

Value Property (Calendar Control) (Microsoft Access)

You can bind the Calendar control to a field in the table or query underlying the Microsoft Access form that contains the Calendar control. This is useful if you want the Calendar control to display a date based on data in your application.

To bind the Calendar control to a field, set the Calendar control's **ControlSource** property to the name of the field. The field must be a Date/Time field.

Once you've bound the Calendar control to a field, the Calendar control will display the value of that field for each record on the form. Additionally, setting the **Value** property of the Calendar control sets the value of the field in the underlying table.

DayFont, GridFont Properties Example

The following example sets properties of a Calendar control on a form when the form loads. To try this example, add a calendar control named ActiveXCtl0 to a form, and paste the following code into the form's module:

```
Private Sub Form_Load()  
    Dim ctl As Control  
  
    Set ctl = Me!ActiveXCtl0  
    ' Set font properties for days of week.  
    With ctl.DayFont  
        .Name = "Arial"  
        .Size = 10  
        .Bold = True  
        .Italic = True  
    End With  
  
    ' Set font properties for days of month.  
    With ctl.GridFont  
        .Name = "Times New Roman"  
        .Size = 12  
        .Bold = False  
        .Italic = True  
    End With  
End Sub
```

NextDay Method Example

The following example increments the value of a Calendar control by one day when the user clicks the button named ShowNextDay.

To try this example, create a form with a Calendar control named ActiveXCtl0 and a command button named ShowNextDay. Paste the following code into the form's module. Switch to Form view and click the command button to increment the date.

```
Private Sub ShowNextDay_Click()  
    Me.ActiveXCtl0.NextDay  
End Sub
```

NextMonth Method Example

The following example increments the value of a Calendar control by one month when the user clicks the button named ShowNextMonth.

To try this example, create a form with a Calendar control named ActiveXCtl0 and a command button named ShowNextMonth. Paste the following code into the form's module. Switch to Form view and click the command button to increment the month.

```
Private Sub ShowNextMonth_Click()  
    Me.ActiveXCtl0.NextMonth  
End Sub
```

NextWeek Method Example

The following example increments the value of a Calendar control by one week when the user clicks the button named ShowNextWeek.

To try this example, create a form with a Calendar control named ActiveXCtl0 and a command button named ShowNextWeek. Paste the following code into the form's module. Switch to Form view and click the command button to increment the week.

```
Private Sub ShowNextWeek_Click()  
    Me.ActiveXCtl0.NextWeek  
End Sub
```

NextYear Method Example

The following example increments the value of a Calendar control by one year when the user clicks the button named ShowNextYear.

To try this example, create a form with a Calendar control named ActiveXCtl0 and a command button named ShowNextYear. Paste the following code into the form's module. Switch to Form view and click the command button to increment the year.

```
Private Sub ShowNextYear_Click()  
    Me.ActiveXCtl0.NextYear  
End Sub
```

PreviousDay Method Example

The following example decrements the value of a Calendar control by one day when the user clicks the button named ShowPreviousDay.

To try this example, create a form with a Calendar control named ActiveXctl0 and a command button named ShowPreviousDay. Paste the following code into the form's module. Switch to Form view and click the command button to decrement the date.

```
Private Sub ShowPreviousDay_Click()  
    Me!ActiveXctl0.PreviousDay  
End Sub
```

PreviousMonth Method Example

The following example decrements the value of a Calendar control by one month when the user clicks the button named ShowPreviousMonth.

To try this example, create a form with a Calendar control named ActiveXCtl0 and a command button named ShowPreviousMonth. Paste the following code into the form's module. Switch to Form view and click the command button to decrement the month.

```
Private Sub ShowPreviousMonth_Click()  
    Me.ActiveXCtl0.PreviousMonth  
End Sub
```

PreviousWeek Method Example

The following example decrements the value of a Calendar control by one week when the user clicks the button named ShowPreviousWeek.

To try this example, create a form with a Calendar control named ActiveXctl0 and a command button named ShowPreviousWeek. Paste the following code into the form's module. Switch to Form view and click the command button to decrement the week.

```
Private Sub ShowPreviousWeek_Click()  
    Me!ActiveXctl0.PreviousWeek  
End Sub
```

PreviousYear Method Example

The following example decrements the value of a Calendar control by one year when the user clicks the button named ShowPreviousYear.

To try this example, create a form with a Calendar control named ActiveXCtl0 and a command button named ShowPreviousYear. Paste the following code into the form's module. Switch to Form view and click the command button to decrement the year.

```
Private Sub ShowPreviousYear_Click()  
    Me!ActiveXCtl0.PreviousYear  
End Sub
```

Refresh Method, FirstDay Property Example

The following example enables the user to select the first day to display in a Calendar control. The user selects either Sunday or Monday from a combo box, and the calendar is updated and refreshed.

To try this example, create a form with a Calendar control named ActiveXCtl0 and a combo box named Combo1. Paste the following code into the form's module. Switch to Form view and change the selection in the combo box.

```
' Initialize combo box.
Private Sub Form_Load()
    Dim cbo As ComboBox

    Set cbo = Me!Combo1
    ' Specify type of source of values.
    cbo.RowSourceType = "Value List"
    ' Specify values.
    cbo.RowSource = "Sunday;Monday"
End Sub

Private Sub Combo1_AfterUpdate()
    Dim ctl As Control

    ' Return reference to Calendar control.
    Set ctl = Me!ActiveXCtl0
    'Set FirstDay property based on selection in combo box.
    If Combo1 = "Sunday" Then
        ' Specify Sunday as first day.
        ctl.FirstDay = 1
    Else
        ' Specify Monday as first day.
        ctl.FirstDay = 2
    End If
    ' Refresh Calendar control.
    ctl.Refresh
End Sub
```

Calendar Control

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccsumCalendarDescriptionC"}          {ewc  
HLP95EN.DLL,DYNALINK,"Properties":"ccidxPropList":1}                    {ewc  
HLP95EN.DLL,DYNALINK,"Methods":"ccidxMethList":1}                      {ewc HLP95EN.DLL,DYNALINK,"Events":"ccidxEventList":1}
```

The Calendar control displays a monthly calendar that you can insert in a form.

File Name

Mscal.ocx

Remarks

The Calendar control supports properties that you can use to set and retrieve the date and to control the appearance of the calendar. It supports methods that you can use to set the day, month, and year, to show an About box, and to refresh the calendar. It also supports events that notify your application when the user has moved to a new date on the calendar, when the user has clicked or double-clicked on a date, and when the user has pressed a key.

Calendar Control Properties

{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccidxPropListC"} {ewc
HLP95EN.DLL,DYNALINK,"Methods":"ccidxMethList":1} {ewc HLP95EN.DLL,DYNALINK,"Events":"ccidxEventList":1}

The following table describes the properties for the Calendar control.

| Property | Description |
|----------------------------------|--|
| <u>BackColor</u> | The background color for the Calendar control. |
| <u>Day</u> | The currently selected day of the month. |
| <u>DayFont</u> | The font used to display the days of the week in the Calendar control. |
| <u>DayFontColor</u> | The color used to display the days of the week. |
| <u>DayLength</u> | The format used to display days of the week. |
| <u>FirstDay</u> | The first day of the week that appears in the Calendar control. |
| <u>GridCellEffect</u> | The effect used to display the calendar. |
| <u>GridFont</u> | The font used to display the days of the month in the grid. |
| <u>GridFontColor</u> | The color of the font used to display the days of the month. |
| <u>GridLinesColor</u> | The gridline color for a Calendar control whose GridCellEffect property is set to Flat (0). |
| <u>Month</u> | The month currently displayed in a Calendar control. |
| <u>MonthLength</u> | The format used to display months of the year. |
| <u>ShowDateSelectors</u> | The visibility of the month and year date selectors. |
| <u>ShowDays</u> | The visibility of the days of the week. |
| <u>ShowHorizontalGrid</u> | The visibility of the horizontal gridlines for a Calendar control whose GridCellEffect property is set to Flat. |
| <u>ShowTitle</u> | The visibility of the month/year title. |
| <u>ShowVerticalGrid</u> | The visibility of the vertical gridlines for a Calendar control whose GridCellEffect property is set to Flat. |
| <u>TitleFont</u> | The font used to display the month/year title above the calendar grid. |
| <u>TitleFontColor</u> | The color to use when displaying the month/year title. |
| <u>Value</u> | The date value that corresponds to the selected date in the Calendar control. |
| <u>ValueIsNull</u> | The appearance of the current date as highlighted or not highlighted. |
| <u>Year</u> | The currently selected year. |

Calendar Control Methods

{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccidxMethListC"} {ewc
HLP95EN.DLL,DYNALINK,"Properties":"ccidxPropList":1} {ewc HLP95EN.DLL,DYNALINK,"Events":"ccidxEventList":1}

The following table describes the methods for the Calendar control.

| Method | Description |
|-----------------------------|--|
| <u>AboutBox</u> | Displays the version and copyright information. |
| <u>NextDay</u> | Increments the value of the Calendar control by one day. |
| <u>NextMonth</u> | Increments the value of the Calendar control by one month. |
| <u>NextWeek</u> | Increments the value of the Calendar control by one week. |
| <u>NextYear</u> | Increments the value of the Calendar control by one year. |
| <u>PreviousDay</u> | Decrements the value of the Calendar control by one day. |
| <u>PreviousMonth</u> | Decrements the value of the Calendar control by one month. |
| <u>PreviousWeek</u> | Decrements the value of the Calendar control by one week. |
| <u>PreviousYear</u> | Decrements the value of the Calendar control by one year. |
| <u>Refresh</u> | Repaints the calendar. |
| <u>Today</u> | Sets the current date to today's date. |

Calendar Control Events

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccidxEventListC"} {ewc  
HLP95EN.DLL,DYNALINK,"Properties":"ccidxPropList":1} {ewc  
HLP95EN.DLL,DYNALINK,"Methods":"ccidxMethList":1}
```

The following table describes the events for the Calendar control.

| Event | Description |
|----------------------------|---|
| <u>AfterUpdate</u> | Occurs after the user moves to a new date in the Calendar control and after the calendar has been refreshed. |
| <u>BeforeUpdate</u> | Occurs after the user moves to a new date in the Calendar control but before the calendar has been refreshed. |
| <u>Click</u> | Occurs when the user clicks a date in the Calendar control. |
| <u>DbClick</u> | Occurs when the user double-clicks a date in the Calendar control. |
| <u>KeyDown</u> | Occurs when the user presses a key while the Calendar control has the focus. |
| <u>KeyPress</u> | Occurs when the user presses and releases an ANSI key while the Calendar control has the focus. |
| <u>KeyUp</u> | Occurs when the user releases a key while the Calendar control has the focus. |
| <u>NewMonth</u> | Occurs when the Calendar control date changes to a new month. |
| <u>NewYear</u> | Occurs when the Calendar control date changes to a new year. |

BackColor Property (Calendar Control)

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproBackColorC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproBackColorX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproBackColorA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproBackColorS"}
```

You can use the **BackColor** property to specify the background color that appears in the Calendar control.

Setting

The setting for the **BackColor** property is a **Long** value that represents an RGB color.

You can set this property by using the Calendar control property sheet or Visual Basic.

Remarks

You can specify an RGB color for the **BackColor** property by using the Visual Basic **RGB** function. The valid range for a normal RGB color is 0 to 16,777,215.

You can also specify a value corresponding to a system color, as defined on the **Appearance** tab of the **Display Properties** dialog box in Windows Control Panel. The simplest way to set the **BackColor** property to a system color is to set this property on the **Colors** tab of the Calendar control property sheet.

Microsoft Windows provides a set of negative numbers that specify system colors. You can use these numbers to set the **BackColor** property from Visual Basic. These numbers correspond to the system colors for various parts of an application's window, such as the window frame or the title bar. To determine the value of the number that corresponds to a specific color, you must first read the value of the **BackColor** property when it's set to that color.

If you don't specify any value for the **BackColor** property, the background of the Calendar control is displayed in a system color.

Day Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproDayC"}
{ewc HLP95EN.DLL,DYNALINK,"Applies To":"ccproDayA"}

{ewc HLP95EN.DLL,DYNALINK,"Example":"ccproDayX":1}
{ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproDayS"}

You can use the **Day** property to set the current day of the month in the Calendar control.

Setting

The setting for the **Day** property is an **Integer** data type value specifying the day of the month. Possible values range from 1 to 31 depending on the month and year.

You can set this property by using the Calendar control property sheet or Visual Basic.

DayFont Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproDayFontC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproDayFontX":1}             {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproDayFontA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproDayFontS"}
```

You can use the **DayFont** property to specify the characteristics of the font used to display the days of the week in the Calendar control.

Syntax

```
controlname.DayFont[ = name]  
controlname.DayFont.Name[ = name]  
controlname.DayFont.Size[ = size]  
controlname.DayFont.Bold[ = bold]  
controlname.DayFont.Italic[ = italic]  
controlname.DayFont.Underline[ = underline]  
controlname.DayFont.Strikethrough[ = strikethrough]
```

Setting

The **DayFont** property uses the following settings.

| Setting | Visual Basic |
|----------------------|---|
| <i>controlname</i> | The name of the Calendar control object. |
| <i>name</i> | A string expression specifying the font name. |
| <i>size</i> | An Integer value that specifies the font size. |
| <i>bold</i> | A Boolean value that specifies whether the font is bold. |
| <i>italic</i> | A Boolean value that specifies whether the font is italic. |
| <i>underline</i> | A Boolean value that specifies whether the font is underlined. |
| <i>strikethrough</i> | A Boolean expression that specifies whether to use a strikethrough font. |

You can set this property by using the Calendar control property sheet or Visual Basic.

Remarks

The **DayFont** property returns an object which has additional properties, the **Name**, **Size**, **Bold**, **Italic**, **Underline**, and **Strikethrough** properties. You can use the **DayFont** property to set any of these properties for the days of the week displayed in the Calendar control.

The default property for the object returned by the **DayFont** property is the **Name** property. Therefore, you can set the **DayFont** property to a string specifying the name of a font, without explicitly referring to the **Name** property.

Note If you're using the **WithEvents** keyword to create event procedures for the Calendar control, you must set a reference to the **OLE Automation** library before you can set or return the value of the **DayFont** property. To set this reference, click **References** on the **Tools** menu while in module Design view and select the box next to **OLE Automation**. The file that contains this type library is Stdole2.tlb, and is located by default in the Microsoft Windows System directory.

DayFontColor Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproDayFontColorC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproDayFontColorX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproDayFontColorA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproDayFontColorS"}
```

You can use the **DayFontColor** property to specify the color used to display the days of the week in the Calendar control.

Setting

The setting for the **DayFontColor** property is a **Long** value that represents an RGB color.

You can set this property by using the Calendar control property sheet or Visual Basic.

Remarks

You can specify an RGB color for the **DayFontColor** property by using the Visual Basic **RGB** function. The valid range for a normal RGB color is 0 to 16,777,215.

You can also specify a value corresponding to a system color, as defined on the **Appearance** tab of the **Display Properties** dialog box in Windows Control Panel. The simplest way to set the **DayFontColor** property to a system color is to set this property on the **Colors** tab of the Calendar control property sheet.

Microsoft Windows provides a set of negative numbers that specify system colors. You can use these numbers to set the **DayFontColor** property from Visual Basic. These numbers correspond to the system colors for various parts of an application's window, such as the window frame or the title bar. To determine the value of the number that corresponds to a specific color, you must first read the value of the **DayFontColor** property when it's set to that color.

If you don't specify any value for the **DayFontColor** property, the days of the week are displayed in a system color.

DayLength Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproDayLengthC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproDayLengthX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproDayLengthA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproDayLengthS"}
```

You can use the **DayLength** property to determine whether the days of the week in the Calendar control are displayed in short, medium, or long format.

Setting

The **DayLength** property uses the following settings.

| Setting | Description | Visual Basic |
|----------------|---|---------------------|
| Short | Displays S, M, T, W, T, F, S. | 0 |
| Medium | Displays Sun, Mon, Tue, Wed, Thu, Fri, Sat. | 1 |
| Long | Displays Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday. | 2 |

You can set this property by using the Calendar control property sheet or Visual Basic.

FirstDay Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproFirstDayC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproFirstDayX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproFirstDayA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproFirstDayS"}
```

You can use the **FirstDay** property to determine which day of the week is displayed in the first column of the Calendar control.

Setting

The setting for the **FirstDay** property is an **Integer** value that corresponds to a day of the week, as shown in the following table.

| Setting | Visual Basic |
|----------------|---------------------|
| Sunday | 1 |
| Monday | 2 |
| Tuesday | 3 |
| Wednesday | 4 |
| Thursday | 5 |
| Friday | 6 |
| Saturday | 7 |

You can set this property by using the Calendar control property sheet or Visual Basic.

GridFont Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproGridFontC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproGridFontX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproGridFontA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproGridFontS"}
```

You can use the **GridFont** property to specify the font used to display the days of the month in the grid of the Calendar control.

Syntax

```
controlname.GridFont[ = name]  
controlname.GridFont.Name[ = name]  
controlname.GridFont.Size[ = size]  
controlname.GridFont.Bold[ = bold]  
controlname.GridFont.Italic[ = italic]  
controlname.GridFont.Underline[ = underline]  
controlname.GridFont.Strikethrough[ = strikethrough]
```

Setting

The **GridFont** property uses the following settings.

| Setting | Visual Basic |
|----------------------|---|
| <i>controlname</i> | The name of the Calendar control object. |
| <i>name</i> | A string expression specifying the font name. |
| <i>size</i> | An Integer value that specifies the font size. |
| <i>bold</i> | A Boolean value that specifies whether the font is bold. |
| <i>italic</i> | A Boolean value that specifies whether the font is italic. |
| <i>underline</i> | A Boolean value that specifies whether the font is underlined. |
| <i>strikethrough</i> | A Boolean expression that specifies whether to use a strikethrough font. |

You can set this property by using the Calendar control property sheet or Visual Basic.

Remarks

The **GridFont** property returns an object which has additional properties, the **Name**, **Size**, **Bold**, **Italic**, **Underline**, and **Strikethrough** properties. You can use the **GridFont** property to set any of these properties for the days of the month displayed in the Calendar control.

The default property for the object returned by the **GridFont** property is the **Name** property. Therefore, you can set the **GridFont** property to a string specifying the name of a font, without explicitly referring to the **Name** property.

Note If you're using the **WithEvents** keyword to create event procedures for the Calendar control, you must set a reference to the **OLE Automation** library before you can set or return the value of the **GridFont** property. To set this reference, click **References** on the **Tools** menu while in module Design view and select the box next to **OLE Automation**. The file that contains this type library is Stdole2.tlb, and is located by default in the Microsoft Windows System directory.

GridFontColor Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproGridFontColorC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproGridFontColorX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproGridFontColorA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproGridFontColorS"}
```

You can use the **GridFontColor** property to specify the color used to display the days of the month in the grid of the Calendar control.

Setting

The setting for the **GridFontColor** property is a **Long** value that represents an RGB color.

You can set this property by using the Calendar control property sheet or Visual Basic.

Remarks

You can specify an RGB color for the **GridFontColor** property by using the Visual Basic **RGB** function. The valid range for a normal RGB color is 0 to 16,777,215.

You can also specify a value corresponding to a system color, as defined on the **Appearance** tab of the **Display Properties** dialog box in Windows Control Panel. The simplest way to set the **GridFontColor** property to a system color is to set this property on the **Colors** tab of the Calendar control property sheet.

Microsoft Windows provides a set of negative numbers that specify system colors. You can use these numbers to set the **GridFontColor** property from Visual Basic. These numbers correspond to the system colors for various parts of an application's window, such as the window frame or the title bar. To determine the value of the number that corresponds to a specific color, you must first read the value of the **GridFontColor** property when it's set to that color.

If you don't specify any value for the **GridFontColor** property, the days of the month are displayed in a system color.

GridLinesColor Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproGridLinesColorC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproGridLinesColorX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproGridLinesColorA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproGridLinesColorS"}
```

You can use the **GridLinesColor** property to specify the color used to display the lines in the grid of a Calendar control.

Setting

The setting for the **GridLinesColor** property is a **Long** value that represents an RGB color.

Before setting the **GridLinesColor** property, make sure that the **GridCellEffect** property is set to Flat (0). If the **GridCellEffect** property isn't set to Flat, setting the **GridLinesColor** property has no effect.

You can set this property by using the Calendar control property sheet or Visual Basic.

Remarks

You can specify an RGB color for the **GridLinesColor** property by using the Visual Basic **RGB** function. The valid range for a normal RGB color is 0 to 16,777,215.

You can also specify a value corresponding to a system color, as defined on the **Appearance** tab of the **Display Properties** dialog box in Windows Control Panel. The simplest way to set the **GridLinesColor** property to a system color is to set this property on the **Colors** tab of the Calendar control property sheet.

Microsoft Windows provides a set of negative numbers that specify system colors. You can use these numbers to set the **GridLinesColor** property from Visual Basic. These numbers correspond to the system colors for various parts of an application's window, such as the window frame or the title bar. To determine the value of the number that corresponds to a specific color, you must first read the value of the **GridLinesColor** property when it's set to that color.

If you don't specify any value for the **GridLinesColor** property, the gridlines are displayed in a system color.

GridCellEffect Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproGridLinesFormatC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"ccproGridLinesFormatX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"ccproGridLinesFormatA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproGridLinesFormatS"}

You can use the **GridCellEffect** property to specify the effect used to display the grid of the Calendar control.

Setting

The **GridCellEffect** property uses the following settings.

| Setting | Visual Basic |
|----------------|---------------------|
| Flat | 0 |
| Raised | 1 |
| Sunken | 2 |

You can set this property by using the Calendar control property sheet or Visual Basic.

Remarks

If the **GridCellEffect** property is set to any value other than Flat, setting the **GridLinesColor**, **ShowHorizontalGrid**, and **ShowVerticalGrid** properties has no effect.

Month Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproMonthC"}  
HLP95EN.DLL,DYNALINK,"Example":"ccproMonthX":1}  
{ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproMonthS"}
```

```
{ewc  
{ewc HLP95EN.DLL,DYNALINK,"Applies To":"ccproMonthA"}
```

You can use the **Month** property to specify the month to display in the Calendar control.

Setting

The **Month** property uses the following settings.

| Setting | Visual Basic |
|----------------|---------------------|
| January | 1 |
| February | 2 |
| March | 3 |
| April | 4 |
| May | 5 |
| June | 6 |
| July | 7 |
| August | 8 |
| September | 9 |
| October | 10 |
| November | 11 |
| December | 12 |

You can set this property by using the Calendar control property sheet or Visual Basic.

MonthLength Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproMonthLengthC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"ccproMonthLengthX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"ccproMonthLengthA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproMonthLengthS"}

You can use the **MonthLength** property to specify whether months are displayed in short or long format in the month/year title in the Calendar control.

Setting

The **MonthLength** property uses the following settings.

| Setting | Description | Visual Basic |
|----------------|---|---------------------|
| Short | Displays Jan, Feb, Mar, and so on. | 0 |
| Long | Displays January, February, March, and so on. | 2 |

You can set this property by using the Calendar control property sheet or Visual Basic.

ShowDateSelectors Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproShowDateSelectorsC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproShowDateSelectorsX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproShowDateSelectorsA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproShowDateSelectorsS"}
```

You can use the **ShowDateSelectors** property to specify whether to display the date selector drop-down list boxes for month and year in the Calendar control.

Setting

The setting for the **ShowDateSelectors** property is a **Boolean** value. If the **ShowDateSelectors** property is set to **True** (-1), the date selector drop-down list boxes appear in the upper-right corner of the Calendar control.

You can set this property by using the Calendar control property sheet or Visual Basic.

Remarks

Without the date selectors, you can still set the month and year by using the **PreviousMonth**, **NextMonth**, **PreviousYear**, and **NextYear** methods and their corresponding keyboard equivalents.

ShowDays Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproShowDaysC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproShowDaysX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproShowDaysA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproShowDaysS"}
```

You can use the **ShowDays** property to specify whether the days of the week are displayed in the Calendar control.

Setting

The setting for the **ShowDays** property is a **Boolean** value. If the **ShowDays** property is set to **True** (–1), the Calendar control displays the days of the week.

You can set this property by using the Calendar control property sheet or Visual Basic.

ShowHorizontalGrid Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproShowHorizontalGridlinesC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproShowHorizontalGridlinesX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproShowHorizontalGridlinesA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproShowHorizontalGridlinesS"}
```

You can use the **ShowHorizontalGrid** property to specify whether horizontal gridlines are displayed in a Calendar control.

Setting

The setting for the **ShowHorizontalGrid** property is a **Boolean** value.

If the **ShowHorizontalGrid** property is set to **True** (-1) and the **GridCellEffect** property is set to Flat (0), the Calendar control displays horizontal gridlines. If the **GridCellEffect** property isn't set to Flat, setting the **ShowHorizontalGrid** property has no effect.

You can set this property by using the Calendar control property sheet or Visual Basic.

ShowTitle Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproShowTitleC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproShowTitleX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproShowTitleA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproShowTitleS"}
```

You can use the **ShowTitle** property to specify whether the month/year title is displayed above the calendar grid.

Setting

The setting for the **ShowTitle** property is a **Boolean** value. If the **ShowTitle** property is set to **True** (–1), the Calendar control displays the month/year title.

You can set this property by using the Calendar control property sheet or Visual Basic.

Remarks

The month/year title is displayed at the top of the Calendar control, above the days of the week, and to the left of the month date selector. If the **ShowDateSelectors** property is set to **False** (0), the month/year title is centered above the grid.

ShowVerticalGrid Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproShowVerticalGridlinesC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproShowVerticalGridlinesX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproShowVerticalGridlinesA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproShowVerticalGridlinesS"}
```

You can use the **ShowVerticalGrid** property to specify whether vertical gridlines are displayed in a Calendar control.

Setting

The setting for the **ShowVerticalGrid** property is a **Boolean** value.

If the **ShowVerticalGrid** property is set to **True** (-1) and the **GridCellEffect** property is set to Flat (0), the Calendar control displays vertical gridlines. If the **GridCellEffect** property isn't set to Flat, setting the **ShowVerticalGrid** property has no effect.

You can set this property by using the Calendar control property sheet or Visual Basic.

TitleFont Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproTitleFontC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproTitleFontX":1}             {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproTitleFontA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproTitleFontS"}
```

You can use the **TitleFont** property to specify characteristics of the font displayed in the month/year title above the calendar grid.

Syntax

```
controlname.TitleFont[ = name]  
controlname.TitleFont.Name[ = name]  
controlname.TitleFont.Size[ = size]  
controlname.TitleFont.Bold[ = bold]  
controlname.TitleFont.Italic[ = italic]  
controlname.TitleFont.Underline[ = underline]  
controlname.TitleFont.Strikethrough[ = strikethrough]
```

Setting

The **TitleFont** property uses the following settings.

| Setting | Visual Basic |
|----------------------|---|
| <i>controlname</i> | The name of the Calendar control object. |
| <i>name</i> | A string expression specifying the font name. |
| <i>size</i> | An Integer value that specifies the font size. |
| <i>bold</i> | A Boolean value that specifies whether the font is bold. |
| <i>italic</i> | A Boolean value that specifies whether the font is italic. |
| <i>underline</i> | A Boolean value that specifies whether the font is underlined. |
| <i>strikethrough</i> | A Boolean expression that specifies whether to use a strikethrough font. |

You can set this property by using the Calendar control property sheet or Visual Basic.

Remarks

The **TitleFont** property returns an object which has additional properties, the **Name**, **Size**, **Bold**, **Italic**, **Underline**, and **Strikethrough** properties. You can use the **TitleFont** property to set any of these properties for the month/year title displayed in the Calendar control.

The default property for the object returned by the **TitleFont** property is the **Name** property. Therefore, you can set the **TitleFont** property to a string specifying the name of a font, without explicitly referring to the **Name** property.

Note If you're using the **WithEvents** keyword to create event procedures for the Calendar control, you must set a reference to the **OLE Automation** library before you can set or return the value of the **TitleFont** property. To set this reference, click **References** on the **Tools** menu while in module Design view and select the box next to **OLE Automation**. The file that contains this type library is Stdole2.tlb, and is located by default in the Microsoft Windows System directory.

TitleFontColor Property

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproTitleFontColorC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproTitleFontColorX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproTitleFontColorA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproTitleFontColorS"}
```

You can use the **TitleFontColor** property to specify the color of the font used to display the month/year title in the Calendar control.

Setting

The setting for the **TitleFontColor** property is a **Long** value that represents an RGB color.

You can set this property by using the Calendar control property sheet or Visual Basic.

Remarks

You can specify an RGB color for the **TitleFontColor** property by using the Visual Basic **RGB** function. The valid range for a normal RGB color is 0 to 16,777,215.

You can also specify a value corresponding to a system color, as defined on the **Appearance** tab of the **Display Properties** dialog box in Windows Control Panel. The simplest way to set the **TitleFontColor** property to a system color is to set this property on the **Colors** tab of the Calendar control property sheet.

Microsoft Windows provides a set of negative numbers that specify system colors. You can use these numbers to set the **TitleFontColor** property from Visual Basic. These numbers correspond to the system colors for various parts of an application's window, such as the window frame or the title bar. To determine the value of the number that corresponds to a specific color, you must first read the value of the **TitleFontColor** property when it's set to that color.

If you don't specify any value for the **TitleFontColor** property, the month/year title is displayed in a system color.

Value Property (Calendar Control)

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproValueC"}  
HLP95EN.DLL,DYNALINK,"Example":"ccproValueX":1}  
{ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproValueS"}  
  
{ewc  
{ewc HLP95EN.DLL,DYNALINK,"Applies To":"ccproValueA"}
```

You can use the **Value** property to set or retrieve the currently selected date in a Calendar control.

Setting

The setting for the **Value** property is a date value or **Null**.

You can set this property by using the Calendar control property sheet or Visual Basic.

Remarks

Valid dates for the Calendar control range from 1/1/1900 to 12/31/2100.

The **Value** property changes when you use the arrow keys, the mouse, or the date selector drop-down list boxes to change the date. If you set the **Value** property to **Null**, no date is highlighted.

If your application supports data-bound controls, you may be able to bind the Calendar control to a date field in a database. Check your application's documentation to determine whether it supports data-bound controls.

ValuelsNull Property (Calendar Control)

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproValuelsNullC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccproValuelsNullX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccproValuelsNullA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproValuelsNullS"}
```

You can use the **ValuelsNull** property to specify that the **Value** property of a Calendar control is set to **Null**. When the **Value** property is set to **Null**, no date is selected.

Setting

The setting for the **ValuelsNull** property is a **Boolean** value. By default the **ValuelsNull** property is set to **False** (0).

You can set this property by using the Calendar control property sheet or Visual Basic.

Remarks

When the **ValuelsNull** property is set to **False**, the current date is selected on the Calendar control until the user selects a different date. The **Value** property returns the current date. When the **ValuelsNull** property is set to **True** (-1), the current date isn't selected, and the **Value** property returns **Null**.

If the **Value** property is changed when the **ValuelsNull** property is set to **True**, then the **ValuelsNull** property is set to **False**. Your code may change the **Value** property directly. The user may also change the **Value** property by selecting a date on the Calendar control with the arrow keys, the mouse, or the date selector drop-down list boxes. If you don't want the user to change the setting of the **Value** property at all, set the control's **Enabled** property to **False**.

Set the **ValuelsNull** property to **True** if you want the value of a Calendar control to be **Null** rather than the current date.

Year Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccproYearC"}
{ewc HLP95EN.DLL,DYNALINK,"Applies To":"ccproYearA"}

{ewc HLP95EN.DLL,DYNALINK,"Example":"ccproYearX":1}
{ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccproYearS"}

You can use the **Year** property to specify the year displayed in the Calendar control.

Setting

The setting for the **Year** property is an **Integer** value between 1900 and 2100, inclusive.

You can set this property by using the Calendar control property sheet or Visual Basic.

AboutBox Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccmthAboutBoxC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"ccmthAboutBoxX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"ccmthAboutBoxA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccmthAboutBoxS"}

You can use the **AboutBox** method to display an informational dialog box about the Calendar control, including its version and copyright information.

Syntax

controlname.**AboutBox**

The **AboutBox** method has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

NextDay Method

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccmthNextDayC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccmthNextDayX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccmthNextDayA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccmthNextDayS"}
```

You can use the **NextDay** method to increment the value of a Calendar control by one day and refresh the control.

Syntax

controlname.**NextDay**

The **NextDay** method has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

Remarks

If you apply the **NextDay** method when the value of the Calendar control is the last day of a month or year, the month or the year is also incremented.

Pressing the RIGHT ARROW key on the keyboard when the Calendar control has the focus executes the **NextDay** method.

NextMonth Method

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccmthNextMonthC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccmthNextMonthX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccmthNextMonthA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccmthNextMonthS"}
```

You can use the **NextMonth** method to increment the value of a Calendar control by one month and refresh the control.

Syntax

controlname.**NextMonth**

The **NextMonth** method has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

Remarks

If you apply the **NextMonth** method when the current month is December, the year is also incremented.

Pressing the RIGHT ARROW key while holding down the CTRL key when the Calendar control has the focus executes this method.

NextWeek Method

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccmthNextWeekC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccmthNextWeekX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccmthNextWeekA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccmthNextWeekS"}
```

You can use the **NextWeek** method to increment the value of a Calendar control by one week and refresh the control.

Syntax

controlname.**NextWeek**

The **NextWeek** method has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

Remarks

Pressing the DOWN ARROW key while the Calendar control has the focus executes this method.

NextYear Method

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccmthNextYearC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccmthNextYearX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccmthNextYearA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccmthNextYearS"}
```

You can use the **NextYear** method to increment the value of a Calendar control by one year and refresh the control.

Syntax

controlname.**NextYear**

The **NextYear** method has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

Remarks

Pressing the DOWN ARROW key while holding down the CTRL key when the Calendar control has the focus executes this method.

PreviousDay Method

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccmthPreviousDayC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccmthPreviousDayX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccmthPreviousDayA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccmthPreviousDayS"}
```

You can use the **PreviousDay** method to decrement the value of a Calendar control by one day and refresh the control.

Syntax

controlname.**PreviousDay**

The **PreviousDay** method has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

Remarks

If you apply the **NextDay** method when the value of the Calendar control is the first day of a month or year, the month or the year is also decremented.

Pressing the LEFT ARROW key while the Calendar control has the focus executes this method.

PreviousMonth Method

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccmthPreviousMonthC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccmthPreviousMonthX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccmthPreviousMonthA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccmthPreviousMonthS"}
```

You can use the **PreviousMonth** method to decrement the value of a Calendar control by one month and refresh the control.

Syntax

controlname.**PreviousMonth**

The **PreviousMonth** method has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

Remarks

If you apply the **PreviousMonth** method when the current month is January, the year is also decremented.

Pressing the LEFT ARROW key while holding down the CTRL key when the Calendar control has the focus executes this method.

PreviousWeek Method

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccmthPreviousWeekC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccmthPreviousWeekX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccmthPreviousWeekA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccmthPreviousWeekS"}
```

You can use the **PreviousWeek** method to decrement the value of a Calendar control by one week and refresh the control.

Syntax

controlname.**PreviousWeek**

The **PreviousWeek** method has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

Remarks

Pressing the UP ARROW key while the Calendar control has the focus executes this method.

PreviousYear Method

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccmthPreviousYearC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccmthPreviousYearX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccmthPreviousYearA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccmthPreviousYearS"}
```

You can use the **PreviousYear** method to decrement the value of a Calendar control by one year and refresh the control.

Syntax

controlname.**PreviousYear**

The **PreviousYear** method has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

Remarks

Pressing the UP ARROW key while holding down the CTRL key on the keyboard executes this method.

Refresh Method (Calendar Control)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccmthRefreshC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"ccmthRefreshX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"ccmthRefreshA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccmthRefreshS"}

You can use the **Refresh** method to repaint the Calendar control.

Syntax

controlname.**Refresh**

The **Refresh** method has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

Today Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccmthTodayC"}
HLP95EN.DLL,DYNALINK,"Example":"ccmthTodayX":1}
{ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccmthTodayS"}

{ewc
{ewc HLP95EN.DLL,DYNALINK,"Applies To":"ccmthTodayA"}

You can use the **Today** method to set the value of the Calendar control to today's date.

Syntax

controlname.**Today**

The **Today** method has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

Remarks

Pressing CTRL+T on the keyboard executes this method.

AfterUpdate Event (Calendar Control)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccevtAfterUpdateC"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"ccevtAfterUpdateX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies
To":"ccevtAfterUpdateA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccevtAfterUpdateS"}

The AfterUpdate event occurs after the user selects a new date in the Calendar control and the new date is highlighted.

Syntax

Sub *controlname*_AfterUpdate()

The AfterUpdate event procedure has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

Remarks

This event occurs whenever the user moves to a new date in the Calendar control by using the keyboard keys or the mouse.

The **Value** property of the Calendar control changes to the new date before the AfterUpdate event occurs.

BeforeUpdate Event (Calendar Control)

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccevtBeforeUpdateC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccevtBeforeUpdateX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccevtBeforeUpdateA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccevtBeforeUpdateS"}
```

The BeforeUpdate event occurs after the user selects a new date in the Calendar control but before the new date is highlighted.

Syntax

Sub *controlname*_BeforeUpdate (Cancel As Integer)

The BeforeUpdate event procedure has the following arguments.

| Argument | Description |
|--------------------|---|
| <i>controlname</i> | The name of the Calendar control object. |
| Cancel | If set to True (-1) within the event procedure, the event is canceled. |

Remarks

This event occurs whenever the user moves to a new date in the Calendar control by using the keyboard keys or the mouse.

The **Value** property of the Calendar control changes to the new date before the BeforeUpdate event occurs.

To cancel the BeforeUpdate event and return the Calendar control to its initial state, set Cancel to **True** within the BeforeUpdate event procedure. When Cancel is set to **True**, the new date selected by the user isn't highlighted. Once the event procedure has finished running, the **Value** property is reset to the value it had before the BeforeUpdate event procedure ran.

Click Event (Calendar Control)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccevtClickC"}
{ewc HLP95EN.DLL,DYNALINK,"Applies To":"ccevtClickA"}

{ewc HLP95EN.DLL,DYNALINK,"Example":"ccevtClickX":1}
{ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccevtClickS"}

The Click event occurs when the user clicks on a date in the Calendar control.

Syntax

Sub *controlname*_Click()

The Click event procedure has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

Remarks

Clicking on a date sets the Calendar control's **Value** property to that date.

When the user changes the value of the Calendar control by clicking on a different date, the BeforeUpdate and AfterUpdate events occur before the Click event occurs.

DbIclicK Event (Calendar Control)

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccevtDbIclicK"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccevtDbIclicKX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccevtDbIclicKA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccevtDbIclicKS"}
```

The DbIclicK event occurs when the user double-clicks on a date in the Calendar control.

Syntax

Sub *controlname*_DbIclicK()

The DbIclicK event procedure has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

Remarks

The Click event always occurs immediately before the DbIclicK event occurs. The BeforeUpdate and AfterUpdate events occur before the Click or DbIclicK events occur.

KeyDown, KeyUp Events (Calendar Control)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"acevtGotLostFocus;ccevtKeyDownC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"ccevtKeyDownX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"ccevtKeyDownA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccevtKeyDownS"}

The KeyDown and KeyUp events occur when the user presses or releases a key while the Calendar control has the focus. You can use these events to determine when the user has either pressed a non-alphanumeric key or a combination of keys, such as the SHIFT key with an alphanumeric key.

Syntax

Sub *controlname*_KeyDown(KeyCode As Integer, ByVal Shift As Integer)

Sub *controlname*_KeyUp(KeyCode As Integer, ByVal Shift As Integer)

The KeyDown and KeyUp event procedures have the following arguments.

| Argument | Description | | | | | | | | | | | | | | | | | | |
|--------------------|--|-------------|-----------------------------|------|---|-------|---|------|---|-----|---|------------|---|-----------|---|----------|---|----------------|---|
| <i>controlname</i> | The name of the Calendar control object. | | | | | | | | | | | | | | | | | | |
| KeyCode | An integer that's a key code value. To see a list of possible values, open the Object Browser, select VBA in the Project/Library box, and select KeyCodeConstants in the Classes box. | | | | | | | | | | | | | | | | | | |
| Shift | A value specifying the state of the SHIFT, CTRL, and ALT keys. The following values are possible for the Shift argument. <table><thead><tr><th>Key pressed</th><th>Value of the Shift argument</th></tr></thead><tbody><tr><td>None</td><td>0</td></tr><tr><td>SHIFT</td><td>1</td></tr><tr><td>CTRL</td><td>2</td></tr><tr><td>ALT</td><td>4</td></tr><tr><td>SHIFT-CTRL</td><td>3</td></tr><tr><td>SHIFT-ALT</td><td>5</td></tr><tr><td>CTRL-ALT</td><td>6</td></tr><tr><td>SHIFT-CTRL-ALT</td><td>7</td></tr></tbody></table> | Key pressed | Value of the Shift argument | None | 0 | SHIFT | 1 | CTRL | 2 | ALT | 4 | SHIFT-CTRL | 3 | SHIFT-ALT | 5 | CTRL-ALT | 6 | SHIFT-CTRL-ALT | 7 |
| Key pressed | Value of the Shift argument | | | | | | | | | | | | | | | | | | |
| None | 0 | | | | | | | | | | | | | | | | | | |
| SHIFT | 1 | | | | | | | | | | | | | | | | | | |
| CTRL | 2 | | | | | | | | | | | | | | | | | | |
| ALT | 4 | | | | | | | | | | | | | | | | | | |
| SHIFT-CTRL | 3 | | | | | | | | | | | | | | | | | | |
| SHIFT-ALT | 5 | | | | | | | | | | | | | | | | | | |
| CTRL-ALT | 6 | | | | | | | | | | | | | | | | | | |
| SHIFT-CTRL-ALT | 7 | | | | | | | | | | | | | | | | | | |

Remarks

You can use the KeyDown and KeyUp events to determine which key or combination of keys has been pressed. Within the KeyDown or KeyUp event procedure, check the value of the KeyCode argument to determine which key was pressed. Check the value of the Shift argument to determine whether the SHIFT, CTRL, or ALT key was pressed simultaneously.

KeyPress Event (Calendar Control)

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"acevtGotLostFocus;acevtKeyDownUp;ccevtKeyPressC"} {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccevtKeyPressX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccevtKeyPressA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccevtKeyPressS"}
```

The KeyPress event occurs when the user presses and releases an ANSI key while the Calendar control has the focus.

Syntax

Sub *controlname*_KeyPress(KeyAscii **As Integer**)

The KeyPress event procedure has the following arguments.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |
| KeyAscii | An integer that's a numeric ANSI key code. |

Remarks

You can use the KeyPress event to determine which alphanumeric key the user has pressed. Within the KeyPress event procedure, check the value of the KeyAscii argument.

To return the alphanumeric character represented by the KeyAscii argument, use the Visual Basic **Chr** function.

NewMonth Event

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccevtNewMonthC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccevtNewMonthX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccevtNewMonthA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccevtNewMonthS"}
```

The NewMonth event occurs whenever the value of the Calendar control changes to display a different month.

Syntax

Sub *controlname*_NewMonth

The NewMonth event procedure has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

NewYear Event

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"ccevtNewYearC"}           {ewc  
HLP95EN.DLL,DYNALINK,"Example":"ccevtNewYearX":1}           {ewc HLP95EN.DLL,DYNALINK,"Applies  
To":"ccevtNewYearA"}           {ewc HLP95EN.DLL,DYNALINK,"Specifics":"ccevtNewYearS"}
```

The NewYear event occurs whenever the value of the Calendar control changes to display a different year.

Syntax

Sub *controlname*_NewYear

The NewYear event procedure has the following argument.

| Argument | Description |
|--------------------|--|
| <i>controlname</i> | The name of the Calendar control object. |

