(About) Property

Applies To

Displays version information about the control.

Usage

Click on the ellipses ('...') button next to the property text to display the About dialog box.

Remarks

Version information regarding Calendar Widgets can be obtained by accessing this property.

This property is available only at design time.

(About) Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

(Custom) Property

See Also Applies To

Displays Property Pages for the control.

Usage

Click on the ellipses ('...') button next to the property text to activate the Property Pages dialog box.

Remarks

Property Pages allow you to set all properties, including collections, persisted by the control and properties that are not otherwise available at design time.

This property is available only at design time.

(Custom) Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

See Also

Property Pages

About StyleSets

See Also

To understand **StyleSet** objects and the **StyleSets** collection, you should become familiar with the concept of *collections* .
A StyleSet is an object that contains a set of visual properties. In Calendar Widgets, the MonthView, YearView and DateCombo make use of **StyleSet** objects.

See Also

Object Concepts

Accessing Property Pages

The method you use to access the Property Pages of your control depends on two things: the version of the control you are using, and the host environment in which you are using the control.

Many host environments support the use of the right mouse button to pop up a context-specific menu. In these environments, you simply click on your control with the right mouse button, and choose 'Property Pages' or 'Properties' from the pop-up menu.

If you are using the VBX version of a control, this behavior may not be supported. Instead, use the property sheet of your design environment. You will see a property labeled '(Custom)' in the property sheet. By double-clicking this property or choosing the ellipsis (...) button, you can invoke the Property Pages for the selected control.
If neither of these methods are supported, you will need to consult the documentation of your host environment for information on how to change the properties of objects. You may need to choose a special menu option, or perform a shifted mouse-click or double-click on the control. Try searching your environment’s on-line help file for references to objects, embedded objects, object properties, object settings, OLE custom controls, OCX controls or properties.

Add Method

See Also Example Applies To

Used to add an item to a collection.

Syntax

object . **Add** *begintime* , *endtime* , *text* , *backcolor*

object . **Add** *selecteddate*

object . **Add** *stylesetname*

The **Add** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
begintime	Required. Used with **Tasks**. A string expression specifying the beginning time of the **Task** object you want to add.
endtime	Required. Used with **Tasks**. A string expression specifying the ending time of the **Task** object you want to add.
text	Required. Used with **Tasks**. A string expression specifying the text description of the **Task** object you want to add.
backcolor	Optional. Used with **Tasks**. A value or constant specifying the background color of the **Task** object you want to add.
selecteddate	Required. Used with **SelectedDays**. A string or date expression specifying the date of the **Day** object you want to add.
stylesetname	Required. Used with **StyleSets**. A string expression specifying the name of the **StyleSet** object you want to add.

Remarks

For the **DayView** control's **Task** objects, if the *backcolor* parameter is not set, a color will be automatically assigned. You can modify the color later.

The *begintime* and *endtime* parameters can also be set to 'Noon' and 'Midnight', representing 12:00 PM and 12:00 AM respectively.

Add Method Applies To

SelectedDays Collection
StyleSets Collection
Tasks Collection

Add Method Example

The following code adds a Task object to the **Tasks** collection:

```
SSDay1.X.Tasks.Add "12:30PM", "2:00PM", "Meeting", RGB(255,0,0)
```

The following code adds a Day object to the **SelectedDays** collection:

```
SSMonth1.X.SelectedDays.Add "12/14/95"
```

The following code adds a StyleSet to the **StyleSets** collection:

```
SSDateCombo1.X.StyleSets.Add "Holiday"
```

See Also

- **Count** Property
- **Remove** Method
- **RemoveAll** Method
- **SelectedDays** Collection
- **StyleSets** Collection
- **Tasks** Collection

AllowAdd Property

See Also Applies To

Determines if a new task can be added by the user at run time.

Syntax

object . **AllowAdd** [= *boolean*]

The **AllowAdd** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if a new task can be added to the DayView control.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Tasks can be added at run time by the user.
False	Tasks cannot be added at run time by the user.

AllowAdd Property Applies To

DayView Control

See Also

AllowDelete Property

AllowEdit Property

AllowDelete Property

See Also Applies To

Determines if an existing task can be deleted by the user at run time.

Syntax

object . **AllowDelete** [= *boolean*]

The **AllowDelete** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if an existing task can be deleted from the DayView control.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Tasks can be deleted at run time by the user.
False	Tasks cannot be deleted at run time by the user.

AllowDelete Property Applies To

DayView Control

See Also

AllowAdd Property

AllowEdit Property

AllowEdit Property

See Also Applies To

Determines if the user can directly edit the contents of the control at run time.

Syntax

object . **AllowEdit** [= *boolean*]

The **AllowEdit** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if you can directly edit the contents of the control at run time.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Text can be changed at run time by the user.
False	Text cannot be changed at run time by the user.

AllowEdit Property Applies To

DateCombo Control
DayView Control

See Also

- **AllowAdd** Property
- **AllowDelete** Property
- **EditVisible** Property
- **MaxLength** Property
- **ShowEdit** Event

AllowNullDate Property

See Also Applies To

For **DateCombo**, determines if a blank date can be entered in the edit portion at run time.

For **MonthView** and **YearView**, determines if the selected day can be de-selected.

Syntax

object . **AllowNullDate** [= *boolean*]

The **AllowNullDate** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying whether or not a blank date can be entered, or day can be de-selected.

Settings

The settings for *boolean* are:

Setting	Description
True	Allowed.
False	(Default) Not Allowed.

AllowNullDate Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

Date Property

SelectedDays Collection

An Introduction to OCX Controls

Applying StyleSets

Once the StyleSet is created, it can be applied to an object that has a **StyleSet** property. The following code applies the 'vacation' StyleSet to the **Day** object of a MonthView control:

```
SSMonth1.X.Day(1).StyleSet = "vacation"
```

In Calendar Widgets, **StyleSet** objects can be applied to both the **Day** and **DayOfWeek** objects. For more information, see Example 2: Using a StyleSet in the Guided Tours for the MonthView control.

StyleSet objects have the following properties:

Properties

BackColor	Name	PictureMetaHeight
Font	Picture	PictureMetaWidth
ForeColor	PictureAlignment	CaptionAlignment

Note Not all properties are used by every object that exposes a **StyleSet** property. For example, the **CaptionStyleSet** property only uses the **Font** and **ForeColor** properties.

The following properties and methods can be used with the StyleSets collection:

Properties

Count

Methods

Add	Remove	RemoveAll
Item		

Note If a change is made to a StyleSet, it does not have to be reapplied to an object to take effect. But the control may have to be redrawn by invoking the **Refresh** method.

AutoRestore Property

Applies To

Determines if the selection will be restored to the original value in the database when the ESC key is pressed.

For the **DateCombo** control, this also determines if an invalid date will automatically be restored to the last valid date.

Syntax

object . **AutoRestore** [= *boolean*]

The **AutoRestore** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if date will be restored.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Date will be restored.
False	Date will not be restored.

AutoRestore Property Applies To

DateCombo Control
MonthView Control
YearView Control

AutoSelect Property

See Also Applies To

Determines if giving focus to a date automatically selects the date.

Syntax

object . **AutoSelect** [= *boolean*]

The **AutoSelect** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if the focus date will automatically become the selected date.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default for DateCombo) Focus date will automatically become the selected date.
False	(Default for MonthView and YearView) Focus date will not automatically become the selected date.

AutoSelect Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

Date Property

FocusDate Property

AutoValidate Property

See Also Applies To

Determines if validation of text in the edit portion will occur when the control loses focus.

Syntax

object . **AutoValidate** [= *boolean*]

The **AutoValidate** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if validation is automatically performed when the control loses focus.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Automatic validation occurs.
False	Automatic validation does not occur.

Remarks

If the **AutoValidate** property is set to **True** and an invalid date is in the edit portion when the control loses focus, the DateError event will be triggered.

AutoValidate Property Applies To

DateCombo Control

See Also

DateError Event

BackColor Property

Applies To

For DateCombo, returns or sets the color of the edit portion.

For DayView, returns or sets the default **BackColor** color for the task area.

For the **Task** object, returns or sets the color used to show the duration of the task in the Time Selection Bar.

Syntax

object . **BackColor** [= *color*]

The **BackColor** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
color	A value or constant that determines the background color.

Remarks

For the **Task** object, this property sets the color for the task in the Time Selection Bar

BackColor Property Applies To

DateCombo Control
DayView Control
StyleSet Object
Task Object

BackColorSelected Property

See Also Applies To

Returns or sets the background color of the selected day.

Syntax

object . **BackColorSelected** [= *color*]

The **BackColorSelected** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
color	A value or constant that determines the background color of the selected days or tasks in the control.

Remarks

For the **DateCombo**, this only affects the dropdown calendar portion.

BackColorSelected Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

See Also

- **Date** Property
- **ForeColorSelected** Property
- **SelectedDays** Collection

BeepOnError Property

Applies To

Determines if a beep will sound when you press an invalid key.

Syntax

object . **BeepOnError**[= *boolean*]

The **BeepOnError** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if a beep will sound when you enter an invalid key.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) A beep will sound.
False	A beep will not sound.

BeepOnError Property Applies To

DateCombo Control

BeginTime Property

See Also Example Applies To

Specifies the beginning time of the task.

Syntax

object . **BeginTime** [= *text*]

The **BeginTime** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the beginning time.

Remarks

The **BeginTime** cannot be set earlier than the setting of the **TimeBegin** property of the **DayView** control. However, if **TimeBegin** is subsequently set later than **BeginTime**, the task will still be available even though it will no longer appear on the **DayView** control.

The **BeginTime** property can also be set to 'Noon' or 'Midnight', representing 12:00 PM and 12:00 AM respectively.

BeginTime Property Applies To

Task Object

BeginTime Property Example

This example shows how to set the beginning time of the second task to the ending time of the first task in a **DayView** control:

```
SSDay1.X.Tasks(1).BeginTime = SSDay1.X.Tasks(0).EndTime
```

See Also

- **Duration** Property
- **EndTime** Property
- **Task** Object
- **TimeBegin** Property

BevelColorFace Property

See Also Applies To

Returns or sets the color of the bevel face.

Syntax

object . **BevelColorFace** [= *color*]

The **BevelColorFace** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
color	A value or constant that determines the color of the control’s bevel-face.

Remarks

The **BevelColorScheme** property must be set to '2 - Custom Colors' for this property to take effect.

BevelColorFace Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

See Also

- **BevelColorFrame** Property
- **BevelColorHighlight** Property
- **BevelColorScheme** Property
- **BevelColorShadow** Property

BevelColorFrame Property

See Also Applies To

Returns or sets the color of the bevel frame.

Syntax

object . **BevelColorFrame** [= *color*]

The **BevelColorFrame** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
color	A value or constant that determines the color of the control’s frame.

Remarks

The setting for this property is used both in the border of the control and between months in a multi-month display. The **BevelColorScheme** property must be set to '2 - Custom Colors' for this property to take effect.

BevelColorFrame Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

See Also

- **BevelColorFace** Property
- **BevelColorHighlight** Property
- **BevelColorScheme** Property
- **BevelColorShadow** Property

BevelColorHighlight Property

See Also Applies To

Returns or sets the color of the bevel highlight.

Syntax

object . **BevelColorHighlight [**= *color*]

The **BevelColorHighlight** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
color	A value or constant that determines the color of the control’s bevel-highlight.

Remarks

The **BevelColorScheme** property must be set to '2 - Custom Colors' for this property to take effect.

BevelColorHighlight Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

See Also

- **BevelColorFace** Property
- **BevelColorFrame** Property
- **BevelColorScheme** Property
- **BevelColorShadow** Property

BevelColorScheme Property

See Also Applies To

Returns or sets the color scheme of the bevel areas.

Syntax

object . **BevelColorScheme** [= *number*]

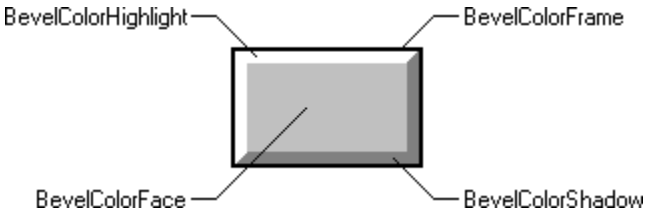
The **BevelColorScheme** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the color scheme to be used.

Settings

The settings for *number* are:

Setting	Description
0	Gray Colors Face = Light Gray Frame = Black Highlight = White Shadow = Dark Gray
1	System Colors Uses Windows-defined bevel colors: Face = Button Face Frame = Window Frame Highlight = Button Highlight Shadow = Button Shadow
2	(Default) Custom Colors Uses colors set in **BevelColorFace**, **BevelColorFrame, BevelColorHighlight** and **BevelColorShadow** properties.



BevelColorScheme Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

See Also

- **BevelColorFace** Property
- **BevelColorFrame** Property
- **BevelColorHighlight** Property
- **BevelColorShadow** Property

BevelColorShadow Property

See Also Applies To

Returns or sets the color of the bevel shadow.

Syntax

object . **BevelColorShadow** [= *color*]

The **BevelColorShadow** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
color	A value or constant that determines the color of the control’s bevel-shadow.

Remarks

The **BevelColorScheme** property must be set to '2 - Custom Colors' for this property to take effect.

BevelColorShadow Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

See Also

- **BevelColorFace** Property
- **BevelColorFrame** Property
- **BevelColorHighlight** Property
- **BevelColorScheme** Property

BevelType Property

Applies To

Sets the type of bevel to be used around the control.

Syntax

object . **BevelType** [= *number*]

The **BevelType** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the type of bevel to use.

Settings

The settings for *number* are:

Setting	Description
0	None
1	(Default for DateCombo) Inset
2	(Default for DayView) Raised

Remarks

For the **DateCombo** control, it sets the bevel type of the combo box.

For the **DayView** control, it sets the bevel type of the outside control area.

BevelType Property Applies To

DateCombo Control
DayView Control

BevelWidth Property

See Also Applies To

Returns or sets the width of the control's bevel in pixels.

Syntax

object . **BevelWidth** [= *number*]

The **BevelWidth** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression that evaluates to the width of the bevel.

Remarks

The valid range for *number* is 0 to 10.

For the **DateCombo** control, it determines the width of the bevel around the combo portion (not the dropdown calendar portion).

For the **DayView**, **MonthView** and **YearView** controls, it determines the width of the bevel around the control.

BevelWidth Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

See Also

BevelType Property

CaptionBevelWidth Property

Binding Across Forms

When using the VBX version of the Calendar Widgets controls to bind across forms, you use the **DataSourceHwnd** property.

The following code binds a **MonthView** control on Form2 to a Visual Basic data control on Form1:

```
Sub Form2_Load( )
   SSMonth1.DataSourceHwnd = SSGetControlHwnd(Form1.Data1)
End Sub
```

The following code illustrates the binding of a **MonthView** control on Form2 to a Visual Basic data control that is bound to a **YearView** control on another form:

```
Sub Form2_Load( )
   SSMonth1.DataSourceHwnd = Form1.SSYear1.DataSourceHwnd
End Sub
```

Consult you host environment’s documentation and the README.TXT file for more information on how to bind across forms when using the OCX version of the controls.

Bold Property

See Also Applies To

Returns or sets the font style of the **Font** object to either bold or nonbold.

Syntax

object . **Bold** [= *boolean*]

The **Bold** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying the font style, as described in Settings.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Turns on bold formatting.
False	Turns off bold formatting.

Remarks

The **Font** object is not directly available at design time. Instead you set the **Bold** property through a control's **Font** property.

At run time, however, you can set **Bold** directly by specifying its setting for the **Font** object.

In Visual Basic 3.0 you set the **Bold** property by selecting a control's **FontBold**, **CaptionFontBold**, **DayFontBold**, **DropDownFontBold** or **TimeSelectionBarFontBold** property in the Visual Basic Properties window.

Bold Property Applies To

Font Object

See Also

Font Object

Font Property

Caption Property

See Also Applies To

Specifies the caption of the object or control.

Syntax

object . **Caption** [= *text*]

The **Caption** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the text displayed as the caption.

Remarks

For the **Day** object, this caption will appear within that day.

For the **DayofWeek** object, this caption will appear as the heading for that day of week for every month.

For the **Month** object, this caption will appear as the heading for that month for every year.

For the **DayView** and **YearView** controls, this appears in the caption area and is available at design time.

For the **Day**, **DayofWeek** and **Month** object, this property is available at run time and through the **Property Pages** at design time.

Caption Property Applies To

Day Object
DayofWeek Object
DayView Control
Month Object
YearView Control

See Also

- **Day** Object
- **DayofWeek** Object
- **Month** Object

CaptionAlignment Property

Applies To

Aligns the caption within the caption area.

Syntax

object . **CaptionAlignment** [= *number*]

The **CaptionAlignment** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the alignment of the caption.

Settings

The settings for *number* are:

Setting	Description
0	Left Top
1	Left Middle
2	Left Bottom
3	Right Top
4	Right Middle
5	Right Bottom
6	Center Top
7	(Default) Center Middle
8	Center Bottom

CaptionAlignment Property Applies To

DayView Control
YearView Control

CaptionAlignmentBeginYear Property

See Also Example Applies To

Aligns the Begin Year caption.

Syntax

object . **CaptionAlignmentBeginYear** [= *number*]

The **CaptionAlignmentBeginYear** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the alignment of the Begin Year caption.

Settings

The settings for *number* are:

Setting	Description
0	Left Justify
1	(Default) Right Justify
2	Left of Caption
3	Right of Caption

Remarks

If the **CaptionAlignmentBeginYear** and **CaptionAlignmentEndYear** properties are both the same, the year caption will be displayed with a hyphen.

CaptionAlignmentBeginYear Property Applies To

YearView Control

CaptionAlignmentBeginYear Property Example

If both the **CaptionAlignmentBeginYear** and **CaptionAlignmentEndYear** are set to '2 - Right of Caption' and **ShowCentury** is set to **True**, the caption will display the following:

See Also

CaptionAlignmentEndYear Property

CaptionAlignmentEndYear Property

See Also Applies To

Aligns the End Year caption if the control spans two years.

Syntax

object . **CaptionAlignmentEndYear** [= *number*]

The **CaptionAlignmentEndYear** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the alignment of the End Year caption.

Settings

The settings for *number* are:

Setting	Description
0	Left Justify
1	(Default) Right Justify
2	Left of Caption
3	Right of Caption

Remarks

If the **CaptionAlignmentBeginYear** and **CaptionAlignmentEndYear** properties are both the same, the year caption will be displayed with a hyphen.

This property has no effect if **StartMonth** is 1.

CaptionAlignmentEndYear Property Applies To

YearView Control

See Also

CaptionAlignmentBeginYear Property

CaptionAlignmentMonth Property

See Also Applies To

Returns or sets the alignment of the month caption.

Syntax

object . **CaptionAlignmentMonth** [= *number*]

The **CaptionAlignmentMonth** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the alignment of the month caption.

Settings

The settings for *number* are:

Setting	Description
0	Left Justify
1	Right Justify
2	(Default) Center

CaptionAlignmentMonth Property Applies To

DateCombo Control
MonthView Control

See Also

CaptionAlignmentYear Property

CaptionAlignmentYear Property

See Also Applies To

Returns or sets the alignment of the year caption.

Syntax

object . **CaptionAlignmentYear** [= *number*]

The **CaptionAlignmentYear** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the alignment of the year caption.

Settings

The settings for *number* are:

Setting	Description
0	Left Justify
1	(Default) Right Justify
2	Left of Month
3	Right of Month

Remarks

When **CaptionAlignmentYear** is set to '2 - Left of Month' or '3 - Right of Month', the year caption is placed just to the left or right of the month caption without a divider between them.

CaptionAlignmentYear Property Applies To

DateCombo Control
MonthView Control

See Also

CaptionAlignmentMonth Property

CaptionBackColor Property

See Also Applies To

Returns or sets the background color in the Caption area.

Syntax

object . **CaptionBackColor** [= *color*]

The **CaptionBackColor** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
color	A value or constant that determines the background color of the caption area.

CaptionBackColor Property Applies To

YearView Control

See Also

CaptionForeColor Property

CaptionBevelType Property

See Also Applies To

.Returns or sets the bevel type for the caption area.

Syntax

object . **CaptionBevelType** [= *number*]

The **CaptionBevelType** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the bevel type of the caption area.

Settings

The settings for *number* are:

Setting	Description
0	None. No bevel is drawn.
1	(Default for DateCombo, DayView and MonthView) Inset. The bevel appears inset on the screen.
2	(Default for YearView) Raised. The bevel appears raised off the screen.

CaptionBevelType Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

See Also

CaptionBevelWidth Property

CaptionBevelWidth Property

See Also Applies To

Returns or sets the width of the caption bevel in pixels.

Syntax

object . **CaptionBevelWidth** [= *number*]

The **CaptionBevelWidth** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression that evaluates to the bevel width of the caption area.

Remarks

The valid range for *number* is 0 to 10.

For the **MonthView** and **YearView** controls, this setting also affects the Selected Date and Today's Date buttons.

For the **YearView** control, this also affects the caption and Month Caption areas.

CaptionBevelWidth Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

See Also

BevelWidth Property

CaptionBevelType Property

CaptionClick Event

See Also Applies To

Occurs when you click the mouse in the Caption area.

Syntax

Sub *object* **_CaptionClick** ([*index* **As Integer**])

The CaptionClick event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
index	An integer expression that uniquely identifies the control if it is in a control array.

Remarks

You can use the **WhereIs** method to determine where in the Caption you have clicked.

CaptionClick Event Applies To

YearView Control

See Also

WhereIs Method

CaptionFont Property

See Also Example Applies To

Returns a **Font** object used for the Caption area.

Syntax

object . **CaptionFont**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

Use the **CaptionFont** property of an object to identify a specific **Font** object to use for displaying text in the caption area.

This property is not available at design time when using the VBX version of the control.

CaptionFont Property Applies To

DayView Control
YearView Control

CaptionFont Property Example

The following code changes the **Bold** property setting of a **Font** object identified by the **CaptionFont** property of a **YearView** control:

```
SSYear1.CaptionFont.Bold = True
```

See Also

- **CaptionFontBold** Property
- **CaptionFontItalic** Property
- **CaptionFontName** Property
- **CaptionFontSize** Property
- **CaptionFontStrikethru** Property
- **CaptionFontUnderline** Property
- **Font** Object

CaptionFont3D Property

See Also Applies To

Returns or sets the 3-D style of text in the Caption area.

Syntax

object . **CaptionFont3D** [= *number*]

The **CaptionFont3D** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the type of 3-D effect to use.

Settings

The settings for *number* are:

Setting	Description
0	(Default) None. Text is displayed flat (not 3-dimensional).
1	Raised w/ light shading. Text appears raised off the screen.
2	Raised w/ heavy shading. Text appears more raised.
3	Inset w/ light shading. Text appears inset on the screen.
4	Inset w/ heavy shading. Text appears more inset.

Settings 2 and 4 (heavy shading) look best with larger, more bold fonts.

CaptionFont3D Property Applies To

DayView Control
YearView Control

See Also

CaptionFont Property

CaptionHeight Property

CaptionFontBold, CaptionFontItalic, CaptionFontStrikethru, CaptionFontUnderline Properties

See Also Applies To

Returns or sets font styles in the following formats for the Caption area: **Bold**, *Italic*, ~~Strikethru~~, and Underline.

Syntax

object . **CaptionFontBold** [= *boolean*]

object . **CaptionFontItalic** [= *boolean*]

object . **CaptionFontStrikethru** [= *boolean*]

object . **CaptionFontUnderline** [= *boolean*]

The **CaptionFontBold**, **CaptionFontItalic**, **CaptionFontStrikethru** and **CaptionFontUnderline** property syntaxes have these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying the font style as described in Settings.

Settings

The settings for *boolean* are:

Setting	Description
True	Turns on the formatting in that style.
False	Turns off the formatting in that style.

Remarks

Use these font properties to format the Caption area's text, either at design time using the Properties window or at run time using code.

When using the VBX version of the controls, the **CaptionFontBold**, **CaptionFontItalic**, **CaptionFontStrikethru**, and **CaptionFontUnderline** properties are available at design time. These properties are supported in the OCX version of the controls for compatibility.

CaptionFontName Property

See Also Applies To

Returns or sets the font used to display text in the Caption area.

Syntax

object . **CaptionFontName** [= *font*]

The **CaptionFontName** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
font	A string expression specifying the font name to use.

Remarks

Use this font property to format the Caption area's text, either at design time using the Properties window or at run time using code.

When using the VBX version of the controls, the **CaptionFontName** property is available at design time. This property is supported in the OCX version of the controls for compatibility.

CaptionFontName Property Applies To

DayView Control
YearView Control

See Also

- **CaptionFont** Property
- **CaptionFont3D** Property
- **CaptionFontBold** Property
- **CaptionFontItalic** Property
- **CaptionFontSize** Property
- **CaptionFontStrikethru** Property
- **CaptionFontUnderline** Property
- **Font** Object
- Fonts

CaptionFontSize Property

See Also Applies To

Returns or sets the size of the font to be used for text in the Caption area.

Syntax

object . **CaptionFontSize** [= *points*]

The **CaptionFontSize** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
points	A numeric expression specifying the font size to use, in points.

Remarks

Use this property to format the Caption area's text in the font size you want.

When using the VBX version of the controls, the **CaptionFontSize** property is available at design time. This property is supported in the OCX version of the controls for compatibility.

CaptionFontSize Property Applies To

DayView Control
YearView Control

See Also

- **CaptionFont** Property
- **CaptionFont3D** Property
- **CaptionFontBold** Property
- **CaptionFontItalic** Property
- **CaptionFontName** Property
- **CaptionFontStrikethru** Property
- **CaptionFontUnderline** Property
- **Font** Object
- Fonts

CaptionFontBold, CaptionFont Italic, CaptionFontStrikethru, CaptionFontUnderline Properties Apply To

DayView Control
YearView Control

See Also

- **CaptionFont** Property
- **CaptionFont3D** Property
- **CaptionFontName** Property
- **CaptionFontSize** Property
- **Font** Object
- Fonts

CaptionForeColor Property

See Also Applies To

Returns or sets the foreground color in the Caption area.

Syntax

object . **CaptionForeColor** [= *color*]

The **CaptionForeColor** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
color	A value or constant that determines the text color of the caption.

CaptionForeColor Property Applies To

YearView Control

See Also

CaptionBackColor Property

CaptionHeight Property

See Also Applies To

Returns or sets the height of the caption area.

Syntax

object . **CaptionHeight** [= *number*]

The **CaptionHeight** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression that evaluates to the height of the caption area.

Remarks

The measurement is in the scale mode of the container. A setting of 0 (default) causes the height to be automatically calculated, based on the font size.

For the **DateCombo**, **MonthView** and **YearView** controls, this setting may affect the Day of Week area.

Depending on this setting, the height of the Day of Week area will be the smaller of either the Caption area or the height of any one of the days in the Days area.

CaptionHeight Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

See Also

- **CaptionFont** Property
- **CaptionFont3D** Property
- **MonthHeight** Property
- **ShowSelectedDate** Property
- **ShowTodaysDate** Property

CaptionPicture Property

See Also Example Applies To

Returns or sets the bitmap picture to appear in the Caption area.

Syntax

object . **CaptionPicture** [= *picture*]

The **CaptionPicture** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
picture	A expression specifying a graphic, as described in Settings.

Settings

The settings for *picture* are:

Setting	Description
(None)	(Default) No picture.
(Bitmap, icon, metafile)	Specifies a graphic. You can load the graphic from the Properties window at design time. At run time, you can also set this property using the name of the file containing the graphic.

CaptionPicture Property Applies To

YearView Control

CaptionPicture Property Example

The following code places a picture in the Caption area of a **YearView** control:

```
SSYear1.CaptionPicture = LoadPicture("MONEY.BMP")
```

The following code also sets the picture in the Caption area:

```
SSYear1.CaptionPicture = Picture1.Picture
```

See Also

CaptionPictureAlignment Property
CaptionPictureMetaHeight Property
CaptionPictureMetaWidth Property
Pictures

CaptionPictureAlignment Property

See Also Applies To

Returns or sets the alignment of the Caption Picture.

Syntax

object . **CaptionPictureAlignment** [= *number*]

The **CaptionPictureAlignment** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the alignment of the Caption Picture.

Settings

The settings for *number* are:

Setting	Description
0	Left Top
1	Left Middle
2	Left Bottom
3	Right Top
4	Right Middle
5	Right Bottom
6	Center Top
7	Center Middle
8	Center Bottom
9	(Default) Left of Caption
10	Right of Caption
11	Above Caption
12	Below Caption
13	Fit to Caption
14	Tile

CaptionPictureAlignment Property Applies To

YearView Control

See Also

CaptionPicture Property
CaptionPictureMetaHeight Property
CaptionPictureMetaWidth Property

CaptionPictureMetaHeight, CaptionPictureMetaWidth Properties

See Also Applies To

Sets the height and width of a metafile selected as a Caption Picture.

Syntax

object . **CaptionPictureMetaHeight** [= *number*]

object . **CaptionPictureMetaWidth** [= *number*]

The **CaptionPictureMetaHeight** and **CaptionPictureMetaWidth** property syntaxes have these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression that evaluates to the height or width of a metafile selected as a Caption Picture.

Remarks

The units specified are based on the scale mode of the container.

The **CaptionPictureMetaHeight** property sets the height of the picture in the Caption Picture area if it is a metafile.

The **CaptionPictureMetaWidth** property sets the width of the picture in the Caption Picture area if it is a metafile.

CaptionPictureMetaHeight, CaptionPictureMethWidth Properties Apply To

YearView Control

See Also

CaptionPicture Property

CaptionPictureAlignment Property

CaptionStyleSet Property

See Also Applies To

Returns or sets a StyleSet override for the Day or Month caption.

Syntax

object . **CaptionStyleSet** [= *text*]

The **CaptionStyleSet** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the name of a StyleSet.

Remarks

The property settings for this object will override any default settings. For example:

```
SSMonth1.X.Day("05/30/95").CaptionStyleSet = "Holiday"
```

-Supersedes-

```
SSMonth1.X.DayCaptionStyleSet = "Default Day"
```

Only the **ForeColor** and **Font** properties of the StyleSet are used.

This property is available at run time and through the **Property Pages** at design time.

CaptionStyleSet Property Applies To

Day Object
Month Object

See Also

- **Day** Object
- Fonts
- **Month** Object
- **StyleSet** Property
- **StyleSets** Collection

Supporting Cast
(in order of appearance)

Cathy
Teddy
Rosanne
Melissa
Michael
Nicole
Dawn
Nick
Haley
Eileen
Clyde
Kym
Lori
Kim
Tony
Dana
Patricia
& The Weasel

ClipMode Property

See Also Example Applies To

Returns or sets the clip mode of the control.

Syntax

object . **ClipMode** [= *number*]

The **ClipMode** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the clip mode of the control.

Settings

The settings for *number* are:

Setting	Description
0	(Default) Includes literals on cut or copy.
1	Excludes literals on cut or copy.

Remarks

The **ClipMode** property of the control determines whether literals, such as date separators, are included in the text during a cut or copy operation.

You can cause a cut or copy by using one of the following keyboard conventions:

Action	Keyboard Convention
Cut	CTRL + X SHIFT + DELETE
Copy	CTRL + C SHIFT + DELETE

ClipMode Property Applies To

DateCombo Control

ClipMode Property Example

If the text from the following **DateCombo** is cut or copied, while the **ClipMode** is set to **0**, the contents of the Windows Clipboard will be '11/14/95'.

If the **ClipMode** is set to **1**, the contents of the Windows Clipboard will be '111495'

See Also

ClipText Property

ClipText Property

See Also Example Applies To

Returns the text that will be placed on the Windows Clipboard during a copy or cut operation.

Syntax

object . **ClipText** [= *text*]

The **ClipText** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the text to be placed on the Clipboard during a copy or cut operation.

Remarks

The format of the text sent to the Clipboard depends on the setting of the **ClipMode** property.

This property is not available at design time and is read-only at run time.

ClipText Property Applies To

DateCombo Control

ClipText Property Example

If the **ClipMode** property is set to 1, the **ClipText** property of the following **DateCombo** would return '111495'. If **ClipMode** were set to 0, **ClipText** would return '11/14/95'.

See Also

- **ClipMode** Property
- **FormattedText** Property
- **RawText** Property

CloseEdit Event

See Also Applies To

Occurs when the edit box in a time slot closes.

Syntax

Sub *object* **_CloseEdit**(*TaskIndex* **As Integer**, *Action* **As Integer**, *Cancelled* **As Integer**, *Changed* **As Integer**)

The CloseEdit event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
TaskIndex	An Integer expression that evaluates to the index of the task in the **Tasks** collection.
Action	An integer expression that indicates whether the task was added (0) or edited (1).
Cancelled	An integer expression that indicates whether the user has cancelled the add or edit by pressing the ESC key.
Changed	An integer expression that indicates whether or not a change was made.

Remarks

This event is the reverse of the ShowEdit event.

CloseEdit Event Applies To

DayView Control

See Also

EditVisible Property
ShowEdit Event
Tasks Collection

CloseUp Event

See Also Applies To

Occurs when the dropdown calendar portion is closed.

Syntax

Sub *object* **_CloseUp**( )

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

This event is the reverse of the **DropDown** event.

Some of the ways the **CloseUp** event is fired when the dropdown calendar portion is down are:

§ When a date is selected from the dropdown calendar.

§ The DropDown button is pressed.

§ The **DateCombo** loses focus.

§ Either F4, ALT + UP ARROW or ALT + DOWN ARROW is pressed.

CloseUp Event Applies To

DateCombo Control

See Also

DropDown Event

DroppedDown Property

Collection Summary

See Also

S

SelectedDays
StyleSets

T

Tasks

Compatibility Issues

Calendar Widgets™

Copyright Notice Credits

The Calendar Widgets Controls

Description of the Calendar Widgets controls and how to use them.

Guided Tours

A step-by-step guide to using Calendar Widgets.

Control Reference

An alphabetical listing of all programming language topics.

Properties
Events
Methods
Objects
Collections
StyleSets
Property Pages
Trappable Errors

Compatibility Issues

A list of considerations when using the VBX or OCX versions of the controls.

Technical Specifications

A list of system requirements and included files.

Technical Support

Getting technical and product support for Sheridan products.

Control Limitations

The limitations for the DayView control are:

§ A maximum of 10 overlapping tasks in the same time slot.

§ A maximum of 255 tasks per day.

Converting from VBX to OCX Controls

Conversion of VBX to OCX controls occurs automatically when you upgrade a Visual Basic project from version 3.0 to version 4.0. When you open the project in Visual Basic 4.0 for the first time, Visual Basic 4.0 will search your system for an OCX version of each VBX used in your Visual Basic 3.0 project. If an OCX version of the control exists, Visual Basic 4.0 will substitute it for the VBX in the upgraded project file.

Note Before upgrading a Visual Basic 3.0 project, you must make sure all the forms in your project are saved **as text** and not in binary format. Forms saved in binary format cannot be reliably upgraded from version 3.0 to 4.0. To save a form as text, select the form, choose "Save File As..." from the Visual Basic 3.0 File menu, check the "Save As Text" checkbox in the File Save dialog and click OK.

This on-line document was produced using Microsoft Word for Windows from Microsoft Corporation and ForeHelp, from ForeFront Incorporated.

Count Property

See Also Example Applies To

Returns the number of items in a collection.

Syntax

object . **Count**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

This property can be used to loop through all of the items in a collection.

Note Collections are zero-based. Therefore the index of the last item in a collection will always be equal to the value of Count - 1.

Count Property Applies To

SelectedDays Collection
StyleSets Collection
Tasks Collection

Count Property Example

Assuming you want to mark a group of selected days on a **MonthView** control as vacation days, the following code loops through the days, sets their **BackColor** to yellow so they stand out.

```
Dim i As Integer
SSMonth1.X.Stylesets("Vacation").BackColor = RGB(255,255,0)
For i = 0 to (SSMonth1.X.SelectedDays.Count - 1)
   SSMonth1.X.SelectedDays(i).Styleset="Vacation"
Next
SSMonth1.X.SelectedDays.RemoveAll
```

See Also

- **Add** Method
- **RemoveAll** Method
- **Remove** Method
- **SelectedDays** Collection
- **StyleSets** Collection
- **Tasks** Collection

Creating StyleSets

In the case of the MonthView control, different StyleSets can be created and applied to specific **Day** or **DayofWeek** objects. Each of these stylesets can in-turn be given characteristics which make one stand out from the other. For example, a 'birthday' StyleSet may have its **Picture** property set to a picture of a birthday cake and its **BackColor** property set to 'red', while a 'vacation' StyleSet may have its **Picture** property set to a picture of a boat and its **BackColor** property set to 'yellow.'

The following is an example of how a 'vacation' StyleSet may be set up:

1. The StyleSet is first added to the StyleSets collection as follows:
   ```
   SSMonth1.X.StyleSets.Add "vacation"
   ```
2. Once added, the properties of a StyleSet may be set as follows:
   ```
   SSMonth1.X.StyleSets("vacation").BackColor = RGB(255,255,0)
   SSMonth1.X.StyleSets("vacation").Picture = "vacation.ico"
   ```

Note Making reference to any of the properties in a StyleSet automatically adds it to the **StyleSets** collection. Therefore, invoking the **Add** method on the **StyleSets** collection is optional.

Calendar Widgets

Starring	Mary S.
Director	Ned R.
Screenplay	Rajeev M.
Lighting	Bill B.
Special Effects	Rob S.
Costumes	Deb S.
Public Relations	Dan W.
Music	Tony A.
Writer	Jim D.
Editor	Jason M.
Producers	Bob W. Joe D. Joe M.
Key Grip	John C.
Best Boy	Saro K.
Gaffer	Camille W.

Supporting Cast

DataSourceHwnd Property

See Also Example Applies To

Returns or sets the window handle (**hWnd**) of a data control to bind this control to at run time.

Syntax

object . **DataSourceHwnd** [= *hWnd*]

The **DataSourceHwnd** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
hWnd	An integer expression that evaluates to the window handle of the data control to bind the control to at run time.

Remarks

Note This property is only available in the VBX version of the controls.

The purpose of this property is to allow you to set the data source of the control at run time. Set this property to the window handle of the data control to bind the control to. If you set the **DataSource** property at design time, this property will automatically be set to the **hWnd** of the data control at run time and will be read-only.

The main benefit of this property is to be able to set the data source to a data control on a different form.

Note Unfortunately, Visual Basic does not supply the **hWnd** property for the data control. To get the **hWnd** for a data control, you can either use the **DataSourceHwnd** property of any Sheridan data-aware control that is bound to a data control, or call an API function called **SSGetControlHwnd** in any of Sheridan's VBX files. See the **Declare** statement for this API in the SSCALWDG.BAS file that comes with Calendar Widgets.

This property is only available at run time.

DataSourceHwnd Property Applies To

DateCombo Control
MonthView Control
YearView Control

DataSourceHwnd Property Example

The following code sets the data source of a MonthView control on Form2 to the data source of a data control on Form1:

```
Sub Form2_Load( )
   SSMonth1.DataSourceHwnd = SSGetControlHwnd(Form1.Data1)
End Sub
```

See Also

Binding Across Forms

Date Property

See Also Example Applies To

Returns or sets the date.

Syntax

object . **Date** [= *text*]

The **Date** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to a date to be set.

Remarks

For **MonthView** and **YearView** controls, it adds the specified date to the **SelectedDays** collection if the **SelectionType** property is set to MultiSelect. It de-selects the previously selected date and selects the specified date if the **SelectionType** property is set to Single Select.

This property is only available at run time.

Date Property Applies To

DateCombo Control
Day Object
MonthView Control
YearView Control

Date Property Example

The following code returns the date for the selected day on a **MonthView** control:

```
ReturnDate = SSMonth1.SelectedDays(0).Date
```

To jump to a specific date in the **YearView** control, you could use the following code:

```
SSYear1.SelectionType = 0    'Single Select
SSYear1.Date = "07/04"       'Selects July 4th of the displayed year
```

See Also

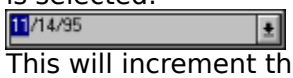
Date Property

SelectedDays Collection

DateCombo (When Dropped Down)

Press	To	Comments
LEFT ARROW	Focus on Previous Day	If previous day is visible, only the focus changes; otherwise, the control scrolls.
RIGHT ARROW	Focus on Next Day	If next day is visible, only the focus changes; otherwise, the control scrolls.
UP ARROW	Focus on Same Day of Previous Week	If previous week is visible, only the focus changes; otherwise, the control scrolls.
DOWN ARROW	Focus on Same Day of Next Week	If next week is visible, only the focus changes; otherwise, the control scrolls.
CTRL+LEFT ARROW	Focus on Same Day of Previous Month	If previous month is visible, only the focus changes; otherwise, the control scrolls to the current day in the previous month.
CTRL+RIGHT ARROW	Focus on Same Day of Next Month	If next month is visible, only the focus changes; otherwise, the control scrolls to the current day in the next month.
PAGE UP	Focus on Same Day of Previous Year	Scrolls to the current month in the previous year.
PAGE DOWN	Focus on Same Day of Next Year	Scrolls to the current month in the next year.
HOME	Focus on First Day of Month	Sets the focus to the first day of the current month.
END	Focus on Last Day of Month	Sets the focus to the last day of the current month.
ENTER / SPACEBAR	Select Day with Focus	Selects the day with focus and closes the dropdown calendar.
ESCAPE	Cancel Selection	Cancels any selection made through the dropdown calendar, restores the date to the date that was in the edit portion before the control was dropped down and closes up the dropdown calendar.
CTRL+S	Focus on Selected Date	Sets the focus on the selected date.
CTRL+T	Focus on Today	Sets the focus on today’s date.
ALT+UP ARROW / ALT+DOWN ARROW / F4	Close Up DateCombo	Closes up the dropdown calendar portion of the control.

DateCombo (When Not Dropped Down)

Press	To	Comments
LEFT ARROW	Move Left	Moves the cursor one position to the left. If the **EditMode** is set to '1 - Month Day, Year' mode, moves cursor to the next area.
RIGHT ARROW	Move Right	Moves the cursor one position to the right. If the **EditMode** is set to '1 - Month Day, Year' mode, moves cursor to the previous area.
UP ARROW / +	Increments Date/ Month/Day/Year	If the **EditMode** property is set to '0 - Whole Date', it increments the entire date by one day. If **EditMode** is set to '1 - Month Day, Year', the part of the date that is currently selected will be incremented. For example, if the month portion of the date is selected:  This will increment the month by one and display: 
DOWN ARROW / -	Decrements Date/ Month/Day/Year	Same as UP ARROW/+ except this decrements.
CTRL+LEFT ARROW / HOME	Go to Beginning	Places the cursor at the beginning of the edit portion when **EditMode** is set to '0 - Whole Date'.
CTRL+RIGHT ARROW / END	Go to End	Places the cursor at the end of the edit portion when **EditMode** is set to '0 - Whole Date'.
ESCAPE	Cancel Selection	If control is bound and AutoRestore is True, resets selected date to the date in the database.
CTRL+T	Select Today's Date	Selects today's date.
ALT+UP ARROW / ALT+DOWN ARROW / F4	Drop Down DateCombo	Drops down the dropdown calendar portion of the control.

The DateCombo Control

See Also Properties Events Methods Objects Collections

The DateCombo control is ideal for creating data entry forms that require date information to be entered. With a variety of options for input masking and date formatting, the DateCombo helps facilitate date selection. You can also use the DateCombo in applications where a monthly view of dates is necessary but screen space is limited. The DateCombo lets you select dates from a dropdown calendar that looks and acts just like the MonthView control. When date selection is complete, the dropdown calendar closes up and out of the way. Like the MonthView and YearView controls, the DateCombo control is also data aware.

File Name SSCALA.VBX, SSCALA16.OCX, SSCALA32.OCX

ObjectType SSDateCombo

Keyboard Interface
Masking and Formatting
Marking Dates
Customizing

Back

DateCombo Control Collections

Collections marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

StyleSets Collection õ

Back

DateCombo Control Events

Events marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

Change Event
Click Event
CloseUp Event
DateError Event
DblClick Event
DragDrop Event
DragOver Event
DropDown Event
FocusChange Event
GotFocus Event
InitMonth Event
KeyDown Event
KeyPress Event
KeyUp Event
LostFocus Event
MonthClick Event
MouseDown Event
MouseMove Event
MouseUp Event
Spin Event
YearClick Event

Back

DateCombo Control Methods

Methods marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

IsDateValid Method õ

Refresh Method

WeekNumber Method õ

Back

DateCombo Control Objects

Objects marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

Day Object õ
DayofWeek Object õ
Font Object
Month Object õ
X Object

Back

DateCombo Control Properties

Properties marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

DragMode Property
DropDownBevelWidth Property
DropDownFont Property
DropDownFont3D Property
DropDownFontBold Property
DropDownFontItalic Property
DropDownFontName Property
DropDownFontSize Property
DropDownFontStrikethru Property
DropDownFontUnderline Property
DropDownForeColor Property
DropDownHeight Property
DropDownMouseIcon Property
DropDownMousePointer Property
DropDownWidth Property
DroppedDown Property
EditMode Property
Enabled Property
FocusDate Property
Font Property
FontBold Property
FontItalic Property
FontName Property
FontSize Property
FontStrikethru Property
FontUnderline Property
ForeColor Property
ForeColorSelected Property
Format Property
FormattedText Property
Height Property
HelpContextId Property
hWnd Property
Index Property
Left Property
Mask Property
MaxDate Property
MinDate Property
Month Property
MouseIcon Property
MousePointer Property
Name Property
NullDateLabel Property
Parent Property
PictureDropDown Property

PromptChar Property
RawText Property
ScrollBar Property
ScrollBarTracking Property
ShowCentury Property
SpinButton Property
StartOfWeek Property
TabIndex Property
TabStop Property
Tag Property
TagVariant Property õ
Text Property
Top Property
Visible Property
VisibleMonth Property õ
Width Property

Customizing

There are many properties in the DateCombo control that let you customize the display to your liking. In addition to bevels, alignment, and color, the DateCombo control contains special properties that help shape the control to look and feel the way you want.

As with the **MonthView** and **YearView** controls, the **StartOfWeek** property can be set so that the display of weeks starts with something other than Sunday. The **Enabled** property can be used to enable and disable specific days or days of the week. The **Visible** property can be used to hide certain days of the week. See Customizing for the MonthView control for more information on how to accomplish these effects.

Marking Dates

Each day on the dropdown calendar portion of the control can be marked so that it carries special meaning. As explained above, templates called StyleSets can be created to store attributes of a type of day. Using properties pertaining to color, font and picture, you can give unique attributes to each StyleSet. Once a Styleset is added to the **StyleSets** collection, it can be applied to any date.

Masking and Formatting

By default, the DateCombo control displays the date in the edit portion according to the Windows International settings. For example, the Windows International settings, found in Control Panel, can be set so that all dates in Windows are displayed in 'Month Day, Year' format. It can also be set so that dates appear in 'Day Month, Year' format. You can override this setting by using the **Mask** property.

The **Format** property can be used to alter the date that is displayed when the DateCombo is not the control with focus. For example, setting the Format property to 'DDDD, MMMM DD, YYYY' will display the date '11/14/95' as 'Tuesday November 14, 1995'.

In addition, the DateCombo provides properties that allow you to obtain the date in different forms. Use the **RawText** property to retrieve the date without date separators ('111495'). Use the **FormattedText** property to retrieve the entire date including all formatting ('Friday, November 14, 1995').

By setting the **DateSeparator** property, you can tell the DateCombo to display '11/14/95' as '11-14-95' or in any other form.

See Also

MonthView Control
YearView Control
DayView Control

DateError Event

See Also Applies To

Occurs when the edit portion of the control contains an invalid date and an attempt was made to drop down the calendar, a spin button was clicked, or the control lost focus with the **AutoValidate** property set to **True**.

Syntax

Sub *object* **_DateError** ([*Index* **As Integer**] *ErrCode* **As Integer,** *ErrString* **As String**, *LastValidDate* **As String,** *Text* **As String,** *RtnDispErrMsg* **As Integer,** *RtnRestore* **As Integer**)

The DateError event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
Index	Uniquely identifies the control if it is in a control array.
ErrCode	An integer expression that evaluates to an error code if the date entered is not valid.
ErrString	A string expression that evaluates to the text contained in the error message.
LastValidDate	A string expression that evaluates to the valid date that was in the edit portion prior to the entering of the invalid date.
Text	A string expression that evaluates to the text in the edit portion.
RtnDispErrMsg	An integer expression that evaluates to a flag that is used to determine if the control is to display an error message after the event procedure is exited if the date value it contains is invalid. This parameter defaults to **True**.
RtnRestore	An integer expression that determines if the control should reset the value of the edit portion if it is invalid. This parameter defaults to the value of the **AutoRestore** property.

Remarks

The **DateError** event is triggered when one of the following conditions is met:

§ The control is dropped down while the date in the edit portion is invalid.

§ One of the spin buttons, UP ARROW, DOWN ARROW, ' + ' or ' - ' key is pressed, while the date in the edit portion is invalid.

§ The control loses focus while the date contained in the edit portion is invalid and the **AutoValidate** property is set to **True**

DateError Event Applies To

DateCombo Control

See Also

AutoRestore Property

AutoValidate Property

DateSeparator Property

See Also Applies To

Returns or sets the character that separates the month, day and year.

Syntax

object . **DateSeparator** [= *text*]

The **DateSeparator** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the date separator that appears in the edit portion.

DateSeparator Property Applies To

DateCombo Control

See Also

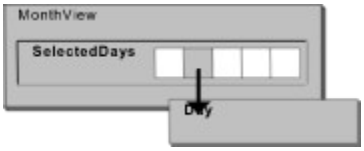
Mask Property

Format Property

Day Object

See Also Applies To

A **Day** object represents a day within the calendar.



Syntax

Day

Remarks

Day objects are accessible either by an index (from 1 to the number of days in the month), date string or date.

A **Day** object has the following properties:

Properties

Caption	DayofWeek	StyleSet
CaptionStyleSet	Enabled	Selected
Date		

Day Object Applies To

Day Property
SelectedDays Collection

See Also

- **DayofWeek** Object
- **Month** Object
- Object Concepts

Day Property

See Also Example Applies To

Returns a **Day** object.

Syntax

object . **Day**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

Use the **Day** property of an object to identify a specific **Day** object whose properties you want to use.

This property is not available at design time.

This is only accessible via the **X** object when using a VBX.

Day Property Applies To

DateCombo Control
Month Object
MonthView Control
YearView Control

Day Property Example

The following code sets the **Caption** property of the 14th day of the current month on a MonthView control:

```
SSMonth1.X.Day(14).Caption = "My Birthday"
```

The following code disables the first day of the current month on the **MonthView** control:

```
SSMonth1.X.Day(1).Enabled = False
```

See Also

- **Day** Object
- **DayofWeek** Object
- **Month** Object
- X Object

DayCaptionAlignment Property

See Also Applies To

Returns or sets the caption alignment of all day-specific captions.

Syntax

object . **DayCaptionAlignment** [= *number*]

The **DayCaptionAlignment** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the alignment of the day caption.

Settings

The settings for *number* are:

Setting	Description
0	Left Top
1	Left Middle
2	Left Bottom
3	Right Top
4	Right Middle
5	(Default) Right Bottom
6	Center Top
7	Center Middle
8	Center Bottom

DayCaptionAlignment Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

- **Caption** Property
- **Day** Object
- **DayCaptionStyleSet** Property
- **DayNumberAlignment** Property
- **DayPictureAlignment** Property

DayCaptionStyleSet Property

See Also Applies To

Returns or sets the default StyleSet applied to all day captions.

Syntax

object . **DayCaptionStyleSet** [= *text*]

The **DayCaptionStyleSet** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the name of a StyleSet to be applied to all day captions.

Remarks

This property specifies a default StyleSet applied to all day captions. It can be overridden by setting the **CaptionStyleSet** property for the **Day** object. Only the **ForeColor** and **Font** properties are used.

This property is only available at run time.

DayCaptionStyleSet Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

- **CaptionSytleSet** Property
- **Day** Object
- **StyleSet** Object
- **StyleSets** Collection

DayCount Property

See Also Example Applies To

Returns the number of days in the month with the current focus.

Syntax

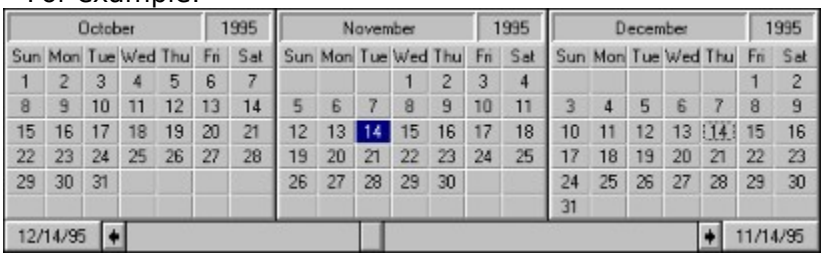
object . **DayCount**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

For the **MonthView** control, if the **NumberOfMonths** property is set larger than 1, the month which has a day with focus will be the month with focus.

For example:



In this case, December would be the month with focus since the focus rectangle is on December 14. Therefore, the **DayCount** property will return 31.

The **DayCount** property can also be used to find the last day of the current month.

This property is not available at design time and is read-only at run time.

DayCount Property Applies To

DateCombo Control
Month Object
MonthView Control
YearView Control

DayCount Property Example

The following code returns the date of the last day of the month:

```
Dim MyDayNumber As Integer
MyDayNumber = SSMonth1.X.VisibleMonth(0).DayCount
    'number of last day in the month
MyLastDay = SSMonth1.X.Day(MyDayNumber).Date
    'date of last day in the month
```

See Also

Month Object

DayFont Property

See Also Example Applies To

Returns a **Font** object.

Syntax

object . **DayFont**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

Use the **DayFont** property to identify a specific **Font** object to use for displaying days and days of the week.

This property is not available at design time in Visual Basic 3.0.

DayFont Property Applies To

YearView Control

DayFont Property Example

The following code changes the **Bold** property setting of a **Font** object identified by the **DayFont** property of a **YearView** control:

```
SSYear1.DayFont.Bold = True
```

See Also

- **DayFont3D** Property
- **DayFontBold** Property
- **DayFontItalic** Property
- **DayFontName** Property
- **DayFontSize** Property
- **DayFontStrikethru** Property
- **DayFontUnderline** Property
- **Font** Object

DayFont3D Property

See Also Applies To

Returns or sets the 3-D style of text for the days on the control.

Syntax

object . **DayFont3D** [= *number*]

The **DayFont3D** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the type of 3-D effect to use.

Settings

The settings for *number* are:

Setting	Description
0	(Default) None. Text is displayed flat (not 3-dimensional).
1	Raised w/ light shading. Text appears raised off the screen.
2	Raised w/ heavy shading. Text appears more raised.
3	Inset w/ light shading. Text appears inset on the screen.
4	Inset w/ heavy shading. Text appears more inset.

Settings 2 and 4 (heavy shading) look best with larger, more bold fonts.

DayFont3D Property Applies To

YearView Control

See Also

CaptionFont3D Property
DayFont Property
Font3D Property

DayFontBold, DayFontItalic, DayFontStrikethru, DayFontUnderline Properties

See Also Applies To

Return or set font styles in the following formats for the days and days of week: **Bold**, *Italic*, ~~Strikethru~~, and Underline.

Syntax

object . **DayFontBold** [= *boolean*]

object . **DayFontItalic** [= *boolean*]

object . **DayFontStrikethru** [= *boolean*]

object . **DayFontUnderline** [= *boolean*]

The **DayFontBold**, **DayFontItalic**, **DayFontStrikethru** and **DayFontUnderline** property syntaxes have these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying the font style as described in Settings.

Settings

The settings for *boolean* are:

Setting	Description
True	Turns on the formatting in that style.
False	Turns off the formatting in that style.

Remarks

Use these font properties to format the text of the days and days of week, either at design time using the Properties window or at run time using code.

When using the VBX version of the control, the **DayFontBold**, **DayFontItalic**, **DayFontStrikethru**, and **DayFontUnderline** properties are available at design time. These properties are supported in the OCX version of the control for compatibility.

DayFontName Property

See Also Applies To

Returns or sets the font used to display text for the days and days of week.

Syntax

object . **DayFontName** [= *font*]

The **DayFontName** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
font	A string expression specifying the font name to use.

Remarks

Use this font property to format the text of the days and days of week, either at design time using the Properties window or at run time using code.

When using the VBX version of the control, the **DayFontName** property is available at design time. This property is supported in the OCX version of the controls for compatibility.

DayFontName Property Applies To

YearView Control

See Also

- **DayFont** Property
- **DayFont3D** Property
- **DayFontBold** Property
- **DayFontItalic** Property
- **DayFontSize** Property
- **DayFontStrikethru** Property
- **DayFontUnderline** Property
- **Font** Object
- Fonts

DayFontSize Property

See Also Applies To

Returns or sets the size of the font to be used for text of the days and days of week.

Syntax

object . **DayFontSize** [= *points*]

The **DayFontSize** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
points	A numeric expression specifying the font size to use, in points.

Remarks

Use this property to format the text of the days and days of week in the font size you want.

When using the VBX version of the control, the **DayFontSize** property is available at design time. This property is supported in the OCX version of the controls for compatibility.

DayFontSize Property Applies To

YearView Control

See Also

- **DayFont** Property
- **DayFont3D** Property
- **DayFontBold** Property
- **DayFontItalic** Property
- **DayFontName** Property
- **DayFontStrikethru** Property
- **DayFontUnderline** Property
- **Font** Object
- Fonts

DayFontBold, DayFontItalic, DayFontStrikethru, DayFontUnderline Properties
Apply To

YearView Control

See Also

- **DayFont** Property
- **DayFont3D** Property
- **DayFontName** Property
- **DayFontSize** Property
- **Font** Object
- Fonts

DayFromPos Method

See Also Applies To

Returns a **Day** object that corresponds to the position indicated by coordinates.

Syntax

object . **DayFromPos**(*x* , *y* , [*scale*])

The **DayFromPos** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
x	Required. A Single-precision integer expression that evaluates to the x-coordinate.
y	Required. A Single-precision integer expression that evaluates to the y-coordinate.
scale	Optional. An integer expression that evaluates to a value in the Settings list.

Settings

The settings for *scale* are:

Setting	Description
0	(Default) Twips
1	Pixels
2	Container
3	HiMetric

Remarks

The *x* and *y* parameters are applied using the top-left of the control as the origin (0,0).

This is only accessible via the **X** object when using a VBX.

DayFromPos Method Applies To

MonthView Control
YearView Control

See Also

- **Day** Object
- **DayLeft** Method
- **DayHeight** Method
- **DayofWeekFromPos** Method
- **DayTop** Method
- **DayWidth** Method
- **MonthFromPos** Method

DayHeight Method

See Also Applies To

Returns the height of the specified day in pixels.

Syntax

object . **DayHeight**(*date*)

object . **DayHeight**(*monthindex* , *dayindex*)

object . **DayHeight**(*dayindex*)

object . **DayHeight**(*dayobject*)

The **DayHeight** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
date	Required. A string expression that evaluates to the date for which you want to return the height.
monthindex	Required. An integer expression specifying the index of the visible month for which you want to return the height of the specified day index.
dayindex	Required. An integer expression specifying the index of the day in the first or specified visible month for which you want to return the height.
dayobject	Required. An object expression that evaluates to a **Day** object for which you want to return the height of the day.

Remarks

This method returns -1 if the specified day is either not visible or off the screen.

This is only accessible via the **X** object when using a VBX.

DayHeight Method Applies To

MonthView Control
YearView Control

See Also

- **Day** Object
- **DayFromPos** Method
- **DayLeft** Method
- **DayofWeekFromPos** Method
- **DayTop** Method
- **DayWidth** Method
- **MonthFromPos** Method

DayLeft Method

See Also Applies To

Returns the left of the specified day in pixels (from 0,0 of the screen).

Syntax

object . **DayLeft**(*date*)

object . **DayLeft**(*monthindex* , *dayindex*)

object . **DayLeft**(*dayindex*)

object . **DayLeft**(*dayobject*)

The **DayLeft** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
date	Required. A string expression that evaluates to the date for which you want to return the left.
monthindex	Required. An integer expression specifying the index of the visible month for which you want to return the left of the specified day index.
dayindex	Required. An integer expression specifying the index of the day in the first or specified visible month for which you want to return the left.
dayobject	Required. An object expression that evaluates to a **Day** object for which you want to return the left of the day.

Remarks

This method returns -1 if the specified day is either not visible or off the screen.

This is only accessible via the **X** object when using a VBX.

DayLeft Method Applies To

MonthView Control
YearView Control

See Also

- **Day** Object
- **DayFromPos** Method
- **DayHeight** Method
- **DayofWeekFromPos** Method
- **DayTop** Method
- **DayWidth** Method
- **MonthFromPos** Method

DayNumberAlignment Property

See Also Applies To

Returns or sets the alignment for the day number for all days.

Syntax

object . **DayNumberAlignment** [= *number*]

The **DayNumberAlignment** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the alignment of the day number.

Settings

The settings for *number* are:

Setting	Description
0	Left Top
1	Left Middle
2	Left Bottom
3	Right Top
4	Right Middle
5	Right Bottom
6	Center Top
7	(Default) Center Middle
8	Center Bottom

DayNumberAlignment Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

DayCaptionAlignment Property

DayPictureAlignment Property

DayPictureAlignment Property

See Also Applies To

Returns or sets the picture alignment for all days.

Syntax

object . **DayPictureAlignment** [= *number*]

The **DayPictureAlignment** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the alignment of the day picture.

Settings

The settings for *number* are:

Setting	Description
0	Left Top
1	Left Middle
2	Left Bottom
3	Right Top
4	Right Middle
5	Right Bottom
6	Center Top
7	(Default) Center Middle
8	Center Bottom
9	Left of Caption
10	Right of Caption
11	Above Caption
12	Below Caption
13	Fit to Caption

DayPictureAlignment Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

- **Day** Object
- **DayCaptionAlignment** Property
- **DayNumberAlignment** Property
- **DayStyleSet** Property
- **StyleSets** Collection

DayStyleSet Property

See Also Applies To

Returns or sets a default StyleSet that is applied to all days.

Syntax

object . **DayStyleSet** [= *text*]

The **DayStyleSet** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the name of a StyleSet.

Remarks

This property specifies a default StyleSet applied to all days. The StyleSet's **ForeColor, BackColor, Picture, PictureMetaHeight, PictureMetaWidth** and **Font** properties are available.

The setting for this property can be overridden by setting the **StyleSet** property of individual **Day** objects.

This property is only available at run time.

DayStyleSet Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

StyleSet Object

StyleSets Collection

DayTop Method

See Also Applies To

Returns the top of the specified day in pixels (from 0,0 of the screen).

Syntax

object . **DayTop**(*date*)

object . **DayTop**(*monthindex* , *dayindex*)

object . **DayTop**(*dayindex*)

object . **DayTop**(*dayobject*)

The **DayTop** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
date	Required. A string expression that evaluates to the date for which you want to return the top.
monthindex	Required. An integer expression specifying the index of the visible month for which you want to return the top of the specified day index.
dayindex	Required. An integer expression specifying the index of the day in the first or specified visible month for which you want to return the top.
dayobject	Required. An object expression that evaluates to a **Day** object for which you want to return the top of the day.

Remarks

This method returns -1 if the specified day is either not visible or off the screen.

This is only accessible via the **X** object when using a VBX.

DayTop Method Applies To

MonthView Control
YearView Control

See Also

- **Day** Object
- **DayFromPos** Method
- **DayHeight** Method
- **DayLeft** Method
- **DayofWeekFromPos** Method
- **DayWidth** Method
- **MonthFromPos** Method

The DayView Control

The DayView control is useful for adding a daily time schedule into an application. The DayView control can display and organize tasks for a variety of applications, whether for an Executive Information System, Personal Information Management (PIM) or group scheduling application.

(Click on the DayView control to explore its parts.)

Filename	SSCALB.VBX, SSCALB16.OCX, SSCALB32.OCX
ObjectType	SSDay

Parts of the DayView Control
Keyboard Interface
Interactive Addition, Update and Deletion
Using the Tasks Collection

DayView Control (When In Edit Mode)

Press	To	Comments
ENTER / TAB / F4	Save Changes	Saves any changes made while in edit mode and closes the edit box.
CTRL + UP ARROW	Move BeginTime Bar Up	This makes the BeginTime of the task earlier.
CTRL + DOWN ARROW	Move BeginTime Bar Down	This makes the BeginTime of the task later.
CTRL + SHIFT + UP ARROW	Move Duration Change Bar Up	This makes the EndTime of the task earlier and the Duration shorter.
CTRL + SHIFT + DOWN ARROW	Move Duration Change Bar Down	This makes the EndTime of the task later and the Duration longer.
CTRL+ENTER	Force Carriage Return	Forces the placement of a carriage return.
ESCAPE	Cancel Changes	Cancels any changes made to a task's description and closes the edit box.

DayView Control (When Not In Edit Mode)

Press	To	Comments
LEFT ARROW	Select Previous Task in Same Time Slot	If there are multiple tasks in the same time slot, this key will select the previous task. It will cycle through the tasks as well as the Time Button itself for that time slot.
RIGHT ARROW	Select Next Task in Same Time Slot	If there are multiple tasks in the same time slot, this key will select the next task. It will cycle through the tasks as well as the Time Button itself for that time slot.
UP ARROW	Select Previous Time Slot	If the previous time slot is not visible, the control scrolls.
DOWN ARROW	Select Next Time Slot	If the next time slot is not visible, the control scrolls.
PAGE UP	Select Previous Page	Scrolls the control up by one page.
PAGE DOWN	Select Next Page	Scrolls the control down by one page.
HOME	Select First Time Slot	If the first time slot is not visible, the control scrolls.
END	Select Last Time Slot	If the last time slot is not visible, the control scrolls.
DELETE	Delete Task	Deletes the task with focus.
ENTER / SPACEBAR / F4	Edit / Add Task	Edits the task with focus. If the focus is on a time button rather than a task description, a new task is added.

Back

DayView Control Collections

Collections marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

Tasks Collection Õ

Back

DayView Control Events

Events marked with a Ô are only accessible via the **X** object when using the VBX version of the control.

Click Event
CloseEdit Event
DblClick Event
DeleteTask Event
DragDrop Event
DragOver Event
GotFocus Event
KeyDown Event
KeyPress Event
KeyUp Event
LostFocus Event
MouseDown Event
MouseMove Event
MouseUp Event
ShowEdit Event
TimeBarClick Event
TimeBtnClick Event
TopIndexChange Event

Back

DayView Control Methods

Methods marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

IndexFromTime Method Õ
Refresh Method
TaskFromPos Method Õ
TaskHeight Method Õ
TaskLeft Method Õ
TaskTop Method Õ
TaskWidth Method Õ
TimeFromIndex Method Õ
TimeFromPos Method Õ
TimeHeight Method Õ
TimeLeft Method Õ
TimeTop Method Õ
TimeWidth Method Õ
WhereIs Method Õ

Back

DayView Control Objects

Objects marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

Font Object

Task Object Õ

X Object

Back

DayView Control Properties

Properties marked with a Ô are only accessible via the **X** object when using the VBX version of the control.

HelpContextId Property
Index Property
Left Property
MaxLength Property
MouseIcon Property
MousePointer Property
Name Property
Picture Property
PictureAlignment Property
PictureMetaHeight Property
PictureMetaWidth Property
ShowTaskColor Property
TabIndex Property
TabStop Property
Tag Property
TagVariant Property õ
TaskSelected Property
TimeBegin Property
TimeEnd Property
TimeInterval Property
TimeSelectionBar Property
TimeSelectionBarFont Property
TimeSelectionBarFontBold Property
TimeSelectionBarFontItalic Property
TimeSelectionBarFontName Property
TimeSelectionBarFontSize Property
TimeSelectionBarFontStrikethru Property
TimeSelectionBarFontUnderline Property
TimeSlotCount Property
TimeSlotIndex Property
Top Property
TopIndex Property
Visible Property
Width Property

Interactive Addition, Update and Deletion

The DayView control supports in-place addition, update and deletion of tasks at run time. The text description, beginning time, ending time and duration of a task can be set directly on the control.

Note The **BackColor** and **Picture** properties of a **Task** object cannot be set directly by the user. However, they can be set through the Task object..

Note In order to be able to add, update or delete a task the **AllowAdd**, **AllowEdit** or **AllowDelete** property must be set to **True** respectively.

To Add a New Task

1. Click on a Time Button indicating the time at which you want to start the new task. This will select the Time Slot to the right of the button and open an Edit box.
2. Type a description for the new task in the Edit box. (To force a carriage-return into the edit box, press CTRL+ENTER.)
3. If you wish to set the beginning time of the task earlier or later than what is indicated by the Time Button to the left, drag the BeginTime Change bar above the Edit box to the desired time.
4. Set the duration of the task by dragging the Duration Change bar below the Edit box. Drag the bar up to decrease the duration or down to increase the duration.

   Note The position of the Duration Change bar represents the EndTime of the task. The EndTime cannot be moved above the BeginTime. The distance between the BeginTime Change bar and the Duration Change bar represents the Duration of the task in the increments previously assigned to the control.
5. When you are done setting the above attributes, either click the mouse outside the edit box or press the ENTER / TAB key.

   At this point the new task will be added to the DayView control, with its description appearing in the Time Slot area. The task will also be added to the Tasks collection.

   The Duration of the task is indicated both by the Duration Fill color in the Time Slot area and the Task BackColor in the Time Selection bar. The color of the Duration Fill is determined by the **DurationFillColor** property of the control. The color of the Task BackColor is determined by the **BackColor** property of the task. When entering tasks interactively as we have just done, the Task BackColor will be assigned automatically. This color can only be changed by setting the **BackColor** property of the **Task** object through code.

To Update a Task

1. Double-click on the description of the task you want to update, or if it has focus press ENTER or F4. This will select the task and open an Edit box.
2. Make any update to the task's description by typing in the Edit box.
3. If you wish to change the beginning time of the task to make it earlier or later than what is indicated by the Time Button to the left, drag the BeginTime Change bar above the Edit box to the desired time or press CTRL + UP/DOWN ARROW.
4. Change the duration of the task by dragging the Duration Change bar below the Edit box up or down or by pressing CTRL + SHIFT + UP/DOWN ARROW.

   Note If you drag either the BeginTime Change or Duration Change bar beyond the range of visible time slots, the DayView control will automatically scroll. This will allow you to choose from the full range of time slots. The DayView control will not scroll

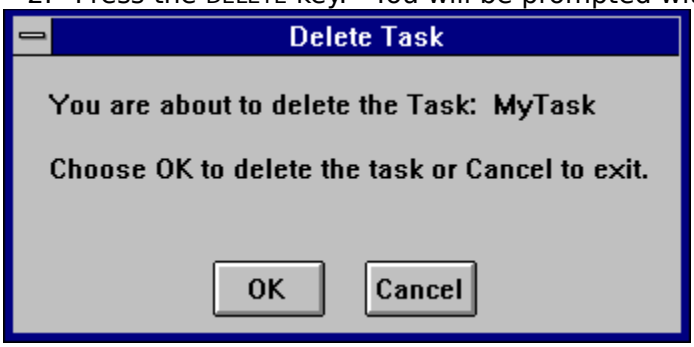
outside the time limits set by the **TimeBegin** and **TimeEnd** properties.

5. When you are done setting the above attributes, either click the mouse outside the edit box or press the ENTER / TAB / F4 key.

   At this point the task will be updated. Any changes made to the task will also be reflected in the **Tasks** collection.

To Delete a Task

1. Click on the description of the task you want to delete. This will highlight the task you have selected.
2. Press the DELETE key. You will be prompted with the following warning message:



3. Press the OK button to delete the task.

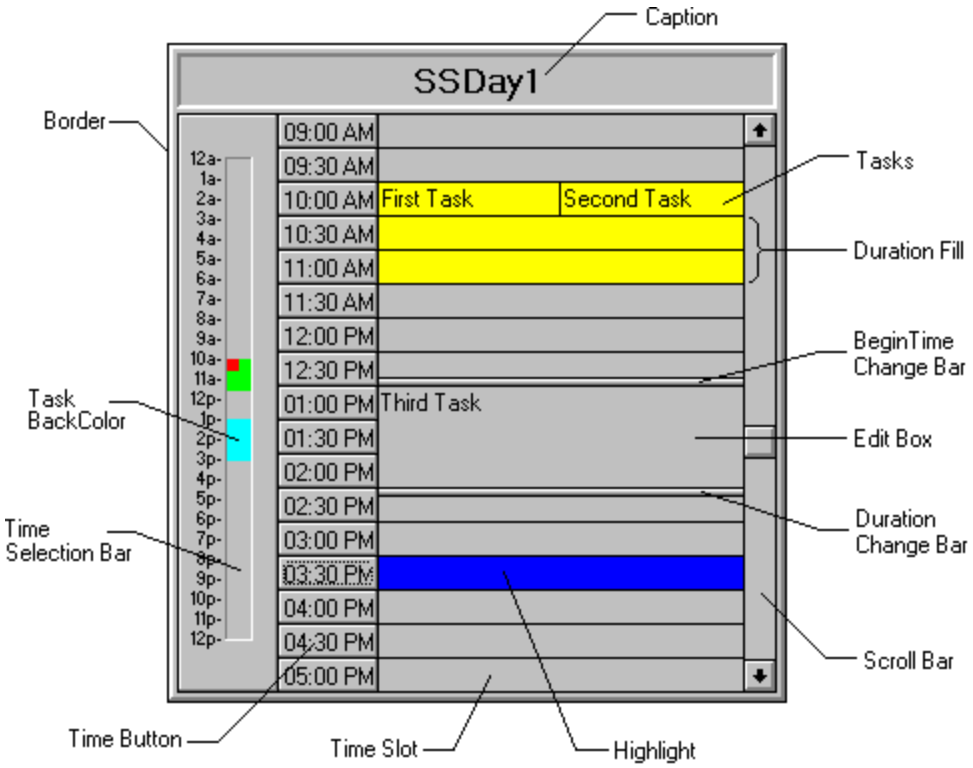
   At this point the task will be deleted. Once a task is deleted, it will no longer appear on the DayView control. The deleted task will also be removed from the **Tasks** collection. The tasks will be re-ordered with tasks occurring after the deleted task moving up one position.

   Note You should never save the tasks position in the **Tasks** collection.

   Note Before a task is deleted, the DeleteTask event is fired. You can cancel the deletion by setting the *RtnCancel* parameter of the event **True**. You can suppress the deletion-warning message by setting the *RtnDispErrMsg* parameter to **False** causing immediate deletion, or you could display your own message.

When Not Dropped Down
When Dropped Down

Parts of the DayView Control



See Also

MonthView Control
YearView Control
DateCombo Control

Using the Tasks Collection

The DayView control also lets you work with tasks through the **Tasks** collection. The Tasks collection is a collection of **Task** objects. Each Task object represents an individual task on the DayView control. Therefore, when a new **Task** object is added to the **Tasks** collection, it will be displayed in the DayView control as a task containing a description, duration, color and possibly a picture.

You can set the following properties of a Task object:

§ Text (description) § EndTime § BackColor
§ BeginTime § Duration § Picture
§ TagVariant

To Add a New Task Through Code

Ÿ Use the **Add** method to add a new task to the **Tasks** collection.

The following code adds a task named 'MyTask,' sets its beginning time to '1:00 PM,' ending time to '2:00 PM,' and background color (optional) to red:

```
SSDay1.X.Tasks.Add "1:00 PM", "2:00 PM", "MyTask", RGB(255,0,0)
```

At this point, the task will be added to the **Tasks** collection. The task will also appear in the DayView control. At this point, the new task can be accessed in the future by referencing its index value in the collection. For example, `SSDay1.X.Tasks(0)` represents the first task in the collection.

Note You should never save the tasks position in the **Tasks** collection.

Note The Tasks collection must be referenced through the **X** object when using the VBX version of the control. Otherwise, the 'X' in code is not necessary, but is supported.

To Update a Task Through Code

Ÿ Use the **Item** method to update a task in the **Tasks** collection.

Each property of a task must be handled separately. The following code checks the text description of the task being clicked. If the task's text is 'Old Task' then it sets the **Text** property to 'New Task.'

```
Sub SSDay1_Click()
Dim ThisTask as Integer
ThisTask = SSDay1.TaskSelected
If SSDay1.TaskSelected > -1 Then
   If SSDay1.X.Tasks(ThisTask).Text = "Old Task" Then
      SSDay1.X.Tasks(ThisTask).Text = "New Task"
   End If
End If
End Sub
```

At this point, the task information will be updated in the **Tasks** collection. The updated task will also appear in the DayView control.

To Delete a Task Through Code

Ÿ Use the **Remove** method to delete a task from the **Tasks** collection.

The following code deletes the first task in the **Tasks** collection:

```
SSDay1.X.Tasks.Remove(0)
```

At this point, the task has been removed from the **Tasks** collection. The task will not appear in the DayView control and cannot be accessed again.

If a task is deleted, the indexes of the tasks in the **Tasks** collection will be re-issued. For example:

Note You should never save the task's position in the **Tasks** collection.

To Remove All Tasks

Ÿ Use the **RemoveAll** method to remove all tasks from the **Tasks** collection.

The follow code will clear the contents of the DayView control:

```
SSDay1.Tasks.RemoveAll
```

At this point, the **Tasks** collection will be empty. The DayView control will be void of all tasks.

DayWidth Method

See Also Applies To

Returns the width of the specified day in pixels.

Syntax

object . **DayWidth**(*date*)

object . **DayWidth**(*monthindex* , *dayindex*)

object . **DayWidth**(*dayindex*)

object . **DayWidth**(*dayobject*)

The **DayWidth** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
date	Required. A string expression that evaluates to the date for which you want to return the width.
monthindex	Required. An integer expression specifying the index of the visible month for which you want to return the width of the specified day index.
dayindex	Required. An integer expression specifying the index of the day in the first or specified visible month for which you want to return the width.
dayobject	Required. An object expression that evaluates to a **Day** object for which you want to return the width of the day.

Remarks

This method returns -1 if the specified day is either not visible or off the screen.

This is only accessible via the **X** object when using a VBX.

DayWidth Method Applies To

MonthView Control
YearView Control

See Also

- **Day** Object
- **DayFromPos** Method
- **DayHeight** Method
- **DayLeft** Method
- **DayofWeekFromPos** Method
- **DayTop** Method
- **MonthFromPos** Method

DayofWeek Object

See Also Applies To

A **DayofWeek** object represents a day of the week within the control.

Syntax

DayofWeek

Remarks

The object is accessible by an index, from 1 to 7. An index value of 1 always represents the first day of the week, based on the **StartOfWeek** property.

A **DayofWeek** object contains the following properties:

Properties

Caption	StyleSet	Visible
Enabled	TagVariant	

This is only accessible via the **X** object when using a VBX.

DayofWeek Object Applies To

DayofWeek Property

See Also

- **Day** Object
- **Month** Object
- **StartOfWeek** Property
- X Object

DayofWeek Property

See Also Example Applies To

Returns a **DayofWeek** object.

Syntax

object . **DayofWeek**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

Use the **DayofWeek** property of an object to identify a specific **DayofWeek** object whose properties you want to use.

This property is not available at design time.

This is only accessible via the **X** object when using a VBX.

DayofWeek Property (Day Object)

See Also Example Applies To

Returns the day of week for a **Day** object

Syntax

object . **DayofWeek**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

The **DayofWeek** property returns a number corresponding to the number of the day of week. This property depends on the value of the **StartOfWeek** property.

DayofWeek Property (Day Object) Applies To

Day Object

DayofWeek Property (Day Object) Example

The following code returns the day-of-week for a given day object:

```
SSMonth1.StartOfWeek = 2     'Monday
MyBirthday = SSMonth1.X.Day("11/14/72").DayofWeek
   'returns the day number for the specified date
   'MyBirthday will be 2 for Tuesday
```

See Also

StartOfWeek Property

DayofWeek Property Applies To

DateCombo Control
MonthView Control
YearView Control

DayofWeek Property Example

The following code disables the first day of week on the **MonthView** control:

```
SSMonth1.X.DayofWeek(1).Enabled = False
```

See Also

- **Day** Object
- **DayofWeek** Object
- **Month** Object
- X Object

DayofWeekFromPos Method

See Also Applies To

Returns a **DayofWeek** object that corresponds to the position indicated by coordinates.

Syntax

object . **DayofWeekFromPos**(*x* , *y* , [*scale*])

The **DayofWeekFromPos** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
x	Required. A Single-precision integer expression that evaluates to the x-coordinate.
y	Required. A Single-precision integer expression that evaluates to the y-coordinate.
scale	Optional. An integer expression that evaluates to a value in the Settings list.

Settings

The settings for *scale* are:

Setting	Description
0	(Default) Twips
1	Pixels
2	Container
3	HiMetric

Remarks

The *x* and *y* parameters are applied using the top-left of the control as the origin (0,0).

This is only accessible via the **X** object when using a VBX.

DayofWeekFromPos Method Applies To

MonthView Control
YearView Control

See Also

DayofWeek Object

DefaultDate Property

See Also Applies To

Returns or sets the initial date selected when an unbound control is loaded.

Syntax

object . **DefaultDate** [= *text*]

The **DefaultDate** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the default date to use when the control is loaded.

Remarks

The default value '(Today)' indicates that the calendar will have today's date selected when loaded.

DefaultDate Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

AllowNullDate Property

DeleteTask Event

See Also Applies To

Occurs when a task is deleted.

Syntax7

Sub *object* **_DeleteTask** (*RtnCancel* **As Integer,** *RtnDispPromptMsg* **As Integer**)

The DeleteTask event has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
RtnCancel	An integer expression that specifies whether or not to cancel deletion.
RtnDispPromptMsg	An integer expression that specifies whether or not the deletion warning message should be displayed upon deletion.

Remarks

Set *RtnCancel* to True (-1) in order to cancel the deletion.

Set *RtnDispErrorMsg* to False (0) to suppress the display of the deletion warning message.

By default, *RtnCancel* = **False** and *RtnDispErrorMsg* = **True**.

DeleteTask Event Applies To

DayView Control

See Also

Tasks Collection

Distribution Notes

See Also

Once you have created a program using Calendar Widgets controls, you must distribute the OCX files with your application. The design time and runtime versions of the controls are not separate, therefore the same OCX files you develop with can be shipped with your application.

Filename	Description
SSCALA.VBX	16-Bit VBX containing SSMonth, SSYear, SSDateCombo controls.
SSCALA16.OCX	16-Bit OCX containing SSMonth, SSYear, SSDateCombo controls.
SSCALA32.OCX	32-Bit OCX containing SSMonth, SSYear, SSDateCombo controls.
SSCALB.VBX	16-Bit VBX containing SSDay control.
SSCALB16.OCX	16-Bit OCX containing SSDay control.
SSCALB32.OCX	32-Bit OCX containing SSDay control.

Support files needed for distribution - 16 Bit

Support files needed for distribution - 32 Bit

Distribution Notes - 16-bit Support Files

Due to the nature of the OLE architecture, the new OCX controls require that a number of supporting files be shipped with your application. They are:

DAO2516.DLL	VAEN21.OLB
MSAJT200.DLL	VB40016.DLL
MSJETERR.DLL	VB4EN16.DLL
MSJETINT.DLL	VBAJET.DLL
OC25.DLL	VBDB16.DLL
SSFM1016.DLL	

Note These files are required when using Calendar Widgets with the Visual Basic host environment. Other host environments may require different files to be included for distribution. Consult the documentation of your host environment for further details on distributing applications that use OLE controls.

Additionally, the following files must be registered via the REGSVR.EXE program:

DAO2516.DLL
OC25.DLL
SSFM1016.DLL

A note about OLE Control file distribution

There are five DLL files needed to support 16-Bit applications created with either the VBX or 16-bit OCX versions of Calendar Widgets. Distribution details on these files are as follows.

Windows 95 and Windows NT

If your application is running under Windows 95 or Windows NT, then the OLE DLLs are part of the operating system and you do not need to install or update these files, provided the version numbers match below:

Windows 95

COMPOBJ.DLL	Version 2.2
OLE2.DLL	Version 2.2
OLE2DISP.DLL	Version 2.1
OLE2NLS.DLL	Version 2.1
STORAGE.DLL	Version 2.2

Windows NT

COMPOBJ.DLL	Version 2.1

OLE2.DLL	Version 2.1
OLE2DISP.DLL	Version 2.1
OLE2NLS.DLL	Version 2.1
STORAGE.DLL	Version 2.1

Windows 3.x and Windows for Workgroups 3.x

If your application is running under Windows 3.x or Windows For Workgroups 3.x, you **must** make sure that these DLLs are installed for any applications created with the 16-bit (VBX or OCX) versions of Calendar Widgets. These DLLs are included with the Calendar Widgets installation disks and are copied into the WINDOWS\SYSTEM directory when you install Calendar Widgets under either of these environments.

Windows 3.x and Windows for Workgroups 3.x

COMPOBJ.DLL	Version 2.03
OLE2.DLL	Version 2.03
OLE2DISP.DLL	Version 2.03
OLE2NLS.DLL	Version 2.03
STORAGE.DLL	Version 2.03

Distribution Notes - 32-bit Support Files

Due to the nature of the OLE architecture, Calendar Widgets controls require that a number of supporting files be shipped with your application. These files must be installed on any machine that runs a Calendar Widgets application.

MFC42.DLL	4.2.6256
MSVCRT.DLL	4.20.6201
OLEAUT32.DLL	2.20.4054
OLEPRO32.DLL	5.0.4055

The above files are automatically installed and registered on your machine by the Calendar Widgets package, provided you do not have later versions installed.
Additionally, the following files must be registered, either via a setup program or via the 32-bit Registration Server Utility (REGSVR32.EXE) available from Microsoft:

MFC42.DLL
OLEAUT32.DLL
OLEPRO32.DLL

See Also

Included Files

DividerStyle Property

See Also Applies To

Returns or sets the style of lines that divide the days.

Syntax

object . **DividerStyle** [= *number*]

The **DividerStyle** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the divider style to use.

Settings

The settings for *number* are:

Setting	Description
0	Black Line
1	Dark Gray Line
2	Raised
3	(Default) Inset
4	ForeColor

DividerStyle Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

DividerType Property

DividerType Property

See Also Applies To

Returns or sets the type of lines used to divide the days.

Syntax

object . **DividerType** [= *number*]

The **DividerType** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the divider type to use.

Settings

The settings for *number* are:

Setting	Description
0	(Default for YearView) None
1	Vertical (separates days of the week)
2	Horizontal (separates weeks)
3	(Default for DateCombo and MonthView) Both

DividerType Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

DividerStyle Property

DropDown Event

See Also Applies To

Occurs when the dropdown calendar portion drops down.

Syntax

Sub *object* **_DropDown** ( )

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

The DropDown event is fired when the dropdown calendar portion is dropped by:

§ Pressing the DropDown button.

§ Setting the **DroppedDown** property to **True**.

§ Pressing the ALT+UP ARROW, ALT+DOWN ARROW, F4 key.

To prevent the dropdown, set the **DroppedDown** property to **False** in this event.

DropDown Event Applies To

DateCombo Control

See Also

CloseUp Event

DroppedDown Property

DropDownBevelWidth Property

See Also Applies To

Returns or sets the width (in pixels) of the bevel around the dropdown calendar.

Syntax

object . **DropDownBevelWidth** [= *number*]

The **DropDownBevelWidth** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression that evaluates to the width of the bevel around the dropdown calendar.

Remarks

The valid range for *number* is 0 to 10.

DropDownBevelWidth Property Applies To

DateCombo Control

See Also

DropDownHeight Property

DropDownWidth Property

DropDownFont Property

See Also Example Applies To

Returns a **Font** object.

Syntax

object . **DropDownFont**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

Use the **DropDownFont** property to identify a specific **Font** object to use for displaying days, days of week, month caption and year caption in the dropdown calendar portion.

This property is not available at design time in Visual Basic 3.0.

DropDownFont Property Applies To

DateCombo Control

DropDownFont Property Example

The following code changes the **Bold** property setting of a **Font** object identified by the **DropDownFont** property of a **MonthView** control:

```
SSDateCombo1.DropDownFont.Bold = True
```

See Also

- **DropDownFont3D** Property
- **DropDownFontBold** Property
- **DropDownFontItalic** Property
- **DropDownFontName** Property
- **DropDownFontSize** Property
- **DropDownFontStrikethru** Property
- **DropDownFontUnderline** Property
- **Font** Object

DropDownFont3D Property

See Also Applies To

Returns or sets the 3-D style of text in the dropdown calendar portion.

Syntax

object . **DropDownFont3D** [= *number*]

The **DropDownFont3D** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the type of 3-D effect to use.

Settings

The settings for *number* are:

Setting	Description
0	(Default) None. Text is displayed flat (not 3-dimensional).
1	Raised w/ light shading. Text appears raised off the screen.
2	Raised w/ heavy shading. Text appears more raised.
3	Inset w/ light shading. Text appears inset on the screen.
4	Inset w/ heavy shading. Text appears more inset.

Remarks

Settings 2 and 4 (heavy shading) look best with larger, more bold fonts.

DropDownFont3D Property Applies To

DateCombo Control

See Also

CaptionFont3D Property
DayFont3D Property
Font3D Property

DropDownFontBold, DropDownFontItalic, DropDownFontStrikethru, DropDownFontUnderline Properties

See Also Applies To

Return or set font styles in the following formats for the dropdown calendar: **Bold**, *Italic*, ~~Strikethru~~, and Underline.

Syntax

object . **DropDownFontBold** [= *boolean*]

object . **DropDownFontItalic** [= *boolean*]

object . **DropDownFontStrikethru** [= *boolean*]

object . **DropDownFontUnderline** [= *boolean*]

The **DropDownFontBold**, **DropDownFontItalic**, **DropDownFontStrikethru** and **DropDownFontUnderline** property syntaxes have these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying the font style as described in Settings.

Settings

The settings for *boolean* are:

Setting	Description
True	Turns on the formatting in that style.
False	Turns off the formatting in that style.

Remarks

Use these font properties to format the text of the dropdown calendar, either at design time using the Properties window or at run time using code.

When using the VBX version of the control, the **DropDownFontBold**, **DropDownFontItalic**, **DropDownFontStrikethru**, and **DropDownFontUnderline** properties are available at design time. These properties are supported in the OCX version of the control for compatibility.

DropDownFontName Property

See Also Applies To

Returns or sets the font used to display text for the dropdown calendar.

Note The property is included for compatibility with Visual Basic 3.0. For additional functionality, use the **Font** object properties.

Syntax

object . **DropDownFontName** [= *font*]

The **DropDownFontName** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
font	A string expression specifying the font name to use.

Remarks

Use this font property to format the text of the dropdown calendar, either at design time using the Properties window or at run time using code.

When using the VBX version of the control, the **DropDownFontName** property is available at design time. This property is supported in the OCX version of the control for compatibility.

DropDownFontName Property Applies To

DateCombo Control

See Also

- **DropDownFont** Property
- **DropDownFont3D** Property
- **DropDownFontBold** Property
- **DropDownFontItalic** Property
- **DropDownFontSize** Property
- **DropDownFontStrikethru** Property
- **DropDownFontUnderline** Property
- **Font** Object
- Fonts

DropDownFontSize Property

See Also Applies To

Returns or sets the size of the font to be used for text of the dropdown calendar.

Syntax

object . **DropDownFontSize** [= *points*]

The **DropDownFontSize** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
points	A numeric expression specifying the font size to use, in points.

Remarks

Use this property to format the text of the dropdown calendar in the font size you want.

When using the VBX version of the control, the **DropDownFontSize** property is available at design time. This property is supported in the OCX version of the control for compatibility.

DropDownFontSize Property Applies To

DateCombo Control

See Also

- **DropDownFont** Property
- **DropDownFont3D** Property
- **DropDownFontBold** Property
- **DropDownFontItalic** Property
- **DropDownFontName** Property
- **DropDownFontStrikethru** Property
- **DropDownFontUnderline** Property
- **Font** Object
- Fonts

DropDownFontBold, DropDownFontItalic, DropDownFontStrikethru, DropDownFontUnderline Properties Apply To

DateCombo Control

See Also

- **DropDownFont** Property
- **DropDownFont3D** Property
- **DropDownFontName** Property
- **DropDownFontSize** Property
- **Font** Object
- Fonts

DropDownForeColor Property

See Also Applies To

Returns or sets the foreground color of text in the dropdown calendar.

Syntax

object . **DropDownForeColor** [= *color*]

The **DropDownForeColor** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
color	A value or constant that determines the text color of the dropdown calendar portion.

Remarks

This property does not affect the color of the text in the edit portion of the **DateCombo**.

DropDownForeColor Property Applies To

DateCombo Control

See Also

DropDownHeight Property

DropDownWidth Property

DropDownHeight Property

See Also Applies To

Returns or sets the height of the dropdown calendar.

Syntax

object . **DropDownHeight** [= *number*]

The **DropDownHeight** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression that evaluates to the height of the dropdown calendar.

Remarks

The measurement is in the scale mode of the container.

A value of 0 will cause the height of the calendar to be calculated, based on the font size.

DropDownHeight Property Applies To

DateCombo Control

See Also

DropDownWidth Property

DropDownMouseIcon Property

See Also Applies To

Specifies a custom icon that will appear when the mouse passes over the dropdown calendar.

Syntax

object . **DropDownMouseIcon** [= *picture*]

The **DropDownMouseIcon** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
picture	An expression specifying a graphic, as described in Settings.

Settings

The settings for *picture* are:

Setting	Description
(None)	(Default) No picture.
(Icon)	Specifies a graphic. You can load the graphic from the Properties window at design time. At run time, you can also set this property using the **LoadPicture** function on an icon.

Remarks

For this property to be active, the **DropDownMousePointer** property must be set to '99 - Custom.'

DropDownMouseIcon Property Applies To

DateCombo Control

See Also

DropDownMousePointer Property

DropDownMousePointer Property

See Also Applies To

Determines the type of mouse pointer that appears when the mouse passes over the dropdown calendar.

Syntax

object . **DropDownMousePointer** [= *number*]

The **DropDownMousePointer** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the type of mouse pointer to use.

Settings

The settings for *number* are:

Setting	Description
0	(Default) Default.
1	Arrow.
2	Cross (cross-hair pointer).
3	I-Beam.
4	Icon (small square within a square).
5	Size (four-pointed arrow pointing north, south, east and west).
6	Size NE SW (double arrow pointing northeast and southwest).
7	Size N S (double arrow pointing north and south).
8	Size NW SE (double arrow pointing northwest and southeast).
9	Size W E (double arrow pointing west and east).
10	Up Arrow.
11	Hour Glass (wait).
12	No Drop.
13	Arrow and Hourglass (32-bit OCX only)
14	Arrow and Question
15	Size All (32-bit OCX only)
99	Custom (Uses DropDownMouseIcon).

DropDownMousePointer Property Applies To

DateCombo Control

See Also

DropDownMouseIcon Property

DropDownWidth Property

See Also Applies To

Returns or sets the width of the dropdown calendar.

Syntax

object . **DropDownWidth** [= *number*]

The **DropDownWidth** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression that evaluates to the width of the dropdown calendar.

Remarks

The measurement is in the scale mode of the container. A value of 0 will cause the width of the calendar to be calculated, based on the font size.

DropDownWidth Property Applies To

DateCombo Control

See Also

DropDownHeight Property

DroppedDown Property

See Also Applies To

Determines if the dropdown calendar is displayed (dropped down).

Syntax

object . **DroppedDown** [= *boolean*]

The **DroppedDown** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if the dropdown calendar portion is displayed.

Settings

The settings for *boolean* are:

Setting	Description
True	Dropdown calendar is dropped down.
False	Dropdown calendar is closed up.

DroppedDown Property Applies To

DateCombo Control

See Also

CloseUp Event

DropDown Event

Duration Property

See Also Example Applies To

Returns or sets the duration of a task in minutes.

Syntax

object . **Duration** [= *number*]

The **Duration** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	A Long integer expression that evaluates to the duration of a task in minutes.

Remarks

It is calculated whenever a task is added.

Duration Property Applies To

Task Object

Duration Property Example

The following code sets the duration of the first task in the **DayView** control to 15 minutes:

```
SSDay1.X.Tasks(0).Duration = 15
```

This code will display the total amount of time allocated to tasks:

```
Dim iTimeTotal as Integer
Dim iC as Integer

iTimeTotal = 0

For ic = 0 to (SSDay1.X.Tasks.Count - 1)
     iTimeTotal = iTimeTotal + SSDay1.X.Tasks(iC).Duration
Next iC

Text1.Text = "Total time for tasks: " & (iTimeTotal/60) & " hrs."
```

See Also

- **BeginTime** Property
- **EndTime** Property
- **Task** Object

DurationFill Property

See Also Applies To

Determines if the time slots including and following a task are filled to show the duration of the task.

Syntax

object . **DurationFill** [= *boolean*]

The **DurationFill** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if the time slots following a task are filled to show duration.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Filled.
False	Not Filled.

DurationFill Property Applies To

DayView Control

See Also

DurationFillColor Property

DurationFillPattern Property

DurationFillColor Property

See Also Applies To

Returns or sets the color used to fill in the duration of a task.

Syntax

object . **DurationFillColor** [= *color*]

The **DurationFillColor** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
color	A value or constant, of type Long, that determines the background color.

Remarks

The **DurationFill** property must be set to **True** for this property to be visible.

DurationFillColor Property Applies To

DayView Control

See Also

DurationFill Property

DurationFillPattern Property

DurationFillPattern Property

See Also Applies To

Determines the pattern used to fill in the duration of a task.

Syntax

object . **DurationFillPattern** [= *number*]

The **DurationFillPattern** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the type of fill pattern to use.

Settings

The settings for *number* are:

Setting	Description
0	(Default) Solid Fill
1	Horizontal Line
2	Vertical Line
3	Upward Diagonal
4	Downward Diagonal
5	Cross
6	Diagonal Cross

The **DurationFill** property must be set to **True** for this property to be visible.

DurationFillPattern Property Applies To

DayView Control

See Also

DurationFill Property

DurationFillColor Property

EditBackColor Property

See Also Applies To

Returns or sets the background color of the edit box in a time slot.

Syntax

object . **EditBackColor** [= *color*]

The **EditBackColor** property syntax has these parts:

Part	Description
object	An object expression.
color	A value or constant that determines the background color of the edit box in a time slot.

Remarks

The **EditBackColor** property defaults to the Windows System background color.

EditBackColor Property Applies To

DayView Control

See Also

EditForeColor Property

EditForeColor Property

See Also Applies To

Returns or sets the color of text in the edit box in a time slot.

Syntax

object . **EditForeColor** [= *color*]

The **EditForeColor** property syntax has these parts:

Part	Description
object	An object expression.
color	A value or constant that determines the color of text in the edit box in a time slot.

Remarks

The **EditForeColor** property defaults to the Windows System text color.

EditForeColor Property Applies To

DayView Control

See Also

EditBackColor Property

EditMode Property

See Also Applies To

Determines the mode in which the spin button will increment the date in the combo box.

Syntax

object . **EditMode** [= *number*]

The **EditMode** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the mode in which the spin button will increment the date in the combo box.

Settings

The settings for *number* are:

Setting	Description
0	Whole Date
1	(Default) Month, Day or Year

Remarks

Whole Date - The date will be incremented or decremented sequentially.

Month, Day or Year - The cursor position will determine which section of the date is to be incremented. That section will be incremented until its maximum number is reached, and will start over again at 1. Once a maximum is reached, the next section will also be incremented.

EditMode Property Applies To

DateCombo Control

See Also

SpinButton Property

EditVisible Property

See Also Applies To

Returns or sets if an edit box is visible.

Syntax

object . **EditVisible** [= *boolean*]

The **EditVisible** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if the Edit Box is displayed.

Settings

The settings for *boolean* are:

Setting	Description
True	Edit Box is visible.
False	Edit Box is not visible.

Remarks

The edit box will allow you to edit the description of a task in a time slot.

This property is not available at design time.

EditVisible Property Applies To

DayView Control

See Also

AllowEdit Property
ShowEdit Event

Enabled Property

See Also Example Applies To

Determines if a control or object is enabled.

Syntax

object . **Enabled** [= *boolean*]

The **Enabled** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if a control or object is enabled.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Control or object is enabled.
False	Control or object is disabled (grayed).

Remarks

When the **Enabled** property is set to **False**, the control or object will be grayed. You cannot click on or use the keyboard to access it.

Enabled Property Applies To

DateCombo Control
Day Object
DayofWeek Object
DayView Control
Month Object
MonthView Control
YearView Control

Enabled Property Example

If the **Enabled** property of a **DayofWeek** object is set to **False** for 'Tuesday' on a **MonthView** control, the following would be the result:

```
SSMonth1.X.DayofWeek(3).Enabled = False
```

Notice that every 'Tuesday' is grayed out. You cannot click on a grayed object or use the keyboard to access it.

See Also

Visible Property

EndTime Property

See Also Applies To

Specifies the ending time of the task.

Syntax

object . **EndTime** [= *text*]

The **EndTime** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the ending time of the task.

Remarks

The **EndTime** cannot be set later than the setting of the **TimeEnd** property. However, if **TimeEnd** is subsequently set earlier than **EndTime**, the Task will still be available even though it will no longer appear on the **DayView** control.

The **EndTime** property can also be set to ‘Noon’ or ‘Midnight’, representing 12:00 PM and 12:00 AM respectively.

EndTime Property Applies To

Task Object

See Also

BeginTime Property
Duration Property
Task Object

Event Summary

See Also

C

CaptionClick
CloseEdit
CloseUp

D

DateError
DeleteTask
DropDown

F

FocusChange

I

InitMonth
InitYear

M

MonthClick

S

SelChange
SelChanged
ShowEdit
Spin

T

TimeBarClick
TimeBtnClick
TopIndexChange

Y

YearClick

Exercise 1: Creating a Data Entry Form

See Also

In this exercise, we create a data entry form using a MonthView control.

1. First, run Visual Basic, start a new project and add the SSCALB.VBX file.
2. Next, place a data control on the form.
3. In the Visual Basic properties window set the following properties:

   DatabaseName = 'C:\SSCALWDG\SAMPLES\CALENDAR.MDB'

   RecordSource = 'Employee'
4. Place a text box on the form and set the following properties:

   DataSource = 'Data1'

   DataField = 'EmpID'
5. Place another text box on the form and set the following properties:

   DataSource = 'Data1'

   DataField = 'Name'
6. Place a MonthView control on the form by double clicking on the tool in the Visual Basic toolbox and set the follow properties:

   DataField = 'EmpDate'

   (The **DataSource** property will automatically be set to 'Data1').
7. Place a label on the form and set its Caption property to 'Employee Since:'.
8. Position and resize the controls on the form so they look like the following:

9. At this point you should save the project.
10. Now we can run the project by pressing the F5 key.
11. Try scrolling through the records in the database by pressing the button on the data control. You should see various records being displayed one at a time. The MonthView control will automatically display the employee's date of employment.
12. Try changing the date for any record by selecting a date from the control, and then scrolling the data control back and forth to see that the date has changed.

Note To set up the MonthView control so that updates cannot be made, set the

RtnCancel parameter to **True** in the **SelChange** event.

13. Try canceling a change by selecting a new date for one of the records and then press the ESCAPE key. The control will revert to the date in the database.

See Also

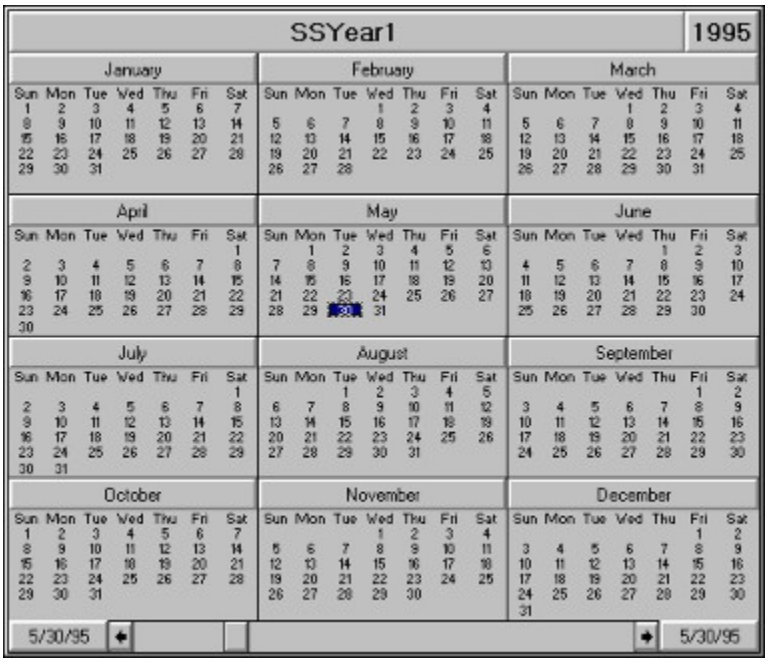
Using Calendar Widgets

Exercise 1: Exploring the YearView Control

See Also

In this exercise, we explore some of the functions of the YearView control.

1. First, run Visual Basic, start a new project and add the SSCALA.VBX file.
2. Place a YearView control on the form by double clicking on the  tool in the Visual Basic toolbox.
3. Resize the control so that it looks similar to the following:



4. Using the Visual Basic Properties window, set the control's **Caption** property to 'Holidays'.
5. Set the **DividerType** property to '3 - Both' and the **DividerStyle** property to '1-Dark gray line'.
5. Set the **MonthAlignment** property to '0 - Left Justify' and the **MonthLayout** property to '1 - Top to Bottom'.
6. Set the **StartMonth** property to '9 - September'.

   At this point the Months will read from top to bottom, and then left to right, starting with September. Notice that the Year display at the top right of the control consists of 2 years (ex. 1995-96)
7. Set the **CaptionAlignmentBeginYear** and **CaptionAlignmentEndYear** properties to '3 - Right of Caption'.
8. At this point you should save the project.
9. Now we can run the project by pressing the F5 key.

   The YearView control should look like the following:

Holidays 1995-96

September

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					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

October

Sun	Mon	Tue	Wed	Thu	Fri	Sat
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	31				

November

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			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		

December

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					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
31						

January

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	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	31			

February

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				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		

March

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					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
31						

April

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	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				

May

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			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	31	

June

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						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						

July

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	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	31			

August

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				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	31

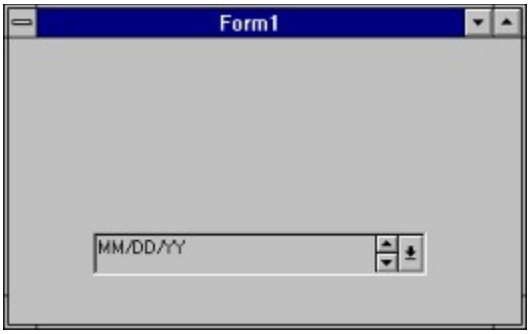
4/11/95 ← → 5/30/96

Exercise 1: Maneuvering Through the DateCombo Control

See Also

In this exercise, we explore some of the functions of the DateCombo control.

1. First, run Visual Basic, start a new project and add the SSCALA.VBX file.
2. Place a DateCombo control on the form by double clicking on the tool in the Visual Basic toolbox.
3. Resize the form and DateCombo so that they look similar to the following:



4. At this point you should save the project.
5. Now you can run the application by pressing the F5 key.

   Running the application now will give you a feel for how different options of the DateCombo affect its appearance and behavior. (Some of these options are explained in the next exercise.)

   While running, the DateCombo should look like the following:

6. Click on the month part of the date ('14') and press either of the spin buttons. You will notice that the date scrolls one day at a time.
7. Click on the year part of the date ('95') and press either of the spin buttons. You will notice that the date scrolls one year at a time.
8. Click on any part of the date in the edit portion of the DateCombo and type in a new value. This is how manual date setting is done.
9. Press the button to drop down the calendar portion of the DateCombo. A dropdown version of the MonthView control is displayed.
10. Make a date selection by clicking on a date.

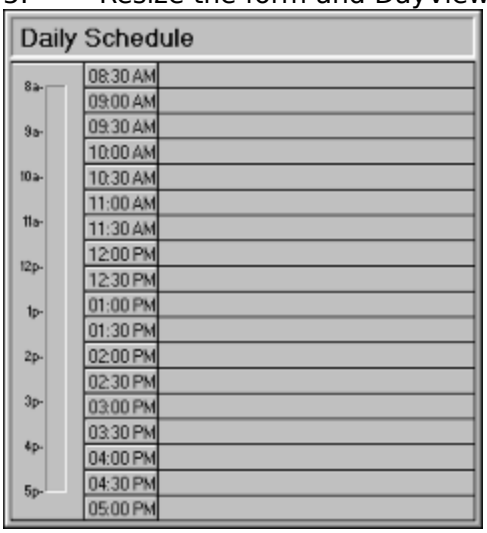
Note Depending on the Windows International settings for your system, the month, day and year may be displayed in an order other than what is shown in the illustration above. However, the International settings can be overridden by setting the **Mask** property.

Exercise 1: Maneuvering Through the DayView Control

See Also

In this exercise, we explore some of the functions of the DayView control.

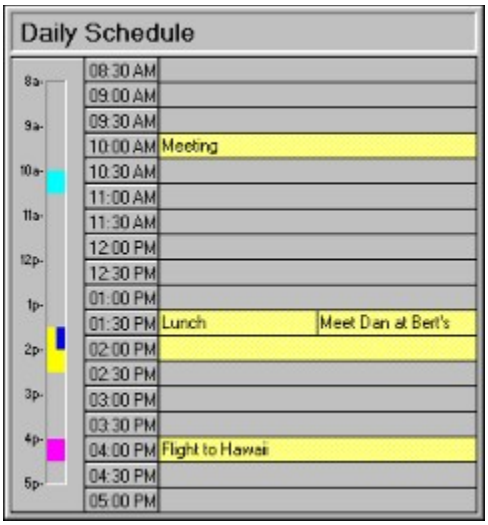
1. First, run Visual Basic, start a new project and add the SSCALB.VBX file.
2. Place a DayView control on the form by double clicking on the  tool in the Visual Basic toolbox.
3. Set the **Caption** property of the DayView control to 'Daily Schedule'.
4. Set the **TimeBegin** property to '8:30 AM' and **TimeEnd** property to '5:30 PM'.
5. Resize the form and DayView control so that they look similar to the following:



6. At this point you should save the project.
7. Now you can run the application by pressing the F5 key.
8. Add a task by clicking on the Time Button labeled '10:00 AM'. An Edit Box will be placed on the 10:00 AM Time Slot as in the following illustration:

9. Type 'Project Meeting' in the Edit Box and press ENTER.
10. Repeat Step **8** and add a new task called 'Lunch' at 12:00 PM, but do not press ENTER this time.
11. Drag the BeginTime Change Bar that appears above the 'Lunch' text down until 'Lunch' appears in the 1:30 PM time slot.
12. Drag the Duration Change Bar that appears below the 'Lunch' caption down until the Lunch task spans 1 hour (2 time slots) and press ENTER.
13. Add a second task at 1:30 PM with text reading 'Meet Dan at Bert's'.
14. Finally, add a task at 4:00 PM with text reading 'Flight to Hawaii'.

The DayView control should look like the following:



15. Move the Meeting from 10:00 AM to 11:00 AM by double-clicking on the task's description ('Meeting') and dragging the BeginTime Change Bar downward. Press ENTER when you are done.
16. Delete the 'Flight to Hawaii' task by clicking it and then pressing the DELETE key.

Exercise 2: Applying a StyleSet

In this exercise, we expand on the previous exercise by creating and applying StyleSets to the YearView control.

1. If you have not completed Exercise 1 above, please do so. The form from the previous exercise will be used for this exercise.
2. Place the following code in the Load event of Form1:

```
SSYear1.X.Stylesets("Holiday").BackColor = RGB(255,255,0)
```

3. Place a command button on the form and set its **Caption** property to 'Show Holidays'
4. Place the following code in the Click event of the command button:

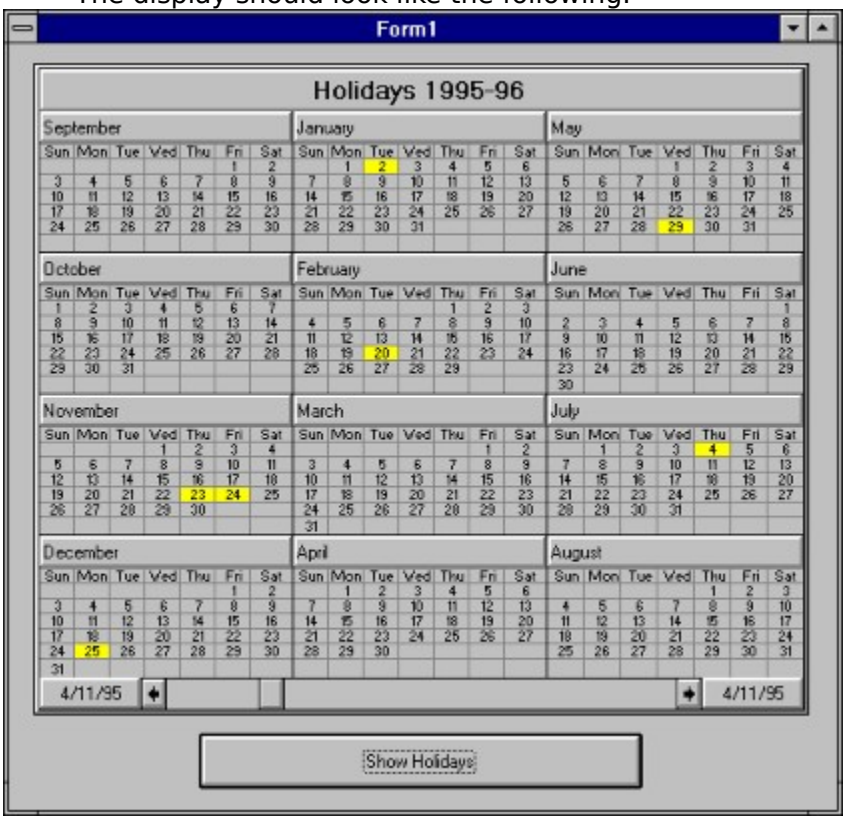
```
SSYear1.X.Day("11/23/95").StyleSet = "Holiday"
SSYear1.X.Day("11/24/95").StyleSet = "Holiday"
SSYear1.X.Day("12/25/95").StyleSet = "Holiday"
SSYear1.X.Day("1/2/96").StyleSet = "Holiday"
SSYear1.X.Day("2/20/96").StyleSet = "Holiday"
SSYear1.X.Day("5/29/96").StyleSet = "Holiday"
SSYear1.X.Day("7/4/96").StyleSet = "Holiday"
SSYear1.X.Day("9/4/96").StyleSet = "Holiday"
```

5 At this point you should save the project.

6 Now we can run the project by pressing the F5 key.

7. Scroll the YearView control until the caption reads 'Holidays 1995-96'.
8. Press the 'Show Holidays' button.

The display should look like the following:

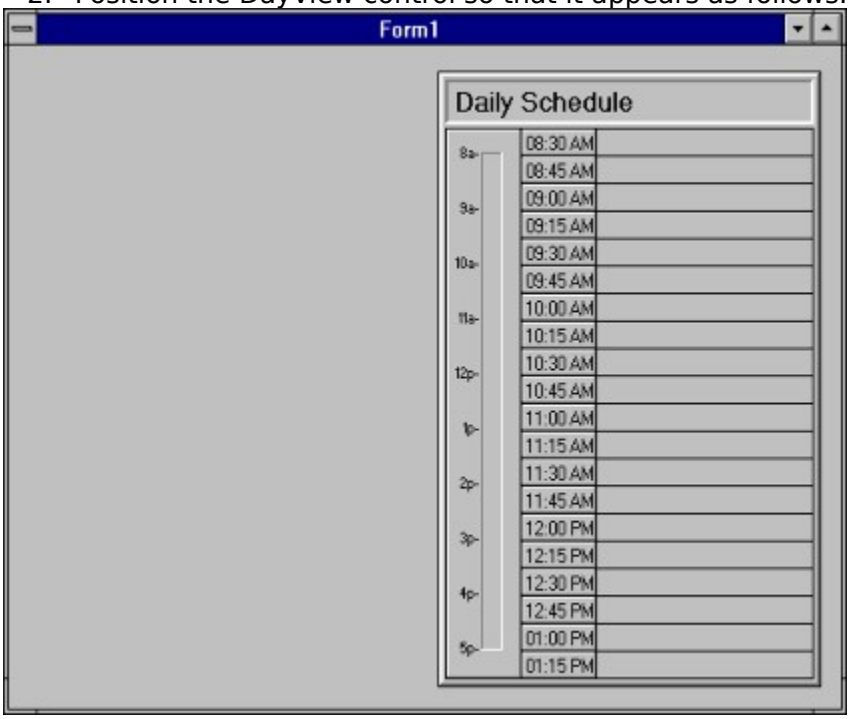


Notice that the days that the 'Holidays' StyleSet was applied to appear yellow.

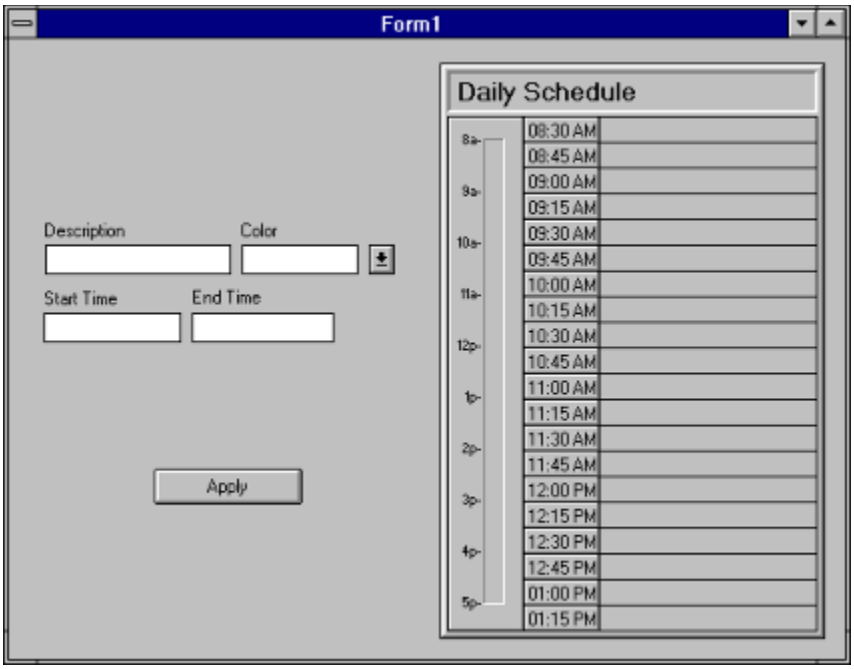
Exercise 2: Creating a Task Entry Form

In this exercise, we will expand on the previous exercise and set tasks through the Tasks collection via code.

1. Using the application created in Exercise 1 above, set the DayView control's **TimeInterval** property to '2 - 15 Minutes'.
2. Position the DayView control so that it appears as follows:



3. Place a text box to the left of the DayView control, with a label above it reading 'Description'.
4. Place a combo box next to the text box, with a label above it reading 'Color'.
5. Place 2 more text boxes underneath the existing text box and combo box, with labels above them reading 'Start Time' and 'End Time'.
6. Place a command button below the series of text box, and set its Caption property to 'Apply'.
7. Position and resize the controls on the form so that it looks like the following:



8. Place the following code in the Click event of the Apply button:

```
Sub Command1_Click()
   Dim MyColor As Long
   Dim AutoFlag As Integer
   Description = Text1.Text
   StartTime = Text2.Text
   EndTime = Text3.Text
   AutoFlag = False
   Select Case Combo1.Text
      Case "Auto":
         AutoFlag = True
      Case "Red":
         MyColor = RGB(255, 0, 0)
      Case "Green":
         MyColor = RGB(0, 255, 0)
      Case "Blue":
         MyColor = RGB(0, 0, 255)
   End Select

   If AutoFlag = False Then
      SSDay1.X.Tasks.Add StartTime, EndTime, Description, MyColor
   Else
      SSDay1.X.Tasks.Add StartTime, EndTime, Description
   End If
End Sub
```

9. Place the following code in the Load event of the form:

```
Sub Form_Load()
   Combo1.AddItem "Auto"
   Combo1.AddItem "Red"
   Combo1.AddItem "Green"
   Combo1.AddItem "Blue"
   Combo1.ListIndex = 0
End Sub
```

10. At this point you should save the project.
11. Now you can run the application by pressing the F5 key.
12. Type 'Meet with Ned' in the Description box, 'Noon' in the Start Time box, '1:30 PM' in the End Time box, and select 'Red' in the Color box.
13. Click on the Apply button. The information you provided is now added to the DayView control as a new task.
14. Add some more tasks by repeating Steps **12** and **13**.

That's all there is to it. Although in this example we added tasks without directly interacting with the DayView control, you can still add, delete, update and move tasks like we did in Exercise 1.

See Also

X Object

Exercise 2: Options of the DateCombo Control

See Also

In this exercise, we expand on the previous exercise by setting various options of the DateCombo control.

1. Using the application created in the previous exercise, set the **SpinButton** property to '2 - SpinButton to left'.
2. Set the **DropDownFont3D** property to '3 - Inset with light shading'.
3. Set the **Format** property to 'DDDD, MMMM DD, YYYY'.
4. Place 2 text boxes above the DateCombo, with a label above each reading 'RawText' and 'FormattedText', and clear the **Text** property of the text boxes.
5. Place the following code in the LostFocus event of the DateCombo:

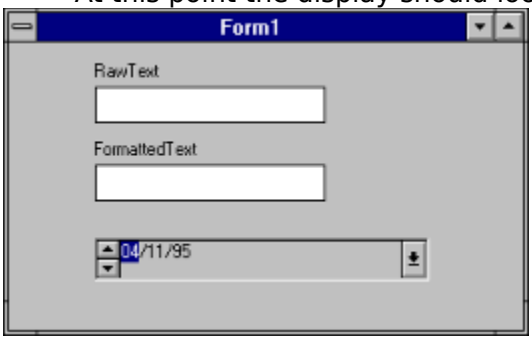
```
Text1.Text = SSDateCombo1.RawText
Text2.Text = SSDateCombo1.FormattedText
```

6. Place the following code in the Load event of the form:

```
SSDateCombo1.X.DayofWeek(1).Enabled = False
SSDateCombo1.X.DayofWeek(7).Enabled = False
```

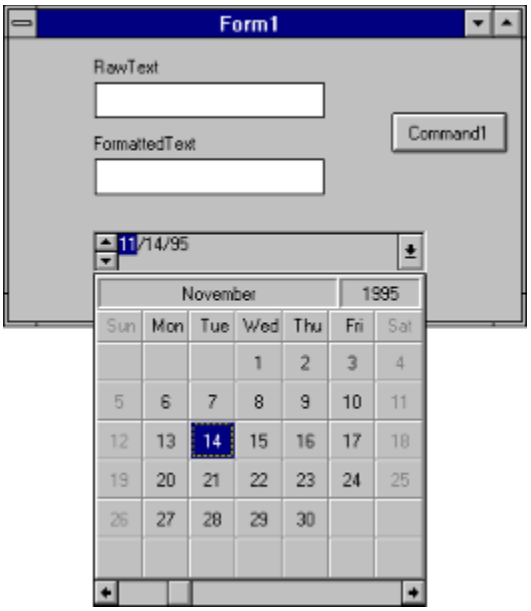
7. At this point you should save the project.
8. Now you can run the application by pressing the F5 key.

   At this point the display should look like the following:



9. Set the date by clicking on either the month, day or year and then click a spin button. Notice that only the selected part of the date changes when a spin button is clicked.
10. Drop down the calendar portion by pressing the button.

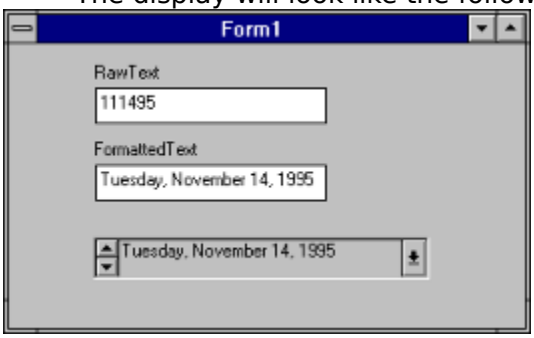
    Notice that Sunday and Saturday are disabled. Any day or day of the week can be disabled by setting its **Enabled** property to **False**.



(A day-of-week can be hidden by setting its **Visible** property to **False**.)

11. Select a date from the calendar portion by clicking on it. The DateCombo will not allow you to click on a Sunday or Saturday.
12. Click on either of the text boxes above the DateCombo.

    The display will look like the following:



The RawText will contain the date of the DateCombo without any formatting or date separators. The FormattedText, on the other hand, will contain the exact contents of the DateCombo's edit portion when the DateCombo does not have focus.

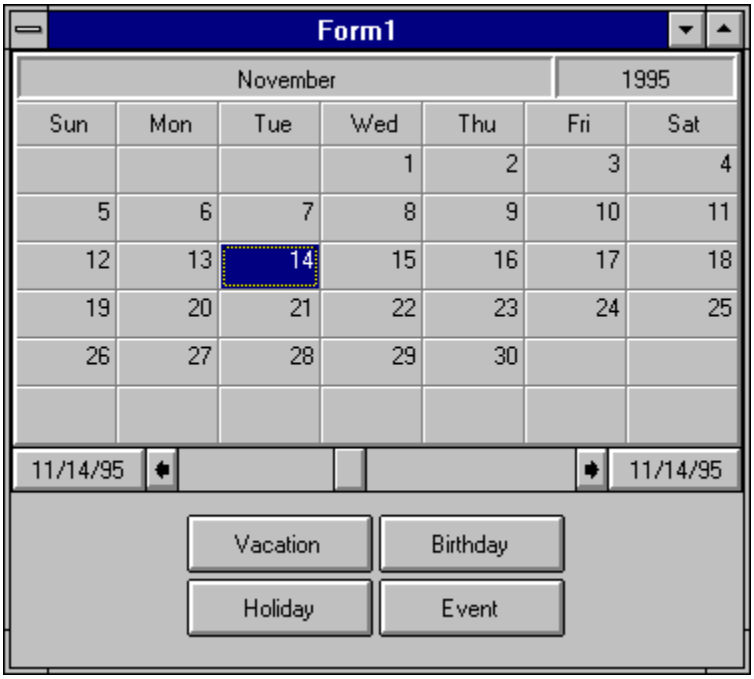
Note Depending on the Windows International settings for your system, the month, day and year may be displayed in an order other than what is shown in the illustration above. However, the International settings can be overridden by setting the **Mask** property.

Exercise 2: Using a StyleSet

See Also

In this exercise, we create and apply 4 different StyleSets to various days in a MonthView control.

1. First, run Visual Basic, start a new project and add the SSCALA.VBX file.
2. Next, place a MonthView control on the form by double clicking on the  tool in the Visual Basic toolbox.
3. In the Visual Basic Properties Box, set the **Align** property of the MonthView control to '1 - Align Top'.
4. Set the **DayNumberAlignment** property to '3 - Right Top'.
5. Set the **DayCaptionAlignment** property to '8 - Center Bottom'.
6. Set the **SelectionType** property to '1 - MultiSelect'.
7. Place 4 command buttons on the form and make them a control array. This is done by setting all of their **Name** properties to the same name. Press the Yes button when prompted if you wish to create a control array.
8. Set the Caption properties of the buttons to 'Vacation,' 'Holiday,' 'Event,' 'Birthday' respectively.
9. Position and resize the controls on the form so they look like the following:



10. Place the following code in the Click event of the Command Button array:

```
Sub Command1_Click(Index As Integer)
   Dim MyCaption As String
   Dim i As Integer
   For i = 0 To (SSMonth1.X.SelectedDays.Count - 1)
      SSMonth1.X.SelectedDays(i).StyleSet = Command1(Index).Caption
   Next
   SSMonth1.X.SelectedDays.RemoveAll
End Sub
```

11. Place the following code in the Load event of Form1:

```
Sub Form_Load()
   SSMonth1.X.StyleSets("Vacation").BackColor = RGB(255, 255, 0)
   SSMonth1.X.StyleSets("Holiday").ForeColor = RGB(255, 0, 0)
   SSMonth1.X.StyleSets("Birthday").Font.Size = 14
   SSMonth1.X.StyleSets("Event").Picture = "C:\SSCALWDG\SAMPLES\
CHAPTER5\CONCERT.BMP"
End Sub
```

12. Now we can run the project by pressing the F5 key.
13. Try selecting multiple dates on the MonthView control by clicking on them. To de-select dates, click on the selected dates again.
14. Once a group of dates are selected, press the Vacation button.

At this point, the dates you have selected should have a background color of yellow. Pressing the Vacation button applies the 'Vacation' StyleSet that was set up in the Load event of the form to all of the days that were selected. The days are de-selected by clearing the **SelectedDays** collection using the **RemoveAll** method.

15. Try selecting other dates and pressing one of the other buttons.
16. Scroll the MonthView control to the next month and back using the scroll bar. You will see that the changes that were made in previous steps are still in effect.

FocusChange Event

See Also Applies To

Occurs when the focus changes from one day to another.

Syntax

Sub *object* **_FocusChange** ([*Index* **As Integer**] *FocusDate* **As String,** *OldFocusDate* **As String**, *MonthNum* **As Integer**, *YearNum* **As Integer**, *DayNum* **As Integer**)

The FocusChange event has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
Index	Uniquely identifies the control if it is in a control array.
FocusDate	A string expression that evaluates to the date that receives focus.
OldFocusDate	A string expression that evaluates to the date that had focus prior to the change in focus.
MonthNum	An integer expression that evaluates to the number of the month (1-12) with the focus day.
YearNum	An integer expression that evaluates to the number of the year (For example: 1995) with the focus day.
DayNum	An integer expression that evaluates to the number (1-31) of the focus day.

Remarks

The **FocusChange** event is fired when a change is made to the focus day.

FocusChange Event Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

FocusDate Property

FocusDate Property

See Also Applies To

Returns or sets the date of the day that has focus.

Syntax

object . **FocusDate** [= *text*]

The **FocusDate** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the date of the day with focus.

Remarks

The date with focus will contain a focus rectangle. **SelectedDays** will contain the **FocusDate** if the **AutoSelect** property is set to **True**.

This property is only available at run time.

FocusDate Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

FocusChange Event

Font Object

See Also Applies To

The **Font** object contains information needed to format text.

The **Font** object supports the following properties:

Properties

Bold	Name	StrikeThrough
Italic	Size	Underline

Syntax

Font

Font Object Applies To

CaptionFont Property
DayFont Property
DropDownFont Property
Font Property
TimeSelectionBarFont Property

See Also

Font Property

Fonts

Font Property

See Also Example Applies To

Returns a **Font** object.

Syntax

object . **Font**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

Use the **Font** property to identify a specific **Font** object to use for the following:

DateCombo

Edit Portion

DayView

Task Description	TimeButtons

MonthView

Day Numbers	Month Caption	Today's Date Button
Day Caption	Selected Date Button	Year Number
DayofWeek Caption		

YearView

Month Caption	Selected Date Button	Today's Date Button

This property is not available at design time when using the VBX version of the control.

Font Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

Font Property Example

The following code changes the **Bold** property setting for the **Font** object of a **MonthView** control:

```
SSMonth1.Font.Bold = True
```

See Also

Font Object

Font3D Property

See Also Applies To

Returns or sets the 3-D style of text on the control

Syntax

object . **Font3D** [= *number*]

The **Font3D** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the type of 3-D effect to use.

Settings

The settings for *number* are:

Setting	Description
0	(Default) None. Text is displayed flat (not 3-dimensional).
1	Raised w/ light shading. Text appears raised off the screen.
2	Raised w/ heavy shading. Text appears more raised.
3	Inset w/ light shading. Text appears inset on the screen.
4	Inset w/ heavy shading. Text appears more inset.

Settings 2 and 4 (heavy shading) look best with larger, more bold fonts.

Remarks

For the **MonthView** control, this affects all text but the selected date.

For the **YearView** control, this affects the month name, the Today's Date button, and the Select Date button.

Font3D Property Applies To

MonthView Control
YearView Control

See Also

CaptionFont3D Property
DayFont3D Property
DropDownFont3D Property

Fonts

See Also

Fonts are supported both through font properties such as **FontBold** and **FontItalic** as well as the **Font** object. Both the VBX and OCX versions of the controls support the **Font** object.

At design time, when you use the VBX version of the control, fonts are set using the individual font properties (For example: **FontBold**, **FontItalic**, **CaptionFontBold**, **CaptionFontItalic**, etc.).

At design time, when you use the OCX version of the control, fonts are set through one of the font properties (For example: **Font**, **TimeSelectionBarFont**). Depending on the development environment you are using, a dialog box containing font information may be available so that you can set properties of the **Font** object. If not, you can set the font properties through the Property Pages. The following properties are supported by the **Font** object:

Properties

Bold	Size	Underline
Italic	StrikeThrough	Weight
Name		

Whether you use the VBX or OCX version of the control, fonts can be set either through the individual font properties or by setting properties of the **Font** object at run time. For example, the following sets of code are equivalent:

```
SSMonth1.FontName = "Arial"
SSMonth1.FontSize = 18
SSMonth1.FontBold = True
SSMonth1.FontItalic = False
SSMonth1.FontStrikethru = True
```

```
SSMonth1.Font.Name = "Arial"
SSMonth1.Font.Size = 18
SSMonth1.Font.Bold = True
SSMonth1.Font.Italic = False
SSMonth1.Font.StrikeThrough = True
```

Note The **Font**, **CaptionFont**, **DayFont** and **TimeSelectionBarFont** properties are not available at design time in Visual Basic 3.0.

See Also

- **CaptionFont** Property
- **DayFont** Property
- **Font** Property
- Property Pages
- **TimeSelectionBarFont** Property

ForeColorSelected Property

See Also Applies To

Returns or sets the text color of the selection.

Syntax

object . **ForeColorSelected** [= *color*]

The **ForeColorSelected** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
color	A value or constant that determines the text color of the selection.

Remarks

For the **DayView** control, this property sets the text color of text for the task that is currently selected.

For the **DateCombo**, **MonthView** and **YearView** controls, this property sets the text color for the selected day number.

ForeColorSelected Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

See Also

Date Property

Format Property

See Also Applies To

Determines the format that the date will be displayed in.

Syntax

object . **Format** [= *text*]

The **Format** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
Text	A string expression that evaluates to the format that the date will be displayed in.

Remarks

You can either specify a format by supplying a string or select one from the list of preset formats.

If no format (default) is specified, the date will be unchanged when the control loses focus.

Format Property Applies To

DateCombo Control

See Also

Mask Property

FormattedText Property

See Also Example Applies To

Returns the formatted text for the edit portion of the DateCombo.

Syntax

object . **FormattedText**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

The **FormattedText** property can be used to obtain the complete contents of the edit portion with the format applied.

This property is not available at design time and is read-only at run time.

FormattedText Property Applies To

DateCombo Control

FormattedText Property Example

If the **Format** property is set to 'DDDD, MMMM DD, YYYY', using the following **DateCombo** and code:


```
MyText = SSDateCombo1.FormattedText
```

The MyText variable would equal 'Tuesday, November 14, 1995.'

See Also

Format Property

RawText Property

Guided Tours

MonthView Control
Sample programs using the MonthView control. (Chapter 5)

Exercise 1: Creating a Data Entry Form
Exercise 2: Using a StyleSet

YearView Control
Sample programs using the YearView control. (Chapter 6)

Exercise 1: Exploring the YearView Control
Exercise 2: Applying a StyleSet

DateCombo Control
Sample programs using the DateCombo control. (Chapter 7)

Exercise 1: Maneuvering Through the DateCombo Control
Exercise 2: Options of the DateCombo Control

DayView Control
Sample programs using the DayView control. (Chapter 8)

Exercise 1: Maneuvering Through the DayView Control
Exercise 2: Creating a Task Entry Form

The sample programs described above have already been written. You will find all of the necessary files in the \SAMPLES subdirectory under the directory in which Calendar Widgets was installed.

How is an OCX control different from a VBX control?

The VBX control specification was designed specifically for use with Visual Basic. Although some other languages offer limited VBX support, the majority of VBX controls function only in Visual Basic. VBX controls are also limited in other ways. Their 16-bit architecture restricts their ability to take full advantage of a 32-bit operating system, such as Windows NT.

The difference between OCX and VBX controls may not even be apparent to you if you program exclusively in Visual Basic. You access the properties of an OCX control at design time and through code just as you do the properties of a VBX. The process of including both types of controls in your project and distributing them is very similar. The similarities end when you move outside of the Visual Basic programming environment.

OCX controls are designed to be supported by a much wider range of platforms, including other languages, database management systems, and productivity applications. OCX controls also have the ability to make full use of the newest 32-bit operating systems, taking advantage of improved memory access, better multi-tasking and increased performance.

Included Files

The following table gives a brief description of the files that are installed on your hard disk during the Setup process.

Files installed in the **\WINDOWS\SYSTEM** directory:

Filename(s)	Description
CMDIALOG.VBX	Property Pages support file
COMPOBJ.DLL	OLE support file (Windows 3.X only)
MFCOLEUI.DLL	OLE support file
MSAJT112.DLL	Data Access support file
MSAJT200.DLL	Data Access support file
MSOUTLIN.VBX	Property Pages support file
OC25.DLL	OLE support file
OLE2.DLL	OLE support file (Windows 3.X only)
OLE2DISP.DLL	OLE support file (Windows 3.X only)
OLE2NLS.DLL	OLE support file (Windows 3.X only)
OLE2PROX.DLL	OLE support file (Windows 3.X only)
SCP.DLL	Support file
SSCALA.VBX	VBX control for MonthView, YearView and DateCombo. (Optional)
SSCALB.VBX	VBX control for DayView. (Optional)
SSFM1016.DLL	16-bit support file
SSFORMFX.VBX	Property Pages support file
SSIDXTAB.VBX	Property Pages support file
SSPP16.DLL	Property Pages support file
STORAGE.DLL	OLE support file (Windows 3.X only)
THREED.VBX	Property Pages support file
TYPELIB.DLL	OLE support file (Windows 3.X only)
VBDB300.DLL	VBX support file (for Visual Basic 3.0 demo applications)
VBOA300.DLL	VBX support file (for Visual Basic 3.0 demo applications)
VBRUN300.DLL	VBX support file (for Visual Basic 3.0 demo applications)

Files installed in the **\WINDOWS\SYSTEM** (Windows 95) or the **\WINDOWS\SYSTEM32** (Windows NT) directory:

Filename(s)	Description
MFC40.DLL	MFC support file
MFC42.DLL	MFC support file
MFCO40.DLL	MFC support file
MSVCRT.DLL	VC++ run-time support file
MSVCRT40.DLL	VC++ run-time support file
OLEAUT32.DLL	32-bit OLE support file
OLEPRO32.DLL	32-bit OLE support file

SSFM1032.DLL	32-bit support file
STDOLE2.TLB	32-bit OLE type library

Files installed in the **\SSCALWDG** directory (or whatever directory you specified during installation):

Filename(s)	**Description**
INSTALL.LOG	Record of Calendar Widgets installation. Needed by the uninstall program.
README.WRI	Contains updated information not found in this manual and the Calendar Widgets on-line Help file.
SSCALWDG.BAS	The declarations for the Calendar Widgets API functions and the constant declarations for various Calendar Widgets settings.
SSCALA16.OCX	16-bit OCX control for MonthView, YearView and DateCombo.
SSCALB16.OCX	16-bit OCX control for DayView.
SSCALA32.OCX	32-bit OCX control for MonthView, YearView and DateCombo.
SSCALB32.OCX	32-bit OCX control for DayView.
SSCALA.VBX	VBX control for MonthView, YearView and DateCombo.
SSCALB.VBX	VBX control for DayView.
SSDCPP.EXE	Property Pages for the DateCombo control
SSDVPP.EXE	Property Pages for the DayView control
SSMNPP.EXE	Property Pages for the MonthView control
SSYRPP.EXE	Property Pages for the YearView control
UNINSTAL.EXE	Used to uninstall Calendar Widgets from your computer.

Help files are installed under the **\SSCALWDG\HELP** directory:

Filename(s)	**Description**
SSCALWDG.HLP	The Calendar Widgets on-line Help file.

Sample projects directories installed under the **\SSCALWDG\SAMPLES** directory:

Filename(s)	**Description**
CHAPTER?*.*	Contains the examples for the MonthView, YearView, DateCombo and DayView controls.
DATEPAD*.*	Contains a sample application created using Calendar Widgets.
SSCWDEMO*.*	Contains a sample application that interactively demonstrates the effects of various Calendar Widgets properties.

Including Calendar Widgets in Your Project

The method you use to add a Calendar Widgets control to your project varies depending on which programming environment you are using. Calendar Widgets controls come in three varieties:

- 32-bit OCX controls, which are compatible with development environments that support OLE custom controls and run in a 32-bit environment, such as Windows NT.
- 16-Bit OCX controls, which are compatible with 16-bit development environments that support 16-bit OLE custom controls.
- VBX controls, which are compatible with development environments that support level 3 VBX controls such as Microsoft Visual Basic version 3.0 and other development environments that support level 3 VBX controls.

IndexFromTime Method

See Also Applies To

Returns the time slot index of the specified time.

Syntax

object . **IndexFromTime**(*text*)

The **IndexFromTime** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the time to be passed into the method.

Remarks

If the time specified is earlier than **TimeBegin** or later than **TimeEnd**, the property will have a value of -1.

IndexFromTime Method Applies To

DayView Control

See Also

TimeBegin Property
TimeEnd Property
TimeFromIndex Method

InitMonth Event

Example Applies To

Occurs when a new month is displayed.

Syntax

Sub *object* **_InitMonth** ([*Index* **As Integer**] *MonthNum* **As Integer**, *YearNum* **As Integer**, *RtnCancel* **As Integer**)

The InitMonth event has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
Index	Uniquely identifies the control if it is in a control array.
MonthNum	An integer expression that evaluates to the number (1-12) of the newly displayed month.
YearNum	An integer expression that evaluates to the number (For example: 1995) of the newly displayed month.
RtnCancel	Determines if the control should abort the display of the new month. Set this parameter to **True** to cancel the process. The parameter default is **False**.

Remarks

This event fires every time a different month appears. This includes scrolling to a different month. For example, if you scroll forward on the **MonthView** control to the next month, the InitMonth event fires. If you then scroll back to the original month, the InitMonth event fires again.

InitMonth Event Applies To

DateCombo Control
MonthView Control

InitMonth Event Example

The following code disables New Year's Day every year:

```
Sub SSMonth1_InitMonth(MonthNum As Integer, YearNum As Integer, RtnCancel As
Integer)
   If MonthNum = 1 Then
       SSMonth1.X.Month(MonthNum, YearNum).Day(1).Enabled = False
   End If
End Sub
```

InitYear Event

Applies To

Occurs when a new year is displayed.

Syntax

Sub *object* **_InitYear** ([*Index* **As Integer**] *YearNum* **As Integer**, *RtnCancel* **As Integer**)

The InitYear event has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
Index	Uniquely identifies the control if it is in a control array.
YearNum	An integer expression that evaluates to the year number.
RtnCancel	Determines if the control should abort the display of the new year. Set this parameter to **True** to cancel the process. The parameter default is **False**.

Remarks

This event fires every time a different year appears. This includes scrolling to a different year. For example, if you scroll forward on the **YearView** control to the next year, the InitYear event fires. If you then scroll back to the original year, the InitYear event fires again.

InitYear Event Applies To

YearView Control

IsDateValid Method

See Also Applies To

Determines if the current date in the edit portion is valid.

Syntax

[*boolean* =]*object* . **IsDateValid**

The **IsDateValid** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if the current date in the edit portion is valid.

Settings

The settings for *boolean* are:

Setting	Description
True	Date is valid, or date is **Null** while **AllowNull** property is **True**.
False	Date is invalid, or date is **Null** while **AllowNull** property is **False**.

Remarks

This method returns **True** if the date currently in the edit portion is valid. It also returns **True** if the date is **Null** while the **AllowNull** property is **True**.

IsDateValid Method Applies To

DateCombo Control

See Also

AllowNullDate Property

Italic Property

See Also Applies To

Returns or sets the font style of the **Font** object to either italic or nonitalic.

Syntax

object . **Italic** [= *boolean*]

The **Italic** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying the font style as described in Settings.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Turns on italic formatting.
False	Turns off italic formatting.

Remarks

In Visual Basic 3.0 you set the **Italic** property by selecting a control's **FontItalic**, **CaptionFontItalic**, **DayFontItalic**, **DropDownFontItalic** or **TimeSelectionBarFontItalic** property in the Visual Basic Properties window.

At run time, however, you can set **Italic** directly by specifying its setting for the **Font** object.

Italic Property Applies To

Font Object

See Also

Font Object

Font Property

When Not Dropped Down
When Dropped Down

LargeChange Property

Applies To

Returns or sets the number of years the display will scroll when you click on the Large Change area of the scroll bar.

Syntax

object . **LargeChange** [= *number*]

The **LargeChange** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression that evaluates to the number of years the display will scroll.

Remarks

The settings for *number* are 1 to 100. The default setting is 10.

LargeChange Property Applies To

YearView Control

Mask Property

See Also Applies To

Determines the input format for the date.

Syntax

object . **Mask** [= *number*]

The **Mask** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the input format of the date.

Settings

The settings for *number* are:

Setting	Description
0	(Default) System Default
1	MM/DD/YY
2	DD/MM/YY
3	YY/MM/DD

This property allows you to enter the date in various formats. This helps in designing applications that use international and other special date formats.

Mask Property Applies To

DateCombo Control

See Also

Format Property

MaxDate Property

See Also Applies To

Returns or sets the maximum date the control will be able to display.

Syntax

object . **MaxDate** [= *text*]

The **MaxDate** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the maximum date the control will be able to display.

Remarks

MaxDate is set to 12/31/9999 by default.

An error will be generated if the date is set to one that is later than the **MaxDate**.

MaxDate Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

MinDate Property

MaxLength Property

See Also Applies To

Returns or sets the number of characters that you can enter in an Edit box.

Syntax

object . **MaxLength** [= *number*]

The **MaxLength** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the number of characters that you can enter in an Edit box. (Default = 32767)

Remarks

Use this property to limit the amount of text you can type in the Edit box of a time slot.

MaxLength Property Applies To

DayView Control

See Also

AllowEdit Property

Method Summary

See Also

A | B | C | D | E | F | G | H | I | J | K | L | M
N | O | P | Q | R | S | T | U | V | W | X | Y | Z

A

Add

D

DayFromPos
DayHeight
DayLeft
DayofWeekFromPos
DayTop
DayWidth

I

IndexFromTime
IsDateValid

M

MonthFromPos

R

Remove
RemoveAll

T

TaskFromPos
TimeFromIndex
TimeFromPos

W

WeekNumber
WhereIs

MinDate Property

See Also Applies To

Returns or sets the minimum date the control will be able to display.

Syntax

object . **MinDate** [= *text*]

The **MinDate** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the minimum date the control will be able to display.

Remarks

MinDate is set to 1/1/100 by default.

An error will be generated if the date is set to one that is earlier than the **MinDate**.

MinDate Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

MaxDate Property

Month Object

See Also Example Applies To

A **Month** object represents a month.

Syntax

Month

Remarks

Except for the **TagVariant** property, all properties of the **Month** object apply to that month in all years.

Month objects are accessible by an index of 1-12 for.

In addition, the **Month** object can be accessed using the **VisibleMonth** property of the **MonthView** an **YearView** controls.

A **Month** object has the following properties:

Properties

Caption	Month	StyleSet (YearView)
CaptionStyleSet (YearView)	Picture (MonthView)	TagVariant
DayCount	PictureStyle (MonthView)	Year
Enabled (YearView)		

Month Object Applies To

Month Property
VisibleMonth Property

Month Object Example

The following code modifies the caption of the first month on a MonthView control:

```
SSMonth1.Month(1).Caption = "Primero"
```

The following code sets the StyleSet for the left-most month on a **MonthView** control displaying 3 months:

```
SSMonth1.VisibleMonth(0).StyleSet = "Summer"
```

The following code disables December on a **YearView** control:

```
SSYear1.Month(12).Enabled = False
```

See Also

- **Day** Object
- **DayofWeek** Object
- Object Concepts

Month Property

See Also Example Applies To

Returns a **Month** object.

Syntax

object . **Month**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

Each month in the Julian calendar is sequentially numbered. (January = 1, February = 2, etc.).

This is only accessible via the **X** object when using a VBX.

Month Property (Month Object)

See Also Applies To

Returns the month number for the **Month** object.

Syntax

object . **Month**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Month Property (Month Object) Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

Month Object

Object Concepts

Month Property Applies To

DateCombo Control
MonthView Control
YearView Control

Month Property Example

The following code returns a **Month** object:

```
Dim MyMonth As Object
Set MyMonth = SSMonth1.X.Month(3) 'March
```

See Also

Month Object

MonthAlignment Property

See Also Applies To

Returns or sets the alignment of the caption within the Month Caption area.

Syntax

object . **MonthAlignment** [= *number*]

The **MonthAlignment** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the alignment of the caption.

Settings

The settings for *number* are:

Setting	Description
0	Left Justify
1	Right Justify
2	(Default) Center

MonthAlignment Property Applies To

YearView Control

See Also

MonthClick Event
MonthFromPos Method
MonthHeight Property
MonthLayout Property

MonthClick Event

Applies To

Occurs when you click the mouse in the Month Caption area.

Syntax

Sub *object* **_MonthClick** ([*Index* **As Integer**] *MonthNum* **As Integer**, *YearNum* **As Integer**)

The MonthClick event has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
Index	An integer expression that uniquely identifies the control if it is in a control array.
MonthNum	An integer expression that evaluates to the number of the month (1-12).
YearNum	An integer expression that evaluates to the number of the year (For example: 1995).

MonthClick Event Applies To

DateCombo Control
MonthView Control
YearView Control

MonthFromPos Method

See Also Applies To

Returns a **Month** object that corresponds to the position indicated by coordinates.

Syntax

object . **MonthFromPos**(*x* , *y* , [*scale*])

The **MonthFromPos** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
x	Required. A Single-precision integer expression that evaluates to the x-coordinate.
y	Required. A Single-precision integer expression that evaluates to the y-coordinate.
scale	Optional. An integer expression that evaluates to a value in the Settings list.

Settings

The settings for *scale* are:

Setting	Description
0	(Default) Twips
1	Pixels
2	Container
3	HiMetric

Remarks

The *x* and *y* parameters are applied using the top-left of the control as the origin (0,0).

This is only accessible via the **X** object when using a VBX.

MonthFromPos Method Applies To

MonthView Control
YearView Control

See Also

- **MonthAlignment** Property
- **MonthClick** Event
- **MonthHeight** Property
- **MonthLayout** Property

MonthHeight Property

See Also Applies To

Returns or sets the height of the MonthCaption area.

Syntax

object . **MonthHeight** [= *number*]

The **MonthHeight** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression that evaluates to the height of the MonthCaption area.

Remarks

The measurement is in the scale mode of the container. A setting of 0 causes the height to be automatically calculated, based on the font size.

MonthHeight Property Applies To

YearView Control

See Also

MonthAlignment Property
MonthClick Event
MonthFromPos Method
MonthLayout Property

MonthLayout Property

See Also Applies To

Returns or sets the layout of the months.

Syntax

object . **MonthLayout** [= *number*]

The **MonthLayout** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the layout of the control.

Settings

The settings for *number* are:

Setting	Description
0	(Default) Left to Right
1	Top to Bottom

MonthLayout Property Applies To

YearView Control

See Also

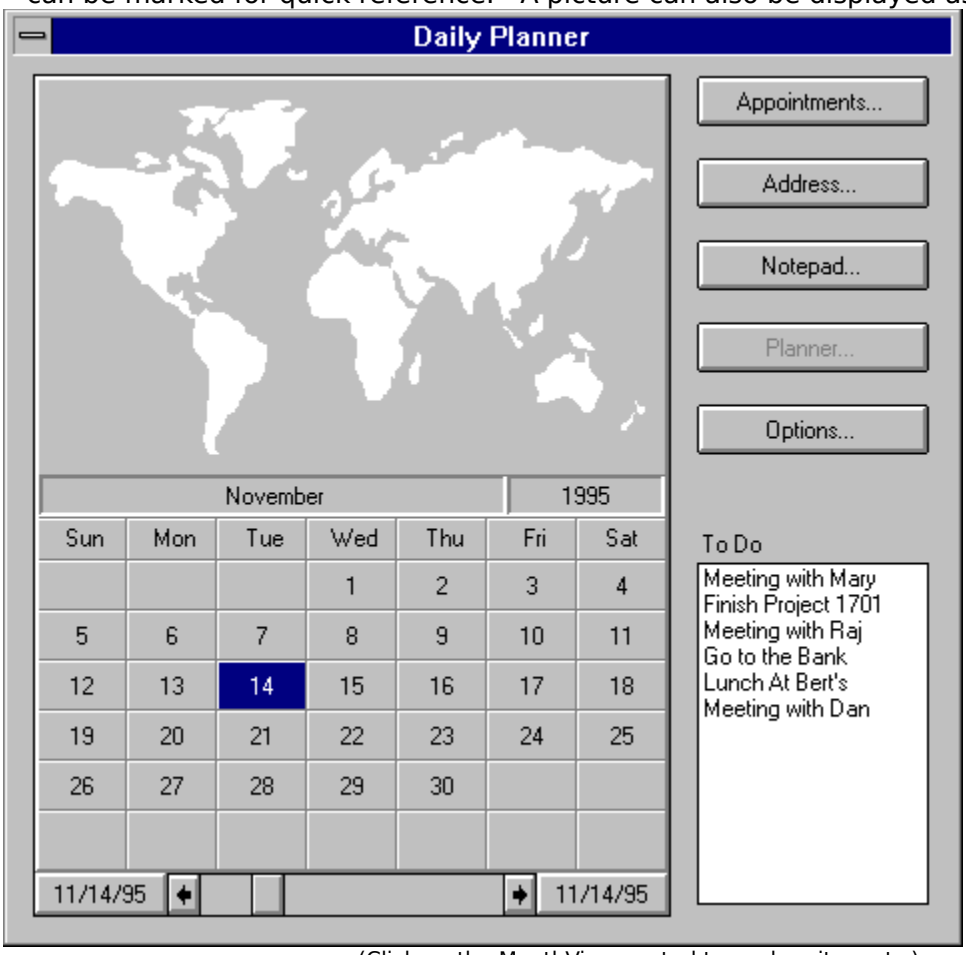
MonthAlignment Property
MonthClick Event
MonthFromPos Method
MonthHeight Property

The MonthView Control

See Also Properties Events Methods Objects Collections

The MonthView control lets you design applications that display date information in a monthly format (up to 3 months at a time).

The MonthView control provides several features which allow you to customize and integrate it into your applications. It is a data aware control so that it can be bound to a database for its source of information. Special days, such as holidays and other events, can be marked for quick reference. A picture can also be displayed as part of the control.



(Click on the MonthView control to explore its parts.)

File Name SSCALA.VBX, SSCALA16.OCX, SSCALA32.OCX
ObjectType SSMonth

Parts of the MonthView Control
Keyboard Interface
Marking Dates
Multiple Selection
Multi-Month Views
Using Pictures

Customizing

Back

MonthView Control Collections

Collections marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

SelectedDays Collection Õ

StyleSets Collection Õ

Back

MonthView Control Events

Events marked with a Ó are only accessible via the **X** object when using the VBX version of the control.

Click Event
DblClick Event
DragDrop Event
DragOver Event
FocusChange Event
GotFocus Event
InitMonth Event
KeyDown Event
KeyPress Event
KeyUp Event
LostFocus Event
MonthClick Event
MouseDown Event
MouseMove Event
MouseUp Event
SelChange Event
SelChanged Event
YearClick Event

Back

MonthView Control Methods

Methods marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

DayFromPos Method õ
DayHeight Method õ
DayLeft Method õ
DayofWeekFromPos Method õ
DayTop Method õ
DayWidth Method õ
MonthFromPos Method õ
Refresh Method
WeekNumber Method õ
WhereIs Method õ

Back

MonthView Control Objects

Objects marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

Day Object õ
DayofWeek Object õ
Font Object
Month Object õ
X Object

Back

MonthView Control Properties

Properties marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

FontItalic Property
FontName Property
FontSize Property
FontStrikethru Property
FontUnderline Property
ForeColor Property
ForeColorSelected Property
Height Property
HelpContextId Property
hWnd Property
Index Property
Left Property
MaxDate Property
MinDate Property
Month Property
MouseIcon Property
MousePointer Property
Name Property
NumberOfMonths Property
Orientation Property
Parent Property
PictureCalendar Property
ScrollBar Property
ScrollBarTracking Property
SelectionType Property
ShowCentury Property
ShowSelectedDate Property
ShowTodaysDate Property
StartOfWeek Property
TabIndex Property
TabStop Property
Tag Property
TagVariant Property õ
Top Property
Visible Property
VisibleMonth Property õ
Width Property

Customizing

There are many properties in the MonthView control that let you customize the display to your liking. In addition to bevels, alignment and color, the MonthView control contains special properties that help shape the control to look and feel the way you want.
The **StartOfWeek** property lets you determine what day of the week the view will start with. The following code sets the control's **StartOfWeek** property to Monday:

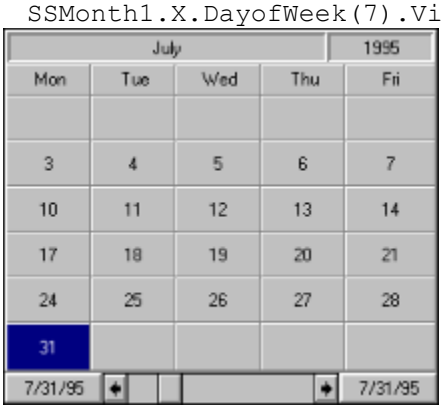
```
SSMonth1.StartOfWeek = 2
```

You can disable specific days or days of the week by setting their **Enabled** property. Once disabled, a day cannot be accessed either with the mouse or keyboard interface. The following code disables Sunday and Saturday:

```
SSMonth1.X.DayofWeek(1).Enabled = False
SSMonth1.X.DayofWeek(7).Enabled = False
```

You can also specify certain days-of-week to be hidden by setting their **Visible** property. Once a day-of-week is hidden, it cannot be accessed either with the mouse or keyboard interface. The following code turns Sunday and Saturday invisible:

```
SSMonth1.X.DayofWeek(1).Visible = False
SSMonth1.X.DayofWeek(7).Visible = False
```

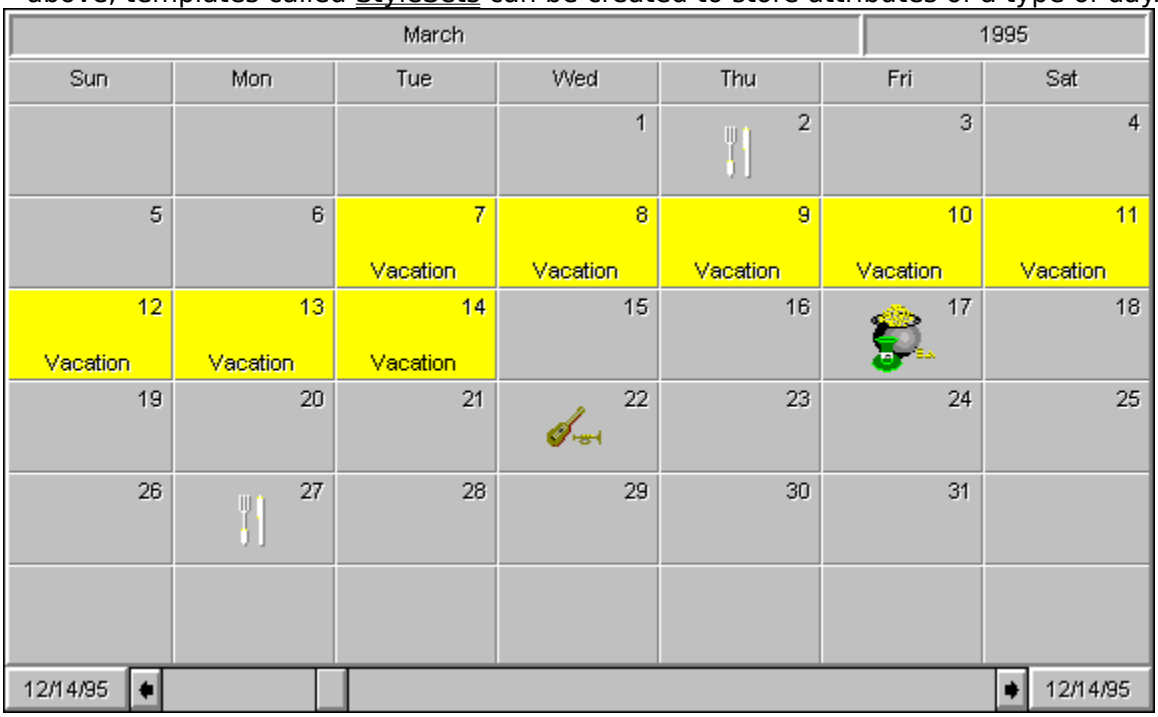


Keyboard Interface for MonthView Control

Press	To	Comments
LEFT ARROW	Focus on Previous Day	If previous day is visible, only the focus changes; otherwise, the control scrolls.
RIGHT ARROW	Focus on Next Day	If next day is visible, only the focus changes; otherwise, the control scrolls.
UP ARROW	Focus on Same Day of Previous Week	If previous week is visible, only the focus changes; otherwise, the control scrolls.
DOWN ARROW	Focus on Same Day of Next Week	If next week is visible, only the focus changes; otherwise, the control scrolls.
CTRL+LEFT ARROW	Focus on Same Day of Previous Month	If previous month is visible, only the focus changes; otherwise, the control scrolls.
CTRL+RIGHT ARROW	Focus on Same Day of Next Month	If next month is visible, only the focus changes; otherwise, the control scrolls.
PAGE UP	Focus on Same Day of Previous Year	Scrolls to the current month in the previous year.
PAGE DOWN	Focus on Same Day of Next Year	Scrolls to the current month in the next year.
HOME	Focus on First Day of Month	Sets the focus to the first day of the current month.
END	Focus on Last Day of Month	Sets the focus to the last day of the current month.
ENTER/ SPACEBAR	Select Day with Focus	Selects the day with focus.
ESCAPE	Reset Date	If control is bound and AutoRestore is True, resets selected date to the date in the database.
CTRL+S	Focus on Selected Day	Sets the focus to the selected date.
CTRL+T	Focus on Today	Sets the focus to today’s date.

Marking Dates

Each day on the control can be marked so that it carries special meaning. As explained above, templates called StyleSets can be created to store attributes of a type of day.



Using properties pertaining to color, font and picture, you will be able to give unique attributes to each StyleSet. Once a Styleset is added to the **StyleSets** collection, it can be applied to any date.

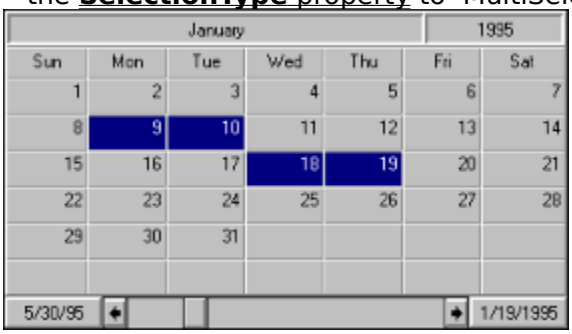
Multi-Month Views

The MonthView control allows you to display up to 3 months at a time. This is useful when you need to see more than one month, but less than a full year's worth of date information. Although you are able to see more than one month, the control operates the same way.

To see up to 3 months at a time, set the **NumberOfMonths** property greater than 1. In addition, the MonthView control allows you to orient the view both horizontally, as above, or vertically using the **Orientation** property.

Multiple Selection

More than one day can be selected at once. This feature will allow you to set properties of an entire group of days at the same time. Multiple date selection is enabled by setting the **SelectionType** property to 'MultiSelect'.

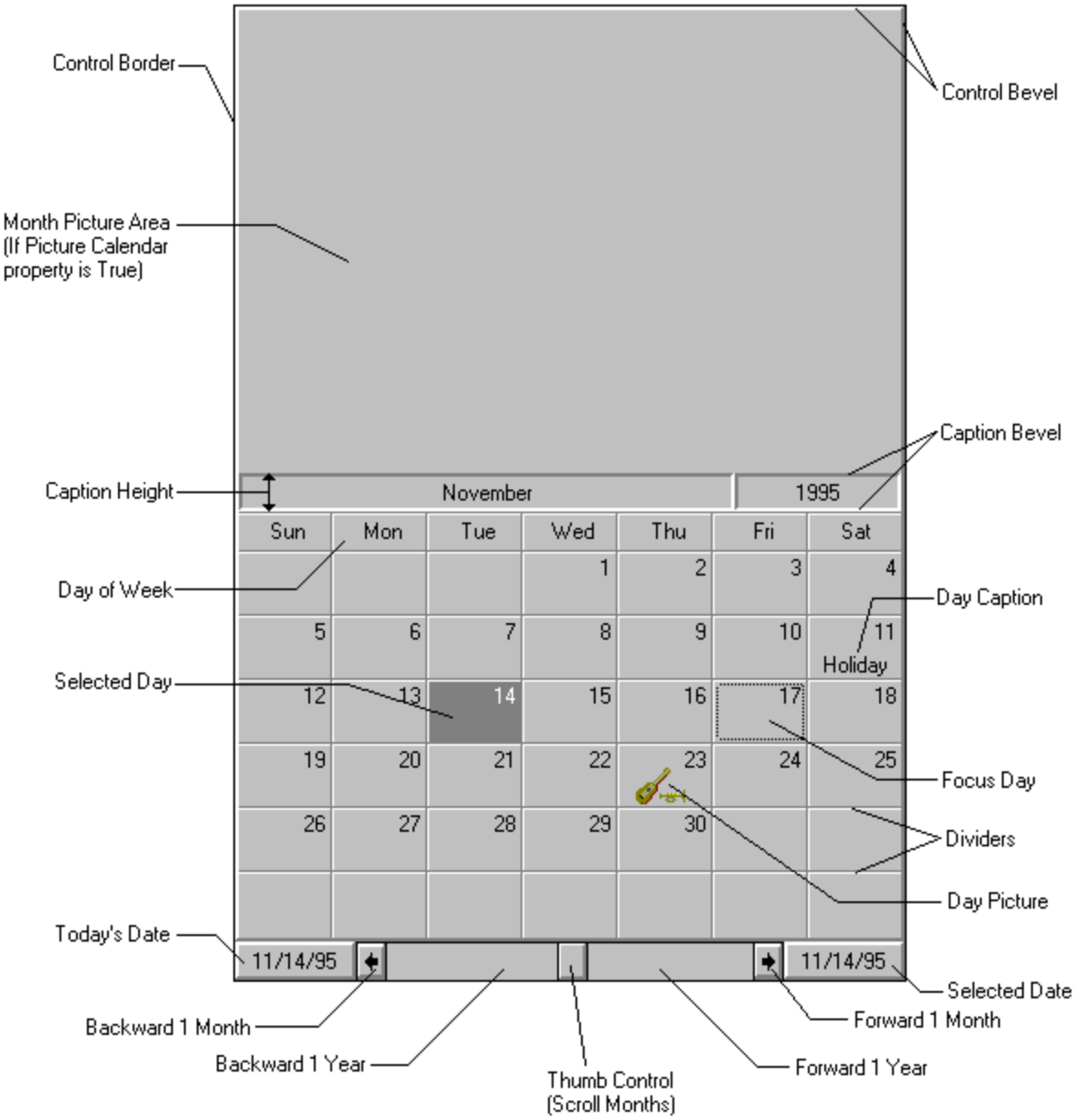


Once a set of days are selected, you can access them through the **SelectedDays** collection. The following code illustrates how you can apply a StyleSet to a group of selected days by iterating through the **SelectedDays** collection:

```
Dim i As Integer
For i = 0 to (SSMonth1.X.SelectedDays.Count - 1)
   SSMonth1.X.SelectedDays(i).Styleset = "Events"
Next
```

Exercise 2: Using a StyleSet in the Guided Tours illustrates how you can take avantage of this feature.

Parts of the MonthView Control



See Also

- **YearView** Control
- **DateCombo** Control
- **DayView** Control

Using Pictures

The MonthView control can contain a picture in the Month Picture for each month. This gives you the ability to have a different picture appear at the top of the calendar every month. In order to have a picture appear on top of each month, the **PictureCalendar** property must be set to **True**. The picture is applied by setting the **Picture** property of the **Month** object. Both of the following lines of code set the picture in the Month Picture area for November:

```
SSMonth1.X.Month(11).Picture = "NOVPIC.BMP"
```

-Or-

```
SSMonth1.X.Month(11).Picture = Picture1.Picture
```

In addition to the picture in the Month Picture area, the MonthView control allows you to place pictures within individual days. This is accomplished by applying a StyleSet, containing a picture, on a **Day** object of the control.

MouseIcon Property

See Also Applies To

Specifies a custom icon that will appear when the mouse passes over the control.

Syntax

object . **MouseIcon** [= *picture*]

The **MouseIcon** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
picture	An expression specifying a graphic, as described in Settings.

Settings

The settings for *picture* are:

Setting	Description
(None)	(Default) No picture.
(Icon)	Specifies a graphic. You can load the graphic from the Properties window at design time. At run time, you can also set this property using the **LoadPicture** function on an icon.

Remarks

For this property to be active, the **MousePointer** property must be set to '99 - Custom.'

Use the **DropDownMouseIcon** and **DropDownMousePointer** properties to specify a custom icon to appear when the mouse passes over the dropdown calendar portion of a **DateCombo**.

MouseIcon Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

See Also

MousePointer Property

MousePointer Property

See Also Example Applies To

Determines the type of mouse pointer that appears when the mouse passes over the control.

Syntax

object . **MousePointer** [= *number*]

The **MousePointer** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the type of mouse pointer that will be used.

Settings

The settings for *number* are:

Setting	Description
0	(Default) Default
1	Arrow
2	Cross
3	I-Beam
4	Icon
5	Size (four-pointed arrow pointing north, south, east and west)
6	Size NE SW (double arrow pointing northeast and southwest)
7	Size N S (double arrow pointing north and south)
8	Size NW SE (double arrow pointing northwest and southeast)
9	Size W E (double arrow pointing west and east)
10	Up Arrow
11	Hour Glass
12	No Drop
13	Arrow and Hourglass (32-bit OCX only)
14	Arrow and Question
15	Size All (32-bit OCX only)
99	Custom (Uses MouseIcon)

Remarks

Use the **DropDownMouseIcon** and **DropDownMousePointer** properties to specify a custom icon to appear when the mouse passes over the dropdown calendar portion of a **DateCombo**.

MousePointer Property Applies To

DateCombo Control
DayView Control
MonthView Control
YearView Control

MousePointer Property Example

Use the **DropDownMouseIcon** and **DropDownMousePointer** properties to specify a custom icon to appear when the mouse passes over the dropdown calendar portion of a **DateCombo**.

See Also

MouseIcon Property

Name Property (Font Object)

See Also Applies To

Returns or sets the font name of the **Font** object.

Syntax

object . **Name** [= *text*]

The **Name** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A string expression specifying the name of the font to be used.

Remarks

The **Font** object is not directly available at design time. Instead you set the **Name** property through a control's **Font** property.

At run time, however, you can set **Name** directly by specifying its setting for the **Font** object.

In Visual Basic 3.0 you set the **Name** property by selecting a control's **FontName**, **CaptionFontName**, **DayFontName**, **DropDownFontName** or **TimeSelectionBarFontName** property in the Visual Basic Properties window.

Name Property (Font Object) Applies To

Font Object

See Also

Font Object

Font Property

Depending on your host environment, this event may be refered to by a different name or in some cases not apply to this control. Refer to your host environment's documentation or help file for further information regarding this event.

Depending on your host environment, this method may be refered to by a different name or in some cases not apply to this control. Refer to your host environment's documentation or help file for further information regarding this method.

Depending on your host environment, this property may be refered to by a different name or in some cases not apply to this control. Refer to your host environment's documentation or help file for further information regarding this property.

NullDateLabel Property

See Also Applies To

Returns or sets the text that appears in the edit portion if no date is selected and the control does not have focus.

Syntax

object . **NullDateLabel** [= *text*]

The **NullDateLabel** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the text displayed as the label in the edit portion of the control if no date is selected.

Remarks

The label is drawn in italics, using the current font.

NullDateLabel Property Applies To

DateCombo Control

See Also

AllowNullDate Property

DefaultDate Property

NumberOfMonths Property

See Also Applies To

Returns or sets the number of months to display at one time.

Syntax

object . **NumberOfMonths** [= *number*]

The **NumberOfMonths** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the number of months to display at once.

Settings

The settings for *number* are:

Setting	Description
1	(Default) One Month
2	Two Months
3	Three Months

Remarks

When using the **Month** object, VisibleMonth(0) is the left-most visible month, VisibleMonth(1) is the center month and VisibleMonth(2) is the right-most visible month.

NumberOfMonths Property Applies To

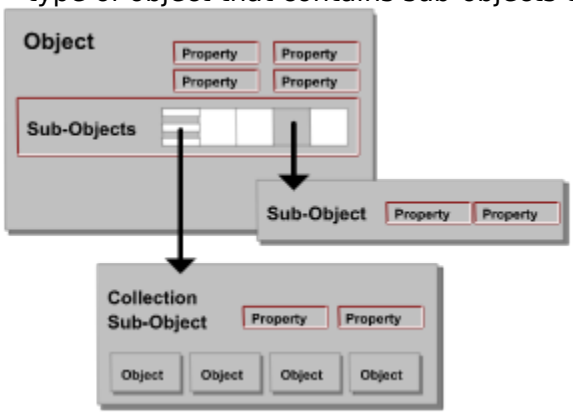
MonthView Control

See Also

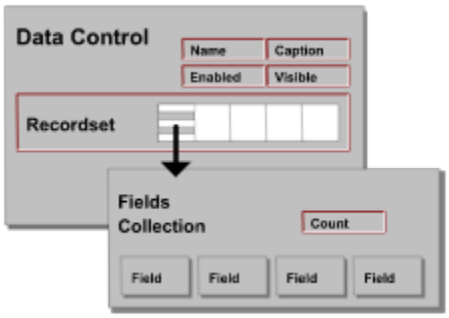
- **Month** Object
- **Orientation** Property
- **VisibleMonth** Property

Object Concepts - Sub-objects and Collections

Calendar Widgets provide an object-oriented approach to programming through the use of *sub-objects* and *collections* . An *object* refers to a single unit or entity that contains both properties and methods. Objects can contain other objects, which have properties and methods of their own that can be examined and changed. A *collection* is a special type of object that contains sub-objects that are all of the same type.



You are probably familiar with the concept of sub-objects if you have used the Visual Basic 3.0 data control. The **Recordset** object is a sub-object of the Visual Basic Data Control. The **Recordset** contains a collection sub-object called the **Fields** object, which contains information that relates to all the fields in the **Recordset** collectively. The **Fields** collection also contains the **Field** objects themselves.



Objects within collections often have this type of "paired" arrangement; a single collection object (**Fields**) which describes and contains the collection as a whole, and multiple member objects (**Field**) which make up the collection. In addition, there is usually a corresponding property of the same name as the object that returns information about the object.

Collections used by Calendar Widgets include the **SelectedDays** collection, which is made up of **Day** objects; the **StyleSets** collection and **StyleSet** objects; and the **Tasks** collections and **Task** objects.

Object Summary

See Also

A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z

D

Day
DayofWeek

F

Font

M

Month

P

Picture

S

StyleSet

T

Task

X

X (DateCombo)
X (DayView)
X (MonthView)
X (YearView)

Orientation Property

See Also Applies To

Returns or sets the orientation of the control.

Syntax

object . **Orientation** [= *number*]

The **Orientation** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the orientation of the control.

Settings

For the **MonthView** control, the settings for *number* are:

Setting	Description
0	(Default) Horizontal
1	Vertical

For the **YearView** control, the settings for *number* are:

Setting	Description
0	1 Column x 12 Rows
1	2 Columns x 6 Rows
2	(Default) 3 Columns x 4 Rows
3	4 Columns x 3 Rows
4	6 Columns x 2 Rows
5	12 Columns x 1 Row

Remarks

For the **MonthView** control:

For horizontal orientations, multiple months are displayed side by side and the scroll bar is placed at the bottom. For vertical orientations, multiple months are displayed one on top of the other and the scroll bar is placed at the right.

For the **YearView** control:

The **Orientation** property arranges the months into one of the 6 formations listed in the Settings List.

Orientation Property Applies To

MonthView Control
YearView Control

See Also

- **MonthLayout** Property
- **NumberOfMonths** Property
- **ScrollBar** Property (DateCombo/YearView)
- **ScrollBar** Property (MonthView)

Other Environments

See Also

Calendar Widgets OCX controls are supported by a variety of host environments. To use Calendar Widgets in other environments, consult your development tool's documentation for information on how to use *OLE Custom Controls* or *OCX Controls* .

To Use	Select (16-bit)	or (32-bit)
MonthView	SSCALA16.OCX	SSCALA32.OCX
YearView	SSCALA16.OCX	SSCALA32.OCX
DateCombo	SSCALA16.OCX	SSCALA32.OCX
DayView	SSCALB16.OCX	SSCALB32.OCX

Once the control is loaded, it should appear as an extension of your environment. Use the control's Property Pages or the environment's property sheet (if available) to set the properties of the control.

If your programming environment supports VBX controls, you must be sure it supports VBX controls compatible with the Visual Basic 3.0 specification. Calendar Widgets custom controls are not compatible with the earlier Visual Basic 1.0 or 2.0 specification.

Note Always use the VBX version of the control if you are developing in Visual Basic 3.0.

See Also

Compatibility Issues

Passing Variant Parameters into Methods

If you experience problems when passing Variant parameters into methods, it may be necessary to use the **CVar** function on the parameter. For example:

```
Dim lButtonLeft As Long
lButtonLeft = TimeLeft(CVar(MyVariantIndex))
```

This will guarantee that the parameter being passed into the method is a Variant. See the Visual Basic documentation or on-line help file for more information on the **CVar** function.

The following methods may be affected:

Methods

TaskFromPos	TaskWidth	TimeTop
TaskHeight	TimeFromPos	TimeWidth
TaskLeft	TimeHeight	WhereIs
TaskTop	TimeLeft	

Picture Object

See Also Applies To

The **Picture** object gives you a way to manipulate bitmaps, icons, and metafiles assigned to objects having a **Picture** property.

The **Picture** object supports the following properties and objects:

Properties

Handle	hPal	Width
Height	Type	

Syntax

Picture

Remarks

You frequently identify a **Picture** object using the **Picture** property of an object that displays graphics.

Picture Object Applies To

CaptionPicture Property
Picture Property
PictureDropDown Property

See Also

- **CaptionPicture** Property
- **Picture** Property
- **PictureDropDown** Property
- **PictureMetaHeight** Property
- **PictureMetaWidth** Property
- Pictures
- **PictureStyle** Property

Picture Property

See Also Applies To

For the **DayView** control, this property returns or sets the bitmap picture to appear in the caption area.

For the **Month** object, this property returns or sets the bitmap picture to appear in the Month Picture area.

Syntax

object . **Picture** [= *picture*]

The **Picture** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
picture	An expression specifying a graphic, as described in Settings.

Settings

The settings for *picture* are:

Setting	Description
(None)	(Default) No picture.
(Bitmap, icon, metafile)	Specifies a graphic. You can load the graphic from the Properties window at design time. At run time, you can also set this property using the name of the file containing the graphic.

Picture Property Applies To

DateCombo Control
DayView Control
Month Object
MonthView Control
Task Object
YearView Control

See Also

- **PictureCalendar** Property
- **PictureDropDown** Property
- **PictureMetaHeight** Property
- **PictureMetaWidth** Property
- **PictureStyle** Property
- Pictures

PictureAlignment Property

Applies To

Returns or sets the alignment of the Caption Picture.

Syntax

object . **PictureAlignment** [= *number*]

The **PictureAlignment** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the alignment of the Caption Picture.

Settings

The settings for *number* are:

Setting	Description
0	(Default) Left of Text
1	Right of Text
2	Fit to Caption
3	Tile

PictureAlignment Property Applies To

DayView Control

PictureCalendar Property

See Also Applies To

Determines if pictures can be displayed on the month view area.

Syntax

object . **PictureCalendar** [= *boolean*]

The **PictureCalendar** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if pictures can be displayed on the Month View area.

Settings

The settings for *boolean* are:

Setting	Description
True	Allows pictures to be displayed on the top of the month.
False	(Default) Month Picture area is not displayed.

Remarks

This property determines if pictures can be displayed in the Month Picture area (top of the month).

PictureCalendar Property Applies To

MonthView Control

See Also

Month Object
Picture Property
PictureStyle Property

PictureDropDown Property

See Also Applies To

Returns or sets a **Picture** object for the picture that will appear on the dropdown button in place of the down arrow.

Syntax

object . **PictureDropDown** [= *picture*]

The **PictureDropDown** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
picture	An expression specifying a graphic, as described in Settings.

Settings

The settings for *picture* are:

Setting	Description
(None)	(Default) No picture.
(Bitmap)	Specifies a graphic. You can load the graphic from the Properties window at design time.

Remarks

The default button that drops down the calendar portion of the **DateCombo** can be set to use a custom picture.

PictureDropDown Property Applies To

DateCombo Control

See Also

Pictures

PictureMetaHeight Property

See Also Applies To

Sets the height of a metafile selected as a picture.

Syntax

object . **PictureMetaHeight** [= *number*]

The **PictureMetaHeight** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression that evaluates to the height of a metafile selected as a picture.

Remarks

The units specified are based on the scale mode of the container.

For the **DayView** control, the **PictureMetaHeight** property sets the height of the picture in the Caption Picture area if it is a metafile.

PictureMetaHeight Property Applies To

DayView Control
StyleSets Collection

See Also

PictureMetaWidth Property

PictureMetaWidth Property

See Also Applies To

Sets the width of a metafile selected as a picture.

Syntax

object . **PictureMetaWidth** [= *number*]

The **PictureMetaWidth** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression that evaluates to the width of a metafile selected as a picture.

Remarks

The units specified are based on the scale mode of the container.

For the **DayView** control, the **PictureMetaHeight** property sets the width of the picture in the Caption Picture area if it is a metafile.

PictureMetaWidth Property Applies To

DayView Control
StyleSets Collection

See Also

PictureMetaHeight Property

PictureStyle Property

See Also Applies To

Returns or sets the style of the picture to display for the Month object.

Syntax

object . **PictureStyle** [= *number*]

The **PictureStyle** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the style of the picture to display for the Month object.

Settings

The settings for *number* are:

Setting	Description
0	(Default) Center.
1	Stretch to Fit (stretch the image so it fills the entire Month Picture area).
2	Tile (repeat the image to that the entire Month Picture area is filled).

Remarks

This property is only available at run time.

PictureStyle Property Applies To

Month Object

See Also

Month Object
Picture Property
PictureCalendar Property

Pictures

As with fonts, both the VBX and OCX versions of the Calendar Widgets controls support the **Picture** object.

The following properties are supported under the **Picture** object:

Properties

Handle	hPal	Width
Height	Type	

Note When you use the VBX version of the controls, control-level picture properties do not return a **Picture** object. The above properties of the **Picture** object can only be set when using the OCX version of the controls or on picture properties of sub-objects using the VBX version of the control.

When you use the VBX version of the controls, the **LoadPicture** function cannot be used to set the **Picture** property of a sub-object. Instead, the **Picture** property must be set to one of the following:

- A literal string containing the name of the file containing the picture.
- The picture property of another control.
- The picture property of another sub-object.
- The handle to a bitmap, icon or metafile.

The following line of code sets the **Picture** property of a **StyleSet** object in the **StyleSets** collection to a string containing the name of a bitmap file:

```
SSMonth1.X.StyleSets(1).Picture = "MYPIC.BMP"
```

The next line of code sets it to the **Picture** property of a Picture control:

```
SSMonth1.X.StyleSets(1).Picture = Picture1.Picture
```

The next line of code sets it to the **Picture** property of another sub-object:

```
SSMonth1.X.StyleSets(1).Picture = SSMonth1.X.StyleSets(2).Picture
```

The next line of code removes a picture by setting the Picture property to an empty string ("""):

```
SSMonth1.X.StyleSets(1).Picture = ""
```

PromptChar Property

See Also Example Applies To

Returns or sets the character that acts as a placeholder for day, month and year numbers when the entry is empty.

Syntax

object . **PromptChar** [= *text*]

The **PromptChar** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that specifies a placeholder character.

Remarks

The **PromptChar** property is set to the underline (' _ ') character by default.

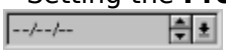
This property is only used during date entry to visually indicate empty spaces. It does not become part of the date text.

PromptChar Property Applies To

DateCombo Control

PromptChar Property Example

Setting the **PromptChar** property to ' - ' will display the following in the DateCombo:



As you type in the month as '03' the display will change to the following:

See Also

FormattedText Property

RawText Property

Property Pages

Property Summary

See Also

A | B | C | D | E | F | G | H | I | J | K | L | M
N | O | P | Q | R | S | T | U | V | W | X | Y | Z

D

See Also

- Collections
- Events
- Methods
- Objects
- Properties

RawText Property

See Also Example Applies To

Returns or sets the text, without separators, for the edit portion of the DateCombo.

Syntax

object . **RawText** [= *text*]

The **RawText** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the text to place in the edit portion.

Remarks

This property is not available at design time and is read-only at run time.

RawText Property Applies To

DateCombo Control

RawText Property Example

For the the following DateCombo:

the **RawText** property would return a value of '111495'.

See Also

FormattedText Property

Remove Method

See Also Applies To

This method is used to remove an item from a collection.

Syntax

object . **Remove** *selecteddate*

object . **Remove** *stylesetname*

object . **Remove** *index*

The **Remove** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
selecteddate	Required. Used with **SelectedDays**. A string expression specifying the date of the **Day** object you want to remove.
stylesetname	Required. Used with **StyleSets**. A string expression specifying the name of the **StyleSet** object you want to remove.
index	Required. Used with **SelectedDays**, **StyleSets** and **Tasks**. An integer expression specifying the index value that uniquely identifies the object you want to remove.

Remarks

Once an item is removed, it will no longer be available.

Remove Method Applies To

SelectedDays Collection
StyleSets Collection
Tasks Collection

See Also

- **Add** Method
- **Count** Property
- **RemoveAll** Method
- **SelectedDays** Collection
- **StyleSets** Collection
- **Tasks** Collection

RemoveAll Method

See Also Example Applies To

Used to remove all items from a collection.

Syntax

object . **RemoveAll**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

Once an item is removed from the collection, it is no longer available.

RemoveAll Method Applies To

SelectedDays Collection
StyleSets Collection
Tasks Collection

RemoveAll Method Example

The following code removes all **Day** objects from the **SelectedDays** collection:

```
SSMonth1.X.SelectedDays.RemoveAll
```

The following code removes all **StyleSet** objects from the **StyleSets** collection:

```
SSMonth1.X.StyleSets.RemoveAll
```

The following code removes all **Task** objects from the **Tasks** collection:

```
SSMonth1.X.Tasks.RemoveAll
```

See Also

- **Add** Method
- **Count** Property
- **Remove** Method
- **SelectedDays** Collection
- **StyleSets** Collection
- **Tasks** Collection

Reset Method

See Also Example Applies To

Used to reset the properties of a **StyleSet** object in the **StyleSets** collection to their default values.

Syntax

object . **Reset**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

This method, in effect, un-applies a StyleSet without actually removing it from the **StyleSets** collection. Once a StyleSet is reset, it is still available from the StyleSets collection.

Reset Method Applies To

StyleSets Collection

Reset Method Example

The following code resets the 'Vacation' StyleSet:

```
SSMonth1.X.StyleSets('Vacation').Reset
```

See Also

- **Add** Method
- **Count** Property
- **Remove** Method
- **RemoveAll** Method
- **StyleSets** Collection

ScrollBar Property (DateCombo/YearView)

See Also Applies To

Specifies the characteristics of the scroll bar.

Syntax

object . **ScrollBar** [= *number*]

The **ScrollBar** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the characteristics of the scroll bar.

Settings

The settings for *number* are:

Setting	Description
0	None
1	(Default) Horizontal
2	Vertical

Remarks

If a horizontal scroll bar is selected, it is placed on the bottom.

If a vertical scroll bar is selected, it is placed on the right.

ScrollBar Property (DateCombo/YearView) Applies To

DateCombo Control
YearView Control

See Also

ScrollBarTracking Property

ScrollBar Property (MonthView)

See Also Applies To

Determines whether or not a scroll bar will be displayed.

Syntax

object . **ScrollBar** [= *boolean*]

The **ScrollBar** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying .

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Scroll bar is displayed.
False	Scroll bar is not displayed.

Remarks

This property determines whether or not a scroll bar will be displayed. If it is not, use the keyboard to navigate.

ScrollBar Property (MonthView) Applies To

MonthView Control

See Also

ScrollBarTracking Property

ScrollBarTracking Property

See Also Applies To

Returns or sets if the control is updated as you scroll using the Thumb Control of the scroll bar.

Syntax

object . **ScrollBarTracking** [= *boolean*]

The **ScrollBarTracking** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if the calendar is updated as you scroll using the Thumb Control.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Calendar is updated during scrolling.
False	Calendar is not updated until scrolling with the mouse is finished by releasing the mouse button.

Remarks

This property determines if the calendar is updated while dragging the thumb of the scrollbar.

ScrollBarTracking Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

ScrollBar Property (DateCombo/YearView)

ScrollBar Property (MonthView)

SelChange Event

See Also Example Applies To

Occurs when the selection status of a day changes.

Syntax

Sub *object* _SelChange ([*Index* **As Integer**] *SelDate* **As String**, *OldSelDate* **As String**, *Selected* **As Integer**, *RtnCancel* **As Integer**)

The SelChange event has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
Index	Uniquely identifies the control if it is in a control array.
SelDate	A string expression that evaluates to the newly selected or de-selected date.
OldSelDate	A string expression that evaluates to the previously selected date.
Selected	Determines if the day is selected or deselected.
RtnCancel	Determines if the control should abort the selection change. The parameter default is False.

Remarks

Set the *RtnCancel* parameter to **True** to cancel a change in selection status.

OldSelDate will return a null string ("") if no date was previously selected.

SelChange Event Applies To

MonthView Control
YearView Control

SelChange Event Example

If 5/30/95 is currently selected and you click on 12/14/95, the SelChange event will fire. At this point the *SelDate* parameter will return 12/14/95, and the *Selected* parameter will return True (-1).

However, if 5/30/95 is de-selected by clicking on it, *SelDate* will return 5/30/95 and *Selection* will return False (0).

See Also

SelChanged Event

SelChanged Event

See Also Applies To

Occurs after the selection status of a day changes.

Syntax

Sub *object* _SelChanged ([*Index* **As Integer**] *SelDate* **As String**, *OldSelDate* **As String**, *Selected* **As Integer**)

The SelChanged event has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
Index	Uniquely identifies the control if it is in a control array.
SelDate	A string expression that evaluates to the newly selected or de-selected date.
OldSelDate	A string expression that evaluates to the previously selected date.
Selected	Determines if the day is selected or deselected.

Remarks

Use this event to perform actions once the selection status has changed.

This event will not fire if the *RtnCancel* parameter of the SelChange event is set to **True**.

SelChanged Event Applies To

MonthView Control
YearView Control

See Also

SelChange Event

Selected Property

See Also Example Applies To

Determines if a specified **Day** object is selected.

Syntax

object . **Selected** [= *boolean*]

The **Selected** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if the specified **Day** object is selected

Settings

The settings for *boolean* are:

Setting	Description
True	Specified **Day** object is selected.
False	Specified **Day** object is not selected.

Remarks

The **SelectionType** property must be set to MultiSelect for more that one **Day** object to be selected.

Once a **Day** object is selected, it is automatically added to the **SelectedDays** collection.

Note Bound controls ignore the **SelectionType** property. They are always Single Select.

Selected Property Applies To

Day Object

Selected Property Example

The following code selects the first three days of the current month in a **MonthView** control:

```
SSMonth1.SelectionType = 1   'MultiSelect
SSMonth1.X.Day(1).Selected = True
SSMonth1.X.Day(2).Selected = True
SSMonth1.X.Day(3).Selected = True
```

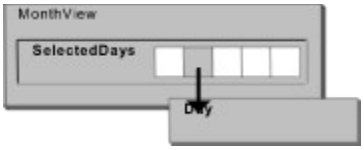
See Also

- **Day** Object
- **SelectedDays** Collection
- **SelectionType** Property

SelectedDays Collection

See Also Example Applies To

Contains a collection of **Day** objects that have been selected.



The **SelectedDays** collection supports the following properties and methods:

Properties

Count

Methods

Add Remove RemoveAll
Item

Syntax

object . **SelectedDays**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

The **SelectedDays** collection is accessible as an object.

The **Day** objects within the **SelectedDays** collection are accessible through the **Item** method and can be iterated over.

Note The **SelectedDays** collection will only contain one day if the **SelectionType** property is set to Single Select.

This is only accessible via the **X** object when using a VBX.

SelectedDays Collection Applies To

MonthView Control
YearView Control

SelectedDays Collection Example

The following code creates a collection of 4 **Day** objects:

```
SSMonth1.X.SelectedDays.Add "12/14/95"
SSMonth1.X.SelectedDays.Add "7/31/95"
SSMonth1.X.SelectedDays.Add "5/30/78"
SSMonth1.X.SelectedDays.Add "11/14/72"
```

The following code removes ‘12/14/95’ from the **SelectedDays** collection:

```
SSMonth1.X.SelectedDays.Remove "12/14/95"
```

The following code removes the first selected date from the **SelectedDays** collection:

```
SSMonth1.X.SelectedDays.Remove 0
```

The following code removes all **Day** objects from the **SelectedDays** collection:

```
SSMonth1.X.SelectedDays.RemoveAll
```

The following code returns the number of Day objects in the **SelectedDays** collection:

```
NumDays = SSMonth1.X.SelectedDays.Count
```

The following code iterates over each **Day** object in the **SelectedDays** collection:

```
Dim DayNum As Integer
For DayNum = 0 to SSMonth1.X.SelectedDays.Count - 1
   SSMonth1.X.SelectedDays(DayNum).Caption = "Vacation"
Next
```

See Also

- **Add** Method
- **Count** Property
- **Date** Property
- **Day** Object
- **Remove** Method
- **RemoveAll** Method
- **SelChange** Event

SelectionType Property

See Also Applies To

Returns or sets the selection type.

Syntax

object . **SelectionType** [= *number*]

The **SelectionType** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the selection type.

Settings

The settings for *number* are:

Setting	Description
0	(Default) Single Select
1	MultiSelect

Remarks

This property is used to determine whether one day or a multiple number of days may be selected. In the Single Select mode, only one day can be selected in the entire calendar. In the MultiSelect mode, many days can be selected.

Note Bound controls ignore the **SelectionType** property. They are always Single Select.

SelectionType Property Applies To

MonthView Control
YearView Control

See Also

- **Day** Object
- **Selected** Property
- **SelectedDays** Collection

ShowCentury Property

See Also Applies To

Determines if the century part of the year is displayed for 1900’s.

Syntax

object . **ShowCentury** [= *boolean*]

The **ShowCentury** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if the century part of the year is displayed.

Settings

The settings for *boolean* are:

Setting	Description
True	Century number is always displayed.
False	(Default) Century number is not displayed for years in the 1900's.

Remarks

If the **ShowCentury** property is set to **False**, '1972' will be displayed as '72'. This is only done for years in the 1900's.

ShowCentury Property Applies To

DateCombo Control
MonthView Control
YearView Control

See Also

- **CaptionAlignmentBeginYear** Property
- **CaptionAlignmentEndYear** Property
- **Format** Property
- **Mask** Property
- **ShowSelectedDate** Property
- **ShowTodaysDate** Property

ShowEdit Event

See Also Applies To

Occurs when the edit box is displayed in a time slot.

Syntax

Sub *object* **_ShowEdit** (*TaskIndex* **As Integer**, *Action* **As Integer**)

The ShowEdit event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
TaskIndex	An Integer expression that evaluates to the index of the task in the **Tasks** collection.
Action	An integer expression that indicates whether the task was added (0) or edited (1).

Remarks

The ShowEdit event is fired when a time slot is placed into edit mode by either double-clicking on it, pressing the SPACEBAR, or setting the **EditVisible** property to **True**.

ShowEdit Event Applies To

DayView Control

See Also

- **CloseUp** Event
- **EditVisible** Property
- **Tasks** Collection

ShowSelectedDate Property

See Also Applies To

Determines if the Selected Date button will be displayed in the lower right corner of the calendar.

Syntax

object . **ShowSelectedDate** [= *boolean*]

The **ShowSelectedDate** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if the Selected Date button will be displayed.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Selected Date button appears.
False	Selected Date button does not appear.

Remarks

This property determines if the Selected Date button will appear on the calendar. If set to **False**, the scroll bar area will be expanded.

Note The Selected Date button can be used to iterate over selected days in MultiSelect mode.

ShowSelectedDate Property Applies To

MonthView Control
YearView Control

See Also

- **Date** Property
- **SelectedDays** Collection
- **SelectionType** Property

ShowTaskColor Property

See Also Applies To

Determines if the Task BackColor will be shown in a rectangle within the Task area.

Syntax

object .**ShowTaskColor** [= *boolean*]

The **ShowTaskColor** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if the Task BackColor of a task appears in the Task area.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Task BackColor is shown in a rectangle in Task area.
False	Task BackColor is not shown in a rectangle in Task area.

Remarks

If this property is set to **True**, a colored rectangle will appear to the left of a task's text and/or picture. The rectangle will be filled with the Task BackColor. The outline of the rectangle will be determined by the **BevelColorFrame** property setting of the control.

ShowTaskColor Property Applies To

DayView Control

See Also

BackColor Property

BevelColorFrame Property

ShowTodaysDate Property

See Also Applies To

Determines if the Today's Date button will be displayed in the lower left corner of the calendar.

Syntax

object . **ShowTodaysDate** [= *boolean*]

The **ShowTodaysDate** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if the Today's Date button will be displayed.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Today's Date button appears.
False	Today’s Date button does not appear.

Remarks

This property determines if the Today's Date button will appear on the calendar. If set to **False**, the scroll bar area will expand.

Note You can click on the Today's Date button to change the focus to today’s date.

ShowTodaysDate Property Applies To

MonthView Control
YearView Control

See Also

DefaultDate Property

Size Property

See Also Applies To

Returns or sets the font size used in the **Font** object.

Syntax

object . **Size** [= *number*]

The **Size** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression specifying the size of the font in points.

Remarks

Use this property to format text in the font size you want. To change the default, specify the size of the font in points.

The **Font** object is not directly available at design time. Instead you set the **Size** property through a control's **Font** property.

At run time, however, you can set **Size** directly by specifying its setting for the **Font** object.

In Visual Basic 3.0 you set the **Size** property by selecting a control's **FontSize**, **CaptionFontSize**, **DayFontSize**, **DropDownFontSize** or **TimeSelectionBarFontSize** property in the Visual Basic Properties window.

Size Property Applies To

Font Object

See Also

Font Object

Font Property

Spin Event

Applies To

Occurs when the user clicks on a spin button to change the date..

Syntax

Sub *object* **_Spin** (*OldDate* **As String**, *NewDate* **As String**)

The Spin event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
OldDate	A string expression that evaluates to the date in the edit portion of the combo before the spin button was clicked.
NewDate	A string expression that evaluates to the date in the edit portion of the combo after the spin button was clicked.

Remarks

The Spin event is fired when a user changes the date in a DateCombo. This gives you the ability to validte the newly selected date.

Spin Event Applies To

DateCombo Control

SpinButton Property

See Also Applies To

Determines the position of the spin button within the combo box.

Syntax

object . **SpinButton** [= *number*]

The **SpinButton** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the position of the spin button.

Settings

The settings for *number* are:

Setting	Description
0	No SpinButton
1	(Default) SpinButton to right
2	SpinButton to left

SpinButton Property Applies To

DateCombo Control

See Also

EditMode Property

StartMonth Property

See Also Applies To

Returns or sets the month that will appear in the upper left position of display.

Syntax

object . **StartMonth** [= *number*]

The **StartMonth** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the month that will appear in the top left position of the control.

Settings

The settings for *number* are:

Setting	Description
1	(Default) January
2	February
3	March
4	April
5	May
6	June
7	July
8	August
9	September
10	October
11	November
12	December

Remarks

This property determines which month will appear as the first month. This is useful for creating special year views such as fiscal years.

StartMonth Property Applies To

YearView Control

See Also

StartOfWeek Property

StartOfWeek Property

See Also Example Applies To

Returns or sets which day of the week will appear in the first (far left) column of each month.

Syntax

object . **StartOfWeek** [= *number*]

The **StartOfWeek** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the day of the week that will appear in the far left column of each month in the control.

Settings

The settings for *number* are:

Setting	Description
1	(Default) Sunday
2	Monday
3	Tuesday
4	Wednesday
5	Thursday
6	Friday
7	Saturday

Remarks

This property determines which day will be used as the start of the week.

StartOfWeek Property Applies To

DateCombo Control
MonthView Control
YearView Control

StartOfWeek Property Example

If **StartOfWeek** is Sunday the display will be:

If **StartOfWeek** is Monday the display will be

See Also

DayofWeek Object

StrikeThrough Property

See Also Applies To

Returns or sets the font style of the **Font** object to either strikethrough or nonstrikethrough.

Syntax

object . **StrikeThrough** [= *boolean*]

The **StrikeThrough** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying the font style as described in Settings.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Turns on strikethrough formatting.
False	Turns off strikethrough formatting.

Remarks

The **Font** object is not directly available at design time. Instead you set the **StrikeThrough** property through a control's **Font** property.

At run time, however, you can set **StrikeThrough** directly by specifying its setting for the **Font** object.

In Visual Basic 3.0 you set the **StrikeThrough** property by selecting a control's **FontStrikethru**, **CaptionFontStrikethru**, **DayFontStrikethru**, **DropDownFontStrikethru** or **TimeSelectionBarFontStrikethru** property in the Visual Basic Properties window.

StrikeThrough Property Applies To

Font Object

See Also

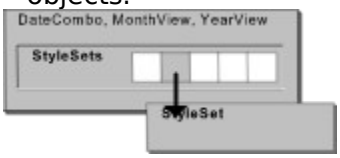
Font Object

Font Property

StyleSet Object

See Also Applies To

The **StyleSet** object contains properties pertaining to **Day**, **DayofWeek** or **Month** objects.



The **StyleSet** object supports the following properties and objects:

Properties

BackColor	Name	PictureMetaHeight
Font	Picture	PictureMetaWidth
ForeColor		

Objects

Font
Picture

Syntax

StyleSet

Remarks

You frequently identify a **StyleSet** object using the **StyleSet** property.

StyleSet Object Applies To

StyleSets Collection

See Also

- **CaptionStyleSet** Property
- **DayCaptionStyleSet** Property
- **StyleSet** Property
- StyleSets
- **StyleSets** Collection

StyleSet Property

See Also Applies To

Returns or sets the name of a **StyleSet** in the **StyleSets** collection.

Syntax

object . **StyleSet** [= *text*]

The **StyleSet** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the name of a StyleSet.

Remarks

This property determines the StyleSet to be used for the **Day**, **DayofWeek** or **Month** objects. Note that the StyleSet specified must be in the **StyleSets** collection, and its properties must be set before the StyleSet can be used.

This property is only available at run time.

Note If a change is made to a StyleSet, the control will need to be refreshed.

Following is a list of properties used in the various StyleSets. Shown, are this **StyleSet** property (with **Day** and **DayofWeek** objects) along with the other StyleSet properties: **CaptionStyleSet**, **DayCaptionStyleSet** and **DayStyleSet**.

StyleSet properties used by (CaptionStyleSet) (DayCaptionStyleSet)

Font	ForeColor

StyleSet properties used by (Day.StyleSet) (DayStyleSet)

BackColor	ForeColor	PictureMetaHeight
Font	Picture	PictureMetaWidth

StyleSet properties used by (DayofWeek) (Month.StyleSet)

BackColor	Font	ForeColor

StyleSet Property Applies To

Day Object
DayofWeek Object
Month Object

See Also

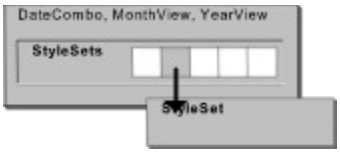
- **Day** Object
- **DayofWeek** Object
- StyleSets
- **StyleSets** Collection

StyleSets

StyleSets Collection

See Also Example Applies To

Contains a collection of **StyleSet** objects.



The **StyleSets** collection supports the following properties, methods and objects:

Properties

Count

Methods

Add	Remove	Reset
Item	RemoveAll	

Objects

StyleSet

Note If a change is made to a StyleSet, the control may have to be refreshed.

Syntax

object . **StyleSets**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

The **StyleSets** collection is accessible as an object.

The **StyleSet** objects within the collection are accessible through the **Item** method and can be iterated over.

The StyleSet properties can be set to distinguish one StyleSet from another. A StyleSet called 'Holiday' may have a **BackColor** of green, a **ForeColor** of red and a **Font** of 'Times New Roman.' This StyleSet can then be applied to any number of **Day** and/or **DayofWeek** objects through the **StyleSet** property. It can also be applied to the caption of all **Day** objects through the **DayCaptionStyleSet** property.

Note that not all properties of a StyleSet are used in every case. For instance, the **DayCaptionStyleSet** can be set to 'Holiday', but only the **ForeColor** and **Font** will actually be reflected in the captions.

This is only accessible via the **X** object when using a VBX.

StyleSets Collection Applies To

DateCombo Control
MonthView Control
YearView Control

StyleSets Collection Example

The following code adds the 'Vacation' StyleSet to the **StyleSets** collection, sets the object's background color to red and then applies the StyleSet:

```
SSMonth1.X.StyleSets.Add "Vacation"
SSMonth1.X.StyleSets("Vacation").BackColor = RGB(255,0,0)
SSMonth1.DayCaptionStyleSet = "Vacation"
```

(The use of the Add method in the above code is optional.)

The following code removes the 'Vacation' StyleSet from the **StyleSets** collection:

```
SSMonth1.X.StyleSets.Remove "Vacation"
```

The following code removes the first StyleSet from the StyleSets collection:

```
SSMonth1.X.StyleSets.Remove 0
```

The following code removes all **StyleSet** objects from the **StyleSets** collection:

```
SSMonth1.X.StyleSets.RemoveAll
```

The following code returns the number of StyleSets in the **StyleSets** collection:

```
NumStyleSets = SSMonth1.X.StyleSets.Count
```

The following code iterates over each **StyleSet** object in the **StyleSets** collection:

```
Dim SSNum As Integer
For SSNum = 0 to SSMonth1.X.StyleSets.Count - 1
   SSMonth1.X.StyleSets(SSNum).BackColor = RGB(255,0,0)
Next
```

See Also

StyleSet Property

System Requirements

§ Microsoft Windows version 3.1 or higher.

§ Microsoft Visual Basic version 3.0 or higher, (See hardware and system requirements for installing Visual Basic in the Visual Basic Programmer's Guide, 'Chapter 1 - Setting Up.') or any development environment that supports either Level 3 VBX controls or 16-bit or 32-bit OCX controls.

§ At least 4 megabytes of available space on your hard disk.

TagVariant Property

See Also Example Applies To

Stores any extra data needed for your program.

You can use this property to attach data of any type, except user defined types, to an object or control.

Syntax

object . **TagVariant** [= *expression*]

The **TagVariant** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
expression	A Variant expression.

Remarks

The **TagVariant** property is similar to the Visual Basic **Tag** property. However, in addition to string expressions, the **TagVariant** property can store any data type including other objects.

Note The **TagVariant** property can store all data types except user defined types.

This property is only available at run time.

This is only accessible via the **X** object when using a VBX.

TagVariant Property Applies To

DateCombo Control
Day Object
DayofWeek Object
DayView Control
Month Object
MonthView Control
Task Object
YearView Control

TagVariant Property Example

The following code illustrates how the **TagVariant** property can be set to a double-precision floating point number:

```
Dim MyTaxRate As Double
MyTaxRate = 0.0825
SSMonth1.TagVariant = MyTaxRate
```

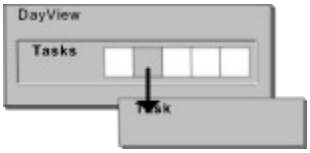
See Also

- **Day** Object
- **DayofWeek** Object
- **Month** Object

Task Object

See Also Applies To

A **Task** object is an object that contains properties relating to a single task.



Syntax

Task

Remarks

A **Task** object contains the following properties and methods:

Properties

BackColor	EndTime	TagVariant
BeginTime	Picture	Text
Duration		

Task Object Applies To

Tasks Collection

See Also

Tasks Collection

TaskFromPos Method

See Also Applies To

Returns a **Task** object that corresponds to the position indicated by coordinates.

Syntax

object . **TaskFromPos**(*x* , *y* , [*scale*])

The **TaskFromPos** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
x	Required. An integer expression that evaluates to the x-coordinate.
y	Required. An integer expression that evaluates to the y-coordinate.
scale	Optional. A Variant expression that evaluates to a value in the Settings list.

Settings

The settings for *scale* are:

Setting	Description
0	(Default) Twips
1	Pixels
2	Container
3	HiMetric

Remarks

The *x* and *y* parameters are applied using the top-left of the control as the origin (0,0). (Useful in the MouseMove event.)

This is only accessible via the **X** object when using a VBX.

TaskFromPos Method Applies To

DayView Control

See Also

- **Task** Object
- **TimeFromPos** Method
- **TaskHeight** Method
- **TaskLeft** Method
- **TaskTop** Method
- **TaskWidth** Method

TaskHeight, TaskLeft, TaskTop, TaskWidth Methods

See Also Applies To

Returns the height, left, top and width of the Task area of a time slot in pixels.

Syntax

object . **TaskHeight**(*time*)
object . **TaskHeight**(*timeindex*)
object . **TaskLeft**(*time*)
object . **TaskLeft**(*timeindex*)
object . **TaskTop**(*time*)
object . **TaskTop**(*timeindex*)
object . **TaskWidth**(*time*)
object . **TaskWidth**(*timeindex*)

The **TaskHeight**, **TaskLeft**, **TaskTop** and **TaskLeft** method syntaxes have these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
time	Required. A Variant expression that evaluates to the time for which you want to return the height of the Task area.
timeindex	Required. An integer expression specifying the index of the time for which you want to return the height of the Task area.

Remarks

This property is not available at design time and is read-only at run time.

See Also

- **TimeHeight** Method
- **TimeLeft** Method
- **TimeTop** Method
- **TimeWidth** Method

TaskSelected Property

Applies To

Returns or sets the number of the task that is selected.

Syntax

object . **TaskSelected** [= *number*]

The **TaskSelected** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression evaluates to the number of the task that is selected.

Remarks

The TaskSelected property is zero-based. Each task is numbered in the order in which it was added to the control. If a task is deleted, the remaining tasks are renumbered around the deleted one, leaving no number unused.

This property returns -1 if no task is selected.

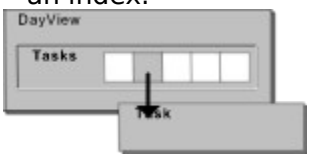
TaskSelected Property Applies To

DayView Control

Tasks Collection

See Also Example Applies To

The **Tasks** collection is a collection of **Task** objects. Each task is accessible by specifying an index.



Syntax

object . **Tasks**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

The **Tasks** collection contains the following properties and methods:

Properties

Count

Methods

Add	Remove	RemoveAll
Item		

The **Tasks** collection is accessible as an object.

The **Task** objects within the collection are accessible through the **Item** method and can be iterated over.

This is only accessible via the **X** object when using a VBX.

Tasks Collection Applies To

DayView Control

Tasks Collection Example

The following code adds 4 tasks to the **Tasks** collection:

```
SSDay1.X.Tasks.Add "10:00AM", "11:00AM", "Proj. 1701-D"
SSDay1.X.Tasks.Add "12:00PM", "1:00PM", "Lunch"
SSDay1.X.Tasks.Add "2:00PM", "3:00PM", "Printing"
SSDay1.X.Tasks.Add "3:30PM", "5:00PM", "Meeting", RGB(255,0,0)
```

The following code removes the first task from the **Tasks** collection:

```
SSDay1.X.Tasks.Remove 0
```

The following code removes all **Task** objects from the **Tasks** collection:

```
SSDay1.X.Tasks.RemoveAll
```

The following code returns the number of tasks in the **Tasks** collection:

```
NumTasks = SSDay1.X.Tasks.Count
```

The following code modifies the duration of a task in the **Tasks** collection:

```
SSDay1.X.Tasks.Duration = 60       '60 minutes (1 hour)
```

See Also

Task Object

TaskHeight, TaskLeft, TaskTop, TaskWidth Methods Apply To

DayView Control

Technical Specifications

All of the following information is subject to change. Please check the README.TXT file for any updates.

System Requirements
A list of requirements for using Calendar Widgets.

Included Files
A list of files included with Calendar Widgets and their locations.

Distribution Notes
A list of files you need to distribute with your applications.

Technical Support

CompuServe

You can obtain technical support on CompuServe by contacting the SYSOP in the Calendar Widgets section of the SHERIDAN forum. You can type GO SHERIDAN at any CompuServe prompt.

Internet

You can send electronic mail to technical support via the Internet. Messages should be addressed to support@shersoft.com

For up-to-the-minute Calendar Widgets information and the latest updates, as well as general information about Sheridan Software Systems Inc. and our products, visit our home page on the World Wide Web. The address is http://www.shersoft.com

BBS (Bulletin Board Service)

For free upgrades to Sheridan products, connect to the Sheridan BBS at (516) 753-5452. For best results, have your modem set to Hayes Compatible, at 8 bit, no parity, 1 stop bit. The BBS supports modems at 28.8 kps and under.

Fax

To fax questions or comments regarding any Sheridan product, dial (516) 753-3661.

Telephone Support

For free technical support for this or any other Sheridan product, contact Sheridan Software at (516) 753-0985. You can either speak to a live technical support representative or get answers using the Automated Fax Service.

Sheridan's support hours are 9AM to 5PM (EST), Monday through Friday.

The Calendar Widgets Controls

Using Calendar Widgets
Explains how to use the Calendar Widgets custom controls in your development environment. A brief overview of the significant differences between the VBX and OCX custom controls is also provided.

MonthView Control
Allows you to display dates in a monthly view.

YearView Control
Like the MonthView control, but allows you to display one year at a time.

DateCombo Control
Especially suited for designing data entry forms.

DayView Control
Displays a daily time schedule.

TimeBarClick Event

See Also Applies To

Occurs when the Time Selection Bar is clicked.

Syntax

Sub *object* _TimeBarClick (*Time* **As String**)

The TimeBarClick event has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
Time	A string expression that evaluates to the time of the task clicked in the Time Selection Bar.

Remarks

Use the Click event to capture a click in the Time Slot area.

Use the TimeBtnClick event to capture a click on a Time Button.

TimeBarClick Event Applies To

DayView Control

See Also

TaskSelected Property

TimeBtnClick Event

TimeBegin Property

See Also Applies To

Specifies the beginning time of the control.

Syntax

object . **TimeBegin** [= *text*]

The **TimeBegin** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the beginning time of the control.

Remarks

This property specifies the beginning time of the control. This time is reflected in the Time Selection Bar, if displayed. In addition to times, 'Midnight' and 'Noon' can be entered to select 12 AM and 12 PM respectively. This property cannot be set to a time later than **TimeEnd**.

TimeBegin Property Applies To

DayView Control

See Also

TimeEnd Property
TimeInterval Property
TimeSelectionBar Property

TimeBtnClick Event

See Also Applies To

Occurs when a Time Button is clicked.

Syntax

Sub *object* _TimeBtnClick (*Time* **As String**)

The TimeBtnClick event has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
Time	A string expression that evaluates to the time indicated on the Time Button that is clicked.

Remarks

Use the TimeBarClick event to capture a click on the Time Selection Bar.

Use the Click event to capture a click in the Time Slot area.

TimeBtnClick Event Applies To

DayView Control

See Also

TaskSelected Property

TimeBarClick Event

TimeEnd Property

See Also Applies To

Specifies the ending time of the control.

Syntax

object . **TimeEnd** [= *text*]

The **TimeEnd** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
text	A string expression that evaluates to the ending time if the control.

Remarks

This property specifies the ending time of the control. This time is reflected in the Time Selection Bar, if displayed. In addition to times, 'Midnight' and 'Noon' can be entered to select 12 AM and 12 PM respectively. This property cannot be set to a time earlier than **TimeBegin**.

TimeEnd Property Applies To

DayView Control

See Also

TimeBegin Property

TimeInterval Property

TimeFromIndex Method

See Also Example Applies To

Returns the time of the specified time index.

Syntax

object . **TimeFromIndex**(*number*)

The **TimeFromIndex** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the time index.

Remarks

This method returns a string that represents the time of the time index specified by *number* . The method will not return a value if the index passed into it is invalid.

TimeFromIndex Method Applies To

DayView Control

TimeFromIndex Method Example

The following code returns the time of the topmost time index shown on the **DayView** control:

```
Dim MyTopTime As String
MyTopTime = SSDay1.TimeFromIndex(SSDay1.TopIndex)
```

See Also

IndexFromTime Method

TopIndex Property

TimeFromPos Method

See Also Applies To

Returns the time that corresponds to the position indicated by coordinates.

Syntax

object . **TimeFromPos**(*x* , *y* , [*scale*])

The **TimeFromPos** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
x	Required. An integer expression that evaluates to the x-coordinate.
y	Required. An integer expression that evaluates to the y-coordinate.
scale	Optional. A Variant expression that evaluates to a value in the Settings list.

Settings

The settings for *scale* are:

Setting	Description
0	(Default) Twips
1	Pixels
2	Container
3	HiMetric

Remarks

The *x* and *y* parameters are applied using the top-left of the control as the origin (0,0). (Useful in MouseMove event.)

The method returns the time whether the mouse pointer is over a Time Button or Time Slot area.

This is only accessible via the **X** object when using a VBX.

TimeFromPos Method Applies To

DayView Control

See Also

- **Task** Object
- **TaskFromPos** Method
- **TimeHeight** Method
- **TimeLeft** Method
- **TimeTop** Method
- **TimeWidth** Method

TimeHeight, TimeLeft, TimeTop, TimeWidth Methods

See Also Applies To

Returns the height, left, top and width of the Time Button of a specified time slot in pixels.

Syntax

object . **TimeHeight**(*time*)

object . **TimeHeight**(*timeindex*)

object . **TimeLeft**(*time*)

object . **TimeLeft**(*timeindex*)

object . **TimeTop**(*time*)

object . **TimeTop**(*timeindex*)

object . **TimeWidth**(*time*)

object . **TimeWidth**(*timeindex*)

The **TimeHeight**, **TimeLeft**, **TimeTop** and **TimeWidth** method syntaxes have these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
time	Required. A Variant expression that evaluates to the time for which you want to return the height of the Time Button.
timeindex	Required. An integer expression specifying the index of the time for which you want to return the height of the Time Button.

Remarks

An error is generated if the *time* or *timeindex* is invalid (falls outside the range specified by the **TimeBegin** and **TimeEnd** properties).

The *timeindex* is zero-based and starts with the first visible time slot. If the time slot specified by *time* or *timeindex* are not visible, a value of -1 is returned.

This is only accessible via the **X** object when using a VBX.

See Also

- **Task** Object
- **TaskFromPos** Method
- **TaskHeight** Method
- **TaskLeft** Method
- **TaskTop** Method
- **TaskWidth** Method
- **TimeBegin** Property
- **TimeEnd** Property
- **TimeFromPos** Method

TimeInterval Property

See Also Applies To

Specifies the intervals between time slots.

Syntax

object . **TimeInterval** [= *number*]

The **TimeInterval** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer specifying the intervals between time slots.

Settings

The settings for *number* are:

Setting	Description
0	5 minutes
1	10 minutes
2	15 minutes
3	(Default) 30 minutes
4	60 minutes

Remarks

This property specifies the time intervals between time slots. If a task's beginning time is not shown, then the task is shown at the closest prior interval.

See Also

BeginTime Property

EndTime Property

TimeInteval Property Applies To

DayView Control

TimeSelectionBar Property

See Also Applies To

Determines if the Time Selection Bar will be displayed.

Syntax

object . **TimeSelectionBar** [= *boolean*]

The **TimeSelectionBar** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if the Time Selection Bar will be displayed.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) TimeSelectionBar is displayed.
False	TimeSelectionBar is not displayed.

TimeSelectionBar Property Applies To

DayView Control

See Also

TimeSelectionBarFont Property

TimeBegin Property

TimeSelectionBarFont Property

See Also Example Applies To

Returns a **Font** object.

Syntax

object . **TimeSelectionBarFont**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

Use the **TimeSelectionBarFont** property to identify a specific **Font** object to use for displaying the Time Selection Bar scale.

This property is not available at design time in Visual Basic 3.0.

TimeSelectionBarFont Property Applies To

DayView Control

TimeSelectionBarFont Property Example

The following code changes the **Bold** property setting of a **Font** object identified by the **TimeSelectionBarFont** property of a **DayView** control:

```
SSDay1.TimeSelectionBarFont.Bold = True
```

See Also

- **Font** Object
- **TimeSelectionBarFontBold** Property
- **TimeSelectionBarFontItalic** Property
- **TimeSelectionBarFontName** Property
- **TimeSelectionBarFontSize** Property
- **TimeSelectionBarFontStrikethru** Property
- **TimeSelectionBarFontUnderline** Property

TimeSelectionBarFontBold, TimeSelectionBarFontItalic, TimeSelectionBarFontStrikethru, TimeSelectionBarFontUnderline Properties

See Also Applies To

Return or set font styles in the following formats for the Time Selection Bar: **Bold**, *Italic*, ~~Strikethru~~, and Underline.

Syntax

object . **TimeSelectionBarFontBold** [= *boolean*]

object . **TimeSelectionBarFontItalic** [= *boolean*]

object . **TimeSelectionBarFontStrikethru** [= *boolean*]

object . **TimeSelectionBarFontUnderline** [= *boolean*]

The **TimeSelectionBarFontBold**, **TimeSelectionBarFontItalic**, **TimeSelectionBarFontStrikethru** and **TimeSelectionBarFontUnderline** property syntaxes have these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying the font style as described in Settings.

Settings

The settings for *boolean* are:

Setting	Description
True	Turns on the formatting in that style.
False	Turns off the formatting in that style.

Remarks

Use these font properties to format the text of the Time Selection Bar, either at design time using the Properties window or at run time using code.

When using the VBX version of the control, the **TimeSelectionBarFontBold**, **TimeSelectionBarFontItalic**, **TimeSelectionBarFontStrikethru**, and **TimeSelectionBarFontUnderline** properties are available at design time. These properties are supported in the OCX version of the control for compatibility.

TimeSelectionBarFontName Property

See Also Applies To

Returns or sets the font used to display text for the Time Selection Bar.

Syntax

object . **TimeSelectionBarFontName** [= *font*]

The **TimeSelectionBarFontName** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
font	A string expression specifying the font name to use.

Remarks

Use this font property to format the text of the Time Selection Bar, either at design time using the Properties window or at run time using code.

When using the VBX version of the control, the **TimeSelectionBarFontName** property is available at design time. This property is supported in the OCX version of the control for compatibility.

TimeSelectionBarFontName Property Applies To

DayView Control

See Also

Fonts
TimeSelectionBarFontBold Property
TimeSelectionBarFontItalic Property
TimeSelectionBarFontSize Property
TimeSelectionBarFontStrikethru Property
TimeSelectionBarFontUnderline Property
Font Object

TimeSelectionBarFontSize Property

See Also Applies To

Returns or sets the size of the font to be used for text of the Time Selection Bar.

Syntax

object . **TimeSelectionBarFontSize** [= *points*]

The **TimeSelectionBarFontSize** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
points	A numeric expression specifying the font size to use, in points.

Remarks

Use this property to format the text of the Time Selection Bar in the font size you want.

When using the VBX version of the control, the **TimeSelectionBarFontSize** property is available at design time. This property is supported in the OCX version of the control for compatibility.

TimeSelectionBarFontSize Property Applies To

DayView Control

See Also

Fonts
TimeSelectionBarFontBold Property
TimeSelectionBarFontItalic Property
TimeSelectionBarFontName Property
TimeSelectionBarFontStrikethru Property
TimeSelectionBarFontUnderline Property
Font Object

TimeSelectionBarFontBold, TimeSelectionBarFontItalic, TimeSelectionBarFontStrikethru, TimeSelectionBarFontUnderline Properties Apply To

DayView Control

See Also

Fonts
TimeSelectionBarFontName Property
TimeSelectionBarFontSize Property
Font Object

TimeSlotCount Property

See Also Applies To

Returns the number of items in the list portion of a control.

Syntax

object . **TimeSlotCount**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Remarks

If no time slot is selected, the **TimeSlotIndex** property value is -1. The first time slot in the list is **TimeSlotIndex** = 0, and **TimeSlotCount** is always one more than the largest **TimeSlotIndex** value.

This property is not available at design time and is read-only at run time.

TimeSlotCount Property Applies To

DayView Control

See Also

TimeSlotIndex Property

TopIndex Property

TimeSlotIndex Property

See Also Applies To

Returns or sets the index of the currently selected time slot in the control.

Syntax

object . **TimeSlotIndex** [= *index*]

The **TimeSlotIndex** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression that evaluates to the index of the currently selected time slot.

Remarks

If no time slot is selected, the **TimeSlotIndex** property value is -1. The first time slot in the list is **TimeSlotIndex** = 0, and **TimeSlotCount** is always one more than the largest **TimeSlotIndex** value.

This property is not available at design time.

TimeSlotIndex Property Applies To

DayView Control

See Also

TimeSlotCount Property

TopIndex Property

TimeHeight, TimeWidth Methods Apply To

DayView Control

TopIndex Property

See Also Applies To

Returns or sets a value specifying which time slot in a DayView control is displayed in the topmost position.

Syntax

object . **TopIndex** [= *number*]

The **TopIndex** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression that evaluates to the number of the time slot that is displayed in the topmost position. The default is 0, or the first time slot in the list.

Remarks

Use this property to scroll through a **DayView** control.

This property is not available at design time.

TopIndex Property Applies To

DayView Control

See Also

TimeSlotCount Property
TimeSlotIndex Property
TopIndexChange Event

TopIndexChange Event

See Also Applies To

Occurs when the index of the top-most time slot changes.

Syntax

Sub *object* _TopIndexChange (*TopIndex* **As Short**)

The TopIndexChange event has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
TopIndex	A Short integer expression that evaluates to the index of the top-most time slot.

Remarks

This event will occur when the control is scrolled.

TopIndexChange Event Applies To

DayView Control

See Also

TopIndex Property

Trappable Errors

The following is a list of trappable errors that could occur at run time when using the Calendar Widgets custom controls. The constant declarations for these values can be found in the SSCALWDG.BAS file that comes with Calendar Widgets.

Code	Description
30904	SS_ERR_DATASOURCEALREADYSET
	Can't set DataSourceHwnd property if DataSource property has been set
	You tried setting the DataSourceHwnd property while the DataSource property was already set.
32001	SS_ERR_DAY_INACTIVE
	The date specified is invisible or disabled and therefore cannot be accessed
	You tried setting the date to one that has its Enabled property or Visible property set to False.
32004	SS_ERR_NO_TODAYSDATE
	Note: Today's date does not fall within the minimum and maximum dates. Change range or set ShowTodaysDate to False
	Today's date is outside the range allowed by MinDate and MaxDate.
32005	SS_ERR_NO_DEFAULTDATE
	Note: The default date does not fall within the minimum and maximum dates. Change range or default date
	You tried setting the DefaultDate property to one that is outside the range allowed by MinDate and MaxDate.
32006	SS_ERR_NO_SETDATE
	Note: The set date does not fall within the minimum and maximum dates
	You tried setting the date to one that is outside the range allowed by MinDate and MaxDate.
32007	SS_ERR_MULTIBOUND
	MultiSelect mode is not allowed for a bound control
	You tried setting the SelectionType property to MultiSelect while the control was bound.
32008	SS_ERR_MULTIAUTO
	MultiSelect mode is not allowed when AutoSelect is True
	You tried setting the SelectionType property to MultiSelect while the AutoSelect property was set to True.
32009	SS_ERR_AUTOMULTI
	AutoSelect must be False in MultiSelect mode
	You tried setting the AutoSelect property to True while the SelectionType property was set to MultiSelect.
32010	SS_ERR_BEVEL
	BevelWidth must be from 0 to 10

You tried setting the BevelWidth property outside the allowed range.

32011 SS_ERR_CAPTIONBEVEL

CaptionBevelWidth must be from 0 to 10

You tried setting the CaptionBevelWidth property outside the allowed range.

32012 SS_ERR_DROPDOWNPIC

PictureDropDown must be a bitmap

You tried setting the PictureDropDown property to something other than a bitmap

32013 SS_ERR_DROPDOWNBEVEL

DropDownBevelWidth must be from 0 to 10

You tried setting the DropDownBevelWidth property outside the allowed range.

32014 SS_ERR_INVALIDDATE

Specified Date is Invalid

You tried referencing an invalid date.

32015 SS_ERR_BADFORMAT

Invalid Format

You tried setting the Format property to a format that is not supported.

32016 SS_ERR_BADPARAM

Invalid Parameter

You tried sending an invalid parameter to a method.

32017 SS_ERR_NULLDEFAULT

The default date must have a value since AllowNullDate is False

You tried setting the DefaultDate property to nothing while the AllowNullDate property was set to False.

32018 SS_ERR_BADEDITDATE

Date currently in edit box is invalid

You tried setting focus to another control while the date in the edit portion of the DateCombo was invalid.

32019 SS_ERR_LARGECHANGE

LargeChange must be from 1 to 100

You tried setting the LargeChange property outside the allowed range.

32020 SS_ERR_BADHOST

Host environment does not support date formatting.

You tried setting the Format property in a development environment that does not support date formatting.

32096 SS_ERR_OVERLAPPING_TASK_LIMIT

A Maximum of 10 overlapping tasks is allowed

You tried placing a task on a time slot that already contains 10 tasks.

32097 SS_ERR_BEGINNING_TIME

Beginning Time cannot be set to occur after the Ending Time

You tried setting the Beginning Time past the Ending Time.

32098 SS_ERR_ENDING_TIME

	Ending Time cannot be set to occur before the Beginning Time
	You tried setting the Ending Time before the Beginning Time.
32099	SS_ERR_EQUAL_TIME
	Beginning Time and Ending Time cannot be equal except for when they both equal 12:00 AM
	You tried setting the Beginning Time and Ending time to the same time.
32100	SS_ERR_INVALID_TIME
	Invalid Time
	You tried setting a task with an invalid BeginTime or EndTime.
32101	SS_ERR_TASK_NOT_SELECTED
	No Task is currently selected
	You tried setting EditVisible property to True while no task was selected.
32103	SS_ERR_TASKS_LIMIT
	The 255 task limit has been reached
	You tried adding a 256th task.
32104	SS_ERR_INVALID_TIME_INDEX
	The time index passed in is either before the Beginning Time or after the Ending Time of the day
	You tried adding a task or setting its Beginning Time or Ending Time past the range defined by the TimeBegin and TimeEnd properties.
32520	SS_MOUSEPTR_ERR_ICONSONLY
	Property supports icons only
	You tried setting the MouseIcon property to something other than an icon.

Underline Property

See Also Applies To

Returns or sets the font style of the **Font** object to either underlined or nonunderlined.

Syntax

object . **Underline** [= *boolean*]

The **Underline** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying the font style as described in Settings.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Turns on underline formatting.
False	Turns off underline formatting.

Remarks

The **Font** object is not directly available at design time. Instead you set the **Underline** property through a control's **Font** property.

At run time, however, you can set **Underline** directly by specifying its setting for the **Font** object.

In Visual Basic 3.0 you set the **Underline** property by selecting a control's **FontUnderline**, **CaptionFontUnderline**, **DayFontUnderline**, **DropDownFontUnderline** or **TimeSelectionBarFontUnderline** property in the Visual Basic Properties window.

Underline Property Applies To

Font Object

See Also

Font Object

Font Property

Using Calendar Widgets

Using the Keyboard Interface

The following sections describe the keyboard interface for each of the Calendar Widgets controls. For the DateCombo, MonthView and YearView controls, assume that the **AutoSelect** property is set to **False** unless otherwise indicated.

If you experience problems when passing Variant parameters into methods, it may be necessary to use the **CVar** function on the parameter. For example:

```
Dim lButtonLeft As Long
lButtonLeft = TimeLeft(CVar(MyVariantIndex))
```

This will guarantee that the parameter being passed into the method is a Variant. See the Visual Basic documentation or on-line help file for more information on the **CVar** function.

Visible Property

See Also Example Applies To

Determines if a control or object is visible.

Syntax

object . **Visible** [= *boolean*]

The **Visible** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
boolean	A Boolean expression specifying if a control or object is visible.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Control or object is visible.
False	Control or object is not shown.

Remarks

When the **Visible** property is set to **False**, the control or object will not be shown. You cannot click on or use the keyboard controls to access it.

However, you can still set properties and apply methods to controls and objects that are not visible.

Visible Property Applies To

DateCombo Control
DayofWeek Object
DayView Control
MonthView Control
YearView Control

Visible Property Example

If the **Visible** property of **DayofWeek** objects is set to **False** for 'Sunday' and 'Saturday' on a **MonthView** control, the following would be the result:

```
SSMonth1.X.DayofWeek(1).Visible = False
SSMonth1.X.DayofWeek(7).Visible = False
```

Notice that 'Sunday' and 'Saturday' do not appear on the control.

Although the two **DayofWeek** objects are no longer visible, they can still be accessed through code. You can still set their properties even though they are invisible.

See Also

Enabled Property

VisibleMonth Property

See Also Example Applies To

Returns a **Month** object of one of the visible months on the display.

Syntax

object . **VisibleMonth**(*index*)

The **VisibleMonth** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
index	An integer expression that evaluates to an index to a particular month on the control.

Remarks

For the **MonthView** control, *index* is a number from 0 to 2.

For the **YearView** control, *index* is a number from 0 to 11.

This is only accessible via the **X** object when using a VBX.

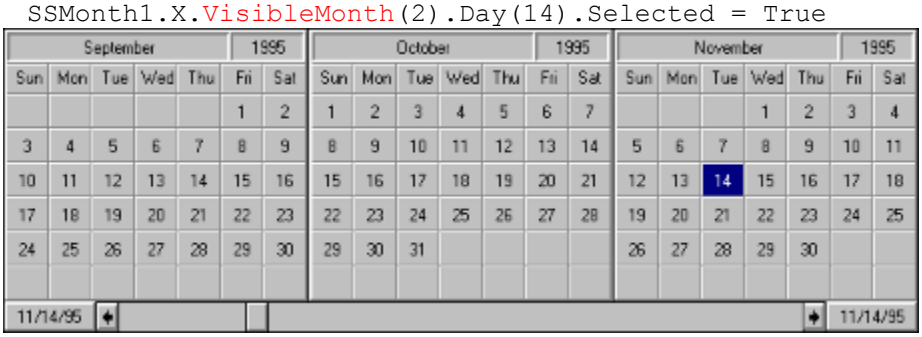
VisibleMonth Property Applies To

DateCombo Control
MonthView Control
YearView Control

VisibleMonth Property Example

The follow code selects the 14th day of the right-most month on a 3-month **MonthView** control.

```
SSMonth1.X.VisibleMonth(2).Day(14).Selected = True
```



See Also

- **Month** Object
- X Object

Visual Basic 3.0

In Visual Basic, custom controls are installed on a project basis. Once you include a custom control in a project, the control will appear in the Visual Basic Toolbox whenever you subsequently open the project. For more information see the 'Custom Control Reference' section of the *Visual Basic Professional Features Book 1*.

To Include Calendar Widgets In Your Project:

1. Open the project in which you want to place Calendar Widgets.
2. Select 'Add File' from the 'File' menu.
3. Either type the name of, or select, the file that contains the control you want to use.

To Use	Select
 MonthView	SSCALA.VBX
 YearView	SSCALA.VBX
 DateCombo	SSCALA.VBX
 DayView	SSCALB.VBX

4. Repeat steps 2 and 3 of you wish to include both VBX files.

The icons for the custom controls you selected will then appear in the Visual Basic Toolbox. Once loaded in the Toolbox, you can use them just like any standard Visual Basic control. Since there are no separate design time and run time versions of the controls, the same files you use in development can be shipped with your application.

WeekNumber Method

See Also Example Applies To

Returns the week number.

Syntax

object . **WeekNumber**(*date* , [*FirstWeek*], [*FirstMonth*])

The **WeekNumber** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
date	Optional. The date to use when returning the number of the week of the year. The value can be any of the following: § A date (in date format). § A string expression that evaluates to a valid date. § A **Day** object. If no parameter is specified, the day with focus will be used to determine the week number.
FirstWeek	Optional. The first week to use.
FirstMonth	Optional. The first month to use. (For **YearView** only)

Settings

The settings for *FirstWeek* are:

Setting	Description
0	First day of year
1	(Default) First four day week from Sunday
2	First four day week from **StartOfWeek**
3	First full week from Sunday
4	First full week from **StartOfWeek**

The settings for *FirstMonth* are:

Setting	Description
0	January
1	(Default) **StartMonth**

Remarks

This is only accessible via the **X** object when using a VBX.

WeekNumber Method Applies To

DateCombo Control
MonthView Control
YearView Control

WeekNumber Method Example

This code displays the week number of the week containing the currently selected day. The week number is based on the first month displayed by the control, as specified in the **StartMonth** property:

```
Text1.Text = "Week Number " & SSYear1.WeekNumber
```

This code displays the week number of the first selected day. The week number is computed from the beginning of the year, regardless of which month appears first in the YearView control:

```
Text1.Text = "Week Number " &
SSYear1.WeekNumber(SSYear1.SelectedDays(0).Date, 0, 0)
```

Assuming that an employee gets paid every other Thursday, the following code determines if the selected Thursday is a pay-day:

```
Dim PayMe As Integer
PayMe = False
If (SSMonth1.X.WeekNumber Mod 2) = 1 Then PayMe = True
```

See Also

FocusDate Property

Weight Property

See Also Applies To

Returns or sets the weight of the characters that make up a **Font** object.

Syntax

object . **Weight** [= *number*]

The **Weight** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
number	An integer expression specifying the weight of the font.

Remarks

The **Font** object is not directly available at design time. Instead you set the **Size** property through a control's **Font** property.

At run time, however, you can set **Weight** directly by specifying its setting for the **Font** object.

Regular and Italic fonts usually have a **Weight** value of 400, and the Bold and Bold Italic fonts usually have a **Weight** value of 700.

This property is not available in Visual Basic 3.0.

Weight Property Applies To

Font Object

See Also

Font Object

Font Property

What are Property Pages?

Sheridan Software custom controls now support a feature known as Property Pages. Property pages provide an interface through which you can view and modify the properties of the controls. The purpose of Property Pages is twofold. First, Property Pages allow you to set properties at design time that would not otherwise be available - the so-called "run time" properties. Second, Property Pages allow you to modify your control in a host environment that does not provide a property sheet.

Note All of the following information is subject to change. Please check the README.TXT file for any updates.

When should I use OCX controls?

OCX controls come in two varieties: 16-bit and 32-bit. 16-bit controls offer compatibility with Windows and Windows for Workgroups 3.1 and 3.11. 32-bit controls work with systems running in 32-bit operating systems such as Windows NT. In general, you should use the most advanced version of the control that is available and is supported by your host environment.

If you are using a 32-bit programming system to develop an application that will run exclusively on a 32-bit platform, use the 32-bit OCX. If you are developing an application that must run on a mixed platform, you can use a 16-bit OCX or VBX, although you will obtain better performance if you develop separate 16-bit and 32-bit versions of your program, using the appropriate OCX controls. If you are developing exclusively for a 16-bit platform, use the 16-bit OCX or VBX.

Note Visual Basic 3.0 only supports the VBX version of the controls.

WhereIs Method

Applies To

Returns a value indicating the position of the mouse pointer, given its coordinates.

Syntax

[*number* =]*object* . **WhereIs**(*x* , *y* , [*scale*])

The **WhereIs** method has these parts:

Part	Description
number	An integer expression that evaluates to the position of the mouse pointer.
object	An object expression that evaluates to an object or control in the **Applies To** list.
x	Required. An integer expression that evaluates to the x-coordinate.
y	Required. An integer expression that evaluates to the y-coordinate.
scale	Optional. A Variant expression that evaluates to a value in the Settings list.

Settings

The settings for *number* are:

Setting	DayView	MonthView	YearView
0	Nothing	Nothing	Nothing
1	Caption	Year	Begin Year
2	TimeSelectionBar	Month	End Year
3	TimeButton	DayofWeek	Month
4	TimeSlot	Day	DayofWeek
5	Selected Date Button		Day
6	Today’s Date Button		Selected Date Button
7		Today’s Date Button	
8		Caption	

The settings for *scale* are:

Setting	Description
0	(Default) Twips
1	Pixels
2	Container
3	HiMetric

Remarks

The *x* and *y* parameters are applied using the top-left of the control as the origin (0,0).

This is only accessible via the **X** object when using a VBX.

WhereIs Method Applies To

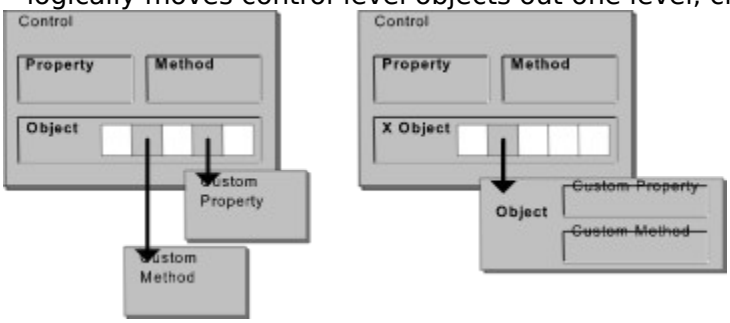
DayView Control
MonthView Control
YearView Control

X Object

A limitation of using VBX controls is that they do not support Variant properties and custom methods that are applied to *control - level objects* . A control-level object is an object that can be directly accessed from the control. For example, the **StyleSets** collection is a control-level object. Therefore, the following code would not work with a VBX:

```
SSMonth1.StyleSets.Add "vacation"
```

For this reason, each of the Calendar Widgets controls provides an **X** object. The **X** object logically moves control-level objects out one level, circumventing the limitation.



Note This limitation only applies to control-level objects. Sub-objects do not have this limitation (see 'Sub-objects and Collections' under 'Object Concepts' earlier in this appendix).

In order to invoke the **Add** method to the **StyleSets** collection using a VBX, it must be done through the **X** object as follows:

```
SSMonth1.X.StyleSets.Add "vacation"
```

In addition to the **StyleSets** collection, the **X** object must also be used when accessing properties or methods in the **SelectedDays** and **Tasks** collections.

Passing a Form/Control to a Visual Basic (BAS) Module

Due to a limitation in Visual Basic, a control's **X** object cannot be directly accessed when passing the control, or form containing the control, to a Visual Basic module. For example, if the control containing the **X** object is located on Form1 and the form was passed to a procedure called AddTask( ) in a BAS module, the following code will generate an error:

```
Sub AddTask(frmForm as Form)
     frmForm.SSDay1.X.Tasks.Add "1 AM", "2 AM", "Meeting"
End Sub
```

In order to achieve the same effect the **X** object must first be set to a variable of type Object as follows:

```
Sub AddTask(frmForm as Form)
     Dim xobj As Object
     Set xobj = frmForm.SSDay1.X
     xobj.Tasks.Add "1am", "2am", "Test"
End Sub
```

Passing Values to Methods Using the X Object

Due to the way Visual Basic handles VBX controls, you cannot pass a value returned from an **X** object to a method that is invoked through an **X** object. For example, the following code will generate an error in VBOA300.DLL when exiting your application:

```
Dim iMyIndex As Integer
iMyIndex = SSDay1.X.IndexFromTime(SSDay1.X.Tasks(0).BeginTime)
```

In order to achieve the same effect the value returned from the **X** object must first be

placed in a variable as follows:

```
Dim sTempTime As String
Dim iMyIndex As Integer
sTempTime = SSDay1.X.Tasks(0).BeginTime
iMyIndex = SSDay1.X.IndexFromTime(sTempTime)
```

Compatibility with OCX Controls

Any code written for use with the VBX version of the controls is fully compatible with the OCX version. The **X** object as used in the VBX version is supported in the OCX version also. This measure was taken so that you can substitute OCX controls for VBX controls without having to make changes to your code. Please see the section on converting from VBX to OCX controls for additional information on using your Visual Basic 3.0 projects in Visual Basic 4.0.

X Object (DateCombo)

See Also Applies To

An **X** object exposes properties, methods, objects and collections that are otherwise unavailable using the VBX version of the control.

Remarks

The following properties, methods, objects and collections can be accessed using the **X** object with the **DateCombo** control:

Properties

TagVariant	VisibleMonth

Methods

IsDateValid	WeekNumber

Objects

Day	DayofWeek	Month

Collections

StyleSets

X Object (DateCombo) Applies To

DateCombo Control

See Also

X Object

X Object (DayView)

See Also Applies To

An **X** object exposes properties, methods, objects and collections that are otherwise unavailable using the VBX version of the control.

Remarks

The following properties and collections can be accessed using the **X** object with the **DayView** control:

Properties

TagVariant

Methods

TaskFromPos	TaskWidth	TimeTop
TaskHeight	TimeFromPos	TimeWidth
TaskLeft	TimeHeight	WhereIs
TaskTop	TimeLeft	

Objects

Task

Collections

Tasks

X Object (DayView) Applies To

DayView Control

X Object (MonthView)

See Also Applies To

An **X** object exposes properties, methods, objects and collections that are otherwise unavailable using the VBX version of the control.

Remarks

The following properties, methods, objects and collections can be accessed using the **X** object with the **MonthView** control:

Properties

TagVariant	VisibleMonth

Methods

DayFromPos	DayofWeekFromPos	MonthFromPos
DayHeight	DayTop	WeekNumber
DayLeft	DayWidth	WhereIs

Objects

Day	DayofWeek	Month

Collections

SelectedDays	StyleSets

X Object (MonthView) Applies To

MonthView Control

X Object (YearView)

See Also Applies To

An **X** object exposes properties, methods, objects and collections that are otherwise unavailable using the VBX version of the control.

Remarks

The following properties, methods, objects and collections can be accessed using the **X** object with the **YearView** control:

Properties

TagVariant	VisibleMonth

Methods

DayFromPos	DayofWeekFromPos	MonthFromPos
DayHeight	DayTop	WeekNumber
DayLeft	DayWidth	WhereIs

Objects

Day	DayofWeek	Month

Collections

SelectedDays	StyleSets

X Object (YearView) Applies To

YearView Control

Year Property

See Also Applies To

Returns the year number for the **Month** object.

Syntax

object . **Year**

The *object* placeholder represents an object expression that evaluates to an object or control in the **Applies To** list.

Year Property Applies To

Month Object

See Also

Month Object

Object Concepts

YearClick Event

Applies To

Occurs when you click the mouse in the Year Caption area of the month display.

Syntax

Sub *object* _YearClick ([*Index* **As Integer**] *YearNum* **As Integer**, *MonthNum* **As Integer**)

The YearClick event has these parts:

Part	Description
object	An object expression that evaluates to an object or control in the **Applies To** list.
Index	Uniquely identifies the control if it is in a control array.
YearNum	An integer expression that evaluates to the year number.
MonthNum	An integer expression that evaluates to the number of the current month (1-12).

Remarks

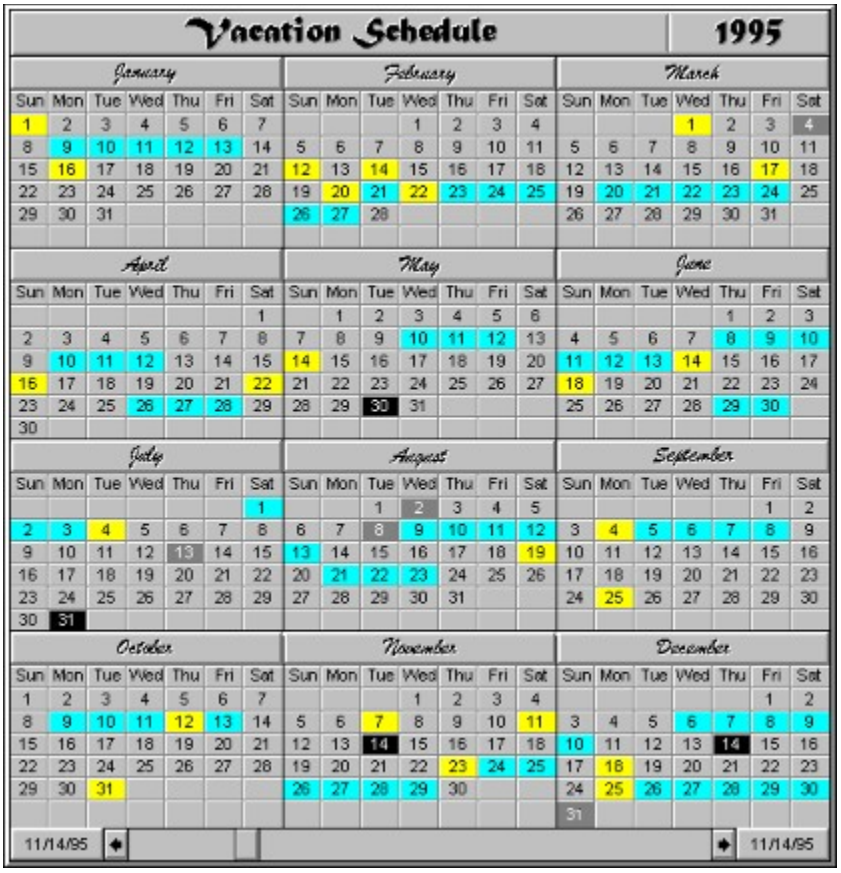
The YearClick event is fired when you click the mouse in the Year Caption area of the month display.

YearClick Event Applies To

DateCombo Control
MonthView Control

The YearView Control

The YearView control has all the features of the MonthView control with the added ability to display an entire year at a time. This control can be used as the foundation of an application with yearly planning capabilities. Like the MonthView control, the YearView control is also data aware.



File Name SSCALA.VBX, SSCALA16.OCX, SSCALA32.OCX

ObjectType SSYear

Keyboard Interface

As a Foundation for Yearly Planners

Marking Dates

Multiple Selection

Customizing

Back

YearView Control Collections

Collections marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

SelectedDays Collection õ

StyleSets Collection õ

Back

YearView Control Events

Events marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

CaptionClick Event
Click Event
DblClick Event
DragDrop Event
DragOver Event
FocusChange Event
GotFocus Event
InitYear Event
KeyDown Event
KeyPress Event
KeyUp Event
LostFocus Event
MonthClick Event
MouseDown Event
MouseMove Event
MouseUp Event
SelChange Event
SelChanged Event

Back

YearView Control Methods

Methods marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

DayFromPos Method õ
DayHeight Method õ
DayLeft Method õ
DayofWeekFromPos Method õ
DayTop Method õ
DayWidth Method õ
MonthFromPos Method õ
Refresh Method
WeekNumber Method õ
WhereIs Method õ

Back

YearView Control Objects

Objects marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

Day Object õ
DayofWeek Object õ
Font Object
Month Object õ
X Object

Back

YearView Control Properties

Properties marked with a Õ are only accessible via the **X** object when using the VBX version of the control.

Name Property
Orientation Property
Parent Property
ScrollBar Property
ScrollBarTracking Property
SelectionType Property
ShowCentury Property
ShowSelectedDate Property
ShowTodaysDate Property
StartMonth Property
StartOfWeek Property
TabIndex Property
TabStop Property
Tag Property
TagVariant Property õ
Top Property
Visible Property
VisibleMonth Property õ
Width Property

As a Foundation for Yearly Planners

The YearView control can act as a foundation for applications that require yearly planning capabilities. It can display an entire year's worth of date information at a time.

You can depict attributes of days on the YearView control by using captions, colors and pictures. To accomplish this, you can create StyleSets (templates) to store caption, color, picture and other information (see Exercise 2: Applying a StyleSet of the YearView control in Guided Tours). Once created, these StyleSets can be applied to individual days on the YearView control.

The YearView control can also be used as a control center that exposes greater information detail when you click on it. For instance, when you click on a specific day on the YearView control, a box with a DayView control can be made to pop up, revealing hourly information for that day.

Customizing

There are many properties in the YearView control that let you customize the display to your liking. In addition to bevels, alignment, and color, the YearView control contains special properties that help shape the control to look and feel the way you want.

As with the **DateCombo** and **MonthView** controls, the **StartOfWeek** property can be set so that the display of weeks starts with something other than Sunday. The **Enabled** property can be used to enable and disable specific days or days of the week. The **Visible** property can be used to hide certain days of the week. See Customizing for the MonthView control for more information on how to accomplish these effects.

YearView Control

Press	To	Comments
LEFT ARROW	Focus on Previous Day	If previous day is visible, only the focus changes; otherwise, the control scrolls.
RIGHT ARROW	Focus on Next Day	If next day is visible, only the focus changes; otherwise, the control scrolls.
UP ARROW	Focus on Same Day of Previous Week	If previous week is visible, only the focus changes; otherwise, the control scrolls.
DOWN ARROW	Focus on Same Day of Next Week	If next week is visible, only the focus changes; otherwise, the control scrolls.
CTRL+LEFT ARROW	Focus on Same Day of Previous Month	If previous month is visible, only the focus changes; otherwise, the control scrolls.
CTRL+RIGHT ARROW	Focus on Same Day of Next Month	If next month is visible, only the focus changes; otherwise, the control scrolls.
PAGE UP	Focus on Same Day of Previous Year	Displays the previous year.
PAGE DOWN	Focus on Same Day of Next Year	Displays the next year.
CTRL+PAGE UP	Focus on Same Day of Past Year Based on LargeChange	Sets the focus to a specified number of years into the past, based on the setting of the LargeChange property (default = 10 years).
CTRL+PAGE DOWN	Focus on Same Day of Future Year Based on LargeChange	Sets the focus to a specified number of years into the future, based on the setting of the LargeChange property (default = 10 years).
CTRL+S	Focus on Selected Day	Sets the focus on the selected date.
CTRL+T	Focus on Today	Sets the focus on today’s date.
HOME	Focus on First Day of Month	Sets the focus to the first day of the current month.
END	Select Last Day of Month	Sets the focus to the last day of the current month.
CTRL+HOME	Focus on First Day	Sets the focus to the first visible day on the control.
CTRL+END	Focus on Last Day	Sets the focus to the last visible day on the control.
ENTER / SPACEBAR	Select Day with Focus	Selects the day with focus.
ESCAPE	Reset Date	If control is bound and AutoRestore is True, resets

selected date to the date in
the database.

Marking Dates

Each day on the control can be marked so that it carries special meaning. As explained above, templates called StyleSets can be created to store attributes of a type of day. Using properties pertaining to color, font and picture, you can give unique attributes to each StyleSet. Once a Styleset is added to the **StyleSets** collection, it can be applied to one or more days.

Multiple Selection

The **YearView** control lets you select more than one day at a time. See Multiple Selection for the **MonthView** control for more information on how to accomplish this.

See Also

MonthView Control
DateCombo Control
DayView Control

Zodiac

version 2.01.0005

version 2.03.0142

version 4.1.6038

version 4.02.6256

version 4.00.5157

version 2.0

version 1.99.1605

version 2.50.1606

version 03.00.0033

version 4.20.6201

version 4.10.5270

version 2.53

version 2.03.0142

version 2.03

version 2.03

version 2.02.0121

version 2.20

version 5.0

version 2.0

version 1.00.0003

version 1.00.0003

version 1.00.0004

version 1.00.0004

version 1.0

version 2.20

version 2.03.0140

version 3.0

version 2.1

version 3.00.0030

version 03.00.0025

version 03.00.0038