being the last day of the month. The font used for the year label is System Bold 10 Point Black.

November							1995
Su	Mo	Tu	We	Th	Fr	Sa	
29	30	31	1	2	3	4	
5	6	7	8	9	10	11	
12	13	14	15	16	17	18	
19	20	21	22	23	24	25	
26	27	28	29	30	1	2	
3	4	5	6	7	8	9	

In the example above the YearLabelFont is set to System Bold 10 Point Black.

See Also : [MonthLabelFont](#)

SelectedCellFont

This property defines the font type, size, and color of the selected day in the calendar.



The selected day above is using a System Bold 10 point font.

See Also : [CellFont](#), [CombosFont](#), [MonthLabelFont](#), [YearLabelFont](#)

MonthLabelFont

This property sets the Font Type, Font Size, and Color for the month label.



The image shows a calendar for November 1995. The month label 'November' and the year label '1995' are displayed in a bold, black font. The days of the week are labeled 'Su', 'Mo', 'Tu', 'We', 'Th', 'Fr', and 'Sa' in a blue font. The dates are displayed in a black font, with the 30th of the month being bolded.

November							1995
Su	Mo	Tu	We	Th	Fr	Sa	
29	30	31	1	2	3	4	
5	6	7	8	9	10	11	
12	13	14	15	16	17	18	
19	20	21	22	23	24	25	
26	27	28	29	30	1	2	
3	4	5	6	7	8	9	

In the example above the MonthLabelFont is set to System Bold 10 Point Black.

See Also : [YearLabelFont](#)

Text3dCells

With this property you can make the text that is in the cells look 3 dimensional.

tsRaised	- Raised text style..
rsRecessed	- Recessed text style
rsNone	- Normal text

Note : If the calendar is very small and the text only just fits in the cells the 3d effect can overlap the border for the cell.

See Also : [Text3dHighlight](#), [Text3dWeekDayHeader](#)

Text3dHighlight

With this property you can make the text in the highlighted (Selected) cell look 3 dimensional.

tsRaised	- Raised text style..
rsRecessed	- Recessed text style
rsNone	- Normal text

Note : If the calendar is very small and the text only just fits in the cells the 3d effect can overlap the border for the cell.

See Also : [Text3dCells](#), [Text3dWeekDayHeader](#)

Text3dWeekDayHeader

With this property you can make the text in the week day header look 3 dimensional.

tsRaised - Raised text style..
tsRecessed - Recessed text style
tsNone - Normal text



Su Mo Tu We Th Fr Sa

The example above is draw using tsRecessed.

See Also : [Text3dCells](#), [Text3dHighlight](#)

SpaceBetweenButtonsAndBottom

This property sets the space between the button bar and the bottom of the calendar control.

SpaceBetweenLabelsAndWeekDayHeader

This property sets the space between the month/year labels and the week day header.

SpaceBetweenWeekDayHeaderAndCells

This property sets the space between the week day header and the calendar day cells.

StartOfWeek

This property defines the starting day of the week for the calendar.

swSunday - The first day of the week is a Sunday
swMonday - The first day of the week is a Monday

The calendar is drawn using the starting day as the first day of the week.

SpaceFromSides

This property sets the space from both the left and right sides of the calendar control to the edge of the calendar week day cells.



In the example above the property SpaceFromSides is set to 5.



In the example above the property SpaceFromSides is set to 20.

SpaceBetweenCellsAndButtons

This property sets the space between the calendar day cells and the button bar.

SpaceBetweenTopAndLabels

This property sets the space between the top of the calendar control and the top of the month/year labels.

SpaceAtTop

This property sets additional space at the top of the calendar. The space is before the weekday header if it is visible, else it is before the top of the calendar cells.

You would use this extra space to put other controls into the calendar making them look like they are part of the calendar.

The sample program does this with a button on calendar in the default tab.

See Also : [SpaceAtBottom](#), [SpaceAtLeft](#)

SpaceAtBottom

This property sets additional space at the bottom of the calendar. The space is below the button bar if it is visible, else it is below the bottom of the calendar cells.

You would use this extra space to put other controls into the calendar making them look like they are part of the calendar.

See Also : [SpaceAtTop](#), [SpaceAtLeft](#)

SpaceAtLeft

This property sets additional space at the left side of the calendar.

Since the calendar cells are all equal in width, the left and right space from the calendar border edges can sometimes look not right, ie: there is more space on the right, than the left. With this property you can make the calendar look more centered.



November						1995
Su	Mo	Tu	We	Th	Fr	Sa
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

In the above example the space at the right is more than the left, by setting the property SpaceAtLeft you can make the calendar more centered within the borders. To line the above example SpaceAtLeft := 3 would do it.....

See Also : [SpaceAtTop](#), [SpaceAtBottom](#)

ShowSundayColor

This property can enable or disable the showing of the specific color for days that fall on a Sunday.

If TRUE then all sunday's in the current calendar month will be drawn using the ColorSunday property color.

If FALSE then all sunday's in the current calendar month will be draw using the ColorMonFri property color.

See Also : [ShowSaturdayColor](#), [ColorMonFri](#), [ColorSaturday](#), [ColorSunday](#)

ShowSaturdayColor

This property can enable or disable the showing of the specific color for days that fall on a Saturday.

If TRUE then all Saturday's in the current calendar month will be drawn using the [ColorSaturday](#) property color.

If FALSE then all Saturdays's in the current calendar month will be draw using the [ColorMonFri](#) property color.

See Also : [ShowSundayColor](#), [ColorMonFri](#), [ColorSaturday](#), [ColorSunday](#)

SelectPreviousNextMonthDate

This property sets if the user can select dates that are in the previous and next month. The user can only select dates in the previous and next months if the [ShowPrevNextMonth](#) property is set to TRUE.

See Also : [ShowPrevNextMonth](#)

SelectMode

This property defines how the OnDateChange events get triggered.

smOnlyAfterMouseUp - A OnDateChange event is only triggered when the user releases the mouse button.

smDuringMouseMove - A OnDateChange event is triggered while the mouse moves around the cells, this is dependant on the property MouseMode.

PreviousYearCaption

This property sets the caption for the Previous Year button when the Button Bar is visible. By default it is set to Y but can be changed to anything you require.

See Also : [NextYearCaption](#), [PreviousMonthCaption](#), [NextYearCaption](#)

PreviousMonthCaption

This property sets the caption for the Previous Month button when the Button Bar is visible. By default it is set to M but can be changed to anything you require.

See Also : [NextMonthCaption](#), [NextYearCaption](#), [PreviousYearCaption](#)

NextMonthCaption

This property sets the caption for the Next Month button when the Button Bar is visible. By default it is set to M but can be changed to anything you require.

See Also : [PreviousMonthCaption](#), [NextYearCaption](#), [PreviousYearCaption](#)

NextYearCaption

This property sets the caption for the Next Year button when the Button Bar is visible. By default it is set to Y but can be changed to anything you require.

See Also : [PreviousYearCaption](#), [NextMonthCaption](#), [PreviousMonthCaption](#)

CursorNonSelectable

This property defines the cursor shape when the user clicks on a valid day in the calendar. You can choose from three different cursors for this property.

cuDefault Standard Delphi default cursor shape.
cuClick The cursor that is the Click Cursor from the AD3DCAL.RES file.
cuSelectable The cursor that is the Selectable Cursor from the AD3DCAL.RES file.
cuNonSelectable The cursor that is the NonSelectable Cursor from the AD3DCAL.RES file.

This property is only valid when the ShowCustomCursors property is enabled.

The custom cursor is stored in the Ad3dCal.RES file and if you want to change it is the place to do it.

See Also : [ShowCustomCursors](#), [CursorClick](#), [CursorSelectable](#)

CursorSelectable

This property defines the cursor shape when the user moves the mouse over a selectable cell in the calendar. You can choose from three different cursors for this property.

cuDefault Standard Delphi default cursor shape.
cuClick The cursor that is the Click Cursor from the AD3DCAL.RES file.
cuSelectable The cursor that is the Selectable Cursor from the AD3DCAL.RES file.
cuNonSelectable The cursor that is the NonSelectable Cursor from the AD3DCAL.RES file.

This property is only valid when the [ShowCustomCursors](#) property is enabled.

The custom cursor is stored in the Ad3dCal.RES file and if you want to change it is the place to do it.

See Also : [ShowCustomCursors](#), [CursorClick](#), [CursorNonSelectable](#)

ColorWeekDayHeaderSaturday

This property defines the color for the Saturday text in the week day header.



In the above example ColorWeekDayHeaderSaturday is set to clRED.

See Also : [ColorWeekDayHeaderMonFri](#), [ColorWeekDayHeaderSunday](#)

CursorClick

This property defines the cursor shape when the user clicks on a valid day in the calendar. You can choose from three different cursors for this property.

cuDefault Standard Delphi default cursor shape.
cuClick The cursor that is the Click Cursor from the AD3DCAL.RES file.
cuSelectable The cursor that is the Selectable Cursor from the AD3DCAL.RES file.
cuNonSelectable The cursor that is the NonSelectable Cursor from the AD3DCAL.RES file.

This property is only valid when the ShowCustomCursors property is enabled.

The custom cursor is stored in the Ad3dCal.RES file and if you want to change it is the place to do it.

See Also : [ShowCustomCursors](#), [CursorSelectable](#), [CursorNonSelectable](#)

ColorWeekDayHeaderMonFri

This property defines the color for the Monday - Friday text in the week day header.



In the above example ColorWeekDayHeaderMon-Fri is set to clBLUE

See Also : [ColorWeekDayHeaderSaturday](#), [ColorWeekDayHeaderSunday](#)

ColorWeekDayHeaderSunday

This property defines the color for the Saturday text in the week day header.



In the above example ColorWeekDayHeaderSaturday is set to cIRED.

See Also : [ColorWeekDayHeaderSaturday](#), [ColorWeekDayHeaderMonFri](#)

ShowCustomCursors

This property controls if custom cursors are displayed when the user moves the mouse over cells in the calendar, or selects a cell in the calendar.

If this property is set to false all custom cursors are disabled.

See Also : [CursorClick](#), [CursorSelectable](#), [CursorNonSelectable](#)

CellWidth

Style

This property sets the drawing style for the calendar.

odStandard - Normal the calendar controls the drawing of the inside of the cells and the text in the cells.

odOwnerDraw - You control the drawing of the inside of the cells, and the text or graphics in the cells.

When the OwnerDraw option is being used you can display anything you like in the cells, like bitmaps, text..... And when you are using owner draw you can even decide not to draw the cell and let the default drawing routine draw it.

See also : [OnDrawCell](#), [Canvas](#)

OnNewDate

OnDrawCell

This event gets called for every cell that needs to be drawn in the calendar when the property Style is set to `odOwnerDraw`.

This event gives you lots of details about the cell to be drawn including the rectangle of the cell, the actual day number, the ID of the cell starting from 1, if the cell is in the current month, if the cell is the selected cell.

You can even decide not to draw the cell and let the default drawing routines handle it. ie: you might not want to be bothered with drawing cells that are not in the current calendar month.

Example taken from the sample program.

```
procedure TSDIAppForm.Adrock3dCalendar5DrawCell(Sender: TObject; CellRect: TRect; Day,
CellID: Integer; CellInCurrentMonth, Selected: Boolean; var UseDefaultDrawing: Boolean);
Var
    ImageWidth  : Integer;
    ImageHeight : Integer;
begin
    Adrock3dCalendar5.Canvas.Font.Name := 'SmallFonts';
    Adrock3dCalendar5.Canvas.Font.Size := 6;

    if (Selected = TRUE) then
        Adrock3dCalendar5.Canvas.brush.Color := cIRED
    else if (CellInCurrentMonth = FALSE) then
        Adrock3dCalendar5.Canvas.brush.Color := CIGREEN
    else
        Adrock3dCalendar5.Canvas.brush.Color := cIBLUE;

    Adrock3dCalendar5.Canvas.Pen .Color := cIWHITE;
    Adrock3dCalendar5.Canvas.FillRect(CellRect);
    Adrock3dCalendar5.Canvas.TextOut(CellRect.Left, CellRect.Top, IntToStr(Day));

    if (Cells[CellID] > '') then
        begin
            Adrock3dCalendar5.Canvas.Font.Color := cIWhite;
            Adrock3dCalendar5.Canvas.TextOut(CellRect.Left, CellRect.Bottom-
Adrock3dCalendar5.Canvas.textHeight('A'),
            Cells[CellID]);
            Adrock3dCalendar5.Canvas.Font.Color := cIRed;
        end;

        if (CellsImage[CellID] = 'Alarm') then
            begin
                ImageWidth := Image1.Picture.Bitmap.Width;
                ImageHeight := Image1.Picture.Bitmap.height;
                Adrock3dCalendar5.Canvas.Draw(CellRect.Right-ImageWidth, CellRect.Top,
Image1.Picture.Bitmap);
            end;
    end;
```

PrevDay

Declaration

Procedure NextDay;

Description

This method will force the calendar to make the Previous Day the current day, redrawing the calendar as needed.

NextWeek

Declaration

Procedure NextDay;

Description

This method will force the calendar to make the Next Week the current day, redrawing the calendar as needed.

PrevWeek

Declaration

Procedure NextDay;

Description

This method will force the calendar to make the Previous Week the current day, redrawing the calendar as needed.

GetDateFromIndex

Declaration

```
Function GetDateFromIndex(Index : Integer) : TDateTime;
```

Description

Given a index (1-42) this routine will return the date for the specified cell.

NextMonth

Declaration

Procedure NextDay;

Description

This method will force the calendar to make the Next Month the current day, redrawing the calendar as needed.

PrevMonth

Declaration

Procedure NextDay;

Description

This method will force the calendar to make the Previous Month the current day, redrawing the calendar as needed.

NextYear

Declaration

Procedure NextDay;

Description

This method will force the calendar to make the Next Year the current day, redrawing the calendar as needed.

PrevYear

Declaration

Procedure NextDay;

Description

This method will force the calendar to make the Previous Year the current day, redrawing the calendar as needed.

GotoMonth

Declaration

Procedure GotoMonth(fGotoMonth : Integer);

Description

This method will force the calendar to go to the specified month, the specified month must be in the range 1- 12.

GotoYear

Declaration

Procedure GotoYear(fGotoYear : Integer);

Description

This method will force the calendar to go to the specified year.

IsIndexInCurrentCalendarMonth

Declaration

Function IsIndexInCurrentCalendarMonth(nIndex : Integer) : Boolean;

Description

This method will return TRUE if the cell referenced by nIndex is in the current calendar month, it is mainly used internally but someone might have a need for it in Owner Draw. nIndex ranges from 1 - 42.

GetIndexFromPoint

Declaration

```
Function GetIndexFromPoint(nLeft : Integer ; nTop : Integer) : Integer;
```

Description

This method will return the Index of a cell if the nLeft and nTop parameters are within a calendar day cell.

GetDayFromIndex

Declaration

Function GetDayFromIndex(Index : Integer) : Integer;

Description

Given a index (1-42) this routine will return the day for the specified cell.

GetIndexFromDay

Declaration

Function GetIndexFromDay(Day : Integer) : Integer;

Description

Given a day in the current month this routine will return the index of the cell for that day.

OwnerDraw

Below is actual code taken from the OnDragOver, OnDragDrop, and OnDrawCell events of the Drag & Drop tab in the sample program.

```
{ Drag Over Event : This allows the object to be dropped }  
procedure TSDIAppForm.Adrock3dCalendar2DragOver(Sender, Source: TObject; X, Y: Integer;  
State: TDragState; var Accept: Boolean);  
begin  
    if ((Source = ListBox3) or (Source = ListBox5)) and  
(Adrock3dCalendar5.GetIndexFromPoint(x,y) <> -1) then  
        Accept := TRUE  
    else  
        Accept := FALSE;  
end;
```

```
{ Drag Drop Event for the calendar }  
Procedure TSDIAppForm.Adrock3dCalendar5DragDrop(Sender, Source: TObject; X,  
Y: Integer);  
var  
    Index    : Integer;  
    CurrMonth : String;  
begin  
    Index := Adrock3dCalendar5.GetIndexFromPoint(x,y);  
    if (Adrock3dCalendar5.IsIndexInCurrentCalendarMonth(Index) = TRUE) then  
        CurrMonth := 'TRUE'  
    else  
        CurrMonth := 'False';  
  
    if ((Source = ListBox3) or (Source = ListBox5)) and (Index <> -1) then  
        begin  
            ListBox4.Items.Add(  
                'X = '+IntToStr(x)+  
                ', Y = '+IntToStr(y)+  
                ', Day = '+IntToStr(Adrock3dCalendar5.GetDayFromIndex(Index))+  
                ', Cell = '+IntToStr(Index)+  
                ', Date = '+DateToStr(Adrock3dCalendar5.GetDateFromIndex(Index))+  
                ', CurrentMonth = '+ CurrMonth);  
            if (Source = ListBox3) then  
                begin  
                    Cells[Index] := ListBox3.Items[ListBox3.ItemIndex];  
                    Adrock3dCalendar5.RepaintCellByIndex(Index);  
                end  
            else  
                begin  
                    CellsImage[Index] := ListBox5.Items[ListBox5.ItemIndex];  
                    Adrock3dCalendar5.RepaintCellByIndex(Index);  
                end;  
            end;  
            ShowMessage('Something has been dropped on the calendar.');
```

```
{ onDrawCell Event for the calendar - Used only when ownerdrawing }  
Procedure TSDIAppForm.Adrock3dCalendar5DrawCell(Sender: TObject; CellRect: TRect; Day,  
CellID: Integer; CellInCurrentMonth,
```

```

    Selected: Boolean; var UseDefaultDrawing: Boolean);
Var
    ImageWidth  : Integer;
    ImageHeight : Integer;
begin
{ if (Day > 10) and (Day < 20) then
    begin
        UseDefaultDrawing := TRUE;
        exit;
    end;
}
Adrock3dCalendar5.Canvas.Font.Name := 'SmallFonts';
Adrock3dCalendar5.Canvas.Font.Size := 6;

if (Selected = TRUE) then
    Adrock3dCalendar5.Canvas.brush.Color := cIRED
else if (CellInCurrentMonth = FALSE) then
    Adrock3dCalendar5.Canvas.brush.Color := CIGREEN
else
    Adrock3dCalendar5.Canvas.brush.Color := cIBLUE;
    Adrock3dCalendar5.Canvas.Pen .Color := ciWHITE;
    Adrock3dCalendar5.Canvas.FillRect(CellRect);
    Adrock3dCalendar5.Canvas.TextOut(CellRect.Left, CellRect.Top, IntToStr(Day));

    if (Cells[CellID] > "") then
        begin
            Adrock3dCalendar5.Canvas.Font.Color := ciWhite;
            Adrock3dCalendar5.Canvas.TextOut(CellRect.Left, CellRect.Bottom-
Adrock3dCalendar5.Canvas.textHeight('A'),
            Cells[CellID]);
            Adrock3dCalendar5.Canvas.Font.Color := ciRed;
            end;

        if (CellsImage[CellID] = 'Alarm') then
            begin
                ImageWidth := Image1.Picture.Bitmap.Width;
                ImageHeight := Image1.Picture.Bitmap.height;
                Adrock3dCalendar5.Canvas.Draw(CellRect.Right-ImageWidth, CellRect.Top,
Image1.Picture.Bitmap);
            end;
        end;
end;

```

Cells[] is an array declared in the sample program and it is used to hold the details for the particular cell when it is needed to draw. This is only a very simple example.....

CellsImage[] is used to hold a reference to cells that have had the picture dropped on them. The Image1.Picture.Bitmap is on the main form of the Sample Program, it is used to store the bitmap image, and it is then drawn on the cell using the Canvas.Draw command.

