ToolTip V1.4

Properties Events

The ToolTip control allows you to add MS style ToolTips and a context sensitive status bar to your Visual Basic applications quickly and easily. The following facilities are provided:

- ToolTips
- Mouse Tracking
- Menu Tracking
- Focus Tracking

A single ToolTip control placed on the main form of the application is all that is normally required to provide context sensitive information throughtout.

The ToolTip control is fully integrated with the TabFrame VBX which can be used to provide tabbed dialog boxes to your Visual Basic applications.

Application Notes

History

Registration

ToolTip

A ToolTip is a small popup window which appears next to the cursor when the mouse pauses over a user interface element such as a tooolbar button and is used to provide a brief description its purpose

ToolTips were introduced by Microsoft with MS Word V6 and MS Excel V5 and are rapidly becoming a de facto standard in user interface design.

TabFrame

The TabFrame control allows you to add tabbed dialog boxes to your Visual Basic applications. It includes full design time support to allow you to switch between tabs in the design environment using the right mouse button.

A demonstration version of the TabFrame control is available for download from the MSBASIC forum on Compuserve - see file TABFRAME.ZIP in library 17 (3rd Party Products) or search on the keyword TABFRAME.

ToolTips Overview

The ToolTip control tracks movement of the mouse over forms and controls belonging to the application and, if the mouse pauses over any control for longer than the period specified by the Delay1 property, the GetToolTip event will be fired.

You specify the tool tip text for each control in the application by placing the appropriate Visual Basic code in the GetToolTip event handler. If the ToolTips property is set to 2 (Use Tag as Default) then the Tag property of the control for which the tool tip is being generated is used as the default value for the tool tip text.

The tool tip window is removed as soon as the mouse leaves the control or a key is pressed. Once a tool tip has been displayed the mouse pause period required to display a new tool tip is reduced to the value specified by the Delay2 property to allow users to browse toolbars without having to wait for each tool tip to appear.

Properties are provided to control the foreground and background colours of the tool tip window as well as the font used to display the tool tip text. The window is sized and positioned automatically by the ToolTip control.

The Scope property determines which forms and controls are watched by the ToolTip control. If the Scope is set to 0 (Application) then the GetToolTip event will be generated for all forms and controls in the application **except** those contained on forms which have their own ToolTip control. If the Scope is set to 1 (Form) then the GetToolTip event will only be generated for controls on the same form as the ToolTip control.

Mouse Tracking Overview

The ToolTip control tracks movement of the mouse over forms and controls belonging to the application and generates MouseEnter and MouseExit events. These events are typically used to update the text displayed in a status bar to indicate the function of the control currently under the mouse cursor.

The MouseEnter and MouseExit events will only be generated if the MouseTracking property is set to True.

The Scope property determines which forms and controls are watched by the ToolTip control. If the Scope is set to 0 (Application) then the MouseEnter and MouseExit events will be generated for all forms and controls in the application **except** those contained on forms which have their own ToolTip control. If the Scope is set to 1 (Form) then the MouseEnter and MouseExit events will only be generated for controls on the same form as the ToolTip control.

Menu Tracking Overview

The ToolTip control monitors menu selections and generates a MenuSelect event each time a new menu item is selected. This event is typically used to update the text displayed in a status bar to indicate the function of the current menu item.

The MenuSelect event will only be generated if the MenuTracking property is set to True.

The MenuSelect event will only be generated for the menu on the form containing the ToolTip control. If several of your forms include menus (except MDI applications - see below) then you should include a ToolTip control on each.

In the case of an MDI application a single ToolTip control placed on the MDI form will generate MenuSelect events for both the MDI menu and any menus defined on MDI child forms - remember that the MDI child menu replaces the main MDI menu when the child form is active. Any form which contains a menu and is not an MDI child form still requires its own ToolTip control.

Focus Tracking Overview

The ToolTip control monitors focus changes and generates GotFocus and LostFocus events. These events are typically used to update the text displayed in a status bar to indicate the function of the control which currently has the focus.

The GotFocus and LostFocus events will only be generated if the FocusTracking property is set to True.

The Scope property determines which forms and controls are monitored by the ToolTip control. If the Scope is set to 0 (Application) then the GotFocus and LostFocus events will be generated for all forms and controls in the application **except** those contained on forms which have their own ToolTip control. If the Scope is set to 1 (Form) then the GotFocus and LostFocus events will only be generated for controls on the same form as the ToolTip control.

Properties

The following properties are supported by the ToolTip control.

Standard Properties

BackColor	FontSize	Left
Enabled	FontStrikethru	Tag
FontBold	FontUnderline	Top
FontItalic	ForeColor	
FontName	Index	

Custom Properties

About	MenuTracking	Shadow
AutoPosition	MouseTracking	SubclassForm
Delay1	OffsetX	ToolTips
Delay2	OffsetY	ToolTipText
FocusTracking	Scope	

Standard Properties

For complete documentation on the standard properties supported by the TabFrame control please refer to the Visual Basic Language Reference. The comments below indicate where the TabFrame control differs from other controls in the use of these properties.

Enabled - If the ToolTip control is disabled then no events will be generated and no tool tips will be displayed.

Font Properties and ForeColor/BackColor - These properties relate to the tool tip window displayed at runtime and have no visible effect at design time.

About Property

Description

Use this property to display the About dialog box which shows the ToolTip version number. Available only at design time.

Scope Property

Description

Determines the scope of the ToolTip control - i.e. which forms and controls it can provide tool tips, mouse tracking and focus tracking for. This property is read-only at runtime.

Usage

[form.]ToolTip.Scope

Remarks

The Scope property settings are:

Setting	Description
0	(Default) Application - the ToolTip control will provide tool tips, mouse tracking and focus tracking for controls on all forms except those which contain another ToolTip control.
1	Form - the ToolTip control will only provide tool tips, mouse tracking and focus tracking for controls on the same form.

If an application contains more than one ToolTip control the control for which an event will be fired is determined as follows :

I) If the form containing the control to which the event relates contains a ToolTip control then the event will be fired for that control.

ii) If the form does not contain a ToolTip control then the event will be fired for a control which has application scope.

III) If there is more than one control with application scope then the one for which the event will be fired is indeterminate - this situation should be avoided.

Data Type

Integer (Enumerated)

ToolTips Property

Description

Determines if the GetToolTip event is fired when the mouse pauses over a form or control.

Usage

[form.]ToolTip.ToolTips [= *numericexpression*]

Remarks

The ToolTips property settings are:

Setting	Description
0	Disabled - the GetToolTip event will never be fired.
1	(Default) Ignore Tag - the GetToolTip event will be fired with the ToolTip parameter set to a null string.
2	Use Tag as Default - the GetToolTip event will be fired with the ToolTip parameter set to the Tag property of the control for which the tool tip is being generated.

Data Type

Integer (Enumerated)

See Also

GetToolTip Event

Delay1 and Delay2 Properties

Description

Determines the delay (in msec) after which the tool tip window will be displayed.

Usage

[form.]ToolTip.Delay1 [= *numericexpression*]

[form.]ToolTip.Delay2 [= *numericexpression*]

Remarks

The Delay1 property determines the primary delay (i.e. how long the mouse must remian over a control) before a tool tip window is displayed. Once a tool tip has been displayed the Delay2 property determines the secondary delay before another tool tip is displayed. If no new tool tip is displayed within the time set by the secondary delay the ToolTip control will revert to the primary delay.

The default settings of 600 msec and 200 msec respectively mean that there is a delay of just over 1/2 second before the first tool tip will appear but subsequent tool tips will appear more quickly to facilitate browsing controls - e.g. the buttons on a toolbar.

Values of less than 50 msec should not be used.

Data Type

Integer

AutoPosition Property

Description

Determines if control over the tool tip window position is controlled automatically or manually.

Usage

[form.]ToolTip.AutoPosition [= {True|False}]

Remarks

The AutoPosition property settings are:

Setting	Description
True	(Default) The ToolTip control will position the tool tip window automatically based on the current cursor shape.
False	The tool tip window will be offset from the current cursor position as determined by the OffsetX and OffsetY properties.

If you are using non-standard cursor shapes (including the large cursor used on some LCD screens) the automatic positioning may not be optimum and you should take control of the tool tip window position by setting this property to False.

Data Type

Integer (Boolean)

See Also

OffsetX Property, OffsetY Property

OffsetX and OffsetY Properties

Description

Determines the position of the toll tip window relative to the current cursor position when the AutoPosition property is set to False.

Usage

[form.]ToolTip.OffsetX [= *numericexpression*]

[form.]ToolTip.OffsetY [= *numericexpression*]

Remarks

The offsets are expressed in pixels and determine the position of the top left corner of the tool tip window relative to the cursor position. Positive values move the window down and to the right; negative values up and to the left.

These properties are ignored when the AutoPosition property is set True.

Data Type

Integer

See Also

AutoPosition Property

Shadow Property

Description

Determines the size of the drop shadow used to highlight the tool tip window.

Usage

[form.]ToolTip.Shadow [= *numericexpression*]

Remarks

This property determines the size, in pixels, of the shadow displayed below and to the right of the tool tip window.

Data Type

Integer

ToolTipText Property

Description

This property can be used at runtime to control the text in the tool tip window. Not available at design time.

Usage

[form.]ToolTip.ToolTipText [= *stringexpression*]

Remarks

This property, which is only available at runtime, can be used to change the text displayed in a tool tip window. Setting the value to a non-empty string will cause the string to be displayed in a tool tip window; setting to an empty string will destroy any tool tip window which is currently displayed.

This facility is very useful if you wish to display separate tool tip text for different regions of the same control. If the ToolTips property is set to 0 (disabled) then you are responsible for destroying any tool tip window which you create, otherwise, the tool tip window will be destroyed in the normal way - i.e. when the mouse leaves the current control, a key or mouse button is pressed etc.

Data Type

String

See Also

ToolTips Property

MouseTracking Property

Description

Determines if the MouseEnter and MouseExit events will be generated as the mouse moves into and out of forms and controls.

Usage

[form.]ToolTip.MouseTracking [= {True|False}]

Remarks

The MouseTracking property settings are:

Setting	Description
True	The MouseEnter and MouseExit events will be fired each time the mouse moves into or out of a form or control.
False	(Default) The MouseEnter and MouseExit events will never be fired.

If enabled, the MouseExit event for one control will always occur before the MouseEnter event for the next control. The only exception to this is when the form or control which the mouse has left is destroyed in which case the MouseExit event will not be generated.

Data Type

Integer (Boolean)

See Also

MouseEnter Event, MouseExit Event

MenuTracking Property

Description

Determines if the MenuSelect event will be generated when a menu item is selected.

Usage

[form.]ToolTip.MenuTracking [= {True|False}]

Remarks

The MenuTracking property settings are:

Setting	Description
True	The MenuSelect event will be fired each time a menu item is selected from a menu on the form containing the ToolTip control.
False	(Default) The MenuSelect event will never be fired.

Data Type

Integer (Boolean)

See Also

MenuSelect Event

FocusTracking Property

Description

Determines if the GotFocus and LostFocus events will be generated when the input focus moves from one control to another.

Usage

[form.]ToolTip.FocusTracking [= {True|False}]

Remarks

The FocusTracking property settings are:

Setting	Description
True	The GotFocus and LostFocus events will be fired each time a form or control receives or loses the input focus.
False	(Default) The MenuSelect event will never be fired.

If enabled, the LostFocus event for one control will always occur before the GotFocus event for the next control. The only exception to this is when the form or control which is losing the focus is destroyed in which case the LostFocus event will not be generated.

Data Type

Integer (Boolean)

See Also

GotFocus Event, LostFocus Event

SubclassForm Property

Description

Determines if the ToolTip control will subclass the form in order to monitor menu selections.

Usage

[form.]ToolTip.SubclassForm [= {True|False}]

Remarks

The SubclassForm property settings are:

Setting	Description
True	(Default) The ToolTip control will subclass the form in order to monitor menu selections.
False	The ToolTip control will not subclass the form. When the control is not subclassing the form the MenuSelect event will never be triggered and the tool tip and mouse tracking functions may behave in an unexpected fashion when a menu is open.

This property is provided to overcome problems encountered when the ToolTip control is included on a form together with another control which subclasses the form - e.g. a control which allows you to intercept messages such as Desawares SBC.VBX and SBCEASY.VBX. See the application notes for more information.

Data Type

Integer (Boolean)

See Also

Use with other subclassing controls

Events

The following events are supported by the ToolTip control.

- GetToolTip
- GotFocus
- LostFocus
- MenuSelect
- MouseEnter
- MouseExit

GetToolTip Event

Description

Occurs when the ToolTip control is about to display a tool tip window.

Syntax

Sub *ctlname*_GetToolTip([Index as Integer], FrmName as String, CtlName as String, ToolTip as String)

Remarks

The GetToolTip event uses these arguments:

Argument	Description
Index	Identifies the control if it is part of a control array.
FrmName	The name of the form containing the control for which the tool tip is about to be displayed.
CtlName	The name of the control for which the tool tip is about to be displayed. If the tool tip is to be displayed for the form then this argument will be an empty string.
ToolTip	The text to be displayed as a tool tip for the specified form or control. This value should be set in the GetToolTip event handler. If the tool tip text contains line feed (Chr$(10)) characters then the text will be shown on multiple lines.

The value of the ToolTip parameter on entry is determined by the current setting of the ToolTips property. If this property is set to 1 (Ignore Tag) then the ToolTip parameter will be set to a null string; if set to 2 (Use Tag as Default) then the ToolTip parameter will be set to the Tag property of the control for which the tool tip is being generated.

If the ToolTip argument is set to a null string on exit then no tool tip window will be displayed.

See Also

ToolTips Property

MouseEnter Event

Description

Occurs when the mouse enters a form or control being monitored by the ToolTip control.

Syntax

Sub *ctlname*_MenuEnter([Index as Integer], FrmName as String, CtlName as String, TagText as String)

Remarks

The MouseEnter event uses these arguments:

Argument	Description
Index	Identifies the control if it is part of a control array.
FrmName	The name of the form containing the control which the mouse has entered.
CtlName	The name of the control which the mouse has entered. If the mouse is over the form then this argument will be an empty string.
TagText	The Tag property of the form or control which the mouse has entered.

This event will always occur after the MouseExit event for the form or control from which the mouse has moved (unless that form or control has been destroyed).

See Also

MouseTracking Property, MouseExit Event

MouseExit Event

Description

Occurs when the mouse exits a form or control being monitored by the ToolTip control.

Syntax

Sub *ctlname*_MenuExit([Index as Integer], FrmName as String, CtlName as String)

Remarks

The MouseExit event uses these arguments:

Argument	Description
Index	Identifies the control if it is part of a control array.
FrmName	The name of the form containing the control which the mouse has exitted.
CtlName	The name of the control which the mouse has exitted. If the mouse was over the form then this argument will be an empty string.

This event will always occur before the MouseEnter event for the form or control into which the mouse has moved (unless the form or control which the mouse has exitted has been destroyed).

See Also

MouseTracking Property, MouseEnter Event

MenuSelect Event

Description

Occurs when the user selects a menu item from a menu on the form containing the ToolTip control.

Syntax

Sub *ctlname*_MenuSelect([Index as Integer], MenuName as String, MenuCaption as String, TagText as String)

Remarks

The MenuSelect event uses these arguments:

Argument	Description	
Index	Identifies the control if it is part of a control array.	
MenuName	The value of this parameter determines the type of menu item selected:	
	Visual Basic Menu control	control name
	System menu item	System
	Unknown menu item	Unknown
	Menu closed	(Null string)
MenuCaption	If the menu item corresponds to a Visual Basic menu control this parameter will contain the Caption property of the menu control. If the menu item does not correspond to a Visual Basic menu control (e.g. a system menu item) this parameter will contain the menu caption string **without** the mnemonic key definition. If the menu has been closed this parameter will contain a null string.	
TagText	If the menu item corresponds to a Visual Basic menu control this parameter will contain the Tag property of the menu control. If the menu item does not correspond to a Visual Basic menu control (e.g. a system menu item) or the menu has been closed this parameter will contain a null string.	

This event can be used to reset status bar text to its normal condition when a menu is closed as indicated by a null string for the MenuName parameter.

When an MDI child form is maximised its control menu appears as the first menu item on the MDI forms menu bar. When an item is selected from the child forms control menu the MenuName parameter will be Unknown; when an item is selected from the system (or control) menu of the form on which the ToolTip control is placed the MenuName parameter will be System. This allows you to distinguish between the MDI form control menu and the child form control menu when the child form is maximised.

See Also

MenuTracking Property

GotFocus Event

Description

Occurs when a form or control being monitored by the ToolTip control receives the input focus.

Syntax

Sub *ctlname*_GotFocus([Index as Integer], FrmName as String, CtlName as String, TagText as String)

Remarks

The GotFocus event uses these arguments:

Argument	Description
Index	Identifies the control if it is part of a control array.
FrmName	The name of the form containing the control which has received the focus.
CtlName	The name of the control which has received the focus. If it is the form itself which has received the focus then this argument will be an empty string.
TagText	The Tag property of the form or control which has received the focus.

This event will always occur after the LostFocus event for the form or control which has lost the focus (unless that form or control has been destroyed).

See Also

FocusTracking Property, LostFocus Event

LostFocus Event

Description

Occurs when a form or control being monitored by the ToolTip control loses the input focus.

Syntax

Sub *ctlname*_LostFocus([Index as Integer], FrmName as String, CtlName as String)

Remarks

The MouseExit event uses these arguments:

Argument	Description
Index	Identifies the control if it is part of a control array.
FrmName	The name of the form containing the control which has lost the focus.
CtlName	The name of the control which has lost the focus. If it is the form itself which has lost the focus then this argument will be an empty string.

This event will always occur before the GotFocus event for the form or control which is receiving the input focus (unless the form or control losing the focus has been destroyed).

See Also

FocusTracking Property, GotFocus Event

Application Notes

The following topics describe techniques which are commonly used with the ToolTip control.

- Using multiple ToolTip controls
- Using the Tag property
- Implementing a status bar
- Use with the TabFrame control
- Use with other subclassing controls
- Menu tracking in MDI applications

Using Multiple ToolTip Controls

ToolTip controls are only active when the form on which they are placed is loaded into memory. If there is no single form in your application which is always loaded then you will need to use multiple ToolTip controls. You may also wish to use mulitple controls so that the tool tip management is kept local to each form - e.g. each form containing a ToolTip control to handle the controls on that form.

A good approach is to include a ToolTip control on the main application form with the Scope property set to 0 (Application) and to add ToolTip controls to all the other major forms (typically dialog boxes) with the Scope property set to 1 (Form). Each ToolTip control should handle the controls on its own form and the one on the main application form can also handle any other controls which need tool tips.

You should avoid placing multiple ToolTip controls on the same form or including more than one control with Application scope. Failure to observe these rules could lead to unpredictable behaviour.

If you wish to provide menu tracking for multiple forms (except MDI child forms) then you must use a ToolTip control on each.

Using the Tag Property

The Tag property of a control is passed to several of the ToolTip control event procedures and can be used to simplify the code needed in these procedures. Some simple examples are given below to suggest ways in which the Tag property could be used.

- Automatic tool tips
- Automatic tool tips and status bar hints
- Automatic menu hints
- Automatic focus hints

Example 1 - Automatic Tool Tips

Set the ToolTips property to 2 (Use Tag as Default) and set the Tag property of each control for which you wish to provide a tool tip to the tool tip text. No code is required in the GetToolTip event procedure since the Tag property will be passed as the initial value of the ToolTip parameter and is returned unchanged. Any control which has a null Tag property will not have a tool tip.

Example 2 - Automatic Tool Tips and Status Bar Hints

Set the ToolTips property to 2 (Use Tag as Default) and set the Tag property of each control for which you wish to provide a tool tip and status bar hint as follows :

```
        Tag := <tool tip> :<hint>
```

i..e. the tool tip text followed by a colon (or any other separator character) and the status bar hint text.

Include the following code in the ToolTip control event procedure:

```
Sub ctlname_GetToolTip(FrmName As String, CtlName As String, ToolTip As String)
        Dim n As Integer
        n = InStr(ToolTip, :)
        If n > 0 Then ToolTip = Left$(ToolTip, n - 1)
End Sub
Sub ctlname_MouseEnter(FrmName As String, CtlName As String, TagText As String)
        Dim n As Integer
        n = InStr(ToolTip, :)
        if n > 0 Then lblStatus = Mid$(ToolTip, n + 1)
End Sub
Sub ctlname_MouseExit(FrmName As String, CtlName As String)
        lblStatus =
End Sub
```

Note: the MouseTracking property must be set to True.

Example 3 - Automatic Menu Hints

Set the Tag property of each menu control for which you wish to provide a status bar hint to the status bar hint text. This can be done at design time by selecting the menu control in the property window or at runtime in the Form_Load event.

Include the following code in the ToolTip control event procedure:

```
Sub ctlname_MenuSelect(MenuName As String, MenuCaption As String, TagText As String)
        lblStatus = TagText
End Sub
```

Note: the MenuTracking property must be set to True.

Example 4 - Automatic Focus Hints

Set the Tag property of each control which can receive the input focus to the text which you would like to appear on the status bar when that control receives the focus.

Include the following code in the ToolTip control event procedure:

```
Sub ctlname_GotFocus(FrmName As String, CtlName As String, TagText As String)
        lblStatus = TagText
End Sub
Sub ctlname_LostFocus(FrmName As String, CtlName As String)
        lblStatus =
End Sub
```

Note: the FocusTracking property must be set to True.

If you are also using the tag property to provide automatic tool tips and/or status bar hints triggered by mouse movement over the control then you will need to encode the Tag property accordingly. For example the Tag property could be defined as :

```
        <tool tip> : <mouse hint> : <focus hint>
```

The GetToolTip, MouseEnter and GotFocus event handlers will then need to extract the relevant component of the Tag property in a manner similar to that demonstrated in the example for automatic tool tips and status bar hints.

Implementing a Status Bar

A status bar is often included at the bottom of a window and used to display various messages concerning the current status of the application. One common use of a status bar is to show hints describing the purpose of a menu item or other user interface element such as a toolbar button. The ToolTip control makes implementing this type of status bar simple by providing the MenuSelect, MouseEnter/MouseExit and GotFocus/LostFocus events.

Code placed in these event procedures can be used to update the text on a status bar as the user selects menu items, moves the mouse over various user interface elements or moves the input focus. The default status bar text (e.g. Ready) can be restored in the MouseExit/LostFocus events and the MenuSelect event when both the MenuName and MenuCaption parameters are null indicating that the menu has been closed.

Use with the TabFrame Control

When used in conjunction with the TabFrame VBX the ToolTip control can provide individual tool tip and mouse tracking for each tab. The GetToolTip, MouseEnter and MouseExit events are generated as the mouse moves over each tab (the client area of the active tab is ignored).

Use with Other Subclassing Controls

Care is needed if the ToolTip control is placed on a form which is subclassed by another control - e.g. a control which intercepts Windows messages such as Desawares SBC.VBX and SBCEASY.VBX. In order to ensure that the subclassing and unsubclassing takes place in the correct order you may need to use the SubclassForm property.

Normally the ToolTip control will subclass the form on which it is placed as soon as the control is created and the form will remain subclassed until the control is destroyed. This is necessary so that the control can monitor menu activity in order, among other things, to generate the MenuSelect event. If another control also subclasses the form then it is essential that the subclassing is removed in the reverse order in which it is installed.

In order to control the subclassing order you should set the SubclassForm property to False in the design environment and then include code in the Form_Load and Form_Unload event to ensure things are done in the correct order:

```
Sub Form_Load()
         install subclassing
        SubClass1.CtlParam = Form1
        GSToolTip1.SubclassForm = True
End Sub

Sub Form_Unload(Cancel As Integer)
         remove subclassing in reverse order
        GSToolTip1.SubclassForm = False
        SubClass1.CtlParam =
End Sub
```

Failure to ensure that subclassing is removed in the correct order could result in a GPF.

Note This problem is inherent in subclassing technology and is not the fault of any control which uses this technique.

Menu Tracking in MDI Applications

All menu selections in an MDI application can be tracked by a single ToolTip control placed on the MDI form with the exception of system menu commands for non-maximised child forms. The menu name can be used to determine what type of menu item has been selected:

Visual Basic menu control	control name
MDI form system menu	System
Maximised child form control menu	Unknown
Menu closed	(null string)

Items on the system/control menus may be identified from the MenuCaption parameter.

When a child form is not maximised the control menu selections can only be trapped by placing a ToolTip control on the child form. The only menu selections trapped by this control will be system menu selections while the form is not maximised - i.e. the ToolTip control on the MDI form will continue to trap VB child menu items and child control menu items when the form is maximised.

Change History

Date	Description
15-Mar-1994	V1.0.0 - first version.
21-Mar-1994	V1.1.0 - functional enhancements:
	a) Added Scope property to improve handling of multiple controls.
	b) Provided user control over dwell periods through the Delay1 and Delay2 properties.
	c) Provided user control over tool tip window position via the AutoPosition, OffsetX and OffsetY properties.
	d) Added support for multi-line tool tips.
29-Mar-1994	V1.2.0 - functional enhancements:
	a) Added facility to use Tag property of a control as the default tool tip text.
	b) Added mouse tracking with MouseEnter and MouseExit events.
	c) Added menu tracking with MenuSelect event.
13-Apr-1994	V1.3.0 - functional enhancements:
	a) Added focus tracking with GotFocus and LostFocus events.
	b) Added drop shadow with Shadow property.
	c) Integrated with TabFrame V1.3 to allow each tab to have its own tool tip.
	d) Changed AutoPosition to show tool tip above cursor if it would otherwise be off the bottom of the screen.
	e) Fixed problems with MDI child menus.
	f) Fixed bug which failed to recognise disabled controls.
21-Apr-1994	V1.3.1 - maintenance release
	a) Fixed minor bugs.
	b) Added ToolTipText property for manual control of tool tips.
	c) Added SubclassForm property to control subclassing.

Registration

The demonstration version of the ToolTip control is fully functional but may only be used in the development environment. Any attempt to use this version in conjunction with an EXE file will display a dialog box identifying it as a demonstration version and the control will fail to load.

If you find the ToolTip control useful then you can receive a full version, which may be distributed with your applications, by registering as follows :

1) In the SWREG forum on Compuserve. The fee is be $29.95 and the registration ID is 2275.

2) By sending a cheque or money order for £20 to :

GC Consulting Services Ltd
Fellsgarth House
Hognaston
Ashbourne
Derbyshire DE6 1PR
ENGLAND

In return for your registration you will receive the latest version of the ToolTip control and be eligible for product support (via the MSBASIC forum on Compuserve) and free product upgrades for a period of 12 months.

Any questions should be sent to Graham Cockell (Compuserve ID 100113,2774) via e-mail or the MSBASIC forum.