MhAlarm

Custom Control MHTI200

Description: The MhAlarm is an audio-visual "attention getter" that comes in three styles: telephone, alarm clock or wrist alarm. You can display MhAlarm controls on your forms and use the associated bitmaps in place of your form's icon when the form is minimized. The MhAlarm bitmaps are shown below.

Class Name: MhIAlarm

Remarks: The MhAlarm control is useful when you want to let the user know that your program has completed a task or needs attention. The ability to display an MhAlarm icon in place of the form's icon is especially helpful when the user minimizes your program to work with other applications.

The control remains in an idle state until you turn it on, using the .RingOn property.

You can use the .WindowState property to instruct the control to display its bitmaps where the icon is normally displayed when a form is minimized. When you replace the form's icon, the MhAlarm control isn't displayed until the alarm triggers for the first time.

Note: The .TimerInterval property applies only to the toggling of bitmap displays when the .RingOn property is non-zero. The sound effects are controlled via the other properties having names beginning with "Ring."

CAUTION: When you set the .Style property, default values are placed into
the .RingTime, .PauseTime, .RingTone, .RingMode, .RingLength and .Interval properties. If you wish to use different values for any of those properties, you must explicitly set each one after setting the .Style property.

CAUTION: If you want to take advantage of the control's ability to display its bitmaps in place of your form's icon, be sure to set the AutoSize property to TRUE.

Properties

AutoSize	BackColor	CtlName	Enabled
Height	Index	*Interval	Left
MousePointer	Parent	*PauseTime	*RingLength
*RingMode	*RingOn	*RingTime	*RingTone
*Style	TabIndex	Tag	Top
Visible	Width	*WindowState	

Events

Change	Click	DblClick	GotFocus
KeyDown	KeyPress	KeyUp	LostFocus
MouseDown	MouseMove	MouseUp	*Ring

Custom Properties

Note: The default values for most of the custom properties depend on the style of MhAlarm control you choose. Please consult the table shown with the .Style property for a complete list of default values.

Interval *(Long Integer)*
  Description: This property sets or returns the interval, in milliseconds, between changes in the bitmap displays when .RingOn is non-zero.

  Usage: [*form*.]*MhAlarm*.**Interval** = *milliseconds&*

  Remarks: The range is 1 - 2,147,483,647

PauseTime *(Integer)*

    Description: The .PauseTime property sets or returns the length, in milliseconds, of the silence between ringing sounds.

    Usage: [*form*.]*MhAlarm*.**PauseTime** = *integer%*

    Remarks: The range is 0-32,767.

RingLength *(Integer)*

    Description: Sets or returns the length of the note sent to the speaker.

    Usage: [*form*.]*MhAlarm*.**RingLength** = *integer%*

    Remarks: The .RingLength property corresponds to the Play% function's "L" length. The range is 1 - 64.

RingMode *(Integer)*

    Description: This property sets or returns the type of ring sound.

    Usage: [*form*.]*MhAlarm*.**RingMode** = *setting%*

    Remarks: The valid settings for the .RingMode property are:

Setting	Description
0	Normal
1	Legato
2	Staccato

Note: The Windows 3.0 sound driver has a bug that causes the RingMode property to have no effect. See Play in this appendix for more information.

RingOn *(Integer)*

    Description: The .RingOn property sets or returns whether the MhAlarm control is active and, if so, how frequently the alarm rings.

    Usage: [*form*.]*MhAlarm*.**RingOn** = *setting%*

    Remarks: The default value is zero. The valid settings are:

Setting	Description
0	Ring off - do not sound tones or toggle bitmaps.
1	Flash ring - single ring only, using RingTime.
2	Ring on - continuous ring.

RingTime *(Integer)*

    Description: The .RingTime property sets or returns the length of time, in milliseconds, during which tones are sent to the speaker.

    Usage: [*form*.]*MhAlarm*.**RingTime** = *integer%*

    Remarks: The range is 0-32,767. (Also see RingLength.)

RingTone *(Integer)*

    Description: The .RingTone property sets or returns the note number that is sent to the Windows API sound driver.

    Usage: [*form*.]*MhAlarm*.**RingTone** = *notenumber%*

    Remarks: This property is equivalent to the Play% function's "N" (note). The range is 1-84. See Play in this appendix for more information.

Style *(Integer)*

    Description: This property sets or returns the icon the control displays.

Usage: [*form*.]*MhAlarm*.**Style** = *setting%*

Remarks: The range is 0-2. The table below lists the bitmaps MhAlarm displays for each setting.

Setting	Description
0	Telephone
1	Alarm clock
2	Wrist alarm

The table below shows the default values of the properties affected by the .Style property:

Property	Style		
	Phone	Alarm clock	Wrist Watch
.PauseTime	1000	500	1000
.RingMode	2-Staccato	2-Staccato	0-Normal
.RingLength	34	32	34
.RingTime	1000	2000	1000
.RingTone	41	45	72
.TimerInterval	29	62	29

WindowState *(Integer)*

Description: The .WindowState property sets or returns whether you want the MhAlarm bitmaps to be displayed (instead of the form's icon) when the form is minimized.

Usage: [*form*.]*MhAlarm*.**WindowState** = *setting%*

Remarks: If you set the .WindowState property to one when your form is minimized, the control displays its bitmaps in place of the Icon associated with the form, provided that the MhAlarm control is activated. At all other times, this property should be zero. The best time to set this property is in the Form_Resize event, as shown in the Example section below. The default value is zero.

Setting	Description
0	Normal display
1	Display alarm bitmaps in place of form icon

Example:

```
Sub Form1_Resize
  If Form1.WindowState = 1 Then
    MhAlarm1.WindowState = 1
  Else
    MhAlarm1.WindowState = 0
  End If
End Sub
```

Custom Events

Ring

Syntax: *CtlName*_Ring()

Description: The Ring event occurs approximately every .RingTime milliseconds. You can use it to update your program's status, or execute any other task you want to perform, while waiting for the user to respond to the alarm.

Example: The MHALARM project