

Contents

Thank you for purchasing Software FX, Inc. products!

The following topics are available:

- 1) [LICENSE AGREEMENT](#)
- 2) [PROPERTIES REFERENCE](#)
- 3) [EVENTS REFERENCE](#)
- 4) [UNDERSTANDING THE PARTS OF A CALENDAR](#)
- 5) [DATABOUND](#)
- 6) [INTERNATIONAL SUPPORT](#)
- 7) [INTEGRATING CALENDAR TO YOUR DEVELOPMENT TOOL](#)

IMPORTANT REGISTRATION INFORMATION. (PLEASE READ FIRST)

This calendar control is an add-on to Chart FX 3.0 original package and it's bound to the same License Agreement. Please note that this calendar control is also a Shareware product. Therefore, If you are not a Chart FX 3.0 registered user you must register this copy before you can integrate it and redistribute it with your application.

Technical Support will be provided only to Chart FX 3.0 registered users or Shareware registered users. So please be sure you enter in either one of these categories before you contact Software FX, Inc.

In order to register as a shareware copy you must contact our offices:

- 1) **Software FX, Inc.**
Ph: (407) 998-2377
Fax: (407) 998-2383
Toll Free: 1-800-392-4278

- 2) **Software FX, Inc.**
CompuServe ID: 74032,2412

Price: \$49.95
Royalty Free for distribution
We accept: VISA/MC/AMEX

If you're a Chart FX 3.0 registered user, you may use this Calendar control freely. Technical support will not be provided through Software, Inc. Toll Free Number.

For Help on Help, Press F1

License Agreement

Software FX, Inc. LICENSE CARD (Calendar Control)

READ CAREFULLY BEFORE INSTALLING AND/OR DISTRIBUTING THE SOFTWARE.

By opening the sealed packet(s) containing Software FX, Inc. software. (hereinafter "the Software" or "Software") , you are accepting the following License Agreement.

LICENSE AGREEMENT

This is a legal agreement between you (either an individual or an entity) and Software FX, Inc. By installing and/or distributing the software you are agreeing to be bound by the terms of this agreement. If you do not agree to the terms of this agreement, promptly return the uninstalled software packet(s) and the accompanying items (including written materials and binders or other containers) to the place you obtained them for a full refund.

SOFTWARE FX, INC. - SOFTWARE LICENSE

- 1. GRANT OF LICENSE.** This License agreement permits you to use one copy of the enclosed software program (hereinafter "The SOFTWARE" or "SOFTWARE") on a single computer. The SOFTWARE is in "use" on a computer when it is loaded into temporary memory (i.e. RAM) or installed into permanent memory (e.g. hard disk, or other storage device) of that computer.
- 2. COPYRIGHT.** The SOFTWARE is owned by Software FX, Inc. or its suppliers and is protected by United States copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE like any other copyrighted material (e.g. a book or a musical recording) except that you may either (a) make one copy of the SOFTWARE solely for backup or archival purposes. or (b) transfer the SOFTWARE to a single hard disk provided you keep the original solely for backup or archival purposes. You may not make multiples copies of SOFTWARE nor the written materials accompanying the SOFTWARE.
- 3. OTHER RESTRICTIONS.** You may not rent or lease the SOFTWARE, but you may transfer the SOFTWARE and accompanying written materials on a permanent basis provided you retain no copies and the recipient agrees to the terms of this Agreement. Upon such transfer, you will notify Software FX, Inc. of the transfer and the name and address of recipient. You may not reverse engineer, decompile, or disassemble the SOFTWARE. If the SOFTWARE is an Update or has been updated, any transfer must include the most recent update and all prior versions.
- 4. DUAL-MEDIA SOFTWARE.** If the SOFTWARE package contains both 3.5" and 5.25" disks, then you may use only the disks appropriate for your single-user computer. You may not use the other disks on another computer or loan, rent, lease, or transfer them to another user except as part of the permanent transfer (as provided above) of all SOFTWARE and written materials.
- 5. LIBRARY SOFTWARE.** You have a royalty-free right to distribute only the "run-time modules " with the executable files created in any other vendor product (Language or Development Tool) limited as hereinafter set forth in paragraph a through d. Software FX, Inc. grants you a royalty-free distribution if : (a) you distribute the "run time" modules only in conjunction with the executable files that make use of them as a part of your software product; (b) you do not use the Software FX, Inc. name, logo or trademark to market your software product; (c) The SOFTWARE end users do not use the "run time" modules or any other SOFTWARE components for development purposes. and, (d) you agree to indemnify, hold harmless, and defend Software FX, Inc. and its suppliers from and against any and all claims or lawsuits including attorney's fees, that arise or result from the use or distribution of your software product. If any of the conditions set forth in paragraphs a through d are breached, such breach shall constitute an unlawful use of the SOFTWARE, and you shall be prosecuted to the full extent of the law. Furthermore, you shall be liable to Software FX, Inc.

for all damages caused by such breach and unlawful use of the software, including attorney's fees and costs incurred in any action, lawsuit or claim brought or filed to redress the breach of this agreement. The "run time modules" are those files included in the SOFTWARE package that are required during execution of your software program.

Understanding the parts of a calendar

Click on the corresponding section you want to learn about:



Calendar Background

The calendar background can be set to a desired color through the [BackColor](#) property.

Default:

White (RGB(255,255,255))

The calendar is always enclosed in a rectangle with a background, which can be set to any desired color.

Related Properties:

[Like3D](#)

[RGBSundays](#)

[RGBWeekDays](#)

[RGBDelimiter](#)

[RGBGrid](#)

[RGBJumpDays](#)

[RGBTopBarText](#)

Sundays

Sundays are displayed in the Calendar control with a different color for a better-user interface. Nevertheless, this color can be changed to a desired color for further customization

Default:

Red (RGB(255,0,0))

When applying a special color the text associated to sundays (See [DaysTags](#) property) the letter associated to this day is also displayed in this color.

Related Properties:

[RGBSundays](#)

[DaysTags](#)

Selected Date

Calendar Control provides you with the ability to select a date at running time or from a databounded calendar. (Simple selection only).

Please note that this feature will allow you to retrieve the selected date for further processing. The property associated to setting a date is called ([SelDate](#) property).

Also, when you change the displayed month and year, the Selected date does not necessarily change. Evenmore, when calling the SelDate property, the displayed month and year will remain the same unless you have the [AutoSelection](#) property turned on or set the [CurrentMonth](#) and [CurrentYear](#) properties manually.

Default Selected Date:

Today's Date

Related Properties:

[CurrentMonth](#)

[CurrentYear](#)

[CurrentGrayed](#)

Jumping Days

The days that appear before and/or after the first and last day of the current month called Jumping Days. Basically, when you click in any of the Jumping Days the calendar control will automatically jump to the previous or next month accordingly.

When not shown (Jumping Days), the calendar control will not have squared form. Instead, those rectangles will be suppressed from the current month and the end user will not be able to click on these days to change the actual displayed month.

The other way to change the actual month is interacting with the Buttons or ComboBoxes located in the [TopBar](#)

Default:

Show

Related Properties:

[RGBJumpDays](#)

Days & DayTags

The days displayed in the calendar can be customized to be shown in any font type, color and even change if they display leading zeros and grid.

Default:

Blue (RGB(0,0,255))

Related Properties:

[RGBWeekDays](#)

[LeadingZero](#)

[Grid](#)

[DayGapX](#)

[DayGapY](#)

Delimiters

The Chart control displays two delimiters to separate the different elements in the calendar control. You can choose whether to show Top, Bottom or Both delimiters in the calendar. Color changing is also available for delimiters.

You can also control the gap (in device units, pixels) or separation of the delimiters to each element in the calendar control.

Default:

Both delimiters
3D appearance

Related properties:

[RGBDelimiter](#)

[Delimiters](#)

[DelimiterGap](#)

TopBar

The TopBar is an element that allows you to change the current month or year displayed in the chart. The [TopBar](#) setting has three different settings:

0.- None:

No TopBar is displayed

1.- ComboBoxes:

In this mode the TopBar will display two comboboxes containing the month and year for user-interaction. You can control how many years (and the beginning year) displayed in this comboboxes.

2.- Buttons:

Instead of two comboboxes, buttons are displayed in order to change the current month. This mode is more suitable when you don't want to change the current year.

3.- TextOnly:

No user-interaction to change current month or year is presented. Only a Text showing the current month and year is displayed.

Default:

Buttons

Related Properties

[TopBar](#)

International Support

For those of you who want to translate Calendar Control resources to your language, we have included all text used in this control in a String Table inside the VBX.

In order to change this string table you must have a resource editor to access it. Most development tools are accompanying with such editors. The string table contains the following text:

- 1) One string per month of the year (January through December)
- 2) One string containing seven letters corresponding to the week days (Sunday through Saturday)

Basically, you will change all strings to your language and the control will be ready to display text in the altered resource.

Suggested Resource Editors:

Microsoft AppStudio (bundled with Visual C++)

Borland WorkShop (bundled with Borland C/C++)

Databound Calendar

The following sample was produced using Microsoft Visual Basic Professional Edition version 3.0. You must check your development tool documentation for DataBound support. We strongly suggest you use Visual Basic for DataBound Calendar control.

Important Note:

The calendar control is a simple selection control, which means that it will only take one field per record in your table to display/update the selected date automatically from the table specified.

In order to bind a calendar control to a database you must do the following:

- 1) Create the Calendar control in your form by dragging the calendar icon (from the toolbox) to your form area.
- 2) Create a Data control (Data1) from the Toolbos, with the following characteristics:
 - 2.1) Fill the **DatabaseName** property with the appropriate database
 - 2.2) Specify in the **RecordSource** property the SELECT statement or the table you want to link the data control to.
- 3) If you want the Calendar control to update the table in your database, you must set the **ReadOnly** property of the Data Control to FALSE (Default). If you fail to set this property on the Data Control, whenever you change the selected date you will affect the current record of the database.
- 4) Write the field that contains the date value in your chart control by setting the [DataField](#) property in your calendar control property sheet.
- 5) Link the Data Control to the Calendar control by setting the [DataSource](#) property of the Calendar control to the name of the created Data Control (Data1).
- 6) Turn on the [AutoSelection](#) property to TRUE, so everytime the data control browse through records, the calendar control automatically changes the [CurrentMonth](#) and [CurrentYear](#) so the selected date is always visible.

Results:

When you navigate the table through the data control (Data1), for every record that has a valid date the calendar control will select and display (and also update) such date.

Integrating Calendar control to your development tool

Please follow these steps to fully integrate the calendar control to your development tool:



Borland Delphi

* *Include the Chart FX VBX in your Delphi:*

- From the Options menu select Install Components... option.
- Press the VBX... button
- From the Install VBX File Dialog
- Choose sfxcalen.vbx from your Windows directory.
- From the Install VBX Dialog press the OK button
- From the Install component Dialog press the OK button again.



Visual Basic

* *Include the Calendar VBX in your program*

- From the File menu select Add File... option.
- Choose sfxcalen.vbx from your Windows directory.



Visual C++

* *Enable the VBX engine at your application startup code.*

- In the InitInstance member function add the following code:
EnableVBX();

- To ensure proper response when your application cannot find the VBX file add the following code after the EnableVBX Call

```
if (LoadVBXFile("SFXCALEN.VBX") > HINSTANCE_ERROR)
    UnloadVBXFile("SFXCALEN.VBX");
else {
    AfxMessageBox("Cannot Load SFXCALEN.VBX",MB_OK);
    return FALSE;
}
```

* *Include the Chart FX VBX button in the AppStudio's ToolBar.*

- From the File menu select Install Controls... option.
- Choose sfxcalen.vbx from the Windows directory.

About Property

The About property will allow you to retrieve important information about Calendar Control development team and Software FX, Inc. company information.

AutoSelection Property

By setting Autoselection property to TRUE, the calendar control will automatically change the displayed year and month when selecting a date with the [SelDate](#) property. This property is very useful when setting dates and those need to be shown in the calendar control.

Visual Basic

```
[form.] Chart1.Autoselection [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("Autoselection");  
sfxCalen1->SetNumProperty("Autoselection", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.Type [ := setting& ];
```

Remarks

Data Type

Boolean

Default

FALSE

See Also

[SelDate](#), [CurrentMonth](#), [CurrentYear](#), [SelMonth](#), [SelDay](#), [SelYear](#)

BackColor Property

This property sets the background color of the calendar control.

Visual Basic

```
[form.] Chart1.BackColor [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("BackColor");  
sfxCalen1->SetNumProperty("BackColor", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.BackColor [ := setting& ];
```

Remarks

If [Like3D](#) property is turned on, is strongly suggested to set the BackColor property to gray (RGB(128,128,128)) to obtain a full 3D display

Data Type

RGB

Default

RGB(255,255,255) White

See Also

[RGBDelimiter](#), [RGBGrid](#), [RGBJumpDays](#), [RGBSundays](#), [RGBTopBarText](#), [RGBWeekDays](#), [Like3D](#)

CurrentGrayed Property

This property sets the selected date to be grayed when selecting a date in the calendar control. By turning on this property the selected date will appear like a 3D button, when turned off the selected date will appear in a negative (black background) display.

Visual Basic

```
[form.] Chart1.CurrentGrayed [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("CurrentGrayed");  
sfxCalen1->SetNumProperty("CurrentGrayed", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.CurrentGrayed [ := setting& ];
```

Remarks

When creating the calendar control, the selected date will be Today's date, and displayed month and year correspond to this date. To change current selected date, please refer to [SelDate](#) property.

Data Type

Boolean

Default

TRUE

See Also

[CurrentMonth](#), [CurrentYear](#) , [SelDate](#)

CurrentMonth Property

This property sets the month to be displayed in the calendar control. CurrentMonth property only will change the displayed month while keeping the selected date to the same.

Visual Basic

```
[form.] Chart1.CurrentMonth [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("CurrentMonth");  
sfxCalen1->SetNumProperty("CurrentMonth", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.CurrentMonth [ := setting& ];
```

Remarks

When creating the calendar control, the selected date will be Today's date, and displayed month and year correspond to this date. To change current selected date, please refer to [SelDate](#) property.

Data Type

Integer

Default

Today's date month.

See Also

[CurrentYear](#), [CurrentGrayed](#)

CurrentYear Property

This property sets the year to be displayed in the calendar control. CurrentYear property only will change the displayed year while keeping the selected date to the same.

Visual Basic

```
[form.] Chart1.CurrentYear [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("CurrentYear");  
sfxCalen1->SetNumProperty("CurrentYear", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.CurrentYear [ := setting& ];
```

Remarks

When creating the calendar control, the selected date will be Today's date, and displayed month and year correspond to this date. To change current selected date, please refer to [SelDate](#) property.

Data Type

Integer

Default

Today's date year.

See Also

[CurrentMonth](#), [CurrentGrayed](#), [SelDate](#)

DataField Property

When binding a calendar control to a database, you must specify which field contains (name as string) a valid date to be selected in the calendar control.

Visual Basic

```
[form.] Chart1.DataField [ = setting& ]
```

Visual C++

N/A

Borland Delphi

N/A

Remarks

This field can be, either:

- 1) A date type
- 2) A string type containing a valid date string.

Databound calendars are only available from Microsoft Visual Basic

Data Type

String

Default

None

See Also

[DataSource](#), [DataBound Calendar](#)

DataSource Property

The DataSource property, allows you to specify which data control has the SELECT statement in order to grab the field from the database automatically. This property must be used in conjunction with the [DataField](#) property.

Visual Basic

```
[form.] Chart1.DataSource [ = setting& ]
```

Visual C++

N/A

Borland Delphi

N/A

Remarks

Databound calendars are only available from Microsoft Visual Basic

Data Type

string

Default

None

See Also

[DataField](#), [DataBound Calendar](#)

DayGapX Property

This property sets the horizontal gap to be used in the rectangle enclosing the days in the calendar control

Visual Basic

```
[form.] Chart1.DayGapX [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("DayGapX");  
sfxCalen1->SetNumProperty("DayGapX", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.DayGapX [ := setting& ];
```

Remarks

This property is also maintained when you change font types or sizes.

Data Type

Integer

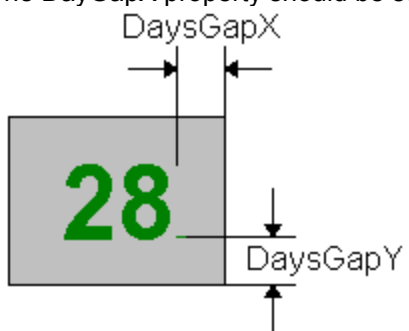
Default

3 pixels

See Also

[DayGapY](#), [Grid](#), [RGBGrid](#)

The DayGapX property should be set considering the following diagram:



DayGapY Property

This property sets the vertical gap to be used in the rectangle enclosing the days in the calendar control

Visual Basic

```
[form.] Chart1.DayGapY [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("DayGapY");  
sfxCalen1->SetNumProperty("DayGapY",lSetting);
```

Borland Delphi

```
[Tform.] Chart1.DayGapY [ := setting& ];
```

Remarks

This property is also maintained when you change font types or sizes.

Data Type

Integer

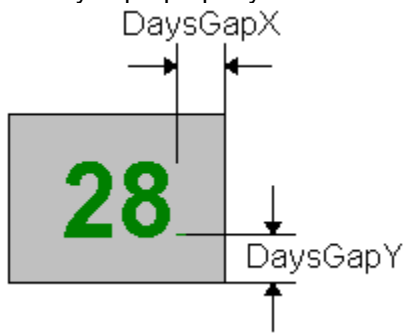
Default

2 pixels

See Also

[DayGapX,Grid,RGBGrid](#)

The DayGapY property should be set considering the following diagram:



DaysTags Property

This property show/hide the days header in the calendar control. The DaysTags are the letters representing every day of the week in the calendar control.

Visual Basic

```
[form.] Chart1.DaysTags [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("DaysTags");  
sfxCalen1->SetNumProperty("DaysTags", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.DaysTags [ := setting& ];
```

Remarks

See [international support](#) section for changing the Days letters.

Data Type

Boolean

Default

TRUE

See Also

[RGBWeekDays](#), [JumpDays](#)

DelimiterGap Property

This property sets the gap between delimiters shown in the calendar control

Visual Basic

```
[form.] Chart1.DelimiterGap [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("DelimiterGap");  
sfxCalen1->SetNumProperty("DelimiterGap",lSetting);
```

Borland Delphi

```
[Tform.] Chart1.DelimiterGap [ := setting& ];
```

Remarks

Data Type

Integer

Default

5 pixels

See Also

[Delimiters](#), [RGBDelimiters](#)

Delimiters Property

This property shows the desired delimiters in the calendar control.

Visual Basic

```
[form.] Chart1.Delimiters [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("Delimiters");  
sfxCalen1->SetNumProperty("Delimiters",lSetting);
```

Borland Delphi

```
[Tform.] Chart1.Delimiters [ := setting& ];
```

Remarks

The setting can be any of the following:

0 = None

1 = Top

2 = Bottom

3 = Both

Data Type

Integer

Default

Both (3)

See Also

[DelimiterGap](#), [RGBDelimiter](#)

FirstYear Property

This property sets the year to start with in the calendar control. When you're working with Comboboxes in the TopBar you can specify how many years the combo will have and what's the first year to start with. This way the end-user is able to select a desired year starting in the year set in this property.

Visual Basic

```
[form.] Chart1.FirstYear [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("FirstYear");  
sfxCalen1->SetNumProperty("FirstYear", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.FirstYear [ := setting& ];
```

Remarks

This property will be set with the current year (today's date). To control the number of years appearing the years combobox please refer to [YearsNumber](#) property.

Data Type

Integer

Default

Today's date year

See Also

[YearsNumber](#), [TopBar](#)

Grid Property

This property show/hide the grid lines inside the days in the calendar control

Visual Basic

```
[form.] Chart1.Grid [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("Grid");  
sfxCalen1->SetNumProperty("Grid", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.Grid [ := setting& ];
```

Remarks

The setting can be any of the following:

- 0 = None
- 1 = Horizontal
- 2 = Vertical
- 3 = Both

Data Type

Integer

Default

Both (3)

See Also

[RGBGrid](#), [DayGapX](#), [DayGapY](#)

JumpDays Property

This property show/hide the jumping days in the calendar control. The jumping days are those shown in the calendar that represent other months date and by clicking those days the calendar control will automatically jump to the appropriate month (Previous/Next).

Visual Basic

```
[form.] Chart1.JumpDays [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("JumpDays");  
sfxCalen1->SetNumProperty("JumpDays",lSetting);
```

Borland Delphi

```
[Tform.] Chart1.JumpDays [ := setting& ];
```

Remarks

Data Type

Boolean

Default

TRUE

See Also

[RGBJumpDays](#)

LeadingZero Property

This property show/hide leading zeros in the days displayed in the calendar control.

Visual Basic

```
[form.] Chart1.LeadngZero [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("LeadingZero");  
sfxCalen1->SetNumProperty("LeadingZero",lSetting);
```

Borland Delphi

```
[Tform.] Chart1.LeadngZero [ := setting& ];
```

Remarks

Data Type

Boolean

Default

TRUE

See Also

[RGBWeekDays](#), [RGBSundays](#), [Grid](#), [DayGapX](#), [DayGapY](#)

Like3D Property

This property show/hide the calendar control in 3D mode.

Visual Basic

```
[form.] Chart1.Like3D [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("Like3D");  
sfxCalen1->SetNumProperty("Like3D", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.Like3D [ := setting& ];
```

Remarks

In order to get full 3D display we strongly suggest you set the BackColor property to light gray (RGB(128,128,128))

Data Type

Boolean

Default

TRUE

See Also

[BackColor](#)

RGBDelimiter Property

This property sets the delimiters color in the calendar control.

Visual Basic

```
[form.] Chart1.RGBDelimiter [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("RGBDelimiter");  
sfxCalen1->SetNumProperty("RGBDelimiter",lSetting);
```

Borland Delphi

```
[Tform.] Chart1.RGBDelimiter [ := setting& ];
```

Remarks

Data Type

RGB

Default

Dark Gray - RGB(128,128,128)

See Also

[RGBGrid](#), [RGBJumpDays](#), [RGBSundays](#), [RGBTopBarText](#), [RGBWeekDays](#)

RGBGrid Property

This property sets the grid color in the calendar control.

Visual Basic

```
[form.] Chart1.RGBGrid [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("RGBGrid");  
sfxCalen1->SetNumProperty("RGBGrid", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.RGBGrid [ := setting& ];
```

Remarks

Data Type

RGB

Default

Dark Gray - RGB(128,128,128)

See Also

[RGBDelimiter](#), [RGBJumpDays](#), [RGBSundays](#), [RGBTopBarText](#), [RGBWeekDays](#)

RGBJumpDays Property

This property sets the jump days color in the calendar control.

Visual Basic

```
[form.] Chart1.RGBJumpDays [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("RGBJumpDays");  
sfxCalen1->SetNumProperty("RGBJumpDays", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.RGBJumpDays [ := setting& ];
```

Remarks

Data Type

RGB

Default

Dark Gray - RGB(128,128,128)

See Also

[RGBDelimiter](#), [RGBGrid](#), [RGBSundays](#), [RGBTopBarText](#), [RGBWeekDays](#)

RGBSundays Property

This property sets the sundays color in the calendar control.

Visual Basic

```
[form.] Chart1.RGBSundays [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("RGBSundays");  
sfxCalen1->SetNumProperty("RGBSundays",lSetting);
```

Borland Delphi

```
[Tform.] Chart1.RGBSundays [ := setting& ];
```

Remarks

Data Type

RGB

Default

Red - RGB(255,0,0)

See Also

[RGBDelimiter](#), [RGBGrid](#), [RGBJumpDays](#), [RGBTopBarText](#), [RGBWeekDays](#)

RGBTopBarText Property

This property sets the Top Bar Text (Month and Year) when TopBar is shown as buttons color in the calendar control.

Visual Basic

```
[form.] Chart1.RGBTopBarText [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("RGBTopBarText");  
sfxCalen1->SetNumProperty("RGBTopBarText", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.RGBTopBarText [ := setting& ];
```

Remarks

Please refer to [TopBar](#) property in order to set top bar as buttons

Data Type

RGB

Default

Blue - RGB(0,0,255)

See Also

[RGBDelimiter](#), [RGBGrid](#), [RGBJumpDays](#), [RGBSundays](#), [RGBWeekDays](#)

RGBWeekDays Property

This property sets the weekdays color in the calendar control.

Visual Basic

```
[form.] Chart1.RGBWeekDays [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("RGBWeekDays");  
sfxCalen1->SetNumProperty("RGBWeekDays", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.RGBWeekdays [ := setting& ];
```

Remarks

Data Type

RGB

Default

Dark Blue - RGB(0,0,128)

See Also

[RGBDelimiter](#), [RGBGrid](#), [RGBJumpDays](#), [RGBSundays](#), [RGBTopBarText](#)

TopBar Property

This property will set the TopBar mode, in order to allow end users to change actual month, year, or restrict them to change the actual setting. Hiding the TopBar is also available.

Visual Basic

```
[form.] Chart1.TopBar [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("TopBar");  
sfxCalen1->SetNumProperty("TopBar", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.TopBar [ := setting& ];
```

Remarks

The setting can be any of the following:

0 = None

1 = Combos

2 = Buttons

3 = TextOnly

Data Type

Integer

Default

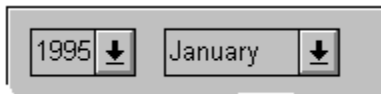
Buttons (2)

See Also

[RGBTopBarText](#), [TopBarGap](#)

The following diagram represent the possible settings:

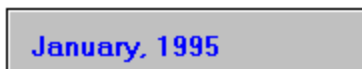
Combos (1)



Buttons (2)



TextOnly (3)



TopBarGap Property

This property sets the gap between top border in the calendar control and the TopBar

Visual Basic

```
[form.] Chart1.TopBarGap [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("TopBarGap");  
sfxCalen1->SetNumProperty("TopBarGap", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.TopBarGap [ := setting& ];
```

Remarks

Data Type

Integer

Default

10 pixels

See Also

[DelimiterGap](#), [DayGapX](#), [DayGapY](#)

Type Property

This property sets the calendar type

Visual Basic

```
[form.] Chart1.TopBarGap [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("TopBarGap");  
sfxCalen1->SetNumProperty("TopBarGap", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.TopBarGap [ := setting& ];
```

Remarks

The setting can be any of the following:

0 = square

1 = line

Data Type

Integer

Default

square (0)

Samples

The following is a sample image of each calendar type:

Square:

January, 1995						
S	M	T	W	T	F	S
01	01	02	03	04	05	06
07	08	09	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	01	02	03
01	02	03	04	05	06	07

Line:

January, 1995																							
M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W
01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

YearsNumber Property

This property sets the number of years that are going to be shown in the combobox when the topbar is shown with comboboxes.

Visual Basic

```
[form.] Chart1.YearsNumber [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("YearsNumber");  
sfxCalen1->SetNumProperty("YearsNumber", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.YearsNumber [ := setting& ];
```

Remarks

In order to show the TopBar with comboboxes please refer to [TopBar](#) property.

Data Type

Integer

Default

5 Years

See Also

[TopBar](#), [FirstYear](#)

SelDate Property

The SelDate property, allows you to set a date at running time, this is the property that allows you to select an specific date in the calendar control. Please note that when you use this property the calendar control will not change the actual setting (Month & Year) unless you have the [Autoselection](#) Property turned on. This means, that if you set an specific date (i.e. 03/25/95) and the displayed month is january the calendar control will not show march automatically unless you set the Autoselection property to TRUE. If you want to change the displayed month & year automatically, please refer to [CurrentMonth](#) and [CurrentYear](#) properties.

Visual Basic

```
[form.] Chart1.SelDate [ = setting$ ]
```

Visual C++

```
sfxCalen1->SetNumProperty("SelDate",lSetting);
```

Borland Delphi

```
[Tform.] Chart1.SelDate [ := setting$ ];
```

Remarks

This property is available at RUN TIME only.

The date string must be in INI format as set in Control Panel.

Data Type

string

Default

Today's date

See Also

[AutoSelection](#), [CurrentMonth](#), [CurrentYear](#) , [SelDay](#), [SelMonth](#), [SelYear](#)

Properties Reference

The following Properties are available in Calendar Control:

[AutoSelection](#)

[BackColor](#)

[CurrentGrayed](#)

[CurrentMonth](#)

[CurrentYear](#)

[DataField](#)

[DataSource](#)

[DayGapX](#)

[DayGapY](#)

[DaysTags](#)

[DelimiterGap](#)

[Delimiters](#)

[FirstYear](#)

[Grid](#)

[JumpDays](#)

[LeadingZero](#)

[Like3D](#)

[RGBDelimiter](#)

[RGBGrid](#)

[RGBJumpDays](#)

[RGBSundays](#)

[RGBTopBarText](#)

[RGBWeekDays](#)

[**SelDate**](#)

[**SelYear**](#)

[**SelMonth**](#)

[**SelDay**](#)

[TopBar](#)

[TopBarGap](#)

[Type](#)

[YearsNumber](#)

SelYear Property

The SelYear is a READ-ONLY property that allows you retrieve the current selected year in the calendar control. This property is very useful when you want to obtain the selected year without having to access the selected date (SelDate property) and trim it to obtain such value.

Visual Basic

```
nYear = [form.] Chart1.SelYear
```

Visual C++

```
nYear = sfxCalen1->GetNumProperty("SelYear");
```

Borland Delphi

```
nYear := [Tform.] Chart1.SelYear ;
```

Remarks

This property is available at RUN TIME only.

Data Type

Integer

See Also

[SelDate](#), [SelDay](#), [SelMonth](#)

SelMonth Property

The SelYear is a READ-ONLY property that allows you retrieve the current selected month in the calendar control. This property is very useful when you want to obtain the selected month without having to access the selected date (SelDate property) and trim it to obtain such value.

Visual Basic

```
nMonth = [form.] Chart1.SelMonth
```

Visual C++

```
nMonth = sfxCalen1->GetNumProperty("SelMonth");
```

Borland Delphi

```
nMonth := [Tform.] Chart1.SelMonth ;
```

Remarks

This property is available at RUN TIME only.

Data Type

Integer

See Also

[SelDate](#), [SelDay](#), [SelYear](#).

SelDay Property

The SelYear is a READ-ONLY property that allows you retrieve the current selected day in the calendar control. This property is very useful when you want to obtain the selected day without having to access the selected date (SelDate property) and trim it to obtain such value.

Visual Basic

```
nDay = [form.] Chart1.SelDay
```

Visual C++

```
nDay = sfxCalen1->GetNumProperty("SelDay");
```

Borland Delphi

```
nDay := [Tform.] Chart1.SelDay ;
```

Remarks

This property is available at RUN TIME only.

Data Type

Integer

See Also

[SelDate](#), [SelYear](#), [SelMonth](#)

LeftGap Property

This property controls the gap between the left side of the control and the calendar itself

Visual Basic

```
[form.] Chart1.LeftGap [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("LeftGap");  
sfxCalen1->SetNumProperty("LeftGap", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.LeftGap [ := setting& ];
```

Remarks

In order to control the Bottom and Right gap, you must drag the bottom-right corner of the control to the desired distance.

Data Type

Integer

Default

10 pixels

See Also

[TopGap](#)

TopGap Property

This property controls the gap between the left side of the control and the calendar itself

Visual Basic

```
[form.] Chart1.TopGap [ = setting& ]
```

Visual C++

```
lAuto = pChart1->GetNumProperty("TopGap");  
sfxCalen1->SetNumProperty("TopGap", lSetting);
```

Borland Delphi

```
[Tform.] Chart1.TopGap [ := setting& ];
```

Remarks

In order to control the Bottom and Right gap, you must drag the bottom-right corner of the control to the desired distance.

Data Type

Integer

Default

10 pixels

See Also

[LeftGap](#)

SelChange event

This event is sent every time the user has changed the current selected date.

The prototype for this event is: (Visual Basic)

```
Sub SFXCalen1_SelChange (nMonth As Integer, nDay As Integer, nYear As Integer)
```

The Selchange event has the following parts:

Part	Description
<i>nMonth</i>	New selected month
<i>nDay</i>	New selected day
<i>nYear</i>	New selected year

See Also

[ShowChange event](#)

ShowChange event

This event is sent everytime the actual month is changed by interacting with the TopBar combos or buttons, and also when the user has pressed any of the jumping days in the calendar control.

The prototype for this event is: (Visual Basic)

```
Sub SFXCalen1_ShowChange (nMonth As Integer, nYear As Integer)
```

The Selchange event has the following parts:

Part	Description
<i>nMonth</i>	New selected month
<i>nYear</i>	New selected year

See Also

[SelChange event](#)

