

## BALLOON.VBX Custom Control

### Description

The BALLOON Control adds a new technique of providing help to the user that is not available any other way. This control allows the developer to provide the user with a dynamic popup help system for all the controls on each form in the VB application. The popup help text information provided is especially useful to new user and is ideal for custom in-house applications where the user may not have a manual to refer to.

When enabled, the control provides a custom event which defines when the user moves the mouse cursor over a form's control. The VB application then passes a help text string to the BALLOON.VBX control which then moves the popup window to the vicinity of the selected control and displays the help text in the popup window. If for a particular control help text is not desired, then no return text should be provided by the VB application.

A separate BALLOON Control must be placed on each form that will have a popup help session available.

A "popup balloon help session" is defined from the time the popup balloon appears on-screen to the point at which the popup balloon window is removed.

### File Name

BALLOON.VBX

### Remarks

When you create and distribute applications that use the BALLOON control you should install the file BALLOON.VBX in the customer's Microsoft Windows \SYSTEM sub directory. All of the properties, events, and methods for this control are listed below. Properties and events that apply only to this control, or require special consideration when used with it, are underlined. They are documented in this help file. See the *Visual Basic Language Reference* or on-line Help for documentation of the remaining properties, events, and methods.

### Properties

Action

BackColor

Balloon Type

Balloon Arrow

CtlName

FontBold

FontName

FontSize

ForeColor

Index

Inverse Video

Left

MessageText

Tag

Timeout

Top

**Events**

BalloonClick

**Methods**

The Action property is used as a pseudo method.

**Utility Functions**

Included control Utility Funtions are for accessing various control properties.

## Typical Problems

1. Some Visual Basic controls do not have hwnd properties.

The Visual Basic control types of Image, Shape and Label do not have hwnds and are not intended to be highlighted and have help text displayed. Use Text boxes and Picture controls as controls that perform the same functions and have hwnd properties. Any control that does not have a hwnd property will not be highlighted and will not trigger a BalloonClick() event.

If any controls without hwnd properties need to have help text displayed, include the control in an enclosing frame which has a hwnd parameter. Use the hwnd of the container control in the Select Case statement of the BalloonClick() event.

2. WARNING ... Once the popup balloon help message session has begun, the VB application MUST NOT take any action or cause any action which would cause a screen update.

For instance; You may NOT send a dialog box to the screen during a help message session. Doing so will cause the help session to immediately terminate, and may leave windows and the VB application in an unstable state.

To properly terminate a help message session, send a Balloon.Action = STOP command to the control. This clears the help session and allows normal application processing to resume.

3. Terminating the help message session from within the VB application can cause the user to become confused. The VB application should ALWAYS allow a help message session to terminate by the user. A mouse click anywhere on the screen or any keyboard key press ends the in progress balloon help session returning normal windows control to the user.

4. It is important to note that during design time you must explicitly set the BackColor and the ForeColor for each BALLOON.VBX control used in a project. VB does not have defaults for palette aware controls.

5. IMPORTANT ... If the colors displayed by a BALLOON.VBX control's popup window are altered during runtime, the program must assume all responsibility for the colors. The colors do not reset between popup window sessions. If the window containing the BALLOON.VBX control is unloaded, then the colors will reset to the value set at design time.

## **MessageText Property, Balloon Control**

### **Description**

Sets the text to be displayed in the popup window.

The initial text to be displayed is set prior to the issuance of the Balloon1.Action property command. If no default text is supplied, then the control uses the internal message to start a balloon help session.

### **Usage**

`[form.]BALLOON.MessageText[ = setting %]`

### **Data Type**

String, maximum of 255 characters.

### **How to use**

The MessageText property moves the popup help balloon window to the location of the control the mouse is over and puts the text in the popup balloon window.

If the mouse is over a control that help text is NOT desired, then do not set the MessageText property. This "do nothing" default does not move the popup window and does not update the text. The next active control the mouse is placed over will continue the move-update sequence. As an example, in the Case Select statement the entry for the window to skip can be omitted, left blank( i.e. "" ) or use an "exit sub" program statement.

## **Timeout Property, BALLOON Control**

### **Description**

Setting the Timeout property determines the time the Balloon remains on screen when the mouse remains immobile. When the internal VBX timer reaches the Timeout time with no mouse motion, the help session is terminated and the balloon popup window is removed.

### **Usage**

`[form.]BALLOON.Timeout[ = setting %]`

### **Settings**

The Timeout property settings are in milliseconds.

The minimum setting is 1000 ( 1 second ).

The maximum setting is 32000 milliseconds ( 32 seconds ).

The default setting is 10000 ( 10 seconds ).

### **Remarks**

Setting the Timeout time to 0 disables the Timeout timer function. The balloon will remain onscreen until the user terminates the balloon help session

## Action Property, BALLOON Control

### Description

Setting the Action property cause the BALLOON.VBX control to display on-screen. If the property BALLOON.MessageText was set prior to the issuance of the Action property, then that text is displayed as the first text in the first Balloon. Typically this would be used for introducing the user to the help session. A phrase such as "Move the mouse over the control of interest to get help prompting displayed for it". If no message text is sent prior to the action property, then an internal default message is displayed.

### Usage

[*form*.]BALLOON.Action[ = *setting* %]

### Settings

The Action property settings are:

Setting	Description
ID_START = 20	Start a popup balloon help session.
ID_STOP = 21	Stop a popup balloon help session.

### Remarks

The code fragment to begin monitoring the form for mouse move events is

```
Form.Balloon1.Action = ID_START
```

The user terminates a popup balloon help session by a mouse click or any keyboard activity. In those rare occasions where it may be necessary to terminate the session from the VB application, then issuing the following command will do so. NOTE: Use caution with this command, as it often results in user confusion, since the program is taking action which effects the program state and the user is not usually notified.

```
Form.Balloon1.Action = ID_STOP
```

### Data Type

Integer

## **BackColor Property, BALLOON Control**

### **Description**

The background color of the popup window is set by the color palette attached to the control's BackColor property.

The BackColor must be explicitly set to a color, the default is not pre-set when the control is loaded.

Each BALLOON.VBX control in a project may have different BackColor properties set.

### **Usage**

`[form.]BALLOON.BackColor[ = setting %]`

### **Settings**

The BackColor property is a color selection palette to choose the popup window's background color. Select the desired color with the mouse, or type in a unsigned long color value.

### **Data Type**

Unsigned long color value. For SDK users, ForeColor uses the same format as the WINDOWS.H COLORREF parameter.

## **Balloon Arrow Property, WSHHELP Control**

The Balloon Arrow Property of the popup balloon help window is set by selecting true/false from the Properties at design time. The Balloon Arrow Property is a screen graphic which points to the control the help text is generated for. This property allows the designer to customize the screen view of the help balloon.

The properties Balloon Arrow, Balloon Type and Inverse Video may be combined to achieve the desired screen effect.

### **Usage**

Not applicable

### **Settings**

True turns pointer on.

False turns pointer off

### **Data Type**

Boolean

## **Balloon Type Property, WSHHELP Control**

The Balloon Type Property of the balloon help window is set by choosing the desired type from the enumeration list in the properties window at design time.

These types of popup balloons available:

- Type 0: Square balloon with drop shadow
- Type 1: Square balloon with NO drop shadow
- Type 2: Square balloon 3D with NO drop shadow
- Type 3: Rounded balloon with NO drop shadow.

The properties Balloon Arrow, Balloon Type and Inverse Video may be combined to achieve the desired screen effect.

When using the "Type 2: Square balloon 3D with NO drop shadow" style balloon, set the backcolor to light gray. The text color can be set to any value.

### **Usage**

Not applicable.

### **Settings**

Choose the appropriate type from the list.

### **Data Type**

Not applicable.

## **Inverse Video Property, WSHHELP Control**

The Inverse Video Property of the popup balloon help window is set by selecting true/false from the properties window at design time. The Inverse Video Property is a screen graphic control which determines whether the selected control is displayed in Inverse Video when the selection is made.

The properties Balloon Arrow, Balloon Type and Inverse Video may be combined to achieve the desired screen effect.

### **Usage**

Not applicable

### **Settings**

True turns Inverse Video on.

False turns Inverse Video off

### **Data Type**

Boolean

## ForeColor Property, BALLOON Control

### Description

The text color of the popup window is set by the color palette attached to the control's ForeColor property.

The ForeColor must be explicitly set to a color, the default is not pre-set when the control is loaded.

Each BALLOON.VBX control in a project may have different ForeColor properties set.

### Usage

*[form.]BALLOON.ForeColor* [= *setting* %]

### Settings

The ForeColor property is a color selection palette to choose the popup window's text color. Select the desired color with the mouse, or type in a unsigned long color (i.e. COLORREF) value.

### Data Type

Unsigned long color value. For SDK users, ForeColor uses the same format as the WINDOWS.H COLORREF parameter.

## FontBold Property, BALLOON Control

### Description

The FontBold property is a standard Visual Basic property. It sets the font bold attribute of the text which appears in the popup window.

### Usage

`[form.]BALLOON.FontBold [= setting %]`

### Settings

The FontBold property is a standard Visual Basic property. See the manuals for the appropriate values

### Data Type

Boolean (integer).

## FontName Property, BALLOON Control

### Description

The FontName property is a standard Visual Basic property. It sets the font name of the text which appears in the popup window.

### Usage

`[form.]BALLOON.ForeColor[ = font]`

### Settings

The FontName property is a standard Visual Basic property. See the manuals for the appropriate values

### Data Type

String

## FontSize Property, BALLOON Control

### Description

The FontSize property is a standard Visual Basic property. It sets the size of the text which appears in the popup window.

### Usage

`[form.]BALLOON.FontSize [= points]`

### Settings

The FontSize property is a standard Visual Basic property. See the manuals for the appropriate values

### Data Type

Integer.

## BalloonClick Event, BALLOON Control

### Description

This event is generated only after a help session is initiated. A popup help window will appear on-screen. When the user moves the mouse over any control on a form, the control will be displayed in inverse video and the popup window moves to the control windows location, sends the BalloonClick Event to the application and awaits the help text to be applied to the popup help window.

### Syntax

**Sub** *BALLOON\_BalloonClick* ( hwndAtMouse **As Integer** )

### Remarks

Code fragments to illustrate the use of BALLOON Control.

Start up the control;

```
Form.Balloon.MessageText = "Initial message text to display"  
Form.Balloon.Action = ID_START
```

The following SUB processes the returned hwndAtMouse and returns the desired help text to the balloon control;

```
SUB BALLOON1_BalloonClick( hwndAtMouse As Integer )  
    Select Case hwndAtMouse  
        Case Control.hwnd-X1  
            Form.Balloon.MessageText = "message text desired"  
        Case Control.hwnd-X2  
            Form.Balloon.MessageText = "message text desired"  
        .  
        .  
        .  
        Case Control.hwnd-Xn  
            ' No text desired  
            Exit Sub  
        Case Else  
            Exit Sub  
    End Sub
```

## Utility Functions

The Balloon control includes the following utility functions;

**Hwnd2ControlTabIndex:**

Retrieves a control's TabIndex property.

Returns a zero length string if property not found.

**Hwnd2CtlName:**

Retrieves a control's name.

Returns a zero length string if property not found.

**Hwnd2HelpContextID:**

Retrieves a control's HelpContextID.

Returns -1 if property not found.

Only works with VB2 and above.

**Hwnd2TagText:**

Retrieves a control's Tag property.

Returns -1 if property not found.

Each utility function takes a single parameter, the hWnd of the control, for with the property is desired. The declares for using these functions are as follows.

### Hwnd2ControlTabIndex

```
Declare Function Hwnd2ControlTabIndex Lib "BALLOON.VBX" (ByVal hwndControl%) As Integer
```

### Hwnd2CtlName

```
Declare Function Hwnd2CtlName Lib "BALLOON.VBX" (ByVal hwndControl%) As String
```

### Hwnd2HelpContextID

```
Declare Function Hwnd2HelpContextID Lib "BALLOON.VBX" (ByVal hwndControl%) As Long
```

### Hwnd2TagText

```
Declare Function Hwnd2TagText Lib "BALLOON.VBX" (ByVal hwndControl%) As String
```

Place all of the declare statement on a single line.

